Proceedings of the Fourth IASTED International Conference
VISUALIZATION, IMAGING, AND IMAGE PROCESSING
September 6-8, 2004, Marbella, Spain

FEATURE SELECTION USING ADABOOST FOR FACE EXPRESSION RECOGNITION

Piyanuch Silapachote, Deepak R. Karuppiah, and Allen R. Hanson
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003, USA
email: {pla, deepak, hanson}@cs.umass.edu

ABSTRACT
We propose a classification technique for face expression recognition using AdaBoost that learns by selecting the relevant global and local appearance features with the most discriminating information. Selectivity reduces the dimensionality of the feature space that in turn results in significant speed up during online classification. We compare our method with another leading margin-based classifier, the Support Vector Machines (SVM) and identify the advantages of using AdaBoost over SVM in this context. We use histograms of Gabor and Gaussian derivative responses as the appearance features. We apply our approach to the face expression recognition problem where local appearances play an important role. Finally, we show that though SVM performs equally well, AdaBoost feature selection provides a final hypothesis model that can easily be visualized and interpreted, which is lacking in the high dimensional support vectors of the SVM.

KEY WORDS
dimensional reduction, feature selection, pattern recognition, machine learning, AdaBoost, Support Vector Machine

1 Introduction

Current classification techniques define image similarities at varying levels of detail. Those that use only global features such as color [1] and texture histograms [2] tend to be expensive in both memory and computation because of the dimensionality of the feature even though such features can lend more flexibility and information to the classification tasks. Applying Principal Component Analysis (PCA) can reduce the dimensionality without sacrificing performance [3]. Even then one cannot avoid a high dimensional feature space altogether since the principal components in PCA are still linear combinations of the original features. Tieu and Viola [4] present a boosting approach to select a small number of features from a very large set allowing fast and effective online classification. Their method applies hand-crafted primitive kernels recursively on subsampled images resulting in a causal structure that is used to discriminate image classes. These selective features, however, are only meaningful in the feature spaces and it is difficult to intuitively interpret the structure of the selected features.

Personal experiences and psychophysical studies in saccadic eye movements [5] indicate that local appearances play crucial roles in learning good classification. More often than not, people can recognize objects because they seek particular regions where discriminating information is located. For example, to classify a car based on make or model, the focus of attention is on small regions at the front/back of the car where the name/symbol is printed. The observation of the shapes of head/rear lights is also significant. On the same note, certain other regions like windshields or tires do not carry that much information. Modeling from this finding, a classification mechanism should be able to discard most of the irrelevant image regions without sacrificing performance.

Techniques that depend only on local regions [6, 7] capitalize on this insight by attempting to segment an image into blobs and focus only on the similarities between blobs using colors or textures. Minut et al. [5] uses eye-saccade data to generate sequential observations that are then used to learn a Hidden Markov Model (HMM) for each face in the database. The observed locations turn out to be more important than the observation sequence, though the latter fits nicely into a HMM framework. Jaimes et al. [8, 9] propose a strong correlation between eye-movements and different semantic categories of images and use this hypothesis to build automatic content-based classifiers.

Following this motivation to look for locally discriminative appearances, we propose to identify from a high dimensional feature space only those dimensions that carry the most information. In this paper, we present an approach using AdaBoost to select features as part of the training phase itself thereby making the feature extraction process in the testing phase very efficient. Our approach differs from previous work in that the reduced set of features contains both locally and globally informative features. Our system automatically singles out the discriminative features and consequently the discriminative image regions without relying on *a priori* domain knowledge. Finally, by a novel combination of feature extraction and feature selection classification techniques, we show that an overlay of these region selections over an original image enables us to visualize actual image regions that carry relevant information that is crucial for the classification task. Thus, the proposed technique not only significantly reduces the prob-

lem complexity and speeds up the online process, but also provides a meaningful interpretation of image regions to the classification process.

In the following section, we define a composite feature that is a concatenation of global and local appearance features extracted from uniformly partitioned regions of an image. The appearance features are derived from either Gabor wavelets or Gaussian derivative filters applied to each partition. We present experimental results about the performance of our approach on the problem of recognizing facial expressions like smile, scream etc. This problem is particularly suited for our approach because local regions of the image are sufficient to determine an expression category. Finally, we compare the results of this approach to classification using a standard technique such as the SVM and point out the essential differences.

2 Features

2.1 Multi-scale Gaussian derivative features

Let I_x be the first order partial derivative with respect to x of image I. I_y is defined similarly and I_{xx}, I_{xy}, I_{yy} denote the second derivatives. These derivatives are more stable when computed by filtering the image with the corresponding normalized Gaussian derivative than by the regular finite differences method [2]. A 2D Gaussian function with zero mean and standard deviation σ, is defined as

$$g_1(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

The parameter σ is also referred to as a scale of the Gaussian function. Local intensity surface orientation and curvature features can be defined as functions of the partial derivatives:

$$\begin{aligned} \text{orientation, } O &= \text{atan2}(I_y, I_x) \\ \text{isophote curvature, } N &= A[2I_xI_yI_{xy} - I_x^2I_{yy} - I_y^2I_{xx}] \\ \text{flowline curvature, } T &= A[I_{xy}(I_x^2 - I_y^2) + I_xI_y(I_{yy} - I_{xx})] \\ \text{shape index, } C &= 0.5 - \frac{1}{\pi} \cdot \tan^{-1}\frac{N+T}{N-T} \\ A &= (I_x^2 + I_y^2)^{-\frac{3}{2}} \end{aligned} \tag{2}$$

The orientation and the shape index computed at every pixel are discretized and histograms representing distributions of these features are constructed. These features are good in modeling dominant local gradients as well as curvatures [10]. Let the feature vector be represented as

$$\phi_\sigma = [C_\sigma^b; O_\sigma^b]_{1\times 2b} \tag{3}$$

where σ is the scale factor, b is the number of histogram bins, C and O, respectively, are the shape index and the orientation histograms. Additionally, instead of concatenation, either C or O could be used as stand alone features. An example of this feature is shown in Figure 1.

2.2 Gabor wavelet features

The Gabor wavelet filter [11] is defined as

$$g_2(x, y, \sigma, u_0, v_0) = g_1(x, y, \sigma) \cdot e^{j\cdot(u_0+v_0)} \tag{4}$$

where (u_0, v_0) is the center frequency of the filter. Let the Gabor filter response at scale σ and location (x, y) on an image be represented by $\psi_\sigma(x, y)$. These filter responses are discretized into b intervals that define the b bins of the histogram. Let b_{min}^k and b_{max}^k, respectively, be the lower and upper limits of the k^{th} bin. Then a value at each bin k, represented by ϕ_σ^k, is equal to

$$\phi_\sigma^k = \sum_{\forall x,y, b_{min}^k \le \psi_\sigma(x,y) < b_{max}^k} \psi_\sigma(x, y) \tag{5}$$

The feature vector ϕ_σ at scale σ is then defined as

$$\phi_\sigma = [\phi_\sigma^1, \phi_\sigma^2, ..., \phi_\sigma^b]_{1\times b} \tag{6}$$

An example of this feature, ϕ_σ, is shown in Figure 1. This feature is different from a regular histogram because it accumulates into each bin the actual filter responses instead of the pixel counts. In a regular histogram of the responses, the bins whose range are near zero will tend to dominate the histogram when significant areas of the image are textureless. Clearly, this is not desirable and therefore these responses need to be weighted less. This happens naturally in our suggested approach by accumulating actual filter responses. Further, it is desirable to pay more attention to the filter responses with extreme values as they indicate a better (or worse) alignment between local image structures and the filter patterns.

2.3 The Composite features

Both the Gaussian functions and the Gabor wavelets can be viewed as bandpass filters whose bandwidths depend on σ, their scale parameter. Consequently, multiple scale filtering is essential in order to extract a wide range of frequencies.

Local features extracted from different image regions may be noticeably different. This distinction is lost if only a global histogram is constructed. We preserve both global and local features by partitioning an image into $P\times Q$ distinct blocks. The number of partitions is determined by the scale at which the features are extracted. Specifically, at the coarsest scale with relatively high σ, a 1×1 partition is sufficient because the coarse filter, having a large footprint, looks for global structures. At finer scales, having smaller filtering footprints, more partitions are needed.

The composite feature vector at a scale σ whose corresponding partition is $P\times Q$ is

$$\phi_\sigma = [\phi_{\sigma,1}, \phi_{\sigma,2}, ..., \phi_{\sigma,P\cdot Q}] \tag{7}$$

where $\phi_{\sigma,p}$ is the appearance feature vector at scale σ extracted from the p^{th} subimage of the partition $P\times Q$. The concatenation over M scales gives the final composite feature vector:

$$\Phi = [\phi_{\sigma_1}; \phi_{\sigma_2}...; \phi_{\sigma_M}] \tag{8}$$

3 Classifiers

We describe two margin-based classification paradigms used in our experiments: the feature selective AdaBoost classifier and the SVM-based classification approach.

In a two-class classification problem, let a set of n data points in an N dimensional feature space be

$$(\Phi_1, y_1), (\Phi_2, y_2), ..., (\Phi_n, y_n), \Phi_i \in \Re^N, y_i \in \{\pm 1\} \tag{9}$$

A pair (Φ_i, y_i) is called a positive instance if y_i=1, and a negative instance, otherwise. A classifier seeks a decision function, or a hypothesis, H: $\Re^N \rightarrow \{\pm 1\}$ that minimizes some loss function.

3.1 AdaBoost

Recall that each image $i \in \{1, 2, ..., n\}$ is represented by Φ_i, a concatenation of global and local appearance features $\phi_{\sigma,p,i}$ extracted at different scales from different subregions. From this composite feature, AdaBoost [12] learns the classification by selecting only those individual features that can best discriminate among classes. We achieve this by designing our weak learner as suggested by Howe [13].

Training on an individual appearance feature $\phi_{\sigma,p}$, a decision boundary hyperplane κ is a bisection between the weighted mean vectors of the positive and negative sample sets.

$$\kappa = \frac{\sum_{\forall i \in \Phi^p} D(i) \cdot \phi_{\sigma,p,i}}{|| \sum_{\forall i \in \Phi^p} D(i) \cdot \phi_{\sigma,p,i} ||} + \frac{\sum_{\forall i \in \Phi^n} D(i) \cdot \phi_{\sigma,p,i}}{|| \sum_{\forall i \in \Phi^n} D(i) \cdot \phi_{\sigma,p,i} ||} \tag{10}$$

where $\Phi^p = \{j | y_j = 1\}$ and $\Phi^n = \{j | y_j = -1\}$, and each sample is weighted by the distribution D.

The positive half-space of κ usually contains a majority of the positive instances. Therefore, if a sample belongs to this half-space, it is classified as positive, and negative otherwise. The decision is flipped when a minority of the positive instances fall into the positive half-space. Symbolically, this weak hypothesis is a function $h \equiv h_{\sigma,p}$: $\phi_{\sigma,p} \rightarrow \{\pm 1\}$ whose empirical error is

$$\epsilon = \sum_{\forall i} D(i) \cdot h_{\sigma,p}(\phi_{\sigma,p,i}) \cdot y_i \tag{11}$$

At each step, every appearance feature parametrized by its scale and its partition, together, form a family of hypotheses $\mathbf{h} \equiv \{h_{\sigma,p}\}$. AdaBoost then chooses a hypothesis that carries minimum error. Effectively, this means that each AdaBoost iteration picks the hypothesis, and in turn the individual feature vector, that contains the most discriminating information allowing a correction of classification errors resulted from previous steps. The feature selective AdaBoost [12] is outlined below.

- Given a training set containing positive and negative samples, where each sample i is (Φ_i, y_i); Φ_i is the composite feature vector of sample i, and $y_i \in \{\pm 1\}$ is the corresponding class label. Initialize sample distribution D_0 by weighting every training sample equally.
- For T iterations do
  - Train a hypothesis for each feature $\phi_{\sigma,p}$.
  - Choose the hypothesis $h_t^\star$ with minimum classification error ϵ_t on the weighted samples.
  - Compute $\alpha_t = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t})$ which weights $h_t^\star$ by its classification performance.
  - Update and normalize the weighted distribution: $D_{t+1}(i) \propto D_t(i) \cdot e^{-\alpha_t y_i h_t^\star(\phi_t^\star)}$.
- The final hypothesis $H(\phi) = sign(\sum_{t=1}^{T} \alpha_t h_t^\star(\phi_t^\star))$ is a linear combination of T hypotheses that are functions of selected features.

3.2 Support Vector Machines

Unlike traditional classification techniques that aim at minimizing the Empirical Risk, SVM approaches the classification problem as an approximate implementation of the Structural Risk Minimization (SRM) induction principle, which is a reduction form of an Expected Risk minimization problem [14, 15]. To this end, a generalization error of a model is minimally bounded and a decision surface is placed in such a way that the margin, which is the distance from a separating hyperplane to the closest positive or negative sample, between different classes is maximized.

SVM approximates the solution to the minimization problem of SRM through a Quadratic Programming optimization. As a result, a subset of training samples is chosen as support vectors that determine the decision boundary hyperplane of the classifier.

Though in principle the hyperplanes can only learn linearly separable datasets, in practice, nonlinearity is achieved by applying an SVM kernel that maps an input vector onto a higher dimensional feature space implicitly.

In this paper, we use SVM [16] with the radial basis function kernel as a black box classifier over the labeled set of composite feature vectors. Doing so yields support vectors in the composite feature space.

3.3 Multi-class classifiers

Both AdaBoost and SVM, as explained above, are suitable only for binary classification. However, they can be easily extended to a multi-class problem by utilizing Error Correcting Output Codes (ECOC) [17].

A dichotomy is a two-class classifier that learns from data labeled with positive (+), negative (-), or (don't care). Given any number of classes, we can relabel them with these three symbols and thus form a dichotomy. Different relabellings result in different two-class problems each of which is learned independently. A multi-class classifier progresses through every selected dichotomy and chooses a class that is correctly classified by the maximum number of selected dichotomies.

Exhaustive dichotomies represent a set of all possible ways of dividing and relabeling the dataset with the three defined symbols. A one-against-all classification scheme on an n-class classification considers n dichotomies each relabel one class as (+) and all other classes as (-).

4 Face Expression Recognition

We applied our integrated feature selection and classification approach to the problem of identifying expressions on faces. The features described in Section 2 are considered appropriate because facial expressions have characteristic local structures that can be mathematically described by edge orientations and curvatures (functions of Gaussian derivatives) or more generally spatial frequencies (Gabor wavelets). Although we look at a person's face as a whole, we focus our attention only on small regions at any instant in time because expressions are mostly localized to regions near the eyes and the mouth. A smile is mostly shown by a person's mouth, while anger is partly shown by a person's eyes. Cheeks and noses contain much less significant information. Since our approach is well suited to single out discriminative features both at the global level and multiple local levels, it is ideal for this problem domain.

4.1 Implementation

We conducted our experiments on the AR face database [18]. We chose face images of 120 people: 55 women and 65 men. Each person shows four expressions: neutral, smile, anger, and scream. There are two images of each person's expression that were taken from two different sessions. Thus in all we have a total of 960 facial images with 240 images for each expression. We manually cropped every face image to remove the influence of the background. This is not an absolute necessary for our method if all the subjects were located at roughly the same region on every image. An example face with four different expressions is shown in the first column of Figure 1.

Both the Gaussian and the Gabor features were extracted at 6 scales, with corresponding partitions of 1×1, 3×3, 3×3, 5×5, 5×5, and 7×7. The first scale corresponds to the lowest frequency with $\sigma_1 = 30$ pixels per cycle. Subsequent frequencies are determined by $\sigma_i = \frac{\sigma_1}{\sqrt{2}^{(i-1)}}$ for $i = \{2, 3, ..., 6\}$. The choice for the first scale, intentionally to be used with a non-partitioned image, is guided by the mean image size of the cropped faces in the database, which is to cover roughly two cycles of the defined filter. On the same basis, higher frequencies determine finer image partitions so that the sizes of subimages are roughly two cycle wide.

The number of histogram bins chosen is 64. While the Gaussian derivative responses are uniformly discretized, the distribution of the Gabor responses is nonlinearly defined where the values ranging from -0.04 to 0.04 are divided equally into 62 intervals making up the middle 62 bins, the first covers all values less than -0.04, and those more than 0.04 fall into the last bin. Examples of the features extracted is shown in Figure 1. The composite feature is very high dimensional. For example, with our configurations, the dimension of the Gaussian composite feature vector is 15,104.

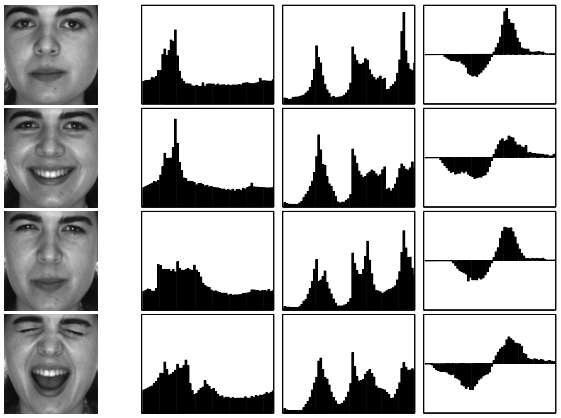

Figure 1. An example face showing four expressions and corresponding features at the coarsest scale. The first column shows the expressions. The normalized shape index and the orientation histograms are shown in the second and third column, respectively. The last column shows the distribution of Gabor filter responses.

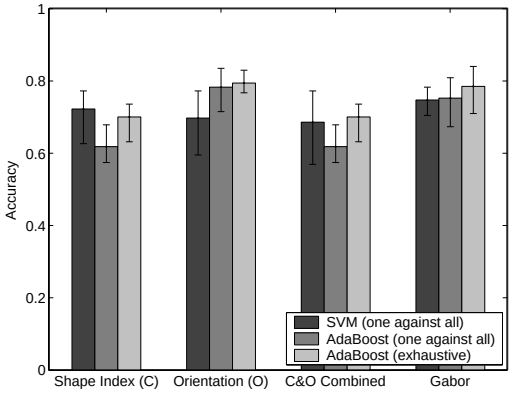

Figure 2. Performance on the test data (anger included)

The number of iterations for AdaBoost is set at 50. Both SVM and AdaBoost performed multi-class classification by employing one-against-all dichotomies. For AdaBoost, we also tested on an exhaustive set of dichotomies. All statistical results of our experiments are based on a 5-fold cross validation analysis [19] where the classifiers were trained on 80% of the data and tested on the other 20% in 5 runs, each run retaining different subsets (20% in each case) of the data as the test set.

4.2 Results and Analysis

The results from two sets of experiments, the first including the anger expression and the second excluding it, are shown

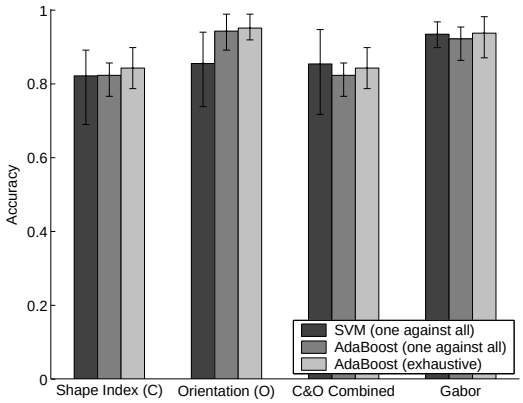

Figure 3. Performance on the test data (anger excluded)

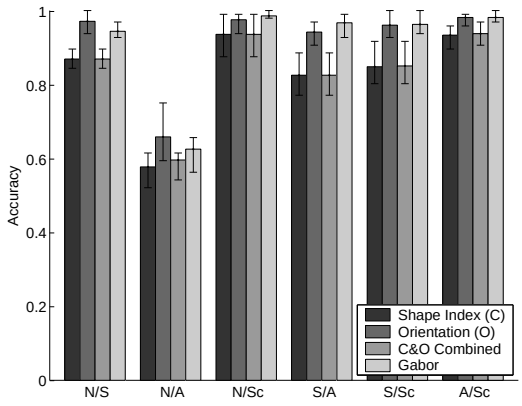

Figure 4. Two-class classifi cation performance of AdaBoost on the test data. On the x-axis, denote the neutral, smile, anger, and scream by N, S, A and Sc, respectively.

Figure 5. Examples showing the similarity of neutral (top row) and anger (bottom row) expressions in the database.

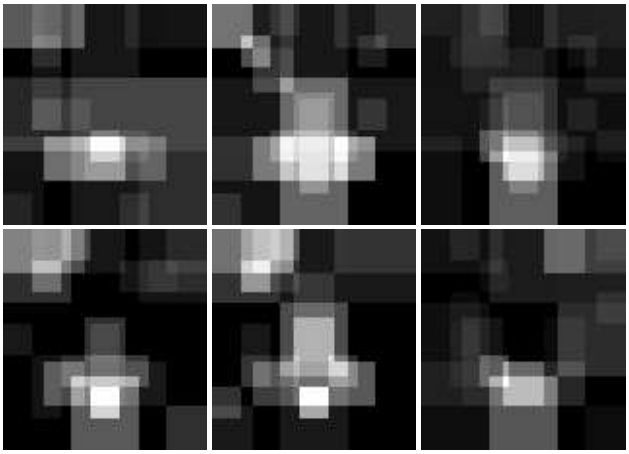

Figure 6. Feature selections by AdaBoost. The dichotomies from top to bottom and left to right are [1 -1 -1], [1 0 -1], [-1 1 -1], [0 1 -1], [1 1 -1], [1 -1 0]. Darker regions represent a low accumulation of α values and their non-discriminative nature. Brighter regions represent their highly discriminative nature for their dichotomies.

in Figures 2 and 3, respectively. In both cases, the best average performance was obtained using the orientation feature with AdaBoost classifier; 79.27% and 94.86% respectively. The reason for the apparently poor performance in the first set of experiments was that neutral and anger as expressed by people in the database were not visibly different in most cases (see Figure 5). This is further substantiated by examining the performance of two class classifiers, illustrated in Figure 4. It is clear that the classifier is having a hard time discriminating between the neutral and anger classes, although it is performing very well on every other case.

Experimental results show that performances of SVM and AdaBoost are comparable. They performed almost equally well with a slight preference toward AdaBoost when an exhaustive set of dichotomies is employed. On average, SVM tends to have higher variances. Another drawback of SVM is its dependency on parameter settings; the choices of kernel function and its parameters are crucial.

It is important to note that while AdaBoost feature selection provides a final hypothesis model that can be easily interpreted, the high dimensional support vectors of SVM approach do not provide any.

Figure 6 illustrates image regions where AdaBoost picked out discriminating features. Each image is a result on a particular dichotomy represented by a code word, where -1, +1 indicate negative and positive samples, and 0 is a don't care label. In a code word, the first number represents a neutral expression, the second and third represent smile and scream. The images show accumulated α values over all iterations. Image regions with higher values contribute more to the final hypothesis. Intuitively, the higher the value a region carries, the more influence it has on the final classification decision, and consequently the more relevant information to a classification task it contains.

As reflected in the results, AdaBoost successfully picked the mouth and the eyes as being most informative and discarded other regions as being irrelevant. This is true because a person's mouth and eyes look different while expressing neutral, smile, or scream. Clearly, an appearance of a mouth region contains significant information. Also in this dataset, people scream with their eyes closed (see Figure 1) which results in the contribution from the eye regions. Additionally, this result draws similarity to how humans naturally perceive and recognize facial expressions.

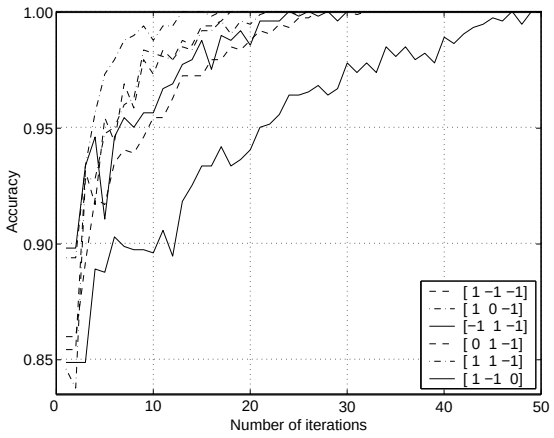

Figure 7. Convergence of feature selective AdaBoost classifiers on the six dichotomies defined in Figure 6.

When AdaBoost with feature selection is employed, the memory requirements and the computational complexity are significantly reduced. During online classification only those features chosen by AdaBoost are needed instead of the composite feature as a whole. Specifically, let S be the total number of appearance features defined and T be the number of iterations run by AdaBoost, then a compression ratio r of S:T is achieved. To be concrete, in our experiment, r = 118:50, which is more than double. Keeping in mind that the actual dimension of the feature vector is $S \times b$ where b is the number of histogram bins (64 in our case), this results in an order of magnitude reduction in dimensionality. Furthermore, the value of T can be set much lower than 50 as the plot in Figure 7 shows that AdaBoost converges quickly in most cases, i.e. within 25 iterations. This further doubles the compression ratio.

5 Conclusions and Future Works

We have proposed a classification system capable of capturing both global and local features and at the same time identifying image regions where distinctive information is located. We successfully applied this technique to the face expression recognition problem.

The main benefit gained from this new feature extraction and image classification approach is the meaningful visualization of informative image regions and the reduction of computational complexity without applying any domain knowledge. The system automatically learns from training data where to look for discriminating information. The reduced feature set then enables fast online classification.

This technique can be applied to a broad range of recognition and classification applications as long as the objects to be classified are at similar location, orientation, and scale in both the training and the testing images. However, if the system is used in conjunction with appropriate segmentation and rectification algorithms then these constraints can be removed.

Acknowledgements

We would like to thank Srinivas Ravela for his MATLAB implementation of the Gaussian derivative features. This work is supported in part by NASA grant NAG9-1445 and DARPA project DABT63-00-1-0004.

References

[1] M. J. Swain and D. H. Ballard, Color indexing, *International Journal of Computer Vision*, 7, 1991, 11–32.

[2] S. Ravela and A. Hanson, On multi-scale differential features for face recognition, *Vision Interface*, Ottawa, Canada, 2001.

[3] A. Pentland, B. Moghaddam, and T. Starner, View-based and modular eigenspaces for face recognition, *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, USA, 1994, 84–91.

[4] K. Tieu and P. Viola, Boosting image retrieval, *IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head Island, USA, 2000, 228–235.

[5] S. Minut, S. Mahadevan, J. M. Henderson, and F. C. Dyer, Face recognition using foveal vision, *IEEE International Workshop on Biologically Motivated Computer Vision*, Seoul, Korea, 2000, 424–433.

[6] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik, Blobworld: A system for region-based image indexing and retrieval, *Third Intl. Conf. on Visual Information Systems*, Amsterdam, The Netherlands, 1999, 509–516.

[7] J. Jeon, V. Lavrenko, and R. Manmatha, Automatic image annotation and retrieval using cross-media relevance models, *26th Intl. ACM SIGIR Conf.*, Toronto, Canada, 2003.

[8] A. Jaimes and S. Chang, Automatic selection of visual features and classifiers, *IS&T/SPIE Storage and Retrieval for Media Databases*, San Jose, USA, 2000, 346–358.

[9] A. Jaimes, J. Pelz, T. Grabowski, J. Babcock, and S. F. Chang, Using human observers' eye movements in automatic image classifiers, *SPIE Human Vision and Electronic Imaging*, San Jose, USA, 2001, 373–384.

[10] S. Ravela, *Differential Representations of Appearance for Image Retrieval*, PhD thesis, Department of Computer Science, University of Massachusetts Amherst, 2003.

[11] C. Palm and T. M. Lehmann, Classification of color textures by gabor filtering, *Machine Graphics and Vision*, 11, 2002.

[12] R. E. Schapire, A brief introduction to boosting, *16th International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 1999, 1401–1406.

[13] N. Howe, A closer look at boosted image retrieval, *International Conference on Image and Video Retrieval*, Urbana-Champaign, USA, 2003, 61–70.

[14] E. Osuna, R. Freund, and F. Girosi, Support vector machines: Training and applications, A.I. Memo 1602, MIT Artificial Intelligence Laboratory, 1997.

[15] V. Vapnik, *Statistical Learning Theory*, (New York: John Wiley & Sons Inc., 1998).

[16] C. C. Chang and C. J. Lin, LIBSVM, 2002.

[17] T. G. Dietterich and G. Bakiri, Solving multiclass learning problems via error-correcting output codes, *Journal of Artificial Intelligence Research*, 2, 1995, 263–286.

[18] A. Martinez and R. Benavente, The AR face database, CVC Technical Report 24, Purdue University, 1998.

[19] T. Mitchell, *Machine Learning*, (USA: WCB/McGraw-Hill, 1997).