Proceedings of
ICISIP - 2005

Combining Multiple Evidence for Video Classification

Vakkalanka Suresh, C. Krishna Mohan, R. Kumaraswamy and B. Yegnanarayana
Speech and Vision Laboratory
Department of Computer Science and Engineering
Indian Institute of Technology Madras, Chennai-600 036, India.
Email:{suresh, ckm, kswamy, yegna}@cs.iitm.ernet.in

Abstract

In this paper, we investigate the problem of video classification into predefined genre, by combining the evidence from multiple classifiers. It is well known in the pattern recognition community that the accuracy of classification obtained by combining decisions made by independent classifiers can be substantially higher than the accuracy of the individual classifiers. The conventional method for combining individual classifiers weighs each classifier equally (sum or vote rule fusion). In this paper, we study a method that estimates the performances of the individual classifiers and combines the individual classifiers by weighing them according to their estimated performance. We demonstrate the efficacy of the performance based fusion method by applying it to classification of short video clips (20 seconds) into six popular TV broadcast genre, namely cartoon, commercial, news, cricket, football, and tennis. The individual classifiers are trained using different spatial and temporal features derived from the video sequences, and two different classifier methodologies, namely Hidden Markov Models (HMMs) and Support Vector Machines (SVMs). The experiments were carried out on more than 3 hours of video data. A classification rate of 93.12% for all the six classes and 97.14% for sports category alone has been achieved, which is significantly higher than the performance of the individual classifiers.

1 Introduction

Content-based video classification deals with the problem of categorizing a given video sequence into one of certain predefined video genre. With the availability of large digital video libraries, it is desirable to classify and categorize video content automatically, so that end users can search, and chose or verify a desired program based on the semantic content thereof. Also, efficient indexing of video data can be achieved, first by categorizing the video, and then employing the domain specific knowledge of the video genre.

There are many approaches to content-based classification of video data. At the highest level of hierarchy, video collections can be categorized into different program genres such as cartoon, sports, commercials, news and music. In a recent approach [1], Li et al used PCA to reduce the dimensionality of the features (low-level audio and visual) of video, and they used Gaussian Mixture Model (GMM) based classifier models. In an another approach [2], Truong et al used semantic aspects of a video genre such as editing, motion and color features and C4.5 decision tree algorithm to build the classifier.

At the next level of hierarchy, domain videos such as sports can be classified into different sub-categories. In [3], Xavier et al classify sports video into four sub categories (ice hockey, basketball, football and soccer) by using motion and color features and HMM based classifier models. In an another approach [4], by using the statistical analysis of camera motion patterns such as fix, pan, zoom and shake, sports videos are categorized into sumo, tennis, baseball, soccer and football.

At a finer level, a video sequence itself can be segmented, and each segment can then be classified according to its semantic content. In [5], sports video segments are first segmented into shots, and each shot is then classified into playing field, player, graphic, audience and studio shot categories. Parsing and indexing of news video [6] and semantic classification of basketball segments into goal, foul and crowd categories [7] by using edge-based features are some of the other works carried out at this level.

This paper is an extension to our earlier work [8], where we addressed the problem of video classification using spatial and temporal features, and support vector machines. This paper addresses the problem of combining evidence from multiple classifiers for video classification. It has been shown in the literature [9, 10, 11, 12] that the combination of several complementary classifiers will improve the performance of individual classifiers. There are at least two reasons justifying the necessity of combining multiple classifiers:

1. For any pattern recognition application, there are a number of classification algorithms developed from

0-7803-8840-2/05/$20.00 ©2005 IEEE

different theories and methodologies. For a specific application problem, each of these classifiers could reach a certain degree of success, but may be none of them is good enough to be employed in practice.

2. Often there are numerous types of features which could be used to represent and recognize patterns. These features are also represented in very diversified forms, and it is hard to lump them together for one single classifier to make the decision.

Given a classification task, there are numerous types of features that can be extracted from the same raw data. Based on each of these features, a classifier or several different classifiers can be trained for the same classification task. As a result, we need schemes to combine the results from these classifiers to produce an improved result for the classification task.

The output information from various classification algorithms can be categorized into three levels:

1. **Abstract level:** Classifier outputs a unique label.

2. **Rank level:** Classifier ranks all labels in a queue with the label at the top being the first choice.

3. **Measurement level:** Classifier attributes to each class a measurement value that reflects the degree of confidence that a specific input belongs to a given class.

Among the three levels, the measurement level contains the highest amount of information, while the abstract level contains the lowest. For this reason, we adopted a well known approach for combining the results of multiple classifiers, at the measurement level. The method is based on the weighted sum of measurements, where the weights are determined by a Bayesian decision rule. In this paper, we address the problem of video genre classification for six classes: cartoon, commercial, news, cricket, football, and tennis. Five different types of spatial and temporal features are extracted from the video data. Two different classifier methodologies, namely Hidden Markov Models (HMMs) and Support Vector Machines (SVMs) are been used to model each feature type.

The rest of this paper is organized as follows: In Section 2, the extraction of spatial and temporal visual features inherent in a video class is described. Section 3 gives a brief introduction to the classifier methodologies. Section 4 describes the weighted Bayesian multi-classifier fusion technique based on the individual classifier performance models. Section 5 describes experiments on video classification on six TV genre, and discusses the performance of the system. Section 6 summarizes the study.

2 Feature Extraction

A feature is defined as a descriptive parameter that is extracted from an image or a video stream. The effectiveness of any classification scheme depends on the effectiveness of attributes in content representation. We extract five different types of visual features ($F1, F2, F3, F4 \; and \; F5$) based on color, texture and motion in the video. The definitions of these features and their intuitive meanings are discussed in the following subsections.

2.1 Color histogram moments ($F1$)

Color is an important attribute for image representation. Color histogram, which represents the color distribution in an image, is one of the most widely used color feature. Since it is not feasible to consider the complete histogram, for each frame in the video, we have considered the second, third moments (variance, skewness respectively) and the dominant color bin value from each of the three histograms for hue, saturation and value planes in HSV color space.

2.2 Color coherence vector ($F2$)

Color coherence vector is similar to color histogram but also takes spatial information of the pixels into consideration. A color's coherence is defined as the degree to which pixels of that color are members of large similarly-colored regions. Color coherence vectors are computed as described in [13]. The RGB(888) color space is quantized into 64 colors by considering only the 2 most significant bits from each plane. For each frame of the video sequence, color coherence vector is obtained with 64 bins. We have considered only the top ten dominant color bin values as our feature from the coherence vector.

2.3 Edge Features ($F3$ & $F4$)

We have considered two different features that can be derived from edge information, namely, edge direction histogram and edge intensity histogram. Edge direction histogram is one of the standard visual descriptors defined in MPEG-7 for image and video, and provides good representation of the non-homogeneous textured images. This descriptor captures the spatial distribution of edges. A given image is first segmented into 2×2 sub-images. The edge information is then calculated for each sub-image using Canny algorithm. The domain of the edge directions $(0 - 180)$ is divided into 5 bins. Thus, an image partitioned into 4 sub-images results in 20 bins.

Also, we derive a 16 bin edge intensity histogram ($F4$) from the edge information as another type of feature.

2.4 Motion feature ($F5$)

Motion is an important attribute of video. Different video genre present different motion patterns. In this paper, we use a simple and effective technique where motion is extracted by pixel-wise differencing of consecutive frames.. We divide each frame into 5 sub-images (top-left, top-right, bottom-left, bottom-right and center), each of size equal to half the width and height of the original image. The motion information is computed by the equation:

$$M(t) = \frac{1}{w * h} \sum_{x=1}^{w} \sum_{y=1}^{h} P_t(x, y) \quad (1)$$

$$\text{where } P_t(x, y) = \begin{cases} 1 & \text{if } |I_t(x, y) - I_{t-1}(x, y)| > \beta \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $I_t(x, y)$ and $I_{t-1}(x, y)$ are the pixel values at pixel location (x, y) in t^{th} and $(t-1)^{th}$ frames, respectively. β is the threshold, and w and h are width and height of the sub-image, respectively. A 5 dimensional feature is derived from each consecutive frame pair.

3 Classifier Methodologies

There are a number of classifier algorithms reported in the literature for various pattern recognition applications. We have chosen Hidden Markov models(HMMs) and Support Vector Machines (SVMs) for our classification task. Given the temporal nature of video, and HMMs being effective tools for modeling time-varying patterns [14], we have chosen HMM as one of the classifier algorithms for our study. SVMs are well-known for their good generalization performance [15], and have been applied to many pattern recognition problems in recent years. In the next subsections, we will give a brief discussion of the two classifier methodologies.

3.1 Hidden Markov Model (HMM)

A Markov model is a finite state machine which changes the state once every time unit, and each time t that a state q_j is entered, a vector $\mathbf{o}_t$ is generated with a probability density $b_{q_j}(\mathbf{o}_t)$. Furthermore, the transition from state q_i to state q_j is also probabilistic, and is governed by the discrete probability $a_{q_i q_j}$. The joint probability that the observation sequence $\mathbf{O} = (\mathbf{o}_1 \mathbf{o}_2 \ldots \mathbf{o}_T)$ of length T is generated by the model λ moving through the state sequence $\mathbf{q} = (q_1 q_2 \ldots q_T)$ is calculated as the product of the transition probabilities and the output probabilities. In practice, only the observation sequence $\mathbf{O}$ is known, and the underlying state sequence $\mathbf{q}$ is hidden. Hence, it is called a hidden Markov model.

Given that the state sequence $\mathbf{q}$ is unknown, the probability of $\mathbf{O}$ (given the model) is obtained by summing the joint probability over all possible state sequence $\mathbf{q}$ as follows:

$$P(\mathbf{O}|\lambda) = \sum_{q1, q2, \ldots, qT} \pi_{q_1} b_{q_1}(\mathbf{o}_1) a_{q_1 q_2} b_{q_2}(\mathbf{o}_2) \ldots a_{q_{T-1} q_T} b_{q_T}(\mathbf{o}_T) \quad (3)$$

where π_{q_1} is the initial state probability. The parameters $\{a_{q_i q_j}\}$ and $\{b_{q_j}(\mathbf{o}_t)\}$ are known for each model λ.

The following steps are implemented for the video classification in the HMM framework.

1. Initialization of models.
2. Baum-Welch Re-estimation of the models, for determining the parameters of HMM. Here independent model for each video genre is built. The model is re estimated until there is no change in the state transition probabilities.
3. Testing of models: This involves testing video clips against the model built.

3.2 Support Vector Machine Model (SVM)

Support vector machines [16] for pattern classification are built by mapping the input patterns into a higher dimensional feature space using a nonlinear transformation (kernel function), and then optimal hyperplanes are built in the feature space as decision surfaces between classes. Nonlinear transformation of input patterns should be such that the pattern classes are linearly separable in the feature space. According to Cover's theorem, nonlinearly separable patterns in a multidimensional space, when transformed into a new feature space are likely to be linearly separable with high probability, provided the transformation is nonlinear, and the dimension of the feature space is high enough [17]. The separation between the hyperplane and the closest data point is called the margin of separation, and the goal of a support vector machine is to find a optimal hyperplane for which the margin of separation is maximized. Fig. 1 illustrates the geometric construction of a hyperplane for two dimension input space. The support vectors constitute a small subset of the training data that lie closest to the decision surface, and are therefore the most difficult to classify.

The separating hyperplane is defined as a linear function of the vectors drawn from the feature space. Construction of this hyperplane is performed in accordance with the principle of structural risk minimization that is rooted in Vapnik-Chervonenkis (VC) dimension theory [17]. By using an optimal separating hyperplane the VC dimension is minimized and generalization is achieved. The number of examples needed to learn a class of interest reliably is proportional to the VC dimension of that class. Thus, in order to have a less

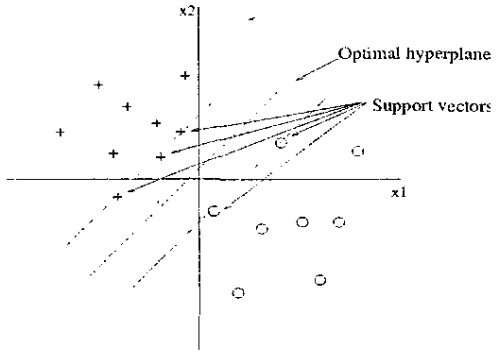

Figure 1: Illustration of the idea of support vectors and an optimal hyperplane for linearly separable patterns.

complex classification system, it is preferable to have those features which lead to lesser number of support vectors.

The optimal hyperplane is defined by:

$$\sum_{i=1}^{N_S} \alpha_i d_i K(\mathbf{x}, \mathbf{x}_i) = 0 \qquad (4)$$

where N_S is the total # of support vectors which lie on either side of the hyperplane, $\{\alpha_i\}_{i=1}^{N_S}$ are the Lagrange multipliers, $\{d_i\}_{i=1}^{N_S}$ are the desired classes and $K(\mathbf{x}, \mathbf{x}_i)$ is the inner-product kernel, and is defined by:

$$\begin{aligned} K(\mathbf{x}, \mathbf{x}_i) &= \varphi^T(\mathbf{x})\varphi(\mathbf{x}_i) \\ &= \sum_{j=0}^{m_1} \varphi_j(\mathbf{x})\varphi_j(\mathbf{x}_i), \ i = 1, 2, \ldots, N_S \end{aligned} \qquad (5)$$

where $\mathbf{x}$ is a vector of dimension m_0 drawn from the input space, and $\{\varphi_j(\mathbf{x})\}_{j=1}^{m_1}$ denotes a set of nonlinear transformations from the input space to the feature space. $\varphi_0(\mathbf{x}) = 1$, for all $\mathbf{x}$. m_1 is the dimension of the feature space. From (4) it is seen that the construction of the optimal hyperplane is based on the evaluation of an inner-product kernel. The inner-product kernel $K(\mathbf{x}, \mathbf{x}_i)$ is used to construct the optimal hyperplane in the feature space without having to consider the feature space itself in explicit form. The design of a support vector machine involves finding an optimal hyperplane. In order to find an optimal hyperplane, it is necessary to find the optimal Lagrange multipliers which are obtained from the given training samples $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N_S}$.

The performance of the pattern classification problem depends on the type of kernel function chosen. Possible choices of kernel function include; polynomial, Gaussian and sigmoidal. In this work, we have used Gaussian kernel, since it was empirically observed to perform better than the other two. SVMs are originally designed for two class classification problems. In our work, multi-class ($M = 6$) classification task is achieved using one-against-rest approach, where an SVM is constructed for each class by discriminating that class against the remaining $(M - 1)$ classes.

4 Combining Evidence

4.1 A classifier

Consider a classifier that distinguishes L different classes, or labels, in a label set $\wedge = \{1, \ldots, L\}$. Any pattern classification application involves two important stages: Training and testing. From a machine learning perspective, the process of learning the patterns involved in the training data for various classes is known as training the classifier. The set of all samples that truly belong to class i is denoted by C_i, so the a priori ground truth "x is truly in class i" is written as

$$x \in C_i \qquad (6)$$

When a test sample x is given to the k^{th} classifier, it outputs measurement/confidence values

$$P_k(x \in C_i \mid x), \ i = 1, \ldots, L \ \text{and} \ k = 1, \ldots, K \qquad (7)$$

to express, what is the probability that the given sample belongs to class i. When the k^{th} classifier assigns sample x to class j, it is written as

$$h_k(x) = j, \ \text{where} \ j = \arg\max_{i \in \wedge} P_k(x \in C_i \mid x) \qquad (8)$$

The goal now is to combine the evidence from all the K classifiers, and to come up with a final measurement/confidence value

$$P_H(x \in C_i \mid x) = \frac{1}{K} \sum_{k=1}^{K} W_k \times P_k(x \in C_i \mid x), \ i = 1, \ldots, L \qquad (9)$$

where W_k is the weight assigned to the k^{th} classifier based on its estimated performance, and a final hypothesis is made based on

$$H(x) = j, \ \text{where} \ j = \arg\max_{i \in \wedge} P_H(x \in C_i \mid x) \qquad (10)$$

4.2 Classifier performance estimation

The performance of a classifier can be described by the probabilistic dependencies between its decisions and actual data memberships, written as conditional probabilities such as

$$P(h_k(x) = j \mid x \in C_i) \qquad (11)$$

to express, for example, the probability that a classifier k assigns a sample x to class j, when in fact x belongs to class i.

For a given test set of samples with known classifications, the classification behavior of the classifier k can be

expressed by its confusion matrix N_k. The number of rows and columns of this matrix equals the number of classes L. Each row corresponds to one class that a sample can be in. Each column represents a classifier decision. The entries of N_k are the co-occurrences of classifier decisions and actual class memberships.

$$n_{k,i,j} = \#\{x | x \in C_i, h_k(x) = j\} \tag{12}$$

In other words, each of the entries $n_{k,i,j}$ of N_k is the number of samples x from class i which classifier k assigned to class j. Given these values, one can estimate the conditional probabilities that describe the classification behavior of a classifier. From its confusion matrix, the probability that a sample x from class i is classified by classifier k as belonging to class j is estimated as

$$\begin{aligned} P(h_k(x) = j \mid x \in C_i, N_k) &= \frac{\#\{x \mid x \in C_i \wedge h_k(x) = j\}}{\#\{x \mid x \in C_i\}} \\ &= \frac{n_{k,i,j}}{n_{k,i,*}} \end{aligned} \tag{13}$$

where the denominator is the row sum of the i^{th} row of N_k, i.e.,

$$n_{k,i,*} = \sum_{j=1}^{L} n_{k,i,j} \tag{14}$$

For computational purpose, we estimate the average performance of each classifier as

$$A_k = \sum_{i=1}^{L} P(h_k(x) = i \mid x \in C_i, N_k), \;\; k = 1, \ldots, K \tag{15}$$

where $P(h_k(x) = i \mid x \in C_i, N_k)$ represents, the probability that a sample x from class i is classified by classifier k as belonging to class i, and is obtained from equation(13).

4.3 Multi-classifier decision fusion

For combining the classification results on the same x by all K classifiers, a simple approach generally followed is to use the average measurement value as an estimation of combined classifier H:

$$P_H(x \in C_i \mid x) = \frac{1}{K} \sum_{k=1}^{K} P_k(x \in C_i \mid x), \;\; i = 1, \ldots, L \tag{16}$$

The combined classifier H with the newly estimated post-probabilities is called an averaged Bayes classifier. The disadvantage with this approach is that, it gives equal preference to all the classifiers. This will be useful only when all the classifiers perform equally well. But, this is not the case with most of the pattern recognition applications. So we propose a weighted Bayesian decision classifier, where the individual classifiers are weighed based on their performance.

The weights are derived from the average classifier performance values given by equation (15) as:

$$W_k = \frac{A_k}{\sum_{j=1}^{K} A_j}, \;\; k = 1, \ldots, K. \tag{17}$$

Now, the weighted Bayesian estimation values of the combined classifier H are obtained as

$$P_H(x \in C_i \mid x) = \frac{1}{K} \sum_{k=1}^{K} W_k \times P_k(x \in C_i \mid x), \;\; i = 1, \ldots, L \tag{18}$$

5 Experimental Results

The experiments were carried out on more than 3 hours of video data ($\approx$ 600 video clips each of 20 seconds captured at 25 frames per second) comprising of cartoon (Ca), commercial (Co), news (Ne), cricket (Cr), football (Fo) and tennis (Te) video categories. The data was collected from different TV channels on different dates and at different times to ensure the variety of data. For each video genre, 40 clips were used for training, 20 clips were used to estimate the performance of the individual classifiers, and the remaining 40 were used for final testing. A 5 state HMM, and a Gaussian kernel based SVM were constructed for each feature type and for each class.

During testing phase, the HMMs will output the log probabilities, given a test clip, representing the a posteriori probability that this clip belongs to a particular class. Given a pattern vector to an SVM model, the result will be a measure of the distance of the pattern vector from the hyper plane constructed as a decision boundary between this class and rest of the classes. The performance of HMM and SVM based classifiers computed as described in Section 4, for each of the five types of features is given in Table 1. For concatenated features ($F1$ to $F5$), the HMM and SVM classifiers resulted in 79% and 85% classification accuracy, respectively. The performances of the average Bayesian HMM classifier and SVM classifier are 80.5% and 90.3%, respectively.

Table 1: Performance of HMM and SVM classifier models for each of the five types of features (in %).

	F1	F2	F3	F4	F5
HMM	57.49	41	77.33	62.35	36.84
SVM	76.11	45	83.8	72.5	57.1

As explained in Section 2, the individual classifiers are weighted and outputs are combined. The weighted combine performance of HMM classifier is 81.78% and of SVM classifier is 91.5%. The measurement values from the two final weighted classifier are normalized to the range 0 to 1. Finally these normalized measurement values from the weighted HMM and weighted SVM are combined, and a classification performance of **93.12%** for all the six classes and a performance of **97.14%** for sports category alone, has been achieved. The confusion matrix for the final classifier (combined HMM and SVM) is given in Table 2.

Table 2: Confusion matrix of video classification results using combined HMM and SVM classifier (in %).

	Ca	Co	Ne	Cr	Fo	Te
Ca	77.42	19.35	0	3.23	0	0
Co	0	100		0	0	0
Ne	0	10	82.5	7.5	0	0
Cr	0	0	0	97.44	2.56	0
Fo	0	2.63	0	0	97.37	0
Te	0	0		0	3.57	96.43

6 Conclusion

We have presented a novel approach to combine evidence from multiple classifiers for video classification based on spatial and temporal features, and Hidden Markov Models and Support Vector Machine models. A video database of TV broadcast programs containing six popular genre namely cartoon, commercial, news, cricket, football and tennis was used for training and testing the models. A correct classification rate of 93.12% percent was achieved. Experimental results indicate that the combined classifier out-performs the individual classifiers, classifiers trained with concatenated features, and average Bayesian classifier. However, in order to achieve better classification performance, evidence from visual features alone may not be sufficient. Evidence from other modalities in a video like audio and text need to be combined with the visual evidence, which will be our future effort.

References

[1] L.-Q. Xu and Y. Li, "Video classification using spatial-temporal features and PCA," in *Int. Conf. Multimedia and Expo*, (Baltimore, MD, USA), pp. 345-348, July 6-9, 2003.

[2] B. T. Truong, S. Venkatesh, and C. Dorai, "Automatic genre identification for content-based video categorization," in *Proc. of Int. conf. Pattern Recognition*, (Barcelona, Spain), vol. 4, pp. 230-233, Sep. 3-8, 2000.

[3] X. Gibert, H. Li, and D. Doermann, "Sports video categorizing using hmm," in *Int. Conf. Multimedia and Expo*, (Baltimore, MD, USA), pp. 345-348, July 6-9, 2003.

[4] S. Takagi, S. Hattori, K. Yokoyama, A. Kodate, and H. Tominaga, "Sports video categorizing method using camera motion parameters," in *Int. Conf. Multimedia and Expo*, (Baltimore, MD, USA), pp. 461-464, July 6-9, 2003.

[5] J. Assflag, M. Bertini, C. Colombo, and A. D. Bimbo, "Semantic annotation of sports videos," *IEEE Multimedia*, vol. 9, pp. 52–60, June 2002.

[6] V. Suresh, S. Palanivel, and B. Yegnanarayana, "Anchorperson indexing and visual-table-of-contents generation for tv news," in *Porc. Workshop on Biometric Challenges Arising out of Theory and Practice*, (Cambridge, UK), pp, 59-62, Aug 22-25, 2004.

[7] M. H. Lee, S. Nepal, and U. Srinivasan, "Edge-based semantic classification of sports video sequences," in *Int. Conf. Multimedia and Expo*, (Baltimore, MD, USA), pp. 157-160, July 6-9, 2003.

[8] V. Suresh, C. K. Mohan, R. Kumaraswamy, and B. Yegnanarayana, "Content-based video classification using support vector machines," in *Int. Conf. Neural Information Processing*, (to be held in Science City, Calcutta), Nov 22-25, 2004.

[9] J. Kettler, M. Hatef, R. P. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 226–239, Mar. 1998.

[10] T. Rohlfing, D. B. Russakoff, and C. R. Maurer, "Performance-based classifier combination in atlas-based image segmentation using expectation-maximization parameter estimation," *IEEE Trans. on Medical Imaging*, vol. 23, pp. 983–994, Aug. 2004.

[11] L. Lam and C. Y. Suen, "Optimal combinations of pattern classifiers," *Pattern Recognition Letters*, vol. 16, pp. 945–954, Sep. 1995.

[12] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Trans. Sys. Man, Cybern.*, vol. 2, pp. 418–435, May 1992.

[13] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *Proc. ACM Conf. Multimedia*, (Boston, MA, USA), pp. 65-73, Nov. 18-22, 1996.

[14] L. Rabiner and B. Juang, "An introduction to hidden markov models," *IEEE Accoustic Speech Signal Processing Magazine*, vol. 3, pp. 4–16, Jan 1986.

[15] R. Collobert and S. Bengio, "Svmtorch: Support vector machines for large-scale regression problems," *Journal of Machine Learning Research*, vol. 1, pp. 143–160, Spring 2001.

[16] V. Vapnik, *Statistical Learning Theory*. New York: John Wiley and Sons, 1998.

[17] S. Haykin, *Neural Networks - A Comprehensive Foundation*. Prentice Hall, 1999.