

Human face detection in video using edge projections

Mehmet Türkan, Berkan Dülek, Ibrahim Onaran, and A. Enis Çetin*

Dept. of Electrical and Electronics Engineering, Bilkent Univ., Bilkent, Ankara, TR 06800

ABSTRACT

In this paper, a human face detection method in images and video is presented. After determining possible face candidate regions using color information, each region is filtered by a high-pass filter of a wavelet transform. In this way, edges of the region are highlighted, and a caricature-like representation of candidate regions is obtained. Horizontal, vertical and filter-like projections of the region are used as feature signals in dynamic programming (DP) and support vector machine (SVM) based classifiers. It turns out that the support vector machine based classifier provides better detection rates compared to dynamic programming in our simulation studies.

Keywords: Human face detection, edge projections, dynamic programming, support vector machines

1. INTRODUCTION

Human face detection problem has received significant attention during the past several years because of wide range of commercial and law enforcement applications. In recent years, many heuristic and pattern recognition based methods have been proposed to detect human faces in still images and video on gray-scale or color. Human face detection techniques based on neural networks^{1,2}, support vector machines^{3,4}, hidden Markov models^{5,6}, Fisherspace/subspace linear discriminant analysis (LDA)⁷, principle component analysis (PCA)⁸, and Bayesian or maximum-likelihood (ML) classification methods² have been described in the literature ranging from very simple algorithms to composite high-level approaches. Rowley et al.¹ presented a neural-network based upright frontal face detection system using a bootstrap algorithm. The performance over a single network is improved by arbitrating the system among multiple networks. Sung and Poggio² described a distribution-based modeling of face and non-face patterns using a multilayer perceptron (MLP) classifier. They developed a successful example-based learning system for detecting vertical frontal views of human faces in complex scenes. Osuna et al.³ demonstrated a decomposition algorithm to train support vector machines for frontal human face detection in images over large data sets. Guo et al.⁴ proposed a binary tree structure for recognizing human faces after extracting features and learning the discrimination functions via SVMs. Nefian and Hayes⁵ described a hidden Markov model (HMM)-based framework using the projection coefficients of the Karhunen-Loeve Transform (KLT) for detection and recognition of human faces. Although the proposed method results slight improvements on the recognition rate, it reduces the computational complexity compared to previous HMM-based face recognition systems, e.g., Samaria⁶ which uses strips of raw pixels. Belhumeur et al.⁷ developed a successful face recognition system using Fisher's linear discriminant which produces well separated classes in a low-dimensional subspace, insensitive to lighting direction and facial expressions. Turk and Pentland⁸ presented an eigenface based human face recognition technique using principal component analysis. They developed a system which tracks the head of a subject and then the person is recognized by comparing characteristics of the face to those of known individuals with a nearest-neighbor classifier. Conceptually detailed literature surveys on human face detection and recognition are conducted by Hjelmås and Low⁹ and Zhao et al.¹⁰, respectively.

Recently, wavelet domain^{12, 13} based face detection methods have been developed and become very popular. The main reason is that a complete framework has been recently built in particular for what concerns the construction of wavelet bases and efficient algorithms for the wavelet transform computation¹¹. Wavelet packets allow more flexibility in signal decomposition and dimensionality reduction as the computational complexity is an important subject for face detection systems. Garcia and Tziritis¹¹ proposed a wavelet packet decomposition method on the intensity plane of the candidate face regions. After obtaining the skin color filtered (SCF) image using the color information of the original image, they

* Contact person: e-mail cetin@bilkent.edu.tr; phone 90 312 290-1477; fax 90 312 266-4192; www.ee.bilkent.edu.tr

This work is supported by The Scientific and Technical Research Council of Turkey (TUBITAK) and, in part, by EU FP-6 Network of Excellence MUSCLE.

extracted feature vectors from a set of wavelet packet coefficients in each region. Then, the face candidate region is classified into either face or non-face class by evaluating and thresholding the Bhattacharyya distance between the candidate region feature vector and a prototype feature vector. Zhu et al.¹⁴ described a subspace approach to capture local discriminative features in the space-frequency domain for fast face detection based on orthonormal wavelet packet analysis. They demonstrated the detail (high frequency) information within local facial areas. For example; eyes, nose, and mouth show noticeable discrimination ability for face detection problem of frontal view faces in a complex background. The algorithm leads to a set of wavelet features with maximum class discrimination and dimensionality reduction. Then the classification is evaluated by a likelihood test. Uzunov et al.¹⁵ described an adequate feature extraction method in a face detection system. The optimal atomic decompositions are selected from various dictionaries of anisotropic wavelet packets using the adaptive boosting algorithm (AdaBoost)^{16,17}. Their method demonstrates a fast learning process with high detection accuracy.

In this study, a human face detection method in images and video is proposed on both gray-scale and color. Our method is based on the idea that a typical human face can be recognized from its edges. In fact, a caricaturist draws a face image in a few strokes by drawing the major edges of the face. Most wavelet domain image classification methods are also based on this fact because wavelet coefficients are closely related with edges¹¹.

After determining all possible face candidate regions in a given video frame or still image, each region is single stage rectangular 2-D wavelet transformed. In this way, wavelet domain edge-highlighted sub-images are obtained. The low-high and high-low sub-images contain horizontal and vertical edges of the region, respectively. The high-high sub-image may contain almost all the edges, if the face candidate region is sharp enough. It is clear that, the detail (high frequency) information within local facial areas, e.g., eyes, nose, and mouth, show noticeable discrimination ability for face detection problem of frontal view faces¹⁴. We take the advantage of this fact by summarizing these wavelet domain sub-images using their projections, and obtain 1-D projection feature vectors corresponding to edge images of face or face-like regions. The advantage of the projections is that they can be easily normalized to a fixed size and this provides robustness against scale changes. The horizontal and vertical projections are simply computed by summing the absolute pixel values in a row and column in a given edge image, respectively. Furthermore, filter-like projections are computed as in Viola and Jones¹⁷ approach as additional feature vectors. The final feature vector for a face candidate region is obtained by combining all the horizontal, vertical, and filter-like projections. These feature vectors are then classified using dynamic programming and support vector machine based classifiers into face or non-face classes. The dynamic programming method is used for measuring the similarity distance between a face candidate region feature vector and a typical prototype face feature vector. The classification is applied by thresholding the resulting distance.

This paper is organized as follows. Section 2 specifies a general block diagram of our face detection system where each block is briefly described for the techniques used in the implementation. In Section 3, the detailed information for dynamic programming and support vector machine based classifiers are given. In Section 4, the detection performance of the dynamic programming is compared with support vector machines and currently available face detection methods. Conclusions are also presented in Section 4.

2. FACE DETECTION SYSTEM

In this paper, a human face detection scheme for frontal pose and upright orientation is developed (Fig. 1). After determining all possible face candidate regions in a given video frame or still image, each region is decomposed into its wavelet domain sub-images as shown in Fig. 2. Face candidate regions can be estimated based on color information in video as described in Section 2.1. The detail (high frequency) information within local facial areas, e.g., eyes, nose, and mouth, is obtained in high-high sub-image of the face pattern. This sub-image is similar to a hand-drawn face image, and in a given region, face patterns can be discriminated using this high-pass filtered sub-image. Other high-band sub-images can be also used to enhance the high-high sub-image. The wavelet domain processing is presented in Section 2.2. For a face candidate region, a feature vector is generated from wavelet domain sub-images using projections.

Firstly, the generated feature vectors are classified using dynamic programming, which is an extensively studied and widely used tool in operations research for solving sequential decision problems in finite vocabulary speech recognition¹⁸ and various communication theory applications, e.g., the Viterbi algorithm. Dynamic programming algorithm generally used for computing the best possible alignment warp between a probe feature vector and a prototype feature vector, and

the associated distortion between them. This property gives us significant freedom on detecting faces which are oriented in the horizontal and vertical directions. Two concepts are to be dealt with when using dynamic programming; the first one is the feature vector that the whole information of the pattern has to be represented in some manner, and the second one is the distance metric to be used in order to obtain a match path. The distance measure between a probe feature vector and a prototype feature vector is calculated using the *Euclidean* distance metric in this paper. There are two types of constraints on distance measure of dynamic programming: local and global. After evaluating the minimum alignment *Euclidean* distance between the probe feature vector and prototype feature vector in dynamic programming, a threshold value is used for classification of the probe feature vector. Dynamic programming based classifiers are reviewed in Section 3.

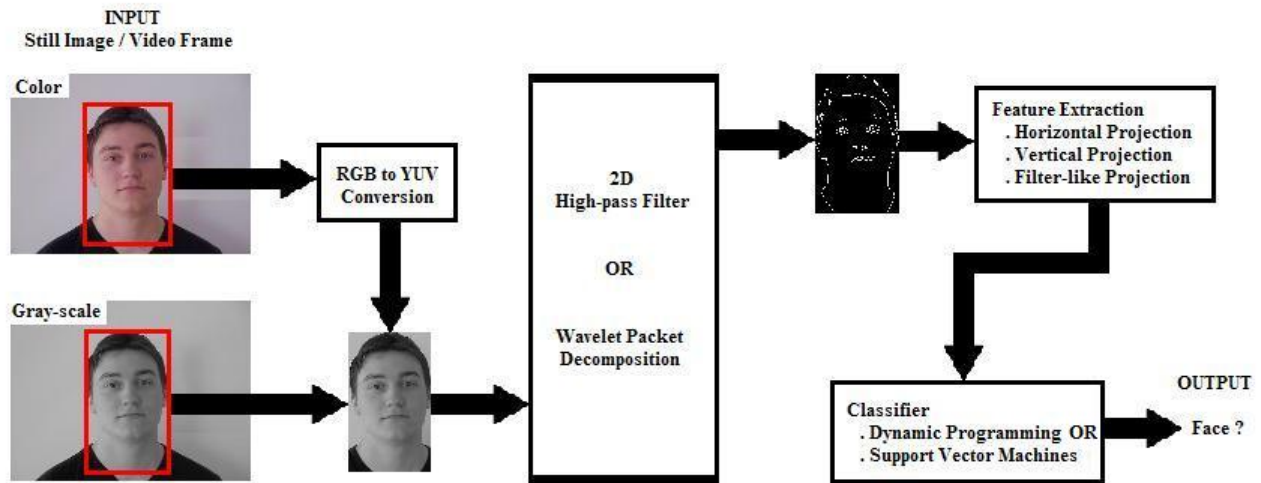


Figure 1: Block diagram of the face detection system.

The second classification approach that we studied is support vector machines (SVMs), which are a brand new and powerful machine learning technique based on structural risk minimization for both regression and classification problems, although the subject can be said to have started in the late seventies by Vapnik^{19,20,21}. SVMs have also been used by Osuna et al.³ for detecting human faces in still images. While training SVMs, support vectors which define the boundary between two or more classes, are extracted. In our face detection case, the extracted support vectors define a boundary between two classes, namely face, labeled as “+1”, and non-face, labeled as “-1”, classes. The main idea behind the technique is to separate the classes with a surface that maximizes the margin between them³. It is obvious that, the main use of SVMs is in the classification block of our system, and it contributes the most critical part of this work. Section 3 also presents a description of support vector machine based classifiers.

2.1 Detection of face candidate regions

Human skin has a characteristic color which is shown to be a powerful fundamental cue for detecting human faces in images or video. It is useful for fast processing and also robust to geometric variations of the face patterns. Most existing human skin color modeling and segmentation techniques contain pixel-based skin detection methods which classify each individual pixel into skin and non-skin categories, independently from its neighbors. Peer et al.²² defined a very rapid skin classifier for clustering an individual pixel into skin or non-skin category through a number of rules in RGB (red, green, blue) colorspace. Although it is a simple and rapid classifier, this method is very much subject to the illumination conditions because of the RGB colorspace. Statistical skin chrominance models^{23,24,25,26} based on (Gaussian) mixture densities and histograms are also studied on different colorspace for pixel-based skin color modeling and segmentation. On the other hand, region-based skin detection methods^{27,28,29} try to take the spatial arrangements of skin pixels into account during the detection stage to enhance the methods performance³⁰.

In this study, especially for real-time implementation, the RGB colorspace with pixel-based skin detection method is chosen for fast processing. Given a color video frame or still image, each pixel is labeled as skin or non-skin using a number of predefined rules²². Then morphological operations are performed on skin labeled pixels in order to have

connected face candidate regions. The candidate regions' intensity images are then fed into a 2-D high-pass filter or a single stage 2-D rectangular wavelet decomposition block.

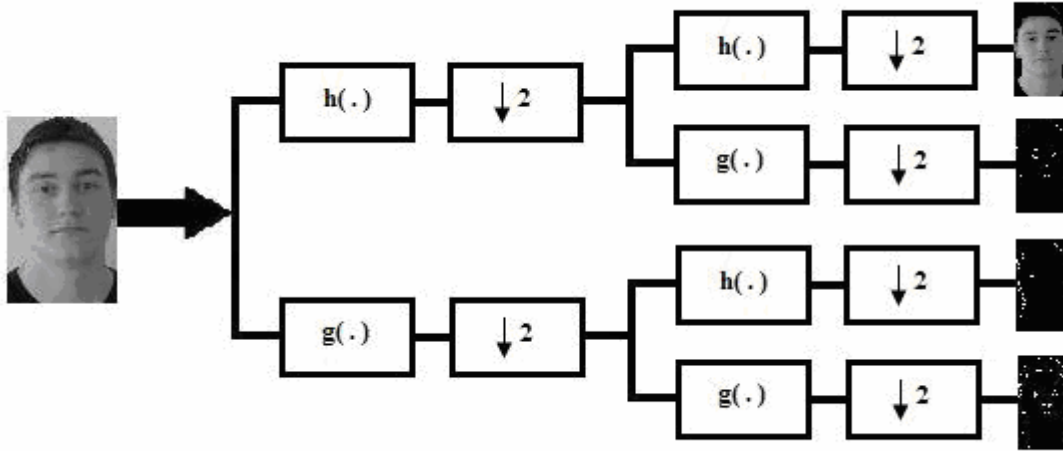


Figure 2: Two-dimensional (2-D) rectangular wavelet decomposition of a face pattern; low-low, low-high, high-low, high-high sub-images. $h(\cdot)$ and $g(\cdot)$ represent 1-D low-pass and high-pass filters, respectively.

2.2 Wavelet decomposition of face patterns

Possible face candidate regions are processed using a two-dimensional filterbank. The regions are first processed row-wise using a 1-D filterbank with a low-pass and high-pass filter pair, $h(\cdot)$ and $g(\cdot)$, respectively. Resulting two image signals are processed column-wise once again using the same filterbank. The high-band sub-images that are obtained using a high-pass filter contain edge information, e.g., the low-high and high-low sub-images in Fig. 2 contain horizontal and vertical edges of the input image, respectively. Therefore, absolute values of low-high, high-low and high-high sub-images can be added to have an image having significant edges of the candidate region. We call this image the *detail* image. Lagrange filterbank³¹ consisting of the low-pass filter $h[n] = \{0.25, 0.5, 0.25\}$, and the high-pass filter $g[n] = \{-0.25, 0.5, -0.25\}$ is used in this paper.

A second approach is to use a 2-D low-pass filter and subtract the low-pass filtered image from the original image. The resulting image also contains the edge information of the original image and it is equivalent to the sum of undecimated low-high, high-low, and high-high sub-images.

2.3 Feature extraction

In this paper, the edge projections of the candidate image regions are used as features. Edge information of the original image is available obtained using the wavelet analysis. The components of the feature vector are the horizontal, vertical, and filter-like projections of the wavelet sub-images. The advantage of the 1-D projection signals is that they can be easily normalized to a fixed size and this provides robustness against scale changes.

Horizontal projection $H(\cdot)$, and vertical projection $V(\cdot)$ are simply computed by summing the absolute pixel values, $d(\cdot, \cdot)$, of the *detail* image in a row and column, respectively as follows,

$$\begin{aligned}
 H(y) &= \sum_x |d(x, y)| \\
 V(x) &= \sum_y |d(x, y)|
 \end{aligned}
 \tag{1}$$

In this way, we take the advantage of the detail (high frequency) information within local facial areas, e.g., eyes, nose, and mouth, in both horizontal and vertical directions. These two projections actually provide us significant discrimination ability for classification. Typical projection vectors after low-pass smoothing are shown in Fig. 4.

Furthermore, filter-like projections, $F_i(\cdot)$, are computed similar to Viola and Jones¹⁷ approach as additional feature vectors. We divide the detail image into two regions, R_1 and R_2 , as shown in Fig. 3(a), and compute projections in these regions. We subtract the horizontal projections in R_1 and R_2 , and obtain a new horizontal projection vector $F_1(y)$. In this way, the symmetry property of a typical human face is considered as a feature also.

$$F_1(y) = \left| \sum_{x \in R_1} |d(x, y)| - \sum_{x \in R_2} |d(x, y)| \right| \quad (2.a)$$

Because of the symmetry property of a face pattern, especially vertical-cut filter-like projections are very close to zero. Similarly, a new vertical projection vector $F_2(x)$ is computed as follows,

$$F_2(x) = \left| \sum_{y \in R_1} |d(x, y)| - \sum_{y \in R_2} |d(x, y)| \right| \quad (2.b)$$

Projection vectors are concatenated to obtain a composite feature vector. A composite feature vector consisting of the projections $H(\cdot)$, $V(\cdot)$, $F_1(\cdot)$ and $F_2(\cdot)$ are used to represent a given image region.

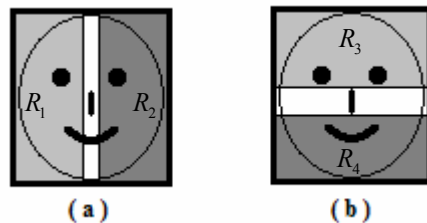


Figure 3: Two-rectangle feature regions. White area between the feature regions is a “dead-zone” in which no summation is carried out, (a) vertical-cut, (b) horizontal-cut of face candidate regions.

We also repeat this process for regions R_3 and R_4 shown in Fig. 3(b) and obtain additional feature vectors.

3. CLASSIFICATION METHODS

After generating feature vectors, the face detection problem is reduced to a classification problem. The extracted feature vectors are then classified using dynamic programming and support vector machine based classifiers into face or non-face classes.

The first method that we used as a classifier is dynamic programming, which is used for measuring a distance metric between a probe feature vector and a typical prototype feature vector. The classification is applied by thresholding the resulting distance. The main reason that we use dynamic programming is that it produces better results than neural networks and HMM in small vocabulary speech recognition.

The second method used is a support vector machine based classifier. An important benefit of the support vector machine approach is that the complexity of the resulting classifier is characterized by the number of support vectors rather than the dimensionality of the transformed space³². Thus, SVMs compensate the problems of overfitting unlike some other classification methods.

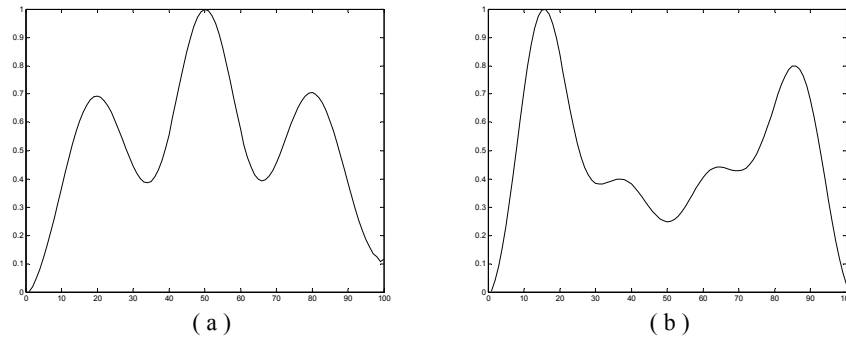


Figure 4: A typical human face (a) horizontal , (b) vertical projection. In horizontal projection, the first peak corresponds to the left eye, the second peak corresponds to nose and mouth, and the third peak corresponds to the right eye, respectively.

3.1 Dynamic programming (DP)

Dynamic programming is an extensively studied and widely used tool in operations research for solving sequential decision problems in finite vocabulary speech recognition¹⁸ and various communication theory applications, e.g., the Viterbi algorithm. Dynamic programming technique generally used in dynamic time-warping (DTW) algorithm for computing the best possible match path between a probe feature vector and a prototype feature vector, and the associated distortion between them. After determining a prototype (template) feature vector for a typical human face, similarity between template and probe feature vectors is determined by aligning them with distortion (Fig. 5(a)). Then, the decision rule classifies the probe feature vector with smallest alignment distortion. Similarity measure is based on the *Euclidean* distance metric as follows,

$$D(v_1, v_2) = \sqrt{\sum_i (v_{1i} - v_{2i})^2} \quad (3)$$

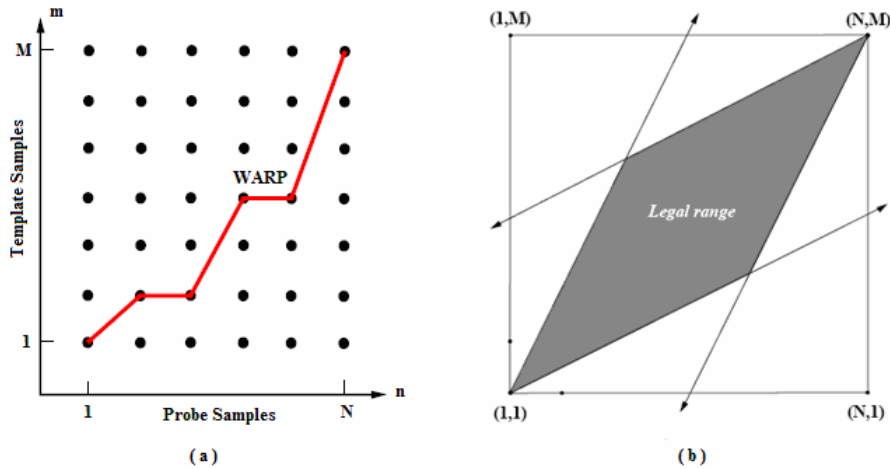


Figure 5: (a) An alignment example, (b) global constraints, range of allowable area for dynamic programming implementation.

There are two types of constraints on distance measure of dynamic programming: local and global. Local constraints are based on the computational difference between a feature of one signal and a feature of the other. Local constraints determine the alignment flexibility (Fig. 6(b)). Global constraints are based on the overall computational difference between an entire signal and another signal of possibly different length. Global constraints exclude portions of search space (Fig 5(b)).

After evaluating the minimum alignment *Euclidean* distance between the probe feature vector and prototype feature vector in dynamic programming, a threshold value is used as a decision rule for classification of the probe feature vector. The threshold value used in our simulations is determined experimentally.

There are several weaknesses of the DP algorithm. It has a high computational cost, i.e., it is not particularly fast. A distance metric must be defined, which may be difficult with different channels with distinct characteristics. Creation of the template vectors from data is non-trivial and typically is accomplished by pair-wise warping of training instances. Alternatively, all observed instances are stored as templates, but this is incredibly slow.

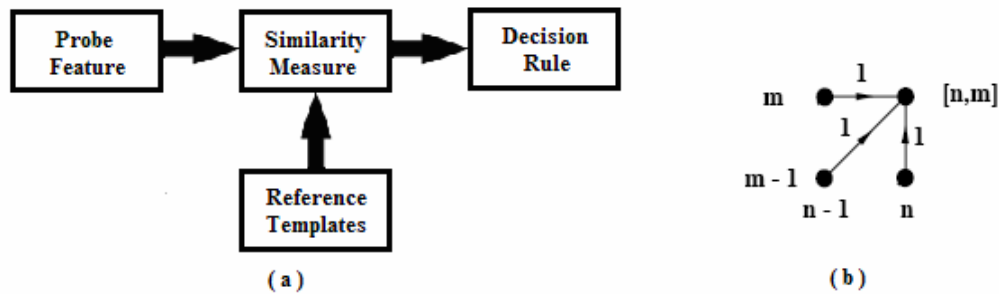


Figure 6: (a) Dynamic programming block diagram, (b) local constraints and slope weights used in our simulations.

3.2 Support vector machines (SVMs)

Support vector machines seek to define a linear boundary between classes such that the margin of separation between samples from different classes that lie next to each other is maximized. Classification by SVMs is concerned only with data from each class near the decision boundary, called *support vectors*. Support vectors lie on the margin and carry all the relevant information about the classification problem. They are informally the hardest patterns to classify, and the most informative ones for designing the classifier³².

This approach is generalized to non-linear case by mapping the original feature space into some other space using a mapping function and performing optimal hyperplane algorithm in this dimensionally increased space. In the original feature space, the hyperplane corresponds to a non-linear decision function whose form is determined by the mapping kernel. Mapping kernels have been developed to compute the boundary as a polynomial, sigmoid, and radial basis function (RBF).

In this paper, we used a library for support vector machines called LIBSVM³³ which is available online for free of charge in either C++ or Java, with interfaces for Matlab, Perl and Python. Our simulations are carried out in C++ with interface for Python using radial basis function (RBF) as kernel with default parameter selection. RBF kernels are computed as follows,

$$k(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}} \quad (4)$$

LIBSVM package provides the necessary quadratic programming routines to carry out classification. It also normalizes each feature by linearly scaling it to the range $[-1, +1]$, and performs cross validation on the training set.

Support vector machines are used for isolated handwritten digit detection, object recognition, and also face detection by several researchers.

4. EXPERIMENTAL RESULTS AND CONCLUSIONS

The proposed algorithm in this paper was evaluated on several face image databases including the Computer Vision Laboratory (CVL) Face Database³⁴. The database contains 797 color images of 114 persons. Each person has 7 different images of size 640x480 pixels; far left side view, 45° angle side view, serious expression frontal view, 135° angle side view, far right side view, smile -showing no teeth- frontal view, and smile -showing teeth- frontal view. We extracted

335 frontal view face dataset from this database by cropping faces of variable size. Furthermore, 100 non-face samples are extracted from color images downloaded randomly from the World Wide Web. These non-face samples include skin colored objects and typical human skin regions such as hands, arms, and legs. The best success rate achieved using dynamic programming is 95.8% over whole face and non-face datasets with concatenation of horizontal, vertical, and vertical-cut filter-like projections as feature vectors. After generating 11 face template vectors, each probe vector is compared to these face templates, and an experimentally determined threshold value is used in dynamic programming. Then, the classification is carried out using majority voting technique. SVMs also applied to this datasets with same feature vectors, and the best success rate of 99.6% is achieved. While training SVMs, 100 face samples and 50 non-face samples used, and then these are also included in test set.

The second experimental setup consists of a real-time human face dataset for real-time implementation. This dataset is collected in our laboratory. The dataset currently contains the video of 12 different people with 30 frames each. A person's face is recorded from 45° side view to 135° side view from different distances to camera with a neutral facial expression under the day-light illumination. Then, SVMs are trained with these data and the resulting modal file of LIBSVM is used for classifying the test features in real-time. The proposed human face detection system is implemented in .NET C++ environment, and it works in real-time with 15 fps on a standard personal computer. We used a Philips web camera with output resolution of 320x240 pixels throughout all our real-time experiments.

The third database that we used in our experiments is the Facial Recognition Technology (FERET) Face Database. We used the same dataset used by Uzunov et al.¹⁵ (Fig. 7). This dataset contains 10,556 gray-scale images of size 32x32 pixels face and non-face samples. There are 3156 face samples where each instance has a single sample of frontal upright human face. These images were collected from the FERET Face Database by Uzunov et al.¹⁵, including human faces from all races with different face expressions, some wearing glasses, having beard and/or mustaches. The non-face dataset contains 7400 samples of random sampling images of size 32x32 of indoor or outdoor scenes which are collected randomly from the World Wide Web. The success rate achieved using a variable threshold on edge images is 99.9%. Our detection rate is better than the best Haar wavelet packet dictionary test results on this dataset which is 99.74% with 150 atoms. Symmlet-2, Symmlet-3, and Symmlet-5 wavelet test results for 150 atoms are 99.25%, 99.51%, and 99.47%, respectively.



Figure 7: Examples of FERET frontal upright face images¹⁵. Wavelet domain sub-images are used for training and testing.

We also tried an AdaBoost classifier on the edge images with the same feature vectors, and achieved 94.9% success rate.

In this paper, a human face detection algorithm is proposed. A set of detailed experiments in both real-time and well-known datasets are carried out. Our experimental results indicate that support vector machines work better than dynamic programming based classifiers and AdaBoost classifiers for our feature extraction method.

REFERENCES

1. H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 23-38, Jan. 1998.
2. K. K. Sung, and T. Poggio, "Example-based learning for view-based human face detection", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 39-51, Jan. 1998.
3. E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: an application to face detection", *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR '97)*, pp. 130-136, 1997.
4. G. Guo, S. Z. Li, and K. Chan, "Face recognition by support vector machines", *Fourth IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pp. 196-201, 2000.
5. A. V. Nefian, and M. H. Hayes III, "Face detection and recognition using hidden markov models", *IEEE Int. Conf. on Image Processing (ICIP '98)*, vol. 1, pp. 141-145, 1998.
6. F. Samaria, and S. Young, "HMM based architecture for face identification", *Image and Computer Vision*, vol. 12, pp. 537-583, Oct. 1994.
7. P. N. Belhumeur, J. P. Hespanha, and D. J. Krigman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 711-720, July 1997.
8. M. A. Turk, and A. P. Pentland, "Face recognition using eigenfaces", *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR '91)*, pp. 586-591, 1991.
9. E. Hjelm, and B. K. Low, "Face detection: a survey", *Computer Vision and Image Understanding* 83, pp. 236-274, 2001.
10. W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: a literature survey", *ACM Computing Surveys*, vol. 35, pp. 399-458, Dec. 2003.
11. C. Garcia and G. Tziritas, "Face detection using quantized skin color regions merging and wavelet packet analysis", *IEEE Trans. on Multimedia*, vol. 1, pp. 264-277, Sept. 1999.
12. I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis", *IEEE Trans. on Information Theory*, vol. 36, pp. 961-1005, May 1990.
13. S. Mallat, "A theory of multi-resolution signal decomposition: the wavelet representation", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674-693, 1989.
14. Y. Zhu, S. Schwartz, and M. Orchard, "Fast face detection using subspace discriminant wavelet features", *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 636-642, June 13-15. 2000.
15. V. Uzunov, A. Gotchev, K. Egiazarian, and J. Astola, "Face detection by optimal atomic decomposition", *SPIE Symposium on Optics and Photonics*, 2005.
16. Y. Freund, and R. Schapire, "A decision-theoretic generalization of online learning and an application to boosting", *Journal of Computer and System Sciences*, vol. 55, pp. 119-139, 1997.
17. P. Viola, and M. Jones, "Rapid object detection using a boosted cascade of simple features", *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR 2001)*, vol. 1, pp. 511-518, 2001.
18. L. Rabiner, and B. H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, Inc., NJ, 1993.
19. B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifier", *Fifth ACM Workshop on Computational Learning Theory*, pp. 144-152, July 1992.
20. C. Cortes, and V. Vapnik, "Support vector networks", *Machine Learning*, 20:1-25, 1995.
21. V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, 1995.
22. J. Kovac, P. Peer, and F. Solina, "Human skin colour clustering for face detection", *The IEEE Region 8 Computer as a tool EUROCON 2003*, vol. 2, pp. 144-148, Sept. 22-24. 2003.
23. M. J. Jones, and J. M. Rehg, "Statistical color models with application to skin detection", *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, vol. 1, p. 280, 1999.
24. J. C. Terrillon, M. N. Shirazi, H. Fukamachi, and S. Akamatsu, "Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images", *Fourth IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pp. 54-61, 2000.
25. H. Greenspan, J. Goldberger, and I. Eshet, "Mixture model for face-color modeling and segmentation", *Pattern Recognition Letters* 22, 2001.

26. S. L. Phung, A. Bouzerdoun, and D. Chai, "Skin segmentation using color pixel classification: analysis and comparison", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 148-154, Jan. 2005.
27. H. Kruppa, M. A. Bauer, and B. Schiele, "Skin patch detection in real-world images", *Lecture Notes in Computer Science*, Springer, 2002.
28. B. Jedydak, H. Zheng, M. Daoudi, and D. Barret, "Maximum entropy models for skin detection", *Lecture Notes in Computer Science*, 2003.
29. M. H. Yang, and N. Ahuja, "Detecting human faces in color images", *Int. Conf. on Image Processing (ICIP'98)*, vol. 1, pp. 127-130, 1998.
30. V. Vezhnevets, V. Sazonov, and A. Andreeva, "A survey on pixel-based skin color detection techniques", *Graphicon-2003*, Moscow, Russia, Sept. 2003.
31. C. W. Kim, R. Ansari, and A. E. Cetin, "A class of linear -phase regular biorthogonal wavelets", *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 673-676, 1992.
32. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley & Sons, Inc., Canada, 2000.
33. C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines", 2002, <http://www.csie.ntu.edu.tw>
34. CVL Face Database, Peter Peer, <http://www.lrv.fri.uni-lj.si/>