MPLab Technical Report TR 2001.06

A SNoW-Based Automatic Facial Feature Detector

Evan C. Smith

University of California at San Diego
Department of Cognitive Science
Institute for Neural Computation

1 Background

A major goal in machine vision is recognizing human facial expressions. At some point in between locating a face and reading its expression it is necessary that we locate the eyes, nose and other facial features. This information can be used to determine the orientation of a face, help to normalize for differences between faces and locate facial regions used in expression recognition. While a wide number of systems have been applied to the detection of facial features most of these efforts were not systematic nor did they include an evaluation of the strengths and weaknesses of each system. In this paper we evaluate a feature recognition system based on the Sparse Network of Winnows (SNoW) architecture[1]. This choice was motivated by the success reported by Yang et al. in the development of a SNoW based face detector [5]. Their system was reported to outperform a variety of other face detection methods such as naïve Bayes and support vector machines. Their measure for comparison was somewhat subjective and difficult to generalize as they only presented the data from each method which they felt were best. None the less, the results were very promising.

In testing our detector we varied the following parameters: (1) the degree of low pass filtering, (2) the size of image patches, (3) the extent of sub-sampling, and (4) the number of intensity levels after histogram equalization. We measured performance of our feature detector using the A' statistic, a non-parametric measure of sensitivity commonly used in the psychophysical literature.

1.1 The FERET Database

We used the FERET face database [2], a set of images collected by the US Army Research Laboratory to develop, test, and evaluate face recognition algorithms. Every image is 385 x 256 pixels, grayscale (with 8 bits per pixel), and contains a single centered face.

A set of 446 frontal images was selected from this database. Each image was labeled by hand to encode the location of fifteen separate features of interest (fiducial points.) In our experiments we used the center of each iris and the philtrum as the features of interest (see Figure 1).

Figure 1: The three fiducial point used by our system: the two irises and the philtrum.

2 The SNoW System

The SNoW architecture has been widely applied to tasks in computational linguistics, the arena for which it was originally designed [1]. Its first application to the visual processing domain was the face detector of Yang et al. [5]. Superficially similar to the classic perceptron the sparse network of winnows is considerably more powerful. A perceptron consists of a set of input nodes, $\{x_1, x_2, \ldots, x_n\}$, connected to an output unit by a set of weights, $\{w_1, w_2, \ldots, w_n\}$. The net output of the network, y, is the summation of the inputs multiplied by their weights such that $y = \Sigma\, w_i x_i$. In perceptrons the relationship between values of x_i are purely scalar and y is linear, and, so, these networks are capable of making only linear separations between categories. Alternatively, SNoW networks connect their inputs to outputs by functions, which are learned during training, such that $y = \Sigma\, F_j(i_j)$. This allows a SNoW network to learn any linearly separable problem as well as a wide variety of problems which can only be separated nonlinearly.

The means by which SNoW generates the functions connecting inputs to output is based on how the inputs are represented. Rather than inputting images into the system simply as I(x, y), the intensity level at a given pixel, individual pixels are expanded into collection of input nodes [5]. Each of these collections is a Boolean representation of intensity at a given pixel (see Figure 2). For an image of 16 intensity values each pixel would be represented by a vector of 16 elements with a single 1 in the element corresponding to the actual intensity and zeros everywhere else. As a separate and independent weight links each intensity value of each pixel to the output, the distribution of weights associated with that pixel can approximate nonlinear functions of the intensity value. For a 20x20 pixel image (400 total pixels) with 256 intensity values the snow representation is a 400x256 input matrix, but with only 400 active elements.

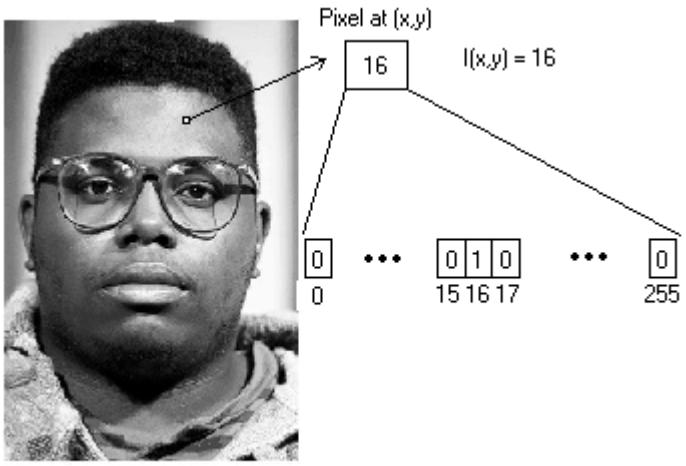

Figure 2: SNoW's Boolean representation of a single pixel with intensity 16. The row of boxes is a vector with 1 at its 16th element.

Separate networks were generated and trained for the left eye, right eye, philtrum and "none-of-the-above". Each subnetwork consisted of a set of all input nodes linked to a single output node. The output indicated the presence or absence of its target feature by taking a value smaller or larger than the given threshold. The links between inputs and output are embodied by a set of weights. Each weight is a non-negative, real number whose magnitude indicates how strongly the network associates the linked input with the presence of its target feature. So, $w_{i,j,k}$ is the weight, w, connecting pixel (i,j) with intensity k to the output node. Initially the weights are set to 1.0 for any input which is actually present in the training set and zero otherwise. In other words, in all training images if $I(23, 45)$ is never 100 then $w_{23,45,100} = 0$. Because weights are changed in multiplicative fashion inputs assigned a weight of zero will never be active in either training or testing. The output for a given image is calculated simply by summing the weights connecting inputs active in the image. A given facial feature is considered to be present if and only if $y>\theta$ for its network. During training the threshold, θ, was set equal to the total number of pixels in the input image patches.

2.1 Winnow

The update rule for SNoW is multiplicative [1]. There are two updating parameters: a promotion parameter $\alpha>1$ and a demotion parameter $0<\beta<1$. During training, if the output is less than or equal to threshold when it should be above (i.e., a feature is present but not detected) then all weights activated by that input are increased by multiplying them by α. Alternatively, if the output exceeds threshold but no target feature is present then all weights associated with that input are decreased via multiplication by β. In all other cases (which equate to correct predictions) the weights are unchanged. Although some alternates were tried, default values [1] of $\alpha=1.5$ and $\beta=0.7$ were used in our system.

3 Method

3.1 Independent Variables

To investigate the potential of SNoW as a facial feature detector we focused on four independent variables. The first was the degree of low pass filtering of the images. The filter was Gaussian, which produces a weighted average of intensity values over some area described by *sigma*, σ, its standard deviation in pixels. Figure 3 shows representations of Gaussian kernels of varying sigmas. Averages are taken across the extent of the kernel with brighter pixels weighted more strongly.

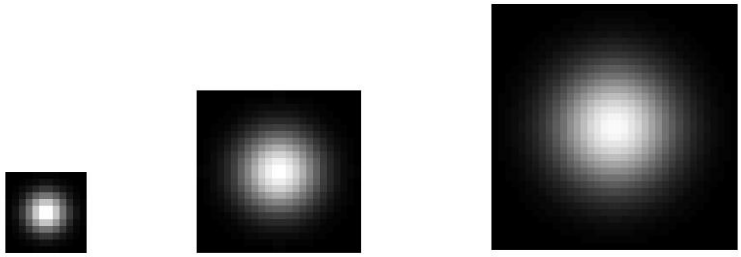

Figure 3: Gaussian kernels with standard deviations of 1/5 iris width (2 pixels), 2/5 iris width (4 pixels) and 3/5 iris width (6 pixels.) The size each square matrix is 12x12 pixels, 24x24 pixels and 36x36 pixels respectively.

By varying the magnitude of sigma we could alter the amount of high frequency information which was filtered from the images. One benefit of low pass filtering is the removal of person specific information from the image; however, too much filtering will remove of useful information. The chosen sigma values were 2, 4 and 6 pixels. In terms of the faces in the database these values represent approximately one fifth of an iris width, two fifths of an iris width and three fifths iris width. The results of filtering at these levels can be seen in Figure 4.

Figure 4: The same FERET image through three different low pass filters.
From left to right the standard deviation of the Gaussian kernels are 2, 4 and 6 pixels.

Our second independent variable addressed the question of how much of an image the system needs to "see" in order to optimize its performance. Neither entire FERET images nor single filtered pixels were presented as input to the detector; rather, image patches of differing sizes, centered on a given pixel were extracted from the images. These patches were the actual inputs. We tested patches 40x40 pixels and 80x80 pixels (Figure 5). For reference, an 80x80 patch represents about 8 iris widths per patch.

Figure 5: Examples of the patch sizes used as input: 40x40 and 80x80. Iris centered patches are on the left and philtrum centered patches on the right.

Our third independent variable was the degree of sub-sampling. This is a process of eliminating pixels from an image, in effect shrinking the image size. One can remove pixels from a filtered image with no loss of information as long as the rate of sampling is twice the highest spatial frequency found in the image. Sub-sampling might also focus the network on a limited set of regular inputs and, at the same time, reduce the complexity of the network. The 80x80 pixel patches were sub-sampled to one fourth, one eighth and one sixteenth size. The 40x40 pixel patches were sub-sampled to one forth and one eighth size. A 40x40 pixel patch sub-sampled to one eighth size becomes a 10x10 pixel patch.

Our last independent variable varied the number of intensity values used to represent a patch represents another attempt to reduce to complexity of the input. In fact, this operation performs a similar function in the intensity domain as the Gaussian filter performs in the spatial domain. We performed histogram equalization using either 64, 128 or 256 intensity values.

3.2 Performance Measures

Our dependent variable was A', a non-parametric measure of sensitivity commonly used in the psychophysical literature. The A' represents the optimal performance achievable by a detector on a 2-alternative forced choice task. We calculate the A' for each detector (each subnetwork trained to detect a specific feature) using standard receiver operating characteristic, or ROC, curves: during testing we can produce a distribution of detector performance value by varying the threshold. With a threshold of zero all patches tested activate the subnetwork; the network never misses a true feature but it says yes to everything. Conversely, with the threshold set arbitrarily high the network never activates erroneously, but it never recognizes true targets either. The maximum threshold could be set equal to the maximum output of the system for the given test set. To complete the idea, a whole range of threshold values can be generated, to some arbitrary degree of accuracy, between zero and the maximum. For each value of the threshold parameter we plot the probability that a feature was detected given that the

feature was present, *P*(detect feature | feature present), versus the probability that a feature was detected given that the feature was absent, *P*(detect feature | feature absent). The area taken by integrating under the ROC curve is the A'. An A' of 0.5 indicates a detector of zero sensitivity (chance), while an A' of 1.0 indicates perfect sensitivity (the detector locates all features with no false positives.)

We implemented and evaluated our SNoW-based feature detectors within the Matlab computing environment. The inputs to the system were our selected FERET images. We obtained three separate A' measures: one for eye recognition, one for philtrum recognition, and the last for the discrimination of patches which are neither eyes nor philtrum. We tested generalization performance using a standard cross validation approach, "training" on 357 images of the total 446 and then testing on the remaining 89 images. This process was repeated five times, and each time the 89 test (or generalization) images consisted of a different fifth of the total faces. This gave us an estimate of A' for each combination of parameters based on all 446 images.

For each test image, we first apply the Gaussian filter. Depending on the trial the kernel of the Gaussian filter had a standard deviation of 1/5, 2/5, or 3/5 iris width (2, 4 or 6 pixels.) After filtering we randomly chose three points per feature that were within one seventh of the distance between eyes from the center of the eye, which typically encompasses the entire iris. This produced the points for each eye (total of six) and three points for the philtrum. In addition, we also randomly chose three points that were known not to be within any tested feature (Figure 6). This choice of test locations made it easy to calculate *P*(detect feature | feature present) and *P*(detect feature | feature absent) and helped to expand the size of our data set.

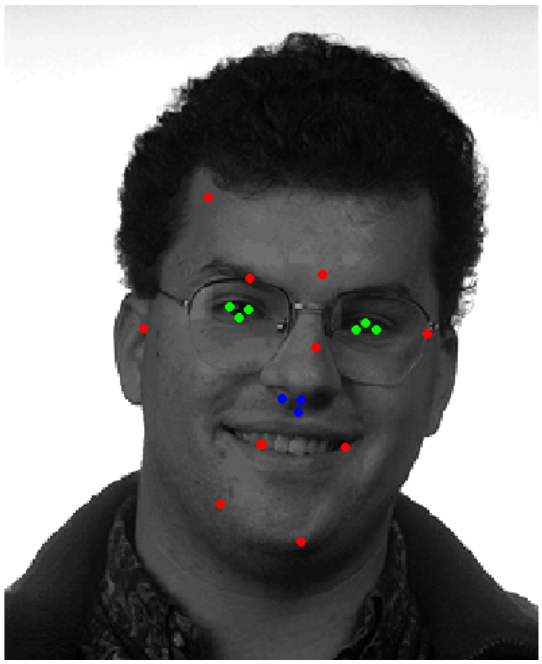

Figure 6: To test performance, three points are chosen from each feature, plus a set of random points known to not fall within those same feature. Using these points we estimated the hit and false alarm specifications of the system as a function of the threshold θ.

For each point a feature patch of either 40x40 pixels or 80x80 pixels was extracted from the original image. These patches were sub-sampled to one sixteenth, one eighth, one fourth or one half of their original size, or not at all. The sub-sampling rate was not completely independent as the original size of the patch delimited the amount of sub-sampling possible. For example 40x40 patches were maximally sub-sampled to an eighth, however, 80x80 patches were sub-sampled to as much as a sixteenth its original size. Histogram equalization was the final step in preprocessing the image. Only the patch, not the entire image, was histogram equalized. Before the patches can be input they must be recast into the SNoW representation. Specifically, in our system this representation was a matrix whose rows are individual pixels and whose columns are intensity levels. A 40x40 pixel patch, sub-sampled by to one eighth its original size (so now it is 10x10 pixels), with only sixty-four gray levels becomes a 100x64 matrix. An 80x80 patch, without sub-sampling or change in number of intensity levels, becomes a 1600x256 matrix. Having collected inputs from the database, starting weights were then generated for each subnetwork. Any weight without an active feature in the training set (for all images $w_{i,j,k} = 0$) is set to zero and all other weights are sent to one.

Each of the twelve patches from a given image was a positive example for the network for which it was the target and a negative example for all other networks. For example, a patch from a philtrum is used by the eye and "none" subnetworks as negative examples. If any of them output above the training threshold ($\Sigma w_{active} > \theta$) then the weights involved in that output are multiplied by 0.7. For the philtrum network the philtrum patch is a positive example. If it fails to output above threshold then all weights involved in the output are multiplied by 1.5. Patches from opposing eyes were not used as either positive or negative examples for each eye subnetwork. Weights were adjusted patch by patch. A sweep refers to a pass over all of the patches of all of the images once. Each network was trained for a total of five sweeps. Though this is short compared to many gradient decent learning algorithms SNoW has been shown to be a highly efficient learner [1]. Early investigation on our system indicated that learning had plateaued by the fifth sweep.

After training was finished we tested performance on the 89 images reserved for testing. These were processed in exactly the same manner as the training images. To plot the ROC curves, the maximum threshold was set to the maximum output by any sub-network during this testing session. We divided the interval between the zero threshold and the maximum into 1000 evenly spaced steps. Each individual output was compared against all of these thresholds, and we recorded the proportion of correct positives and of false positives. The proportions at each threshold were plotted against one another with false positives on the horizontal axis. A trapezoidal integrative method produced the A' based on the resulting curves. We report the mean A' and variance boxed on the five cross-validation runs in Appendix A.

4 Results

The most significant effects on A' were due to the combination of patch size and the extent of sub-sampling. The optimal patch size in our study was 80x80 pixels in size, the largest tested. Decreasing the image sizes via sub-sampling, on the other hand, had a negative effect which could be seen in both patch sizes. Averaged across filtering and

number of intensity levels, our maximum A' for all features were for 80x80 patches with sub-sampling of ¼. In terms of low pass filtering, a Gaussian kernel with a standard deviation (sigma) of 4 pixels produced slightly higher A's than did standard deviations of 2 or 6 pixels. These effects were not dramatic and only held for the optimal results produced by the largest patches. Finally, the system generally produced a larger A' for 256 intensity levels for eye recognition and philtrum recognition . The "other" network showed the higher A' at 64 intensity levels. The differences as a function of intensity levels are fairly consistent across other parameters. In the end our largest A' was 0.9836 for the philtrum, 0.9837 for the eyes and 0.9246 for the "other" subnetwork. All of these results were achieved with an 80x80 pixel patch, sub-sampled to ¼. The filter had a sigma of 4 pixels (approximately 2/5 iris width). The patches had 128 intensity levels for the eye value, 256 intensity levels for the philtrum value and 64 intensity levels for the "none" value. A complete table of results can be found in Appendix A.

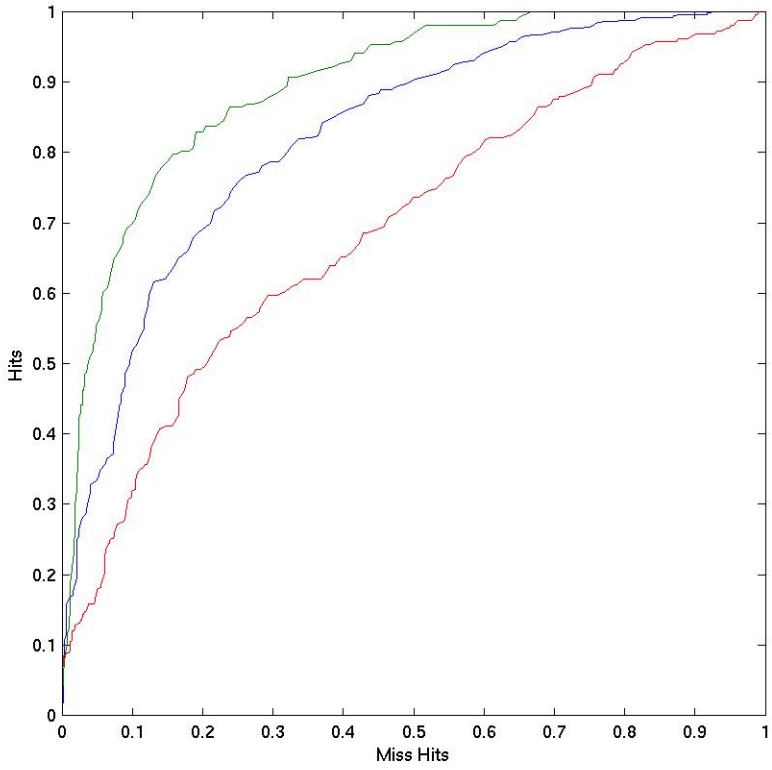

Figure 7: An ROC curve produced by our most accurate system. The *P*(detect feature | feature present) is plotted on the vertical axis. The *P*(detect feature | feature absent) is plotted on the horizontal axis. Green is philtrum, blue is eye and red is "other".

5 Discussion

Our SNoW feature detector preformed quite well in comparison to the Gabor jet representation used by Wiskott et al. [4]. Our maximum philtrum A' of 0.9038 surpassed the 0.6141 produced by the original configuration of Wiskott et al. as well as an optimization of the Gabor filters used by Wiskott, which produced an A' of 0.8664. Further comparisons with detection systems made from standard perceptrons are ongoing

An important result from our evaluation of the SNoW face detector is the effects of patch size on A'. For example, the system locates the philtrum by looking for the mouth and nose (Figure 8). Unfortunately, this might make the system sensitive of occlusion; if the area around an eye is occluded a system using larger patches will be more negatively effected than one using smaller patches. We've begun to generate A's for 100x100 pixel patches and, so far, have not found a significant increase in the A' over those for 80x80. This is likely because the larger images include additional "noise" which negates any small gains from additional information. The degree of filtering also indicates a balance between noise and information. Little filtering allows too much subject specific information into the network which decrements its ability to generalize. On the other hand, too much filtering removes information which the system might use to identify its targets.

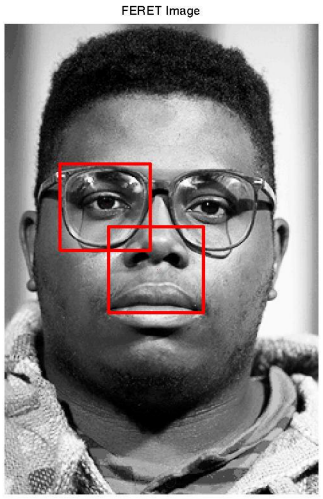

Figure 8: The size of an 80x80 pixel patch on a face.

We preformed an analysis of the weights learned by each network. As noted earlier, the SNoW system describes the relationship between a given pixel and an output by a function rather than a single weight. The weight vector for each pixel (a row out of the entire weight matrix) describes the function used by the network for a particular pixel. Surprisingly, these functions were very noisy and difficult to interpret. This outcome isn't completely unexpected in that SNoW treats each intensity value at each pixel as completely independent. This independence allows for the highly discontinuous functions seen in Figure 9. It is surprising that the system is able to generalize. Note that in a perceptron the function would be lines with slopes equal to the weight learned for that pixel. We expected this ability to be derived from the learning of regular properties

in faces which might be described by a more regular function. We are currently developing tools to better analyze these functions.

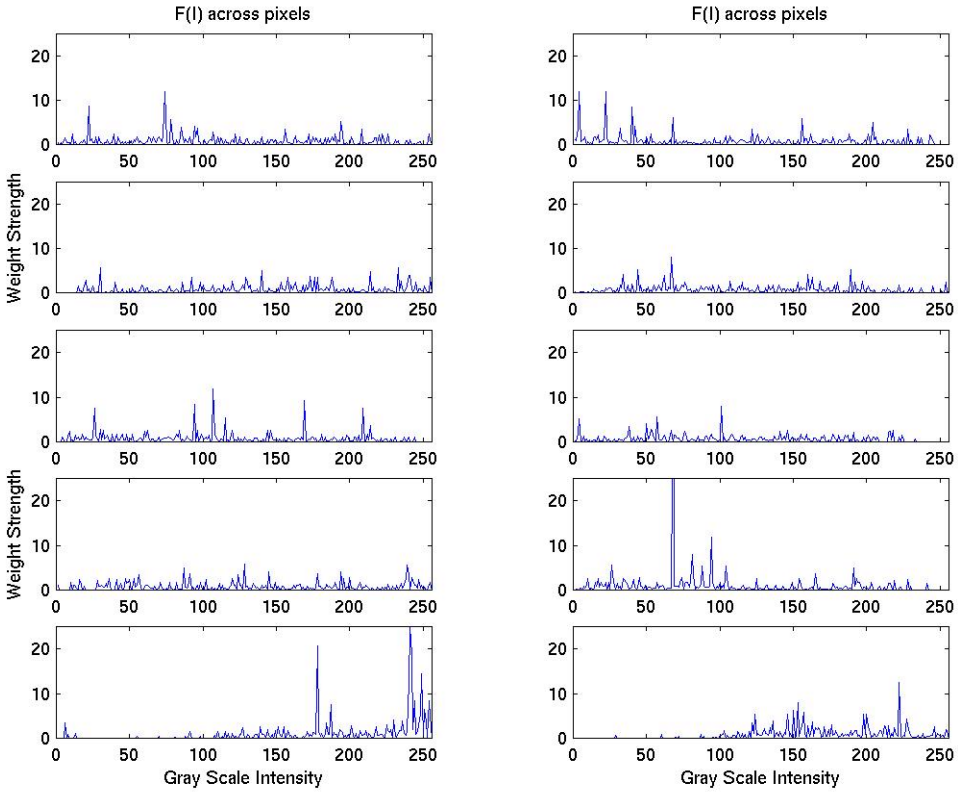

Figure 9: A plot of weight magnitudes (vertical) vs. intensity value (horizontal). for selected pixels from a philtrum detector.

5.1 Future Research

We have begun to evaluate the use of regularization procedures in order to improve SNoW's performance by eliminating much of the noise in the weight functions. Initial results are promising, showing up to a 14% increase in A' for some subnetworks. In one method currently being evaluated, detectors were trained by promoting and demoting not only active weights but also weights adjoining them. For example, if $w_{23,45,100}$ is promoted this method also promotes a small range of intensity values at this pixel to a lesser degree. The regularizing used here essentially relies on a one dimensional Gaussian filter to distribute the weight changes to surrounding weights. Other forms of smoothing are possible and will be applied to our system.

In order to further understand the relative importance of differing pixels to a given network's discrimination we will begin a "lesion" study of highly effective detector subnetworks. By selectively eliminating or altering weights for one or more pixels we should learn more about the importance of specific weights. This might also tell us something about functions which possibly exist distributed among the weights. Although the system currently treats each pixel and intensity value independently, they don't

necessarily vary independently with respect to the images. The networks might encode these relationships across its weights.

We will soon begin to evaluate SNoW using more complex forms of image processing. For example, rather than using the simple Gaussian low pass filters we will make use of Gabor wavelets. Recent work in our lab [2] revealed optimum spatial frequency and orientation for Gabor filter methods using a "nearest neighbor" recognition engine. Given the success of this work as well as previous research into Gabor filter banks we expect further improvement in SNoW's A'. Other areas of investigation might be the use of Haar wavelets and the inclusion of prior information in the systems selection process.

References

1. A Carlson, C Cumby, J Rosen, and D Roth, SNoW User's Guide. UIUC Tech report UIUC-DCS-R-99-210.
2. I Fasel, E Smith, M Bartlett, and J Movellan. *A comparison of Gabor filter methods for automatic detection of facial landmarks*, Machine Perception Lab Tech Report, UCSD, 2000.
3. P Phillips, H Wechsler, J Juang, and P Rauss. The feret database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing Journal*, 16(5):295-306, 1998.
4. L Wiskott, J Fellous, N Kruger, and C van der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7): 775-779, 1997.
5. M Yang, D Roth, and N Ahuja. A SNoW-based face detector. NIPS-12, Dec, 1999.

Appendix A

The following table contains the cross-validated A' values for each combination of parameters. Also included is the variance of the five cross-validation sets which were averaged to produce the A' values given here. The upper left corner of each table shows the patch size and amount of sub-sampling; for example, "80/8" indicates data for an 80x80 pixel patch sub-sampled to 1/8th size.

EYE

	80/4	Intensity Levels 256	128	64
sigma	1	0.9816	0.9778	0.9748
	5	0.0004	0.0006	0.0008
	2	0.9825	0.9837	0.9758
	5	0.0004	0.0004	0.0007
	3	0.9166	0.9139	0.9183
	5	0.0004	0.0003	0.0007

PHIL TRUM

	80/4	Intensity Levels 256	128	64
	1	0.9831	0.9832	0.9829
	5	0.0002	0.0003	0.0002
	2	0.9836	0.9799	0.9824
	5	0.0003	0.0006	0.0003
	3	0.9166	0.9139	0.9183
	5	0.0002	0.0006	0.0004

OTHER

	80/4	Intensity Levels 256	128	64
	1	0.9185	0.9244	0.9231
	5	0.0024	0.0024	0.0019
	2	0.9228	0.9234	0.9246
	5	0.0029	0.0015	0.0019
	3	0.9166	0.9139	0.9183
	5	0.0026	0.0027	0.0022

	80/8	Intensity Levels 256	128	64
sigma	1	0.9745	0.9693	0.9669
	5	0.0005	0.0009	0.0012
	2	0.9776	0.9742	0.9699
	5	0.0006	0.0006	0.0009
	3	0.9764	0.9745	0.9710
	5	0.0006	0.0007	0.0008

	80/8	Intensity Levels 256	128	64
	1	0.9799	0.9798	0.9778
	5	0.0003	0.0004	0.0004
	2	0.9804	0.9790	0.9775
	5	0.0005	0.0005	0.0006
	3	0.9775	0.9755	0.9763
	5	0.0005	0.0007	0.0007

	80/8	Intensity Levels 256	128	64
	1	0.8908	0.9044	0.9095
	5	0.0024	0.0014	0.0016
	2	0.8961	0.9080	0.9064
	5	0.0018	0.0016	0.0019
	3	0.8761	0.8917	0.8930
	5	0.0025	0.0023	0.0022

	80/16	Intensity Levels 256	128	64
sigma	1	0.9323	0.9523	0.9515
	5	0.0017	0.0013	0.0012
	2	0.9429	0.9581	0.9601
	5	0.0011	0.0011	0.0008
	3	0.9400	0.9570	0.9495
	5	0.0009	0.0008	0.0013

	80/16	Intensity Levels 256	128	64
	1	0.9429	0.9639	0.9691
	5	0.0009	0.0008	0.0005
	2	0.9445	0.9640	0.9631
	5	0.0011	0.0010	0.0017
	3	0.9362	0.9543	0.9526
	5	0.0012	0.0010	0.0010

	80/16	Intensity Levels 256	128	64
	1	0.8086	0.8603	0.8746
	5	0.0025	0.0018	0.0017
	2	0.8321	0.8638	0.8874
	5	0.0023	0.0016	0.0019
	3	0.7959	0.8490	0.8792
	5	0.0010	0.0015	0.0012

	40/4	Intensity Levels 256	128	64
sigma	1	0.9661	0.9650	0.9609
	5	0.0003	0.0004	0.0004
	2	0.9684	0.9681	0.9626
	5	0.0003	0.0003	0.0006
	3	0.9705	0.9685	0.9685
	5	0.0003	0.0004	0.0003

	40/4	Intensity Levels 256	128	64
	1	0.9622	0.9616	0.9554
	5	0.0006	0.0006	0.0005
	2	0.9639	0.9633	0.9601
	5	0.0007	0.0007	0.0005
	3	0.9598	0.9607	0.9607
	5	0.0008	0.0008	0.0004

	40/4	Intensity Levels 256	128	64
	1	0.9160	0.9222	0.9191
	5	0.0014	0.0012	0.0010
	2	0.9182	0.9181	0.9246
	5	0.0011	0.0011	0.0016
	3	0.9119	0.9043	0.9043
	5	0.0016	0.0017	0.0012

		Intensity Levels		
	40/8	256	128	64
sigma	1	0.9036	0.9369	0.9394
	5	0.0009	0.0003	0.0006
	2	0.9159	0.9217	0.9217
	5	0.0006	0.0003	0.0003
	3	0.9288	0.9484	0.9388
	5	0.0004	0.0003	0.0001

	Intensity Levels		
40/8	256	128	64
1	0.9135	0.9422	0.9374
5	0.0010	0.0004	0.0006
2	0.9100	0.9338	0.9358
5	0.0019	0.0014	0.0003
3	0.8918	0.9247	0.9258
5	0.0017	0.0012	0.0007

	Intensity Levels		
40/8	256	128	64
1	0.8425	0.8812	0.8932
5	0.0009	0.0012	0.0015
2	0.8325	0.8858	0.9102
5	0.0007	0.0008	0.0008
3	0.8260	0.8658	0.8905
5	0.0007	0.0013	0.0005