FEATURES FOR VIDEO SHOT BOUNDARY DETECTION AND CLASSIFICATION

A THESIS

submitted by

C. KRISHNA MOHAN

for the award of the degree

of

DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS

DECEMBER 2007

THESIS CERTIFICATE

This is to certify that the thesis entitled **Features for Video Shot Boundary Detection and Classification** submitted by **C. Krishna Mohan** to the Indian Institute of Technology, Madras for the award of the degree of Doctor of Philosophy is a bonafide record of research work carried out by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Chennai - 600 036
Date:

Prof. B. Yegnanarayana
Dept. of Computer Science and Engg.

Dr. C. Chandra Sekhar
Dept. of Computer Science and Engg.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Prof. B. Yegnanarayana for providing me with the opportunity to do my research work under his guidance. His emphasis on steady and committed effort has motivated me during the course of the research work. I have immensely benefited from the excellent research environment that he has created and nurtured.

I am grateful to Dr. C. Chandra Sekhar for his guidance and encouragement throughout my research work. His concern and support have been invaluable to me in the completion of my research work.

I am grateful to Prof. Timothy A. Gonsalves and Prof. S. Raman, chairpersons of the department during my stay for providing excellent facilities in the department for conducting research. I sincerely thank my doctoral committee members Dr. V. Kamakoti, Dr. Deepak Khemani, Dr. S. Mohan, and Dr. V. Srinivas Chakravarthy for their interest in evaluating the progress of my research work. I thank Dr.S. Rajendran and Dr. Hema A. Murthy for their support and help during my stay. I thank all faculty and staff members of the Dept. of Computer Science and Engineering and Center for Continuing Education for their timely help.

I sincerely thank Dhananjaya, Guruprasad, Sri Rama Murty, Suresh and Anand for their help in conducting experimental studies during the research work. Their suggestions have helped in refining the content and presentation of this thesis.

I am extremely thankful to my colleagues Suryakanth V Gangashetty, Mahadeva Prasanna, Sreenivasa Rao, Kumaraswamy, Rajesh Hegde, Nayeem, Palanivel, Anil Kumar Sao, Leena Mary, Shahina, Satish, Dileep for their help, cooperation and support. I wish to thank Chaitanya, Vivek, Anitha, Sharma, Chandrakala, Swapna, Veena, Panuku and Sheetal for extending help at different times. I also thank the families of Sreenivasa Rao, Kumaraswamy, Prasanna for making our stay enjoyable at IIT Madras

campus.

I am profoundly grateful to Prof. K. Chidananda Gowda, former Vice-Chancellor, Kuvempu University, who has been the guiding spirit in my academic career. I would like to thank my teachers, grand parents, parents, family members and friends for making me what I am today.

I express my heartfelt appreciation and gratitude to my wife Anitha for her steadfast support, patience and understanding. Our little daughter Gayathri provided me with precious moments of joy which have been welcome distractions from research.

Finally, I thank everyone who helped me directly or indirectly during my stay at IIT Madras.

C. Krishna Mohan

ABSTRACT

Keywords: *Video content analysis; video segmentation; shot boundary detection; color features; early fusion; late fusion; compressed features; video classification; edge-based features; autoassociative neural networks; event detection; hidden Markov model.*

The objective of this research work is to address some issues in segmentation and classification of video, which are two important tasks involved in organization, retrieval and access of digital videos. The goal of video segmentation is to partition a given video sequence into smaller meaningful units, based on temporal changes in the video sequence. The main issues in video segmentation are the *choice of features* that are robust to illumination and camera/object motion, and measure of dissimilarity for detecting temporal discontinuities. In this thesis, we propose methods for shot boundary detection using significant changes in color features and compressed features. We propose a novel technique for shot boundary detection based on the late fusion of evidence obtained from significant changes in color histogram features. We also propose an algorithm for simultaneous detection of abrupt and gradual transitions, on the basis of dissimilarity between two sets of frames separated by a margin that excludes the region around transition. Second order statistics derived from features extracted around the shot boundaries are used for validation. Bidirectional processing of video is explored in order to reduce the number of missed shot boundaries. Finally, decision due to the proposed late fusion is combined with that due to the traditional early fusion, which relies on the extent of overall change in color features for detecting shot boundaries. Since the color histogram features do not represent the spatial distribution of color, we use color coherence vector as a feature. Additionally, the sparseness of distribution of color coherence feature vectors is exploited by nonlinear projection of the feature vectors on to a lower dimension space. This projection is implemented

using autoassociative neural network (AANN) models. Experimental results show that the proposed methods, in combination with color features, can effectively detect both abrupt and gradual transitions, and are less sensitive to the threshold applied to the dissimilarity measure. The compressed features have also been found to be effective for shot boundary detection.

The problem of video classification is addressed in the context of sports video categorization. Key issues involved are the *selection of features* for adequately representing class-specific information, and developing efficient modeling techniques to capture information present in the features. We propose to model class-specific distributions of two edge-based features, namely, edge direction histogram and edge intensity histogram, using autoassociative neural network (AANN) models. The complementary nature of these features is demonstrated by combining evidence from the individual features. Also, combination of evidence due to different classifiers results in an improvement in the performance of classification. We propose a novel method for classification of sports videos based on events detected from each category, using the framework of hidden Markov models. The detected events which denote significant changes in a temporal sequence, can be viewed as features at a higher level. The sequence of events also act as signature for a given class. The classification system is also able to decide whether a given video clip belongs to one of the predefined categories or not.

In summary, this thesis proposes new methods for video segmentation based on combination of early and late fusion of evidence, and a method for simultaneous detection of abrupt and gradual transitions for shot boundary detection. A nonlinear projection of feature vectors from a high dimension sparse feature space to a lower dimension space is proposed, using autoassociative neural network (AANN) models. The thesis also proposes new edge-based features and AANN models for video classification, along with a method for combining evidence from different classifiers. A new method is proposed for classification of sports videos based on events in each sports category using the framework of hidden Markov models.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

VCA	- Video Content Analysis
PCA	- Principal Component Analysis
JPI	- Joint Probability Image
JPPC	- Joint Probability Projection Centroid
AANN	- Autoassociative Neural Network
SVD	- Singular Value Decomposition
ICA	- Independent Component Analysis
CCV	- Color Coherence Vector
NLPCA	- Nonlinear Principal Component Analysis
HMM	- Hidden Markov Model
AVI	- Audio Video Interleave
EDH	- Edge Direction Histogram
EIH	- Edge Intensity Histogram
QBIC	- Query By Image and Video Content
MoCA	- Movie Content Analysis
CONIVAS	- Content-based Image and Video Access System
MARS	- Multimedia Analysis and Retrieval System
ANSES	- Automatic News Summarization Extraction System
GMM	- Gaussian Mixture Model
TV	- Television

CHAPTER 1

INTRODUCTION TO VIDEO CONTENT ANALYSIS

The amount of multimedia data has grown significantly in the past few years. This growth is primarily due to advances in data acquisition, storage, and communication technologies, aided by advances in processing of audio and video signals. Video has played an important role in this growth, more so in terms of its volume. There is a need to organize large collections of digital videos for efficient access and retrieval. Techniques are needed to organize digital videos into compact and meaningful entities, that human beings can relate to. Such a task is known as video content analysis, which refers to understanding the meaning of a video document. The objective of this thesis is to address issues in two important tasks of video content analysis, namely, video segmentation and video classification. Video segmentation involves partitioning a video sequence into several smaller meaningful units, based on temporal discontinuities in the video sequence. Video classification, on the other hand, is the task of categorizing a given video clip into one of the predefined classes.

The need for automatic algorithms for video content analysis is motivated by the large volume of video data. While human beings are adept at deriving meaningful information from video data, it is a challenging problem to automate this task due to our inability to articulate our perceptual ability in the form of an algorithm. Yet a methodical approach is needed to address the problem of video content analysis. In Section I, we briefly describe various tasks involved in automatic video content analysis. This places in perspective, the role of video segmentation and classification in video content analysis. In Section II, we discuss certain issues related to video segmentation and classification that are addressed in this thesis. Section III outlines the organization of the thesis.

1.1 TASKS INVOLVED IN VIDEO CONTENT ANALYSIS

The objective in video content analysis is to develop techniques to automatically parse video, audio, and text to identify meaningful composition structure of video and to extract and represent content attributes of video sequences. A typical video content analysis (VCA) scheme involves five primary tasks: feature extraction, video structure analysis, abstraction, video classification and indexing. A block diagram illustrating these tasks and their interrelationship is shown in Fig. 1.1.

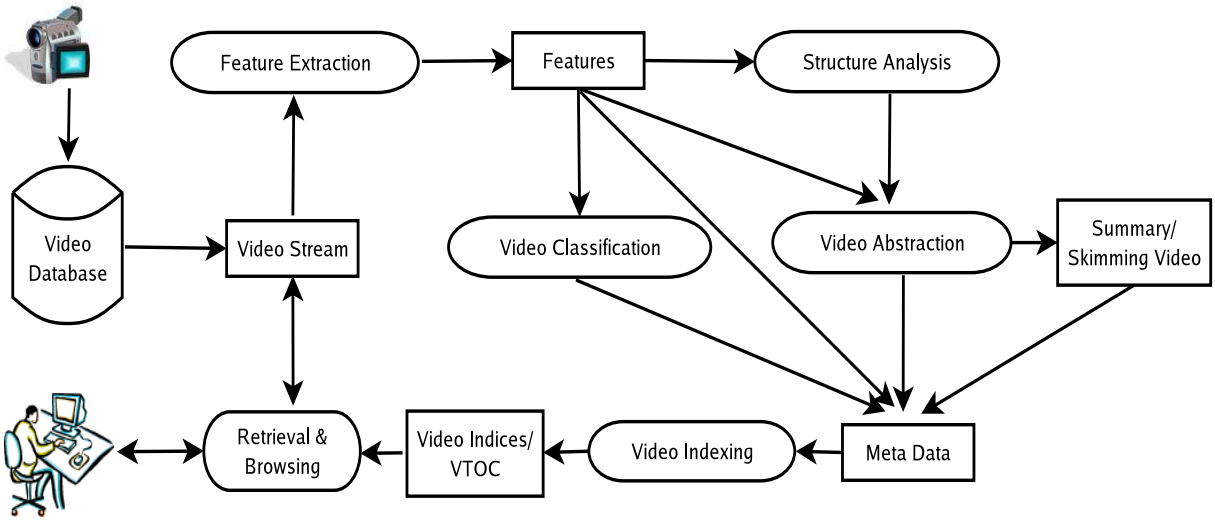


Fig. 1.1: Process diagram for video content analysis.

1.1.1 Feature extraction

A feature is defined as a descriptive parameter that is extracted from an image or a video sequence [1]. The effectiveness of video content analysis depends on the effectiveness of features/attributes used for the representation of the content. Based on the complexity and use of semantics, features can be classified into low-level and high-level features [2].

Low-level features (also known as primitive features) such as color, texture, shape,

object motion (for video), spatial location of image elements (both for image and video), and pitch (for audio) can be extracted automatically. However, these features may not be meaningful from the point of view of human perception. High-level features (also known as logical, semantic features) involve various degrees of semantics depicted in images and video. The features at this level can be objective or subjective. Objective features describe physical objects in images and action in video. Subjective features are concerned with abstract attributes. They describe the meaning and purpose of objects or actions. An event such as a goal in a game of soccer is an example of subjective feature. Interpretation of complex objects or actions, and subjective judgment are required to capture the relationship between video content and abstract concepts.

An important issue is the choice of suitable features for a given task. Effective video content analysis can be achieved by collaboratively using low-level and high-level features. We can use low-level features to segment a video sequence into individual shots and generate representative key frames for each shot. These key frames can then be used for classification and indexing of videos.

1.1.2 Structure analysis

Video structure analysis is the process of extracting temporal and structural information from the video. It involves detection of temporal boundaries and identification of meaningful segments of a video. A video sequence can be viewed as a well organized document and can be parsed into logical units at the following different levels of granularity:

- **Frame level:** A frame represents a single image in a video sequence.
- **Shot level:** A shot is a sequence of frames recorded contiguously from a single camera and representing a continuous action in time or space.
- **Scene level:** A scene is a continuous sequence of shots having a common semantic significance.
- **Sequence/story level:** A sequence/story is composed of a set of scenes.

1.1.2.1 Shot segmentation

Shots are the physical basic layers in video, whose boundaries are determined by editing points or where the camera switches on or off. Shots are analogous to *words* or *sentences* in text documents. The choice of shot as the basic unit for video content indexing provides the basis for constructing a video-table-of-contents. An important issue is the effective detection of different types of shot boundaries.

1.1.2.2 Scene segmentation

The level immediately higher than the shots is called scene. A scene is a continuous sequence of shots having a common semantic significance. The process of detecting video scenes is analogous to paragraphing in parsing of text document and requires a higher level of content analysis. Since a scene is a logical unit, it is often difficult to specify a basis on which a sequence of shots can be grouped together to form a scene.

1.1.2.3 Story segmentation

A story comprises of a set of scenes. Story segmentation needs more semantic understanding of video content. Scenes or stories in video are only logical layers of representation based on subjective semantics, and no universal definition and rigid structure exists for scenes and stories. Hence, grouping a sequence of shots into scene, and grouping a set of scenes into a story require a priori information about the nature of the video program.

1.1.3 Video abstraction

Video abstraction is the process of creating a presentation of the content of a video, which should be much smaller than the original video but which preserves the essential message of the original video. This abstraction process is similar to the extraction of keywords or summaries from text documents. That is, we need to extract a subset

of video data such as key frames or highlights as entries for shots, scenes or stories from the original video. Abstraction is especially important given the vast amount of video data. Video abstraction helps to enable a quick browsing of a large collection of video data and to achieve efficient content representation and access. Combining the structural information extracted from video parsing and key frames extracted during video abstraction, we can build a visual table of contents of a video program.

1.1.4 Video classification

Classification of digital videos into various genres or categories is an important task, and enables efficient cataloging and retrieval with large video collections. Efficient searching and retrieval of video content have become more difficult due to increasing amount of video data. Semantic analysis is a natural way of video classification, since videos of different categories are expected to differ in semantics. However, representation of semantics is a challenging task, since it is not rigidly structured and hence, subjective. Therefore, the problem of video classification is typically addressed through extraction and modeling of low-level features.

1.1.5 Indexing for retrieval and browsing

The structural and content attributes derived during feature extraction, video parsing, abstraction and video classification processes, are often referred to as metadata. Based on this metadata, we can build video indices and table-of-contents. However, a universal solution for video indexing for all video categories is very difficult to achieve. Some of the existing video browsing and retrieval systems are discussed in Appendix A.

1.2 ISSUES ADDRESSED IN THIS THESIS

The previous section briefly described the various issues involved in video content analysis. This research work focuses on features for video segmentation (which is a part of structure analysis of video) and video classification. The problem of shot boundary detection is addressed in video segmentation. This problem deals with the detection of temporal discontinuities in video sequences, where the sequence of frames between two successive discontinuities forms a shot. The key issues are the choice of the features for representation of images, the choice of a similarity/distance metric and an algorithm that is general enough for detection of both abrupt discontinuities and gradual transitions. We address these issues on the basis of significant changes exhibited by a small subset of color features. A novel approach for detection of shot boundaries is proposed based on the late fusion of evidence obtained from the significant changes. We also examine the effect of dimension reduction of feature vectors on the performance of shot boundary detection.

The problem of video classification is addressed in the context of sports videos. Sports videos present a good test case for evaluating algorithms for video classification, since different sports share certain common aspects while retaining their individual identities. An important issue is the representation of video frames, so that resultant features adequately capture class-specific information. Here, edge-based features, namely, edge direction histogram and edge intensity histogram are examined, since different sports are distinctly characterized by edge information. Another issue is the development of effective modeling techniques to capture the information present in the features. These models can either be based on the estimation of probability density function of feature vectors, or based on the estimation of temporal information present in the sequence of feature vectors. Our approach to this problem is twofold. Firstly, the use of autoassociative neural network models is motivated by their ability to capture the density of feature vectors without making assumptions about the shape of the density function. The second approach to video classification is based on the

notion of events in different sports categories. The events denote significant changes in video sequences, and can be viewed as features for representation of class-specific information. The events are not predefined, but instead, they are hypothesized from the changes inherent in the video sequence, using a framework based on hidden Markov models. The hypothesized events are then used to classify a given sports video. The algorithms for video segmentation and classification are verified offline, using video data collected from broadcast channels.

1.3 ORGANIZATION OF THE THESIS

An overview of the existing approaches to video segmentation and classification is presented in Chapter 2. Some research issues are identified in both these tasks which are addressed in this thesis. In Chapter 3, a novel technique called late fusion is proposed for detecting shot boundaries in video sequences. The basis for this method is the significant change exhibited by a few color components over a sequence of frames. A one-pass algorithm for simultaneous detection of abrupt and gradual transitions is also proposed. The sparsity of distribution of color features presents a case for dimension reduction of feature vectors. In Chapter 4, shot boundary detection is performed using feature vectors with reduced dimension. A nonlinear projection of feature vectors from a high dimension sparse feature space to a lower dimension space is performed using autoassociative neural network (AANN) models. In Chapter 5, the problem of video classification is addressed by estimating class-specific densities of edge-based features, using AANN models which are nonparametric. A new method for classification of sports videos based on events in each sports category is proposed in Chapter 6, using the framework of hidden Markov models. Chapter 7 summarizes the research work carried out as part of this thesis, highlights the contributions of the work and discusses directions for future work.

CHAPTER 2

OVERVIEW OF APPROACHES FOR VIDEO SEGMENTATION AND CLASSIFICATION

This chapter reviews some of the existing approaches to video segmentation and video classification. The problem of shot boundary detection is briefly described in Section 2.1. The three important components of algorithms for shot boundary detection, namely, features for representation of video frames, similarity/distance metric, and the algorithm for change detection, are discussed in terms of the commonly made choices for these components. The existing algorithms for shot boundary detection are then reviewed. In Section 2.2, the existing approaches to video classification are reviewed, with particular focus on the classification of sports videos. Some research issues arising out of the review of existing methods are identified, which are addressed in this thesis.

2.1 EXISTING METHODS FOR VIDEO SHOT BOUNDARY DETECTION

Automatic segmentation of video is the first step for organizing a long video sequence into several smaller meaningful units. A typical structure of video is shown in Fig. 2.1. The smallest basic unit is a shot. A shot in a video is a contiguous sequence of video frames recorded from a single camera operation, representing a continuous action in time and space. Relevant shots are typically grouped into a higher level unit called a scene. Each scene is a part of a story. Browsing these scenes unfolds the entire story, enabling users to locate their desired video segments quickly and efficiently.

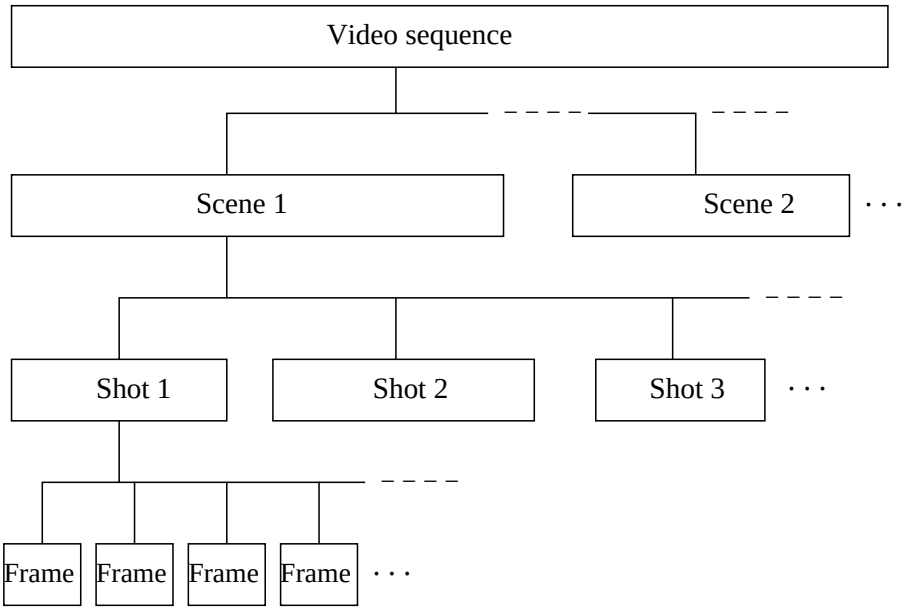


Fig. 2.1: Common video structure

Shot boundary detection is the most basic temporal video segmentation task, as it is intrinsically linked to the way that video is produced. It is a natural choice for segmenting a video into more manageable parts. This is because video content within a shot tends to be continuous, due to the continuity of both the physical scene and the parameters (motion, zoom, focus) of the camera that images it. Therefore, in principle, the detection of a shot change between two adjacent frames requires the computation of an appropriate continuity or similarity metric. However, this premise has three major complications.

The first one is to define a continuity metric for video in such a way that it is insensitive to gradual changes in camera parameters, lighting and physical scene content, easy to compute, and discriminant enough to be useful. For this purpose, one or more scalar or vector features from each frame can be extracted and distance functions can be defined in the feature domain. Alternatively, the features themselves can be used either for clustering the frames into shots or for detecting shot transition patterns. The second complication is deciding which values of the continuity metric

correspond to a shot change and which do not. This is nontrivial, since the variation of feature within certain shots can exceed the respective variation across shots. Decision methods for shot boundary detection include fixed thresholds, adaptive thresholds and statistical detection methods. The third complication is the fact that not all shot changes are abrupt. Using motion picture terminology, changes between shots can be gradual and can belong to the following categories, some of which are illustrated in Fig. 2.2:

1) *Cut:* This is the case of an abrupt change, where one frame belongs to the disappearing shot and the next one to the appearing shot.

2) *Dissolve:* In this case, the last few frames of the disappearing shot temporally overlap with the first few frames of the appearing shot. During the overlap, the intensity of the disappearing shot decreases from normal to zero (fade out), while that of the appearing shot increases from zero to normal (fade in).

3) *Fade:* Here, first the disappearing shot fades out into a black frame, and then the black frame fades into the appearing shot.

4) *Wipe:* This is a set of shot change techniques, where the appearing and disappearing shots coexist in different spatial regions of the intermediate video frames, and the region occupied by the former grows until it entirely replaces the latter.

5) *Other transition types:* Certain special effects are also used in motion pictures. They are, in general, very rare and difficult to detect.

2.1.1 Components of shot boundary detection algorithms

An important component of shot boundary detection algorithms is the set of features extracted from a video frame or from a region of the frame. Another component is the similarity measure that is used to detect the presence of a shot boundary. We present below the different choices that can be made for each component, along with their advantages and disadvantages. A shot boundary detection algorithm can then

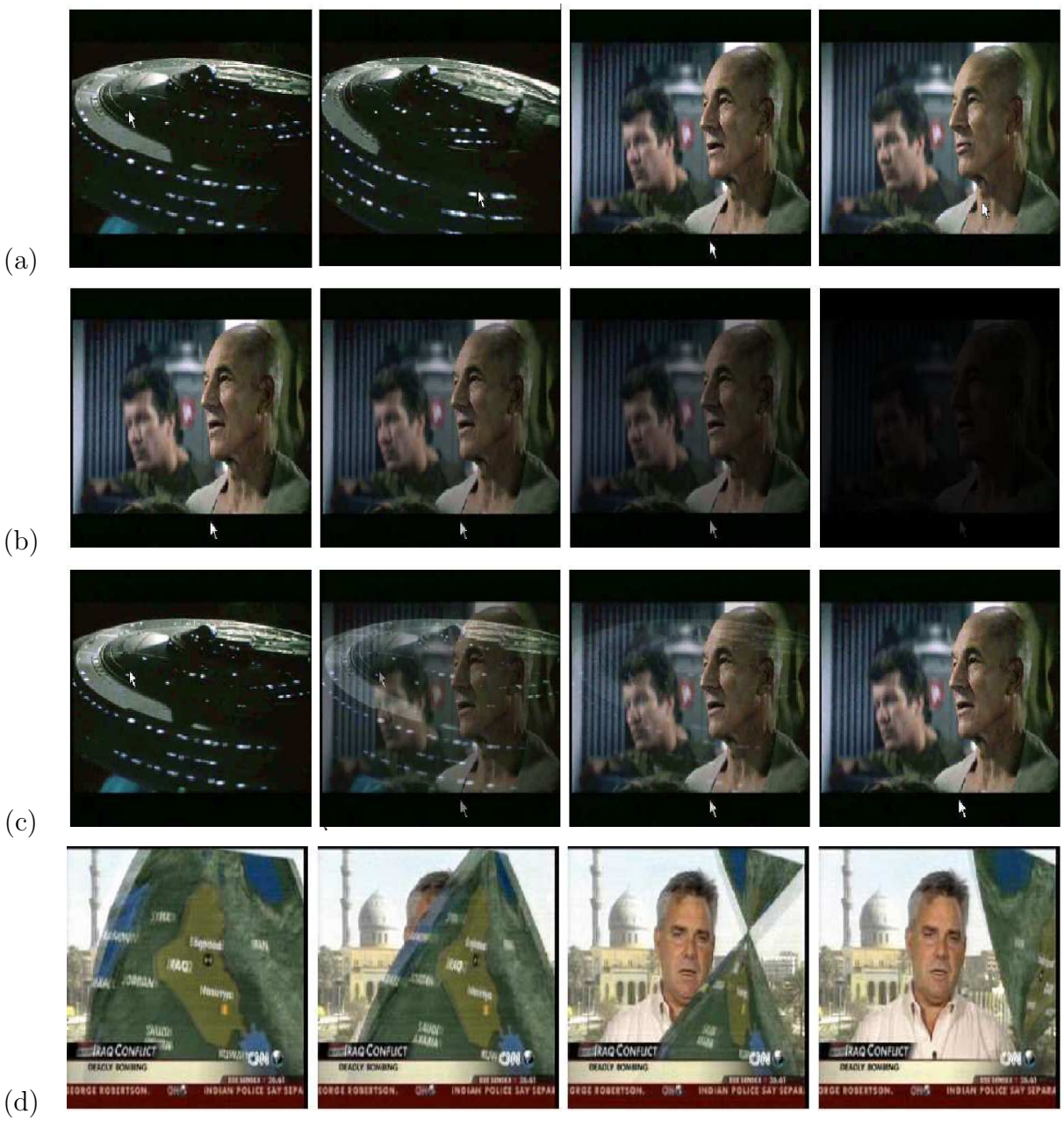


Fig. 2.2: Examples of different types of shot transitions: (a) Cut, (b) fade, (c) dissolve and (d) wipe.

be designed by suitably choosing each component.

2.1.1.1 Features used for representation of video frame

Almost all shot change detection algorithms reduce the large dimensionality of the video domain by extracting a small number of features from one or more regions of interest in each video frame. Such features include the following:

1) *Luminance/color:* The simplest feature that can be used to characterize an image is its average grayscale luminance. This, however, is susceptible to changes in illumination. A more robust choice is to use one or more statistics (e.g., averages) of the values in a suitable color space [3–5], like hue saturation value (HSV).

2) *Luminance/color histogram:* A richer feature for an image is the grayscale or color histogram. Its advantage is that it is discriminant, easy to compute, and mostly insensitive to translational, rotational, and zooming camera motions. For these reasons, it is widely used [6], [7]. However, it does not represent the spatial distribution of color in an image.

3) *Image edges:* Another choice for characterizing an image is its edge information [5], [8]. The advantage of this feature is that it is sufficiently invariant to illumination changes and several types of motion, and is related to the human visual perception of a scene. Its main disadvantage is computational cost, noise sensitivity, and when not post-processed, high dimensionality.

4) *Features in transform domain:* The information present in the pixels of an image can also be represented by using transformations such as discrete Fourier transform, discrete cosine transform and wavelets. Such transformations also lead to representations in lower dimensions. Disadvantages include high computational cost, effects of blocking while computing the transform domain coefficients, and loss of information caused by retaining only a few coefficients.

5) *Motion:* This is sometimes used as a feature for detecting shot transitions, but it is usually coupled with other features, since motion itself can be highly discontinuous within a shot (when motion changes abruptly) and is not useful when there is no motion in the video.

2.1.1.2 Spatial domain for feature extraction

The size of the region from which individual features are extracted plays an important role in the overall performance of algorithms shot change detection. A small region tends to reduce detection invariance with respect to motion, while a large region might lead to missed transitions between similar shots. In the following, we will describe various possible choices:

1) *Single pixel:* Some algorithms derive a feature for each pixel such as luminance and edge strength [5]. However, such an approach results in a feature vector of very large dimension, and is very sensitive to motion, unless motion compensation is subsequently performed.

2) *Rectangular block:* Another method is to segment each frame into equal-sized blocks and extract a set of features (e.g., average color or orientation, color histogram) from these blocks [3], [4]. This approach has the advantage of being invariant to small motion of camera and object, as well as being adequately discriminant for shot boundary detection.

3) *Arbitrarily shaped region:* Feature extraction can also be applied to arbitrarily shaped and sized regions in a frame, derived by spatial segmentation algorithms. This enables the derivation of features based on the most homogeneous regions, thus facilitating a better detection of temporal discontinuities. The main disadvantage is the high computational complexity and instability of region segmentation.

4) *Whole frame:* The algorithms that extract features (e.g., histograms) from the whole frame [7], [9], [10] have the advantage of being robust with respect

to motion within a shot, but tend to have poor performance at detecting the change between two similar shots.

2.1.1.3 Measure of similarity

To evaluate discontinuity between frames based on the selected features, an appropriate similarity/dissimilarity metric needs to be chosen. A wide variety of dissimilarity measures has been used in the literature [7, 11]. Some of the commonly used measures are Euclidean distance, cosine dissimilarity, Mahalanobis distance and log-likelihood ratio. Another example of commonly used metric, especially in the case of histograms, is the chi-square metric. Information theoretic measures like mutual information and joint entropy between consecutive frames are also proposed for detecting cuts and fades [12].

2.1.1.4 Temporal domain of continuity metric

Another important aspect of shot boundary detection algorithms is the temporal window that is used to perform shot change detection. In general, the objective is to select a temporal window that contains a representative amount of video activity. The following cases are typically used:

1) *Two frames:* The simplest way to detect discontinuity between frames is to look for a high value of the discontinuity metric between two successive frames [4], [9], [13], [14]. However, such an approach can fail to discriminate between shot transitions and changes within the shot when there is significant variation in activity among different parts of the video or when certain shots contain events that cause brief discontinuities (e.g., photographic flashes). It also has difficulty in detecting gradual transitions.

2) *N-frame window:* One technique for alleviating the above problems is to detect the discontinuity by using the features of all frames within a suitable temporal window, which is centered on the location of the potential discontinu-

ity [4], [5], [10], [15].

3) *Interval since last shot change:* Another method for detecting a shot boundary is to compute one or more statistics from the last detected shot change up to the current point, and to check if the next frame is consistent with them, as in [3], [7]. The problem with such approaches is that there is often great variability within shots, such that statistics computed for an entire shot may not be representative of its end.

2.1.1.5 Shot change detection method

Having defined a feature (or a set of features) computed from each frame and a similarity metric, a shot change detection algorithm needs to detect where these exhibit discontinuity. This can be done in the following ways:

1) *Static thresholding:* This involves comparing a metric expressing the similarity or dissimilarity of the features computed on adjacent frames against a fixed threshold [7]. This performs well only if video content exhibits similar characteristics over time. The threshold needs to be adjusted for each video.

2) *Adaptive thresholding:* Here, the threshold is varied depending on a statistic (e.g., average) of the feature difference metrics within a temporal window, as in [9] and [15].

3) *Probabilistic detection:* For a given type of shot transition, probability density function of the similarity/dissimilarity metric is estimated a priori, using several examples of that type of shot transition. Then an optimal shot change estimation is performed. This technique is demonstrated in [3] and [4].

4) *Trained classifier:* Another method for detecting shot changes is to formulate the problem as a classification task where blocks of frames are labeled as one of the two classes, namely, "shot change" and "no shot change,". This involves training a classifier (e.g., a neural network) to distinguish between the two classes [10].

2.1.2 Specific algorithms for shot boundary detection

Early work on video shot boundary detection mainly focused on abrupt shot transitions. A comprehensive survey, comparison and performance evaluation of existing shot boundary detection algorithms can be found in [6, 16–19]. In [4], shot detection techniques are reviewed and a statistical detection technique based on motion feature is proposed. Color histogram is a commonly used feature for detecting gradual transitions [20–23]. Luminance [24, 25], chromaticity [11], motion [4] and edge [16] information have also been used for shot boundary detection. Saraceno et al. [26] classify audio into silence, speech, music or noise and use this information to verify shot boundaries hypothesized by image-based features. Boreczky et al. [27] segment the video by using audio-visual features and hidden Markov models (HMM) to hypothesize the various shot transitions. The problem of shot boundary detection is approached by Hanjalic [4] using a probabilistic approach. For detecting abrupt transitions, adjacent frames are compared, while for gradual transitions, frames separated by the minimum shot length are compared. The a priori likelihood functions of the discontinuity metric are obtained using manually labeled data. Thus, different likelihood functions are estimated for each type of shot transition.

Gradual transitions are generally more difficult to detect due to camera and object motion. Detection of gradual transitions, such as fades and dissolves is examined in [20–22]. The approach proposed by Lienhart [10] detects dissolves with a trained classifier (a neural network), operating on either YUV color histograms, magnitude of directional gradients, or edge-based contrast. The classifier detects possible dissolves at multiple temporal scales and merges the results using a winner-take-all strategy. The classifier is trained using a dissolve synthesizer, which creates artificial dissolves from any available set of video sequences. The performance is shown to be superior when compared to simple edge-based dissolve detection methods.

Cernekova et al. [7] perform singular value decomposition (SVD) on the RGB color histograms of each frame to reduce the dimensionality of feature vector to ten. Ini-

tially, video segmentation is performed by comparing the angle between the feature vector of each frame and that of the average of feature vectors of the current segment. If their difference is higher than a static threshold, a new segment is started. Segments whose feature vectors exhibit large dispersion are considered to depict a gradual transition between two shots, whereas segments with small dispersion are characterized as shots. The main problem with this approach is the static threshold applied on the angle between vectors to detect a shot change, especially in the case of large intrashot content variation and small intershot content variation. Independent component analysis is also used in [11] to search for prominent basis functions in the feature space, and thereby reduce the dimension of the feature vector to two. An iterative clustering algorithm based on adaptive thresholding is used to detect cuts and gradual transitions. The reduction in dimension of feature vectors does not result in an appreciable degradation in the performance of shot boundary detection.

Boccignone et al. [15] approach the problem of shot boundary detection using the attentional paradigm for human vision. The algorithm computes for every frame, a set (called a trace) of points of focus of attention in decreasing order of saliency. It then compares nearby frames by evaluating the consistency of their traces. Shot boundaries are hypothesized when the above similarity is below a dynamic threshold.

Lelescu and Schonfeld [3] present a statistical approach for shot boundary detection. They extract the average luminance and chrominance for each block in every frame and then perform principal component analysis (PCA) on the resulting feature vectors. The eigenvectors are computed based only on the first M frames of each shot. The resulting projected vectors are modeled by a Gaussian distribution whose mean vector and covariance matrix are estimated from the first M frames of each shot. A change statistic is estimated for each new frame using a maximum likelihood methodology (the generalized likelihood ratio) and, if it exceeds an experimentally determined threshold, a new shot is started. Since eigenvectors, mean and covariance of the projected vectors are estimated using the first few frames in each shot, the estimates may not be representatives of the end of the shot, more so in the case of shots

with considerable interframe variations.

A measure defined by Li et al. [13] is the joint probability image (JPI) of two frames. The JPI is a matrix whose $[i, j]$ element is the probability that a pixel with color i in the first image has color j in the second image. Different types of shot transitions such as dissolve and fade are observed to have specific patterns of JPI. Also, a one dimensional projection of the JPI called the joint probability projection vector, and a scalar measure of dispersion of the JPI, called joint probability projection centroid (JPPC), are derived, and observed to be useful for shot boundary detection.

Another approach is to use different algorithms for each type of transition, as in [5]. Here, two algorithms are designed, one for dissolves and fades and the other for wipes. Specifically, B-spline interpolation technique is used to determine the presence of fade/dissolve within a temporal window. Further, fades are distinguished from dissolves by additionally checking if the interframe standard deviation is close to zero. Wipes are detected, based on the regular movement of the wipe's edge. For this purpose, two-dimensional wavelet transform of interframe difference is computed to enhance directionality, from which locations of strongest edges in four different directions (horizontal, vertical, and the two diagonals) are computed.

A signal processing approach to detection of cuts in video sequences is proposed in [28] using phase correlation as a measure of similarity between adjacent frames. Phase correlation is shown to be robust to illumination changes and noise.

Information theoretic measures are proposed in [12] for detecting shot boundaries. Mutual information and joint entropy between two successive frames is calculated for each of the RGB components, for detection of cuts, fade-ins and fade-outs.

The approach proposed in [29] is based on mapping the interframe distance values on to a multidimensional space, while preserving the temporal sequence (or frame ordering information). It is shown that detection of boundaries is less sensitive to the choice of threshold in the multidimensional space.

In [30], different types of transitions are observed in different temporal resolutions. Temporal multi-resolution analysis is applied on the video stream, and video frames

within a sliding window are classified into groups such as normal frames, gradual transition frames and cut frames. Then the classified frames are clustered into different shot categories.

It is a difficult task to compare the effectiveness of different algorithms for shot boundary detection, since the performance depends on the choice of features, similarity metric, the algorithm for boundary hypothesis and the video data chosen for evaluation. One attempt is made in [19] where different methods are qualitatively evaluated on the basis of features used, frame difference measures, dimensionality of features, criticality of temporal window size and thresholds, and the ability of the methods to detect different types of shot boundaries.

A survey of core concepts underlying the different schemes of shot boundary detection is presented in [17], while a comprehensive comparison of different shot boundary detection algorithms is discussed in [16].

2.1.3 Issues addressed in shot boundary detection

An observation arising out of the review of the existing approaches is that an algorithm with only one type of feature and/or similarity metric is not general enough to detect different types of shot transitions. Moreover, the choice of window affects the resolution of detection of shot boundary. Finally, most of the algorithms are sensitive to the threshold used on similarity/distance metric. In this thesis, we attempt to address these issues both at the level of features and at the level of the algorithm for shot boundary detection. Our approach to observe significant changes in only a few color features is motivated by the need to derive multiple evidence for detection of shot boundaries. In order to detect different types of shot boundaries using a single general algorithm, we propose a bidirectional processing scheme that exploits the behavior of image frames in the neighbourhood of shot boundaries. This is in contrast to existing approaches that compare a given frame against statistics derived from a window of previous frames. The sparseness of distribution of color feature vectors is exploited

by deriving features of reduced dimension using a nonlinear autoassociative neural network for compression. While any algorithm is not entirely robust to thresholds used on similarity/distance metric, our objective is to reduce the criticality of threshold so that the proposed algorithm performs optimally over range of threshold values.

2.2 REVIEW OF APPROACHES TO VIDEO CLASSIFICATION

Many approaches have been proposed for content-based classification of video data. The problem of content-based classification of video can be addressed at different levels in the semantic hierarchy as shown in Fig. 2.3. For instance, video collections can be

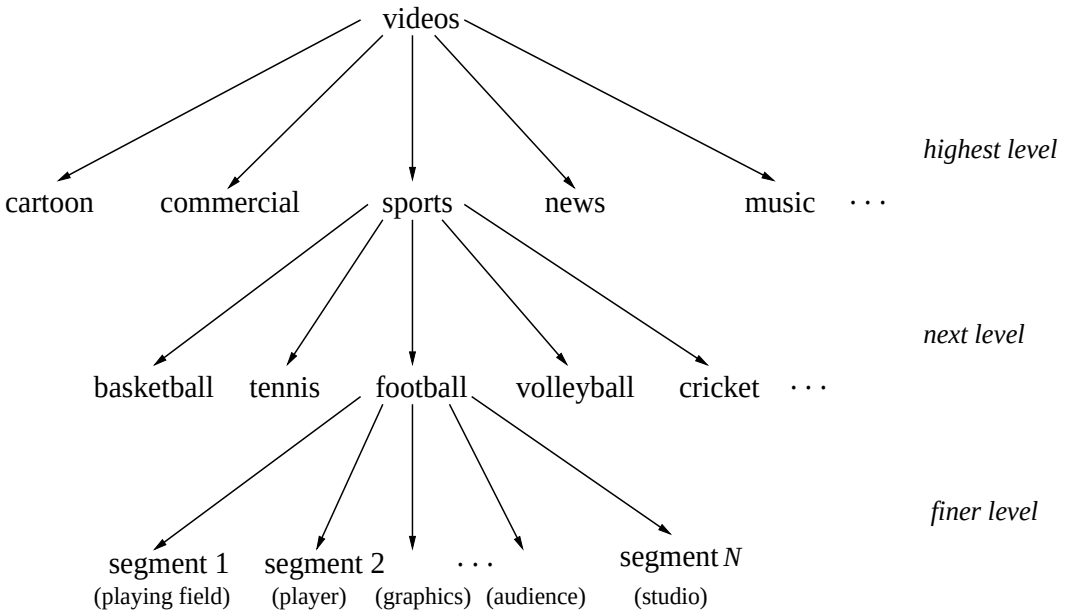


Fig. 2.3: Video classification at different levels

categorized into different program genres such as cartoon, commercials, sports, news, and music. Then, videos of a particular genre, such as sports, can be further classified into subcategories such as basketball, tennis, football and cricket. A video sequence of a given subcategory can then be segmented, and these segments can be classified into semantically meaningful classes. For example, football video sequences can be

segmented into shots and these shots are further classified as belonging to 'playing field', 'player', 'graphics', 'audience', and 'studio' classes.

The general approach to video classification involves the extraction of visual features based on color, shape, and motion, followed by the estimation of class-specific probability density function of the feature vectors [31, 32].

In [33], a criterion based on the total length of edges in a given frame is used. The edges are computed by transforming each block of 8×8 pixels using discrete cosine transform (DCT), and then processing the DCT coefficients. A rule based decision is then applied to classify each frame into one of the predefined semantic categories.

Another edge-based feature, namely, the percentage of edge pixels, is extracted from each keyframe for classifying a given sports video into one of the five categories, namely, badminton, soccer, basketball, tennis, and figure skating [34]. The k-nearest neighbour algorithm was used for classification.

Motion is another important feature for representation of video sequences. In [35], a feature called motion texture is derived from motion field between video frames, either in optical flow field or in motion vector field. These features are employed in conjunction with support vector machines to devise a set of multicategory classifiers.

The approach described in [36] defines local measurements of motion, whose spatio-temporal distributions are modeled using statistical nonparametric modeling. In [37], sports videos are categorized on the basis of camera motion parameters, in order to exploit the strong correlation between the camera motion and the actions taken in sports. The camera motion patterns such as fix, pan, zoom and shake are extracted from the video data.

Motion dynamics such as foreground object motion and background camera motion are extracted in [38] for classification of a video sequence into three broad categories, namely, sports, cartoons and news.

Transform coefficients derived from discrete cosine transform (DCT) and Hadamard transform of image frames are reduced in dimension using principal component analysis (PCA) [39]. The probability density function of the compressed features is then

modeled using a mixture of Gaussian densities.

Dimension reduction of low level features such as color and texture, using PCA, has been attempted in [40, 41], for reducing spatio-temporal redundancy.

Another approach described in [42] constructs two hidden Markov models, from principal motion direction and principal color of each frame, respectively. The decisions are integrated to obtain the final score for classification.

Apart from statistical models, rule-based methods have also been applied for classification. In [43], a decision tree method is used to classify videos into different genres. For this purpose, several attributes are derived from video sequences, such as the length of video clip, number of shots, average shot length and percentage of cuts. A set of decision rules is derived using these attributes.

Another class of algorithms focuses on deriving temporal information from video sequences. Typically, these algorithms are specific to detection of predefined events in videos within the context of a given application.

In [44], features indicating significant events are selected from video sequences. These features include a measure of motion activity and orientation of edges, which help in detection of crowd images, on-screen graphics and prominent field lines in sports videos. The evidence obtained by different feature detectors are combined using a support vector machine, which then detects the occurrence of an event.

In [45], an HMM based framework is suggested to discover the hidden states or semantics behind video signals. The objective is to arrive at a sequence of semantics from a given sequence of observations, by imposing temporal context constraints. The framework is then applied to detect predefined events in sports categories such as basketball, soccer and volleyball.

Another approach [46] models the spatio-temporal behaviour of an object in a video sequence for identifying a particular event. The game of snooker is considered and the movement of snooker ball is tracked using a color based particle filter. A few events are predefined in terms of actions, where an action can be a white ball colliding with a colored ball, or a ball being potted. An HMM is used to model the temporal

behaviour of the white ball, along with algorithms for collision detection.

A symbolic description of events is provided in [47], where events are described in terms of rules using fuzzy hypotheses. These hypotheses are based on the degree of belief of the presence of specific objects and their interrelations, extracted from the video sequence. The method involves the extraction of main mobile objects in video sequences, also called fuzzy predicates. The input is defined on the set of fuzzy predicates, while the output is a fuzzy set defined on the events to be recognized. The association between the fuzzy predicates and the set of events is represented using a neurofuzzy structure. The approach is tested on soccer video sequences for detecting some predetermined events.

In [48], a multilayer framework based on HMMs is proposed for detection of events in sports videos, on the basis that sports videos can be considered as rule based sequential signals. At the bottom layer, event HMMs output basic hypotheses using low-level features. The upper layers impose constraints on the predefined events in basketball.

The deterministic approach to event detection is compared with probabilistic approach, in [49]. While the former depends on clear description of an event and explicit representation of the event in terms of low-level features, the latter is based on states and state transition models whose parameters are learnt through labeled training sequences. It is shown, through automatic analysis of football coaching video, that the probabilistic approach performs more accurately while detecting events, mainly due to their ability to capture temporal patterns and yet, absorb small spatio-temporal variations. Thus, a common feature of most of these approaches is the use of HMM in a traditional framework, for detection of predefined events. While the detected events are used for indexing, retrieval and generation of summary/highlights, they are rarely used for classification of video sequences.

2.2.1 Issues addressed in video classification

The existing approaches to video classification can thus be broadly categorized into those which model the class-specific probability density function of feature vectors, and those which model temporal information present in the sequence of images. While features based on color and its distribution in image frames, and motion, have been explored for classification, we note that the potential of edge-based features is not fully realized. Since we address the problem of video classification in the context of sports, edge-based features play an important role in characterizing entities such as sports areas and motion of players and objects. We propose the use of edge direction histogram and edge intensity histogram, where the former is also used for spatially localizing edge information within an image. We also propose the use of autoassociative neural network models for estimating the distribution of edge-based features. These are nonlinear and nonparametric models that do not make assumptions on the shape of probability density function of feature vectors.

Methods that model temporal information in video sequences focus on the detection of events in video sequences. These are predefined events that require manual effort to identify frame sequences containing those events. Moreover, the detected events are used mostly for indexing and retrieval, and not for classification. In this context, we note that the events being specific to sports categories, can be used as features to classify a given video into one of those categories. We propose a novel method to identify and match events in video sequences, using a framework based on hidden Markov models. Here, the events are not predefined, but hypothesized based on the sequence latent in a given video. Once the events are identified, they are further used for classification of a given video into one of the sports categories.

2.3 SUMMARY

In this chapter, some of the existing approaches to video segmentation and classification were reviewed. The key components of video shot boundary detection are the features used to represent images, and the measure of similarity/distance used to hypothesize a shot boundary. The survey suggests that there is a need for robust features and algorithms which are general enough to detect different types of shot transitions. In this thesis, we propose novel algorithms for shot boundary detection to address this issue, and also examine the effectiveness of features of reduced dimension. In video classification, most algorithms are still based on low-level features, since deriving more meaningful information at a higher level is a challenging task. We explore, low-level edge-based features, and higher level features based on the notion of events for the task of sports video classification. We also study the effect of combining evidence obtained from multiple features and classifiers, on the performance of classification.

CHAPTER 3

VIDEO SHOT BOUNDARY DETECTION BY COMBINING EVIDENCE FROM EARLY AND LATE FUSION TECHNIQUES

Video segmentation is the first step in the analysis of video content for indexing, browsing and retrieval. Segmentation of video can be done at various levels such as shots, scenes and stories. Video shot boundary detection involves a low-level temporal segmentation of video sequences into elementary units called shots. Detection of shot boundaries provides a base for all video abstraction and high-level video segmentation methods. A shot is usually conceived in the literature as a series of interrelated consecutive frames captured contiguously by a single camera operation and representing a continuous action in time and space [19]. The transition between two shots can be either abrupt or gradual. An abrupt transition (hard cut) occurs between two consecutive frames, where as gradual transitions (fades, dissolves and wipes) are spread over several frames. Gradual transitions are harder to detect because the difference between consecutive frames is smaller and gradual transitions can occur even within a shot.

In this chapter, we briefly describe the traditional method of shot boundary detection, which is an early fusion algorithm. Early fusion refers to the combination of evidence due to all the components of a feature vector for detecting shot boundaries. In Section 3.1, we propose two modifications to early fusion algorithm. The first modification is to compute the dissimilarity between frames which are separated by a margin, aimed at simultaneous detection of cuts and gradual transitions. The

second modification is to perform a bidirectional processing of video, aimed at reducing the miss rate. Performance measures for evaluating algorithms for shot boundary detection are described in Section 3.2. A novel method for shot boundary detection, called late fusion is proposed in Section 3.3, which is based on evidence due only to those components of the feature vector that change significantly. The merits of early and late fusion techniques are combined to improve the performance of shot boundary detection, as described in Section 3.4. The performance of the proposed algorithms is evaluated on broadcast video data, that includes both abrupt and gradual transitions. Section 3.5 summarizes the study.

3.1 SHOT BOUNDARY DETECTION BY EARLY FUSION

Shot boundary detection involves testing, at every frame index n of a given video of length N_v frames, the following two hypotheses:

$$\begin{aligned} \mathcal{H}_0 &: \text{A shot boundary exists at frame index } n. \\ \mathcal{H}_1 &: \text{No shot boundary exists at frame index } n. \end{aligned} \tag{3.1}$$

Let $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n, \ldots, \boldsymbol{x}_{N_v}\}$ be the sequence of feature vectors of dimension p representing N_v frames in a video. Testing of the hypotheses at the frame index n involves computation of a dissimilarity value, $d[n]$, between two sequences of N feature vectors $\mathcal{X}_L = \{\boldsymbol{x}_{n-1}, \boldsymbol{x}_{n-2}, \ldots, \boldsymbol{x}_{n-N}\}$ and $\mathcal{X}_R = \{\boldsymbol{x}_n, \boldsymbol{x}_{n+1}, \ldots, \boldsymbol{x}_{n+N-1}\}$ to the left and right of n, respectively. The value of N can vary from one frame to a few frames (corresponding to less than one or two seconds). If the dissimilarity value is greater than a threshold $\tau[n]$ (either fixed or adaptive), the hypothesis $\mathcal{H}_0$, that a shot boundary exists at frame index n, is chosen.

Some of the commonly used dissimilarity measures include Euclidean distance, cosine dissimilarity, Mahalanobis distance and log-likelihood ratio. Mahalanobis distance and log-likelihood ratio are based on probability distributions, and hence are superior to a simple metric like Euclidean distance. But they are usually limited to

second order statistics. Also, they are constrained by the amount of data (number of image frames) available to estimate the parameters of the distributions. Such distance measures are not suitable for detecting gradual transitions, although the number of spurious shot boundaries hypothesized is relatively less. In contrast, measures such as Euclidean distance and cosine dissimilarity can be computed between successive frames or between two sequences of frames, thereby making them suitable for both abrupt and gradual transitions. However, when these distances are computed between successive frames, the number of spurious shot boundaries hypothesized is typically greater than that due to Mahalanobis distance or log-likelihood ratio. The objective is to use a dissimilarity measure that detects both abrupt and gradual transitions and also minimizes the number of spurious shot boundaries. Euclidean distance with an adaptive threshold based on the standard deviation computed over past few frames is similar to the use of Mahalanobis distance with a fixed threshold. In view of this, we use Euclidean distance as the dissimilarity measure with an adaptive threshold computed using the variance of a few frames before the frame at which the hypothesis is tested. Let $d[n] = \mathcal{D}(\boldsymbol{x}_n, \boldsymbol{x}_{n-1})$, where $\mathcal{D}$ denotes the Euclidean distance between two adjacent feature vectors $\boldsymbol{x}_n$ and $\boldsymbol{x}_{n-1}$. If $\sigma_L[n]$ denotes the standard deviation of N frames to the left of n, then the dynamic threshold is computed as $\tau[n] = \beta \times \sigma_L[n]$, where β is a constant scaling parameter.

We now propose a one-pass algorithm which detects cuts and gradual transitions simultaneously. Two modifications are proposed to the traditional method described above. These are: (1) Simultaneous detection of cuts and gradual transitions, by computing the dissimilarity measure between two feature vectors separated by a margin of M frames, and (2) bidirectional processing of the video for reducing the miss rate.

3.1.1 Simultaneous detection of cuts and gradual transitions

The dissimilarity value computed between two adjacent blocks of frames (where the block size can vary from one frame to a few frames) has good evidence to detect cuts,

but fails to detect a significant number of gradual transitions. Figs. 3.1(a) and (b) show the variation of three components of the feature vectors as a function of time, in the vicinity of a shot boundary, for a cut and a gradual transition, respectively. The three-dimensional feature vector shown for illustration is obtained by nonlinear compression of 512-dimensional color histogram. The issue of dimension reduction is discussed in Chapter 4. The dissimilarity values computed between adjacent frames are shown in Figs. 3.1(c) and (d), by solid lines. Also plotted using dotted lines are the dynamic thresholds, computed as described above. A shot boundary is hypothesized, where the dissimilarity value exceeds the dynamic threshold. It can be seen from Fig. 3.1(d) that the dissimilarity values between the adjacent frames is not significant enough compared to the dynamic threshold, and hence the gradual transition is missed. In order to detect gradual transitions simultaneously with cuts, we propose to compute the dissimilarity value between frames separated by a margin of, say M frames. The dissimilarity value and the dynamic threshold are now computed as $d[n] = \mathcal{D}(\boldsymbol{x}_n, \boldsymbol{x}_{n-M})$ and $\tau[n] = \beta \times \sigma_L[n-M]$. The basis for computing dissimilarity value between frames separated by a margin is that the frames in the region of gradual transition do not contribute significantly to the dissimilarity. Thus, by excluding frames in the region of gradual transition, a significant dissimilarity value is obtained. Suppose, if frames in the transition region are excised out, then a gradual transition would resemble an abrupt change. Thus, by comparing frames with a margin greater than the typical duration of a gradual transition, the proposed modification treats the detection of cuts and gradual transitions alike. Figs. 3.1(e) and (f) show the dissimilarity values computed between two frames separated by a margin of $M = 20$ frames. It can be seen from the figures that the evidence for the gradual transition is comparable with that of the cut, and hence can be detected simultaneously with a cut. While the choice of a margin helps in detection of gradual transitions, it does not affect the detection of cuts.

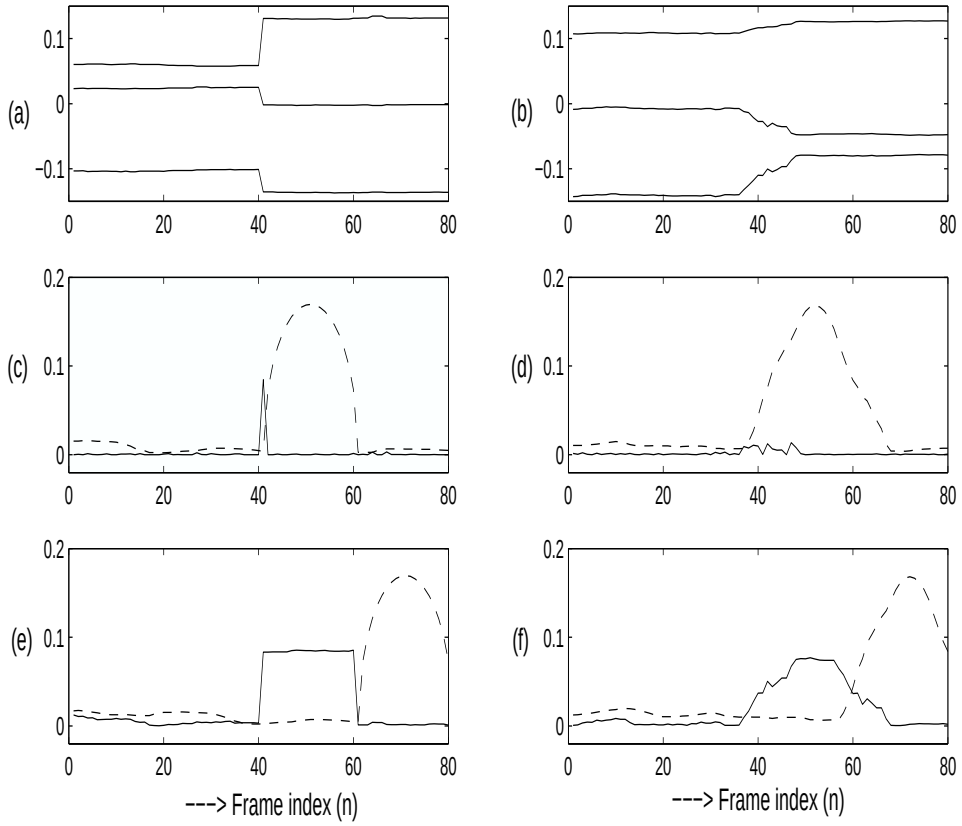


Fig. 3.1: Shot boundary detection without and with margin. Contours of three (3) components of feature vector $\boldsymbol{x}_n$, for (a) a typical cut and (b) a gradual transition. (c) and (d) Dissimilarity values $d(\boldsymbol{x}_n, \boldsymbol{x}_{n-1})$ without margin, corresponding to (a) and (b), respectively, shown in solid line. The dynamic threshold $\beta \times \sigma_L[n]$ with β =4 and N =10 is shown by a dotted line. (e) and (f) Dissimilarity values $d(\boldsymbol{x}_n, \boldsymbol{x}_{n-M})$ with a margin of M =20, corresponding to (a) and (b), respectively. The dynamic threshold $\beta \times \sigma_L[n-M]$ with β =4 is shown by a dotted line.

3.1.2 Bidirectional processing of video

Significant variations in the frames (and hence in the features) just before the shot boundary, results in a high threshold that causes several genuine shot boundaries to be missed. This is a typical problem with methods that process the video only in the forward direction (left to right). This can be overcome by processing the video in the reverse or backward direction (right to left) as well. One such case is shown in Fig. 3.2. The distance and the threshold values computed in the forward direction are given by $d_f[n] = \mathcal{D}(\boldsymbol{x}_n, \boldsymbol{x}_{n-M})$ and $\tau_f[n] = \beta \times \sigma_R[n-M]$. These are same as discussed in Section 3.1.1. The distance and the threshold values in the reverse direction are given by $d_b[n] = \mathcal{D}(\boldsymbol{x}_{n-1}, \boldsymbol{x}_{n+M-1})$ and $\tau_b[n] = \beta \times \sigma_R[n+M-1]$, where $\sigma_R[n+M-1]$ denotes the standard deviation of N frames to the right of the frame whose index is $(n+M-1)$. It can be easily verified that $d_b[n]$ and $\tau_b[n]$ are just the shifted versions of their counterparts in the forward direction, $d_f[n]$ and $\tau_f[n]$. Hence, it is sufficient to compute the dissimilarities and the threshold values only in the forward direction. The evidences due to forward and backward processing appear as two edges (rising and falling respectively) of the distance plot, as is apparent in Fig. 3.1(e). The new condition for hypothesizing a shot boundary at the frame index n becomes

$$d_f[n] > \tau_f[n] \quad | \quad d_b[n] > \tau_b[n] \tag{3.2}$$

where "|" denotes the logical OR operation. This modified condition, that uses the evidence from either side of a shot boundary, reduces the miss rate, but at the same time can increase the number of false alarms.

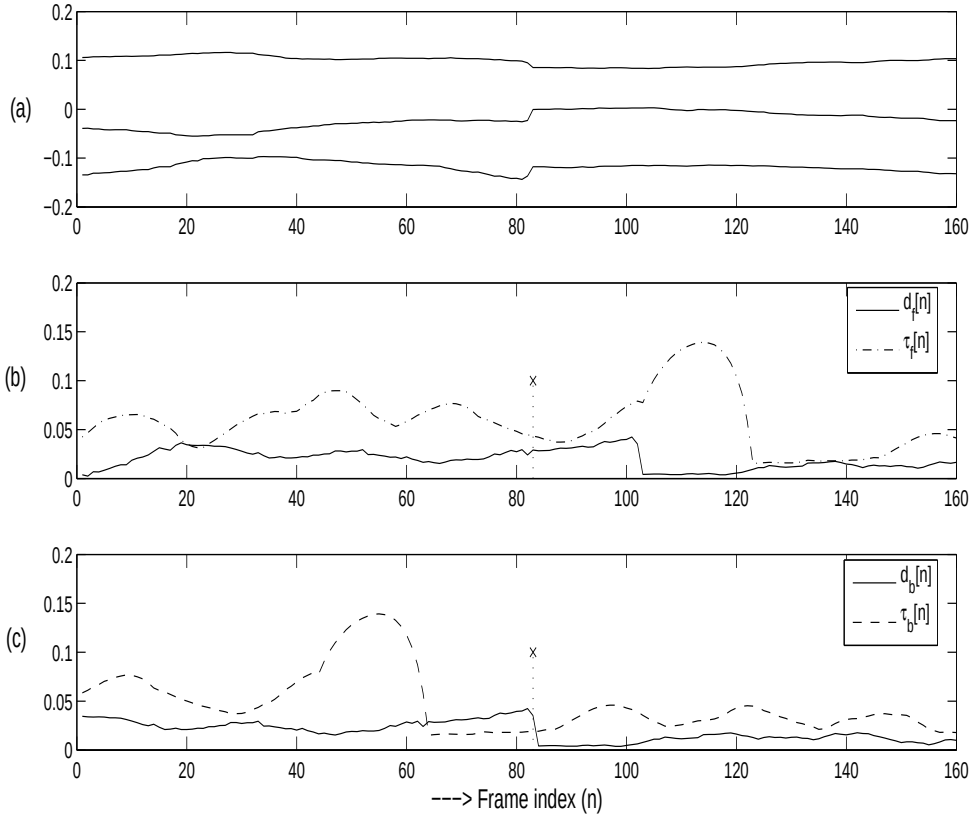


Fig. 3.2: Bidirectional processing of video for shot boundary detection. (a) Contours of three (3) dimensions of feature vector $\boldsymbol{x}_n$, for a cut at $n = 83$ with significant variation to the immediate left of the transition. (b) The dissimilarity value $d_f[n]$ (solid line) and the threshold $\tau_f[n]$ (dashed line). Forward processing fails to detect the shot boundary (marked 'x'). (c) The dissimilarity value $d_b[n]$ (solid line) and the threshold $\tau_b[n]$ (dashed line). Backward processing detects the shot boundary.

3.2 PERFORMANCE EVALUATION

3.2.1 Data set

The performance of shot boundary detection algorithms is evaluated on a database of approximately $2\frac{1}{2}$ hours of news, wild life documentaries and sports video sequences. The database contains a total of 618 cuts and 170 gradual transitions, the details of which are given in Table 3.1. The video clips were captured at a rate of 25 frames per second, at 320 $\times$ 240 pixel resolution, and stored in audio video interleave (AVI) format.

Table 3.1: Video data used for shot boundary detection experiments.

Clip ID	**Duration (min)**	**# frames**	**# cuts**	**# graduals**
BBC	23	33,895	155	28
CNN	24	36,000	76	72
NDTV	27	32,673	135	7
Wild Life	20	30,068	137	21
Sports	15	22,899	115	42
Overall	109	1,55,535	618	170

3.2.2 Features

An image histogram refers to the probability mass function of the image intensities. This is extended for color images to capture the joint probabilities of the intensities of the three color channels, namely, red (R), green (G), and blue (B). More formally, the

color histogram is defined by

$$h_{A,B,C}(a,b,c) = N \times Pr(A = a, B = b, C = c), \tag{3.3}$$

where A , B and C represent the three color channels R,G and B, respectively, and N is the number of pixels in the image. Computationally, the color histogram is formed by discretizing the colors within an image and counting the number of pixels of each color. In our experiments, a 512-dimension RGB color histogram, obtained by quantizing the 3-D color space into an $8 \times 8 \times 8$ grid, is used as the feature vector.

3.2.3 Performance metrics

The performance of the shot boundary detection task is measured in terms of *recall* (R) and *precision* (P) criteria, given by

$$R = \frac{N_c}{N_m}, \tag{3.4}$$

and

$$P = \frac{N_c}{N_c + N_f}, \tag{3.5}$$

where N_m is the total number of actual (or manually marked) shot boundaries, N_c is the number of shot boundaries detected correctly, and N_f is the number of false alarms. A good performance requires both recall and precision to be high, i.e., close to unity. The choice of the threshold factor β is crucial. A small value of β improves the recall, while reducing the precision at the same time. A large value of β has the reverse effect on recall and precision. A compromise between recall and precision is obtained by using a measure combining the recall and precision, given by

$$F_1 = \frac{2 \times R \times P}{R + P}. \tag{3.6}$$

Ideally, F_1 should be close to unity.

3.2.4 Results and discussion

The performance of forward and backward processing using the early fusion algorithm is given in Table 3.2. The optimal threshold factor β_{opt} corresponds to best F_1 measure.

Table 3.2: Performance (in terms of R, P and F_1) of shot boundary detection using forward and backward processing of video by early fusion.

	Forward processing				**Backward processing**			
Clip ID	β_{opt}	R	P	F_1	β_{opt}	R	P	F_1
BBC	10.0	0.881	0.926	0.903	8.0	0.902	0.942	0.921
CNN	8.0	0.878	0.909	0.893	6.5	0.892	0.904	0.898
NDTV	12.5	0.768	0.908	0.832	6.5	0.873	0.780	0.824
Wild Life	9.5	0.852	0.912	0.881	6.5	0.858	0.844	0.851
Sports	11.0	0.867	0.897	0.881	9.0	0.844	0.854	0.849
Overall	9.5	0.854	0.889	0.871	6.5	0.890	0.820	0.853

It is to be noted here that the optimal threshold factor is different for different clips, and also for forward and backward directions of the same clip. Performance after combining the evidence obtained using forward and backward processing is given in Table 3.3. It is seen that the OR logic improves the performance, while the AND logic reduces the optimal F_1 value to 0.59. This is mainly due to a high probability that one of the two sides of a shot boundary has a large variance among the feature vectors.

3.3 SHOT BOUNDARY DETECTION BY LATE FUSION

The early fusion technique described in the previous section was based on the overall change in color histogram between adjacent frames in a video sequence. The dimension

Table 3.3: Performance (in terms of R, P and F_1) of shot boundary detection by combining the evidence obtained using forward and backward processing of video.

Clip ID	Combined (OR)				Combined (AND)			
	β_{opt}	R	P	F_1	β_{opt}	R	P	F_1
BBC	10.0	0.937	0.918	0.927	3.0	0.881	0.592	0.708
CNN	8.0	0.953	0.898	0.925	3.0	0.736	0.407	0.524
NDTV	9.5	0.901	0.837	0.868	3.0	0.852	0.531	0.654
Wild Life	9.5	0.902	0.878	0.889	3.0	0.863	0.583	0.696
Sports	13.0	0.878	0.940	0.908	3.0	0.700	0.240	0.358
Overall	10.0	0.908	0.882	0.895	3.0	0.817	0.465	0.592

of color histogram, 512 in this case, was chosen to provide adequate representation to each color component. However, not all components of the color histogram feature vectors are populated for a given frame of video. Secondly, not all components of the color histogram change significantly in the neighbourhood of a shot boundary. It is observed that in general, a small number of color bins undergo a significant change when there is shot boundary. Figs. 3.3 (a) and (b) show the probability and cumulative distributions of the number of bins changing significantly at the actual shot boundaries, for a threshold factor of β =5. It can be seen from Fig. 3.3 (b) that around 50% of the shot boundaries have 50 or less number of bins changing significantly (approximately 10% of the total number of bins) and around 82% of the shot boundaries have 100 or less number of bins (approximately 20%) changing significantly. Hence we see that a shot boundary can be detected by observing a significant change in a small number of bins.

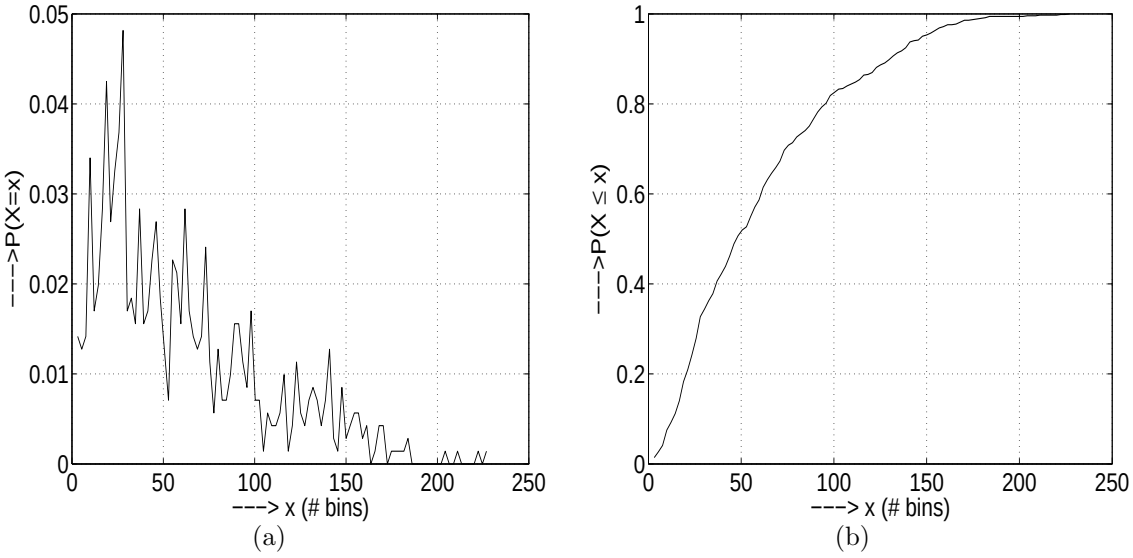


Fig. 3.3: (a) Probability distribution of shot boundaries in terms of the number of color bins changing significantly. (b) Cumulative distribution of (a).

At the same time, if all components of the color histogram are considered for the computation of dissimilarity, as is the case in early fusion, even a small contribution from each component results in a large value of the overall dissimilarity. This is typically the case when frames in a video sequence change gradually due to object/camera motion and intensity variations, even when there is no shot boundary. To overcome the problem of false hypothesis due to small changes accumulated over a large number of bins, we propose to use the number of bins changing significantly as a measure to hypothesize a shot boundary. We call this as late fusion technique, since the components of color histogram are first observed for significant change, and only then included in the process of decision making. The condition for hypothesizing a shot boundary is exactly same as the early fusion technique outlined in the previous section, except that it is applied on individual bins separately. If the number x of bins voting for a shot boundary exceeds a threshold K, a shot boundary is hypothesized. The performance of the late fusion technique is given in Table 3.4. The optimal threshold factor K_{opt} corresponds to best F_1 measure. We observe from the table that less than 20 components of the color histogram are sufficient for detection of shot boundaries, provided that

Table 3.4: Performance (in terms of R, P and F_1) of shot boundary detection by late fusion of decisions along individual dimensions.

Video category	K_{opt}	R	P	F_1
BBC	19	0.930	0.911	0.920
CNN	12	0.865	0.914	0.889
NDTV	17	0.930	0.841	0.883
Wild Life	17	0.880	0.880	0.880
Sports	12	0.900	0.976	0.936
Overall	15	0.888	0.877	0.882

these components change significantly in the vicinity of a shot boundary. Also, comparison of Tables 3.3 and 3.4 indicates that the performance of late fusion algorithm is comparable to that of early fusion (when OR logic is used for combination).

3.4 COMBINING EVIDENCES FROM EARLY AND LATE FUSION TECHNIQUES

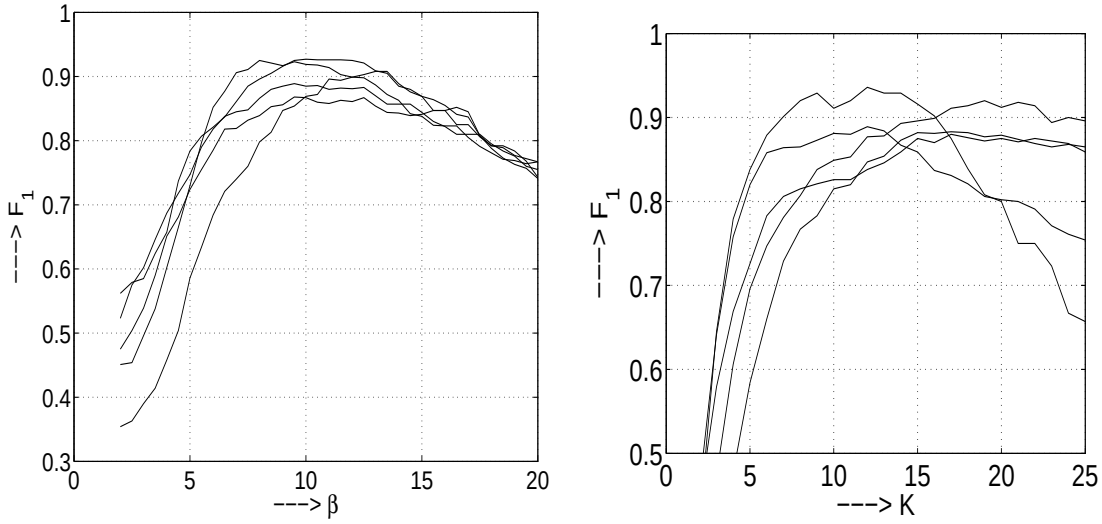
The early fusion technique computes the net changes in all the bins or dimensions, which can be significantly large although the change in the individual bins is small. This can lead to false hypotheses of shot boundaries thereby bringing down the performance. The late fusion technique provides robustness against such cases which are typically caused by illumination changes and camera/object motion. At the same time, it fails to detect genuine boundaries which have similar color content on either side. Thus, early fusion relies on the extent of overall change, while late fusion relies on the number of significant changes. The inherent advantages of these two techniques can be exploited by combining evidence due to these two methods. The performance of the

shot boundary detection task, when the shot boundaries hypothesized by early fusion are validated (equivalent to logical AND operation) by the late fusion method, is given in Table 3.5. It can be seen that there is significant improvement in performance as compared to that of the two techniques individually.

Table 3.5: Performance (in terms of R, P and F_1) of shot boundary detection after combining early and late fusion techniques.

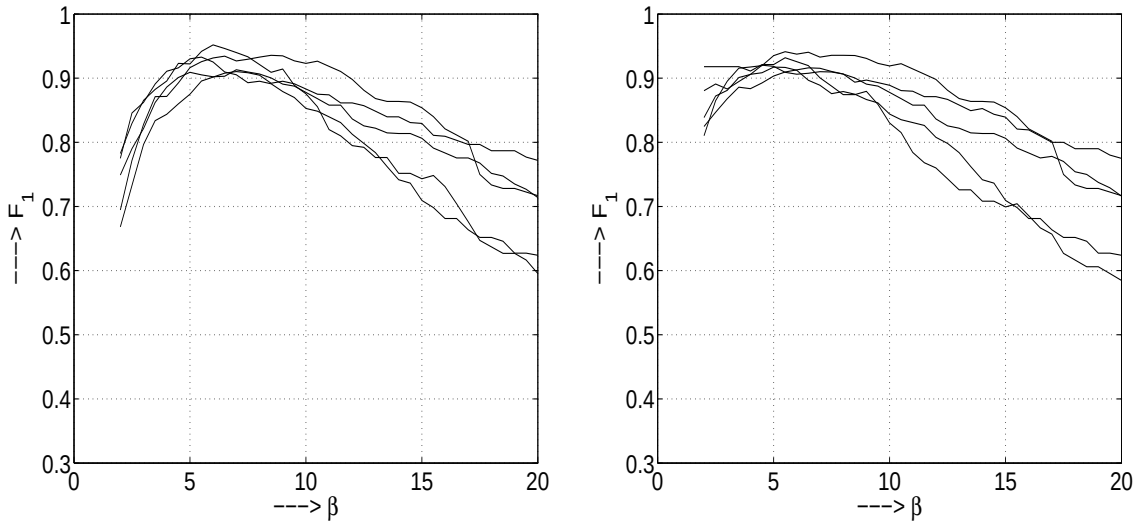
Video category	β_{opt}	K_{opt}	R	P	F_1
BBC	3.5	18	0.9720	0.9329	0.9521
CNN	5.5	3	0.9527	0.9276	0.9400
NDTV	6.5	14	0.9225	0.9225	0.9225
Wild Life	4.5	10	0.9454	0.8964	0.9202
Sports	6.0	5	0.9889	0.9175	0.9519
Overall	5.5	8	0.922	0.921	0.921

Another significant advantage of combining these two techniques is that the criticality of the choice of threshold factors β and K is reduced. It can be seen from Figs. 3.4 (a) and (b), that the F_1 measure drops significantly on either side of the optimal threshold factors β and K, for early and late fusion techniques, respectively. Fig. 3.5 shows that for the combination of early and late fusion, the F_1 measure remains high (around 0.9) over a wide range of β values, for a chosen value of K. Similar trend is observed when the K values are varied for a fixed value of β, as shown in Fig. 3.6. For a fixed value of early fusion threshold factor β =5.0, the overall recall, precision and F_1 values for the entire data set are plotted as a function of the late fusion threshold factor K in Fig. 3.7. Thus, we see that there is a greater flexibility in the choice of the threshold factors.



(a) (b)

Fig. 3.4: Performance curves for five different video clips as a function of the threshold factor. (a) Early fusion technique: F_1 vs β and (b) late fusion technique: F_1 vs K.



(a) (b)

Fig. 3.5: Performance curves for the combination of early and late fusion techniques. F_1 values as a function of early fusion threshold factor β, for five different video clips, and for two different values of late fusion threshold factor K, (a) 5 and (b) 10.

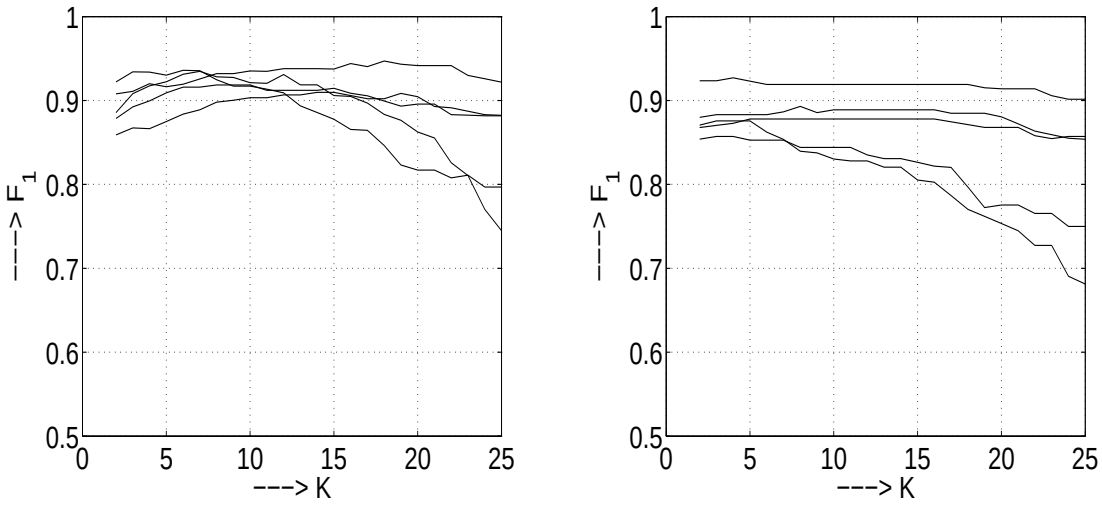


Fig. 3.6: Performance curves for the combination of early and late fusion techniques. F_1 measure as a function of late fusion threshold factor K, for the five different video clips, and for two different values of early fusion threshold factor β, (a) 5 and (b) 10.

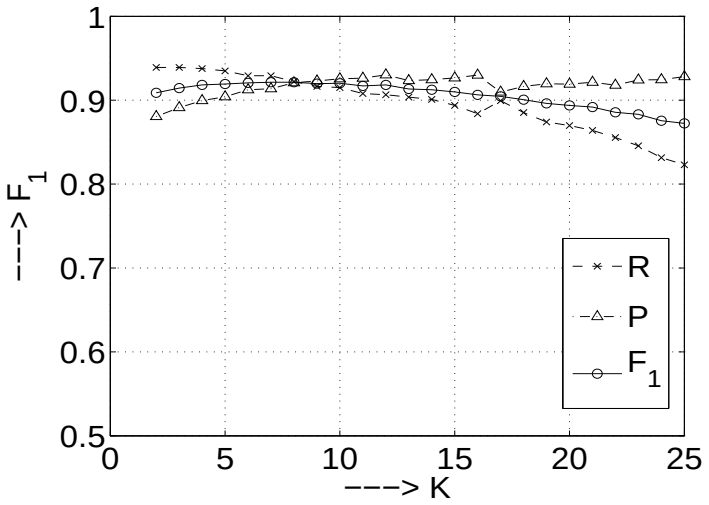


Fig. 3.7: Recall, precision and F_1 *vs* threshold factor K for the combined method, and for optimal β.

The missed shot boundaries are mainly due to lack of evidence (insignificant change in color information, such as cuts within a war footage). Features other than color or intensity information, like edge changes that provide complementary information, need to be used to reduce the miss rate. The false detections fall into two categories: significant changes in captions, graphics or animation and significant camera/object motion within a shot. Strictly speaking, the former should not be considered as false alarms as they indeed correspond to events within a shot, while it is always difficult to eliminate the latter.

3.5 SUMMARY

In this chapter, we proposed a novel method called late fusion of evidence, for detection of shot boundaries. The basis for this method lies in the significant change occurring in a small number of color features, in the neighbourhood of a shot boundary. The technique is robust to illumination changes and camera/object motion within a shot. Also, modifications to the existing early fusion algorithm were suggested. These modifications, namely, processing with a margin and bidirectional processing, make effective use of statistics derived from frames in the neighbourhood of shot boundaries. These modifications were shown to improve the performance of shot boundary detection. The evidence due to late fusion has been combined with the evidence due to early fusion to exploit the advantages of both the methods. It was also observed that such a combination reduces the criticality of the choice of threshold, by yielding good performance over a range of threshold values.

CHAPTER 4

VIDEO SHOT BOUNDARY DETECTION USING FEATURES OF REDUCED DIMENSION

The problem of detection of shot boundaries in video sequences was discussed in the previous chapter, using color histogram as a feature for representation of images. It has been observed that color histogram is sparse, that is, many of the components/bins of color histogram are either small or zero. This sparseness indicates that only a few components of color histogram are significant, and hence, there is a case for reducing the dimension of color histogram feature vector. Secondly, color histogram does not reflect the spatial distribution of colors in an image. Thus, a representation based on color histogram may fail to differentiate between two images which are distinct, but have similar color distributions. Hence, a representation is needed that can capture the spatial distribution of colors. In this chapter, we use color coherence vector as a feature for representation of images. The color coherence vector (CCV) is a histogram-based feature that incorporates information about the spatial distribution of color as well [50]. While the CCV provides additional information for shot boundary detection, it also increases the dimensionality of the feature vector. This can be countered by transforming the high dimension sparse feature space into a low dimension space, while preserving most of the significant information.

There are several choices for providing linear or nonlinear mapping for dimension reduction. Our choice for nonlinear principal component analysis (NLPCA) using autoassociative neural network (AANN) models is motivated by its superior scaling properties, less computational cost, and its ability to capture higher order relations in the data [51, 52]. The nonlinear transformation of the feature vectors to a low di-

mension space preserves information useful for shot boundary detection, and helps to provide better visualization. Hence, we propose the use of autoassociative neural network (AANN) models for nonlinear compression of color coherence feature vectors. We discuss the effect of compression of color coherence vectors on the performance of shot boundary detection. This approach to dimension reduction is compared with those based on singular value decomposition (SVD) and independent component analysis (ICA).

This chapter is organized as follows: In Section 4.1, the extraction of color coherence vector (CCV) from a frame of video is described. In Section 4.2, AANN models and their ability to perform nonlinear compression of feature vectors are discussed. Also, singular value decomposition (SVD) and independent component analysis (ICA) are briefly described in the context of compression. Section 4.4 provides the description of the proposed shot boundary detection algorithm. In Section 4.5, experimental results of the shot boundary detection algorithm are discussed. Section 4.6 summarizes the study.

4.1 COLOR COHERENCE VECTORS

Color histograms are used to compare images in many applications. Their advantages are ease of computation, and insensitivity to small changes in camera viewpoint. However, color histograms lack spatial information, so images with very different appearances can have similar histograms. For example, the images shown in Fig. 4.1 have similar color histograms, despite their rather different appearances [50, 53].

Many applications require simple methods for comparing pairs of images based on their overall appearance. For example, a user may wish to retrieve all images similar to a given image from a large database of images. Color histograms are a popular solution to this problem, and are used in systems like QBIC [54] and Chabot [55]. Color histograms are computationally efficient, and generally insensitive to small changes in camera position. However, a major limitation is that a color histogram provides

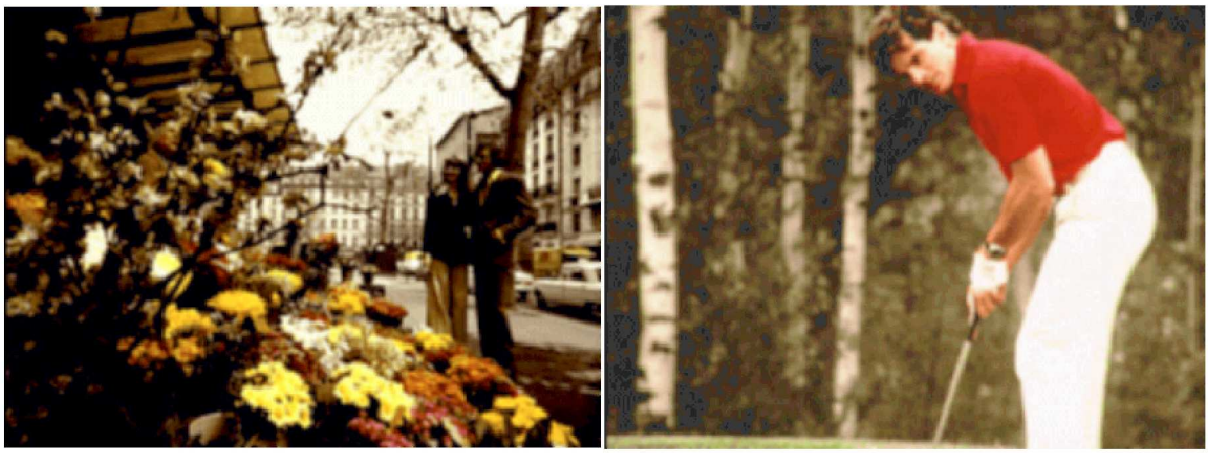


Fig. 4.1: Two images with similar color histograms.

no spatial information; it merely describes which colors are present in an image, and in what proportions. In addition, color histograms are sensitive to both compression artifacts and changes in overall image brightness. To overcome the lack of spatial information, we describe a histogram-based method that incorporates spatial information for representing images. We classify each pixel in a given color bin as either coherent or incoherent, based on whether or not it is part of a large similarly-colored region. A color coherence vector (CCV) stores the number of coherent as well as incoherent pixels with each color. By separating coherent pixels from incoherent pixels, CCVs provide finer distinctions than color histograms. Intuitively, we define a color's coherence as the degree to which pixels of that color are members of large similarly-colored regions. We refer to these significant regions as coherent regions, and observe that they are of importance in characterizing images. CCVs prevent coherent pixels in one image from matching incoherent pixels in another. This allows fine distinctions that cannot be made with color histograms.

4.1.1 Computation of color coherence vector

Color coherence vector is obtained by splitting the number of pixels in each color bin into two parts, coherent and incoherent. The coherent part gives the count of pixels that lie within a neighbourhood of the same color, and the remaining pixels form the incoherent part. By tracking the coherent and incoherent parts separately for each color bin, CCVs provide a finer distinction between images than color histograms. Thus, CCV can detect some additional shot boundaries which may otherwise be missed by color histogram.

The high resolution RGB color space is quantized into a smaller number of color bins, so as to reduce some of the fluctuations in color intensities over adjacent frames from the same scene. The CCV for a frame of video can be computed by constructing connected components or graphs $\mathcal{G} = \{G_1, G_2, \ldots, G_N\}$ linking all adjacent pixels of same color. A connected component $G_i \in G$ is a maximal set of pixels such that for any two pixels $p \equiv (x_1, y_1),\ q \equiv (x_2, y_2) \ \in \ G_i$, there is a path in G_i between p and q. A path in G_i is a sequence of pixels $p_1, p_2, ..., p_n$ such that each pixel is in G_i and any two sequential pixels p_j and p_{j+1} are adjacent to each other. Two pixels are considered to be adjacent, if one pixel is among the eight closest neighbours of the other. Each connected component is of a specific color, can be of varying size, and can be computed in linear time. For each connected component G_i associated with the k^{th} color, the count α_k and β_k of coherent and incoherent pixels , respectively, are updated as follows:

$$
\begin{aligned}
&\text{If} \quad |G_i| > \gamma \quad \text{then} \\
&\qquad \alpha_k = \alpha_k + |G_i| \\
&\text{else} \\
&\qquad \beta_k = \beta_k + |G_i|
\end{aligned}
$$

where γ is a constant that denotes the minimum size of a coherent neighbourhood, and $|G_i|$ denotes the size of the graph G_i. Thus, if the color space is quantized into $m/2$ bins, a CCV of m dimensions $\{(\alpha_1, \beta_1), ..., (\alpha_{m/2}, \beta_{m/2})\}$ is obtained.

4.2 APPROACHES TO DIMENSION REDUCTION

In this section, we discuss three different approaches to reduce the dimension of color coherence feature vectors. The first approach, based on singular value decomposition, attempts to uncover the geometrical structure formed by feature vectors in the input space. The input feature vectors are projected onto orthonormal basis vectors, which in turn are derived from the input data itself. In contrast, independent component analysis attempts to derive basis vectors which need not be orthogonal, but are statistically independent. Finally, autoassociative neural network models are discussed, which attempt to capture nonlinear principal components of the input feature space and thereby help reduce the dimension of the feature vectors.

4.2.1 Singular value decomposition

The singular value decomposition (SVD) of an $M \times N$ matrix $\mathbf{A}$ is any factorization of the form $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U}$ is an $M \times M$ column-orthogonal matrix, $\mathbf{V}$ is an $N \times N$ column-orthogonal matrix, and $\mathbf{\Sigma}$ is an $M \times N$ diagonal matrix with nonnegative elements, given by $\mathbf{\Sigma} = \text{diag}\left(\sigma_1, \sigma_2, \cdots, \sigma_R\right)$ where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_R \geq 0$ and R is the rank of the matrix A. The values σ_i, $i = 1, 2, \cdots, R$ are the singular values, and the first of R columns of $\mathbf{V}$ and $\mathbf{U}$ are called the right and left singular vectors, respectively.

Let $\mathbf{a}_i$ denote an M-dimensional feature vector derived from i^{th} image frame, where $i = 1, 2, \cdots, N$, and N denotes the number of frames in the video sequence. The matrix $\mathbf{A}$ is constructed by arranging the feature vectors $\mathbf{a}_i$, $i = 1, 2, \cdots, N$, along the columns. The column vectors of $\mathbf{A}$ are projected onto the orthonormal basis formed by vectors of the left singular matrix $\mathbf{U}$. The row vectors of $\mathbf{A}$ are projected on to the orthonormal basis formed by vectors of the right singular matrix $\mathbf{V}^T$. Let M be the dimension of the feature vectors and N be the number of feature vectors. By performing SVD, vectors from the M-dimensional feature space are projected onto a K-dimensional ($K < R \leq M$) refined feature space, by preserving only K singular

vectors corresponding to the K largest singular values of $\mathbf{\Sigma}$. A given M-dimensional feature vector $\mathbf{a}_i$ can be compressed into a K-dimensional feature vector $\tilde{\mathbf{a}}_i$ as

$$\tilde{\mathbf{a}}_i = \begin{bmatrix} \mathbf{a}_i^T \mathbf{u}_1 & \mathbf{a}_i^T \mathbf{u}_2 & \cdots & \mathbf{a}_i^T \mathbf{u}_K \end{bmatrix}^T, \tag{4.1}$$

where $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_K$ are the first K column vectors of $\mathbf{U}$ and T denotes the transpose operator.

4.2.2 Independent component analysis

Independent component analysis (ICA) is a statistical and computational technique, which uses higher order statistics for revealing hidden factors that underlie sets of random variables, measurements or signals [56]. ICA is a linear nonorthogonal transform which separates the independent source signals from their linear mixtures without knowing the mixing matrix.

Independent component analysis defines a model for the observed multivariate data, which is typically given as a large database of samples. In the model, the data variables are assumed to be linear or nonlinear mixtures of some unknown latent variables, and the mixing system is also unknown. The latent variables are assumed non-Gaussian and mutually independent and they are called independent components of the observed data. These independent components, also called sources or factors, can be estimated using ICA. The ICA technique aims to find a linear transform for the input data using a basis as statistically independent as possible. While principal component analysis (PCA) tries to obtain a representation based on uncorrelated variables, ICA provides a representation based on statistically independent variables. The features produced by PCA are mutually uncorrelated. However, ICA not only decorrelates the data but also reduces higher-order statistical dependence of data.

Let $\mathbf{x}$ denote an N-dimensional vector and $\mathbf{s}$ denote an M-dimensional vector, whose components are the M statistically independent non-Gaussian source signals. The ICA model can be expressed as

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \tag{4.2}$$

where $\mathbf{A}$ is an $N \times M$ mixing matrix with linearly independent columns. The objective is to estimate the demixing matrix $\mathbf{W}$ using only the observed signals $\mathbf{x}$. The matrix $\mathbf{W}$ is applied on the observed signal such that the components of the output vector $\mathbf{y}$ are as statistically independent as possible, where

$$\mathbf{y} = \mathbf{W}\mathbf{x} = \mathbf{W}\mathbf{A}\mathbf{s}. \tag{4.3}$$

The rows of the output vectors are the independent components. The basis functions learned by ICA form the columns of matrix $\mathbf{A}$. An input feature vector can be compressed by projecting it onto a few independent components.

4.2.3 Autoassociative neural network models

Autoassociative neural network models are multilayer feedforward neural network models that perform a nonlinear identity mapping of the input space [52,57]. The network architecture of these models may have more than one hidden layer, and the input and output layers have the same number of processing units. One of the hidden layers known as the bottleneck layer or the compression layer has a dimension lesser than the input layer. These networks can be trained using backpropagation learning algorithm [52] so as to reconstruct the input data at the output layer. The vectors at the outputs of the compression layer represent projections of the input feature vectors onto significant basis functions learnt by the network [58]. This characteristic of AANN model was exploited extensively for linear and nonlinear compression of input data [59].

The principal component analysis (PCA) projects the input feature vectors onto the first few directions of maximum variances, so that the error due to the representation is optimal in the mean squared sense. This linear transformation uses only the second order correlations in the data, and cannot capture some of the class discriminative information which is significant for pattern recognition tasks. Nonlinear principal component analysis (NLPCA), typically implemented by neural network models, provides a nonlinear generalization of PCA [51, 58, 60]. AANN models and kernel PCA

(KPCA) methods have been commonly used for NLPCA [52]. In the kernel PCA, eigenanalysis is performed in a feature space nonlinearly related to the input space, and whose dimension is directly proportional to the number of input patterns. For a large number of patterns, kernel PCA results in a kernel matrix of large dimension. In such cases, eigenanalysis becomes computationally intensive. Also, from the studies of dimension reduction using AANN models for recognition of consonant-vowel units of speech, it is shown that nonlinear compression using AANN models is superior to linear compression by PCA [61]. Hence, AANNs are an attractive tool for nonlinear compression of input feature vectors.

A five layer AANN model for performing nonlinear compression is shown in Fig. 4.2. It has m nodes in the input layer, p nodes in the compression (third) layer, and m

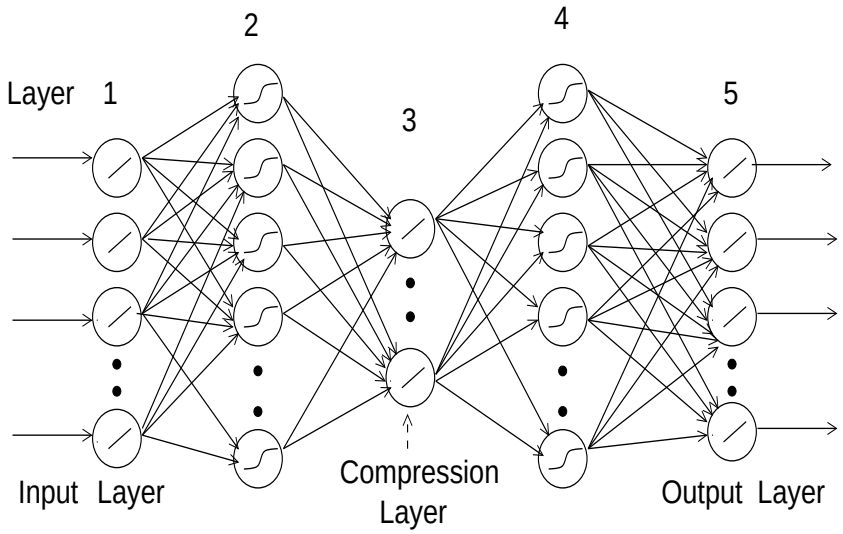


Fig. 4.2: Five layer AANN model used for nonlinear compression of pattern vectors.

nodes in the output layer. The second and fourth layers of the network have more units than the input layer. The compression layer has fewer units than the input and the output layer. The activation functions of the units in the second and fourth layers are nonlinear. The activation functions of the units in the third layer may be linear

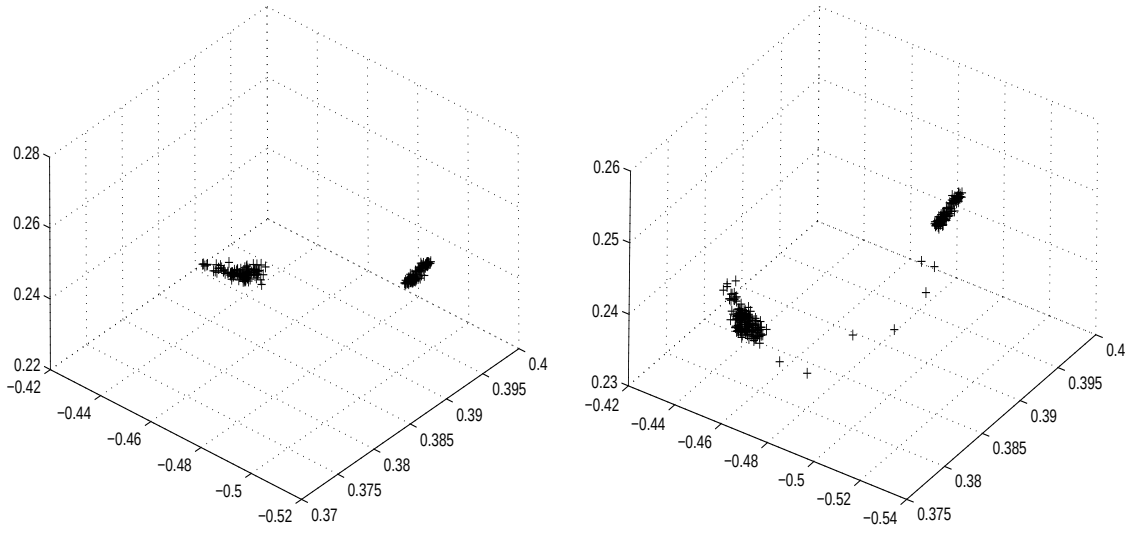
or nonlinear. Once the network has been trained, the m-dimensional input vector is transformed to a p-dimensional ($p < m$) vector at the compression layer. The output values of the units in the compression layer give a reduced dimension representation of the input vector. These reduced dimension vectors are used to detect shot boundaries.

4.3 VISUALIZATION OF EVIDENCE AT THE SHOT BOUNDARY

In our approach, the RGB color space is discretized into 125 ($5 \times 5 \times 5$) colors leading to a 250-dimensional color coherence vector for each frame. The dimension of the feature vector is reduced using an AANN model whose structure is $250L\ 375N\ pN\ 375N\ 250L$, where L denotes linear units and N denotes nonlinear units. The integer values denote the number of units in that particular layer. A video frame is now represented by a point in the p-dimensional ($p = 3$) space and allows for better visualization. The frames with similar color patterns will be mapped close to each other. Thus, in the neighbourhood of an abrupt shot transition, frames on each side result in the formation of two distinct clusters. Such a case is shown in Fig. 4.3(a), where each frame is represented by a 3-dimensional feature vector, obtained after compression using AANN model. The distance between the clusters or the margin of separation depends on the nature of frames in the neighbourhood of the abrupt shot boundary. On the other hand, a gradual transition between two shots, when viewed in a 3-dimensional space, consists of two dense clusters connected by a path, as shown in Fig. 4.3(b). Except for this path from one to cluster to another, a gradual transition is similar to an abrupt one. This reinforces the logic behind the choice of a margin of frames, in the one-pass algorithm discussed in Section. 3.1.1.

4.4 SHOT BOUNDARY DETECTION USING COMPRESSED FEATURES

In this section, we examine the effectiveness of compressed feature vectors for detection of shot boundaries. For this purpose, the compressed feature vectors are first used in



(a) (b)

Fig. 4.3: (a) Cluster patterns for abrupt transition in the low dimension space. (b) Cluster patterns formed during gradual transition in the low dimension space.

the framework of the early fusion algorithm described in the previous chapter. That is, early fusion along with two modifications, namely, one-pass processing and bidirectional processing, is performed using the 3-dimensional feature vectors. We briefly revisit these modifications in this section, and then address two issues, of validation and categorization of shot boundaries.

Shot boundary detection involves testing a hypothesis, at every frame index n of a given video, whether a shot boundary exists or not. Let $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n, \ldots, \boldsymbol{x}_{N_v}\}$ be a sequence of feature vectors, each of dimension p representing N_v frames in the video. Testing of the hypotheses at the frame index n involves computation of a dissimilarity value $d[n]$ between two sequences of N feature vectors $\mathcal{X}_L = \{\boldsymbol{x}_{n-1}, \boldsymbol{x}_{n-2}, \ldots, \boldsymbol{x}_{n-N}\}$ and $\mathcal{X}_R = \{\boldsymbol{x}_n, \boldsymbol{x}_{n+1}, \ldots, \boldsymbol{x}_{n+N-1}\}$ to the left and right of n, respectively. If the dissimilarity value $d[n]$ is greater than a threshold $\tau[n]$ (either fixed or adaptive), the hypothesis that a shot boundary exists, is chosen.

An adaptive threshold is computed using the variance of a few frames before the

frame at which the hypotheses is tested. Let $d[n] = \mathcal{D}(\boldsymbol{\mu}_L, \boldsymbol{\mu}_R)$, where $\mathcal{D}(\boldsymbol{\mu}_L, \boldsymbol{\mu}_R)$ denotes the Euclidean distance between the vectors $\boldsymbol{\mu}_L$ and $\boldsymbol{\mu}_R$ and where $\boldsymbol{\mu}_L$ and $\boldsymbol{\mu}_R$ denote the means of feature vectors in $\mathcal{X}_L$ and $\mathcal{X}_R$, respectively. If $\sigma_L[n] = \sqrt{||\boldsymbol{\Sigma}_L||}$ represents the amount of variability within a block of N frames to the left of n, then the dynamic threshold is computed as $\tau_f[n] = \beta \times \sigma_L[n]$, where $\boldsymbol{\Sigma}_L$ is the covariance matrix of the feature vectors in $\mathcal{X}_L$, and β is a scaling parameter that controls the dynamic threshold.

The first set of shot boundaries is hypothesized using the dynamic threshold based technique described above with a window size of $N = 1$. In order to reduce the number of misses, the video is also processed in the reverse (or backward) direction, which is equivalent to comparing the dissimilarity value with $\tau_b[n]$, the amount of variability to the right of n. The condition for hypothesizing a shot boundary thus becomes

$$d_f[n] > \tau_f[n] \quad | \quad d_b[n] > \tau_b[n] \tag{4.4}$$

where $d_f[n]$ and $d_b[n]$ are the distance values computed in the forward and reverse (or backward) directions, respectively. The use of 'OR' (|) logic in the bidirectional processing of the video increases the number of false hypotheses, which are reduced by validating the hypothesized boundaries using the same condition as in Eq. 4.4, but with a larger window size, say N =10. This modified condition, that uses the evidence from either side of a shot boundary, reduces the miss rate, but at the same time can increase the false alarms. The hypothesized shot boundaries need to be further validated to reduce false detections. An algorithm is now proposed for the same.

4.4.1 Validation of shot boundaries

Let n be the frame index at which a shot boundary is hypothesized. The dissimilarity values between each of the N frames $\{\boldsymbol{x}_n, \boldsymbol{x}_{n+1}, \cdots, \boldsymbol{x}_{n+N-1}\}$ to the right of n and $\boldsymbol{\mu}_L[n-M]$, the mean of N frames to the left of $(n-M)$, are computed. The hypothesized shot boundary is validated if at least $N/2$ of the N dissimilarity values are greater than $\tau_f[n]$. If the validation fails, then a similar process is repeated in the reverse direc-

tion. Hence, the condition for validating a shot boundary at the frame index n becomes

$$\sum_{k=0}^{N-1} h(d_f[n+k]) > N/2 \quad | \quad \sum_{k=0}^{N-1} h(d_b[n-k]) > N/2 \tag{4.5}$$

where

$$h(d_f[n+k]) = \begin{cases} 1, & \text{if } \; d_f[n+k] > \tau_f[n] \\ 0, & \text{if } \; d_f[n+k] \leq \tau_f[n] \end{cases} \tag{4.6}$$

and

$$h(d_b[n-k]) = \begin{cases} 1, & \text{if } \; d_b[n-k] > \tau_b[n] \\ 0, & \text{if } \; d_b[n-k] \leq \tau_b[n]. \end{cases} \tag{4.7}$$

This majority logic, apart from validating the shot boundary, eliminates spurious changes typically caused by bright flashes over a couple of frames.

4.4.2 Categorization of shot boundaries into cuts and gradual transitions

The cuts are identified from gradual transitions using the information that the variance of frames on either side of a cut is small compared to that of a gradual transition. In order to do this, the center of a transition is computed by picking the point of maximum variance around n. Two ratios of standard deviations on either side of the detected shot boundary are computed as

$$v_L = \sigma_C[n]/\sigma_L[n],$$

and

$$v_R = \sigma_C[n]/\sigma_R[n], \tag{4.8}$$

where $\sigma_L[n]$, $\sigma_R[n]$, and $\sigma_C[n]$ are the standard deviations of the N frames to the left, to the right, and around n, respectively. The shot boundary is categorized as a cut if either of the ratios is greater than a predetermined threshold, i.e., if

$$v_L > \delta \quad | \quad v_R > \delta \tag{4.9}$$

In our experiments, $N = 10$ and $\delta = 2$ are used. In addition, every cut should validate both the conditions in (4.4) and (4.5) for a margin of $M = 1$ frame.

The entire process of hypothesizing a shot boundary is summarized in Table 4.1.

Table 4.1: Summary of the algorithm.

1. Compute the m dimension color coherence vector for each frame of the video sequence.
2. Compress the m dimension CCVs to p dimension vectors.
3. At each frame index n, test the hypotheses in (3.1) using (4.4).
4. If H_0 is tested positive, go to step 5 to further validate the shot boundary. Else, increment n by one and go to step 3.
5. Validate the hypothesized shot boundary as per (4.5).
6. If the shot boundary is successfully validated, proceed to step 7. Else, increment n by one and go to step 3.
7. Identify if the shot boundary is a cut by testing the conditions in (4.4) and (4.5) with a margin of $M = 1$, along with the condition in (4.9).

4.5 EXPERIMENTAL RESULTS

The performance of the proposed shot boundary detection algorithm is evaluated on a database of video sequences given in Chapter 3, using *recall* (R), *precision* (P) and F_1 performance measures as described in previous chapter.

The performance of shot boundary detection using feature vectors of reduced dimension is discussed in this section. Table 4.2 lists the performance, when 3-dimensional feature vectors, obtained using AANN models, are used for shot boundary detection. For comparison, the performance due to early fusion algorithm is listed in

Table 4.2: Performance (in terms of R, P and F_1) of shot boundary detection using feature vectors of reduced dimension (p=3) obtained from AANN.

Video category	R	P	F_1
BBC	0.912	0.864	0.887
CNN	0.903	0.855	0.878
NDTV	0.894	0.841	0.866
Wild Life	0.915	0.867	0.890
Sports	0.854	0.785	0.870

Table 4.3. It is observed that the compression of feature vectors leads to a reduction in performance, since dimension reduction invariably results in some loss of information. However, the reduction in performance is not very significant, indicating the degree of sparsity of the input (uncompressed) feature vectors. Comparison of Table 4.2 with Table 4.4 and Table 4.5 indicates that compression using AANN models results in better performance of shot boundary detection, than using SVD or ICA. This can be attributed to the ability of AANN models to learn nonlinear basis functions, compared to the linear basis represented by SVD and ICA. Table 4.6 compares the effect of dimension reduction of feature vectors on the detection of abrupt (cuts) and gradual transition. The slightly poorer performance in the case of gradual transitions is attributed to the lack of evidence available in the reduced dimension feature vectors. In contrast, the dissimilarity metric computed using uncompressed feature vectors show greater evidence for detecting gradual transitions, due to contributions of different components of the feature vector. For cuts, however, the dissimilarity metric computed from only three dimension is enough for detection, primarily due to the extent of change in each dimension.

Table 4.3: Performance (in terms of R, P and F_1) of shot boundary detection by early fusion, using uncompressed feature vectors (250 dimension).

Video category	R	P	F_1
BBC	0.942	0.926	0.934
CNN	0.935	0.889	0.911
NDTV	0.912	0.846	0.877
Wild Life	0.908	0.882	0.894
Sports	0.892	0.934	0.912

Table 4.4: Performance (in terms of R, P and F_1) of shot boundary detection using feature vectors of reduced dimension (p=3) obtained from SVD.

Video category	R	P	F_1
BBC	0.888	0.813	0.846
CNN	0.892	0.812	0.849
NDTV	0.877	0.798	0.835
Wild Life	0.879	0.800	0.837
Sports	0.854	0.785	0.818

Table 4.5: Performance (in terms of R, P and F_1) of shot boundary detection using feature vectors of reduced dimension (p=3) obtained from ICA.

Video category	R	P	F_1
BBC	0.810	0.715	0.759
CNN	0.794	0.729	0.757
NDTV	0.771	0.700	0.733
Wild Life	0.740	0.680	0.708
Sports	0.713	0.660	0.685

Table 4.6: Performance (in terms of R, P and F_1) of shot boundary detection for cut and gradual transitions for the value of $p = 3$ using AANN models of compression.

Video	**Cuts**			**Graduals**		
category	R	P	F_1	R	P	F_1
BBC	0.942	0.886	0.913	0.882	0.843	0.862
CNN	0.934	0.878	0.905	0.872	0.832	0.851
NDTV	0.926	0.854	0.888	0.862	0.828	0.826
Wild life	0.935	0.878	0.906	0.895	0.856	0.874
Sports	0.918	0.862	0.887	0.875	0.832	0.852

4.6 SUMMARY

In this chapter, the task of temporal segmentation of video sequence into shots was performed using feature vectors of reduced dimension. The choice of color coherence vector as feature is based on its ability to represent spatial distribution of color information. Feature vectors of reduced dimension obtained using AANN models were observed to perform better shot boundary detection than those due to SVD and ICA, primarily due to the ability of AANN models to represent nonlinear basis functions from the given data. The reduction in dimension of feature vectors does not result in significant decrease in the performance of shot boundary detection, due to the sparsity of distribution of color coherence vectors. We have also proposed algorithms for categorizing a shot boundary as an abrupt or a gradual transition, and for validating the detected shot boundaries, which help in improving the overall performance of shot boundary detection.

CHAPTER 5

CLASSIFICATION OF SPORTS VIDEOS USING EDGE-BASED FEATURES

In the previous chapters, we proposed methods for detection of shot boundaries in video sequences. Having detected the shot boundaries in a video, the relevant shots can be grouped to form more meaningful units of video. Such units need to be categorized on a basis that enables efficient cataloging and retrieval with large video collections. This requires effective methods for classification of video into different genres.

The objective of video classification is to classify a given video clip into one of the predefined video categories. In this chapter, we address the problem of sports video classification for five classes, namely, cricket, football, tennis, basketball and volleyball. Sports videos represent an important application domain due to their commercial appeal. Classification of sports video data is a challenging problem, mainly due to the similarity between different sports in terms of entities such as playing field, players and audience. Also, there exists significant variation in the video of a given category collected from different television programs/channels. This intra-class variability contributes to the difficulty of classification of sports videos.

Content-based video classification is essentially a pattern classification problem [62] in which there are two basic issues, namely, feature extraction, and classification based on the selected features. Feature extraction is the process of extracting descriptive parameters from video, which will be useful in discriminating between classes of video. The classifier operates in two phases: Training and testing phase. Training is the process of familiarizing the system with the video characteristics of a given category, and testing is the actual classification task, where a test video clip is assigned a class

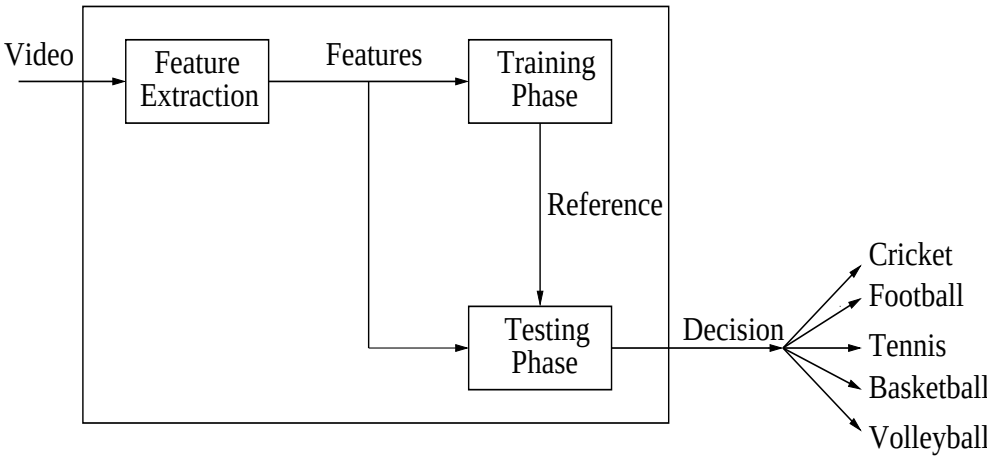


Fig. 5.1: Block diagram of video classification task.

label. A block schematic of video classification task is shown in Fig. 5.1. If there is more than one system for the task based on different features (representations) and/or classifiers, one may combine the evidence from different systems. Hence, an automatic video classification system needs to accomplish the following major tasks:

- Extracting appropriate features from the given video data.
- Generating a model for each class of video.
- Developing a decision logic for classifying a test video.
- Combining the evidence obtained from different features or classifier methodologies.

In this work, we study the effectiveness of edge-based features, namely, edge direction histogram and edge intensity histogram, for sports video classification. We demonstrate that these features provide discriminative information useful for the intended task. Three classifier methodologies, namely, autoassociative neural networks (AANN), hidden Markov models (HMMs), and support vector machines (SVMs) are used for modeling the sports categories. Evidences from the two edge based features are combined using a linear weighting rule. The application of this framework is demonstrated on five sport genre types, namely, cricket, football, tennis, basketball

and volleyball. Also, evidences from multiple classifiers are combined using a linear weighted combination, for improving the classification performance. Finally, the performance of the classification system is examined for test videos which do not belong to any of the above five categories.

This chapter is organized as follows: In Section 5.1, the extraction of edge direction histogram and edge intensity histogram for representing visual features inherent in a video class is described. Section 5.2 gives a brief introduction to the classifier methodologies used for video classification. Section 5.3 describes the combination of evidence from multiple classifiers. Section 5.4 describes experiments on video classification of the five sports categories, and also discusses the performance of the system. Section 5.5 summarizes the study.

5.1 EXTRACTION OF EDGE-BASED FEATURES

Edges constitute an important feature to represent the content of images. Human visual system is sensitive to edge-specific features for image perception. In the context of sports video classification, images that contain the playing field are significant for distinguishing among the classes of sports. This is because, each sport has its own distinct playing field where most of the action takes place. Also, the interaction among subjects (players, referees and audience) and objects (ball, goal, basket) is unique to each sport. A few sample images of each sports category are shown in Fig. 5.2. The corresponding edge images are shown in Fig. 5.3. Each playing field has several distinguishing features such as lines present on the playing field, and regions of different textures. The subjects are also prominent in the images and help in distinguishing between different sports. From Fig. 5.3, we can observe that edge features are important for representing the sports video content and carry sufficient information for human beings to distinguish among classes. These observations suggest that features derived to represent the edge information can be of significant help in automatically differentiating among various categories of sports.

(a) (b) (c) (d) (e)

Fig. 5.2: Sample images from five different sports video categories: (a) Basketball, (b) cricket, (c) football, (d) tennis and (e) volleyball.

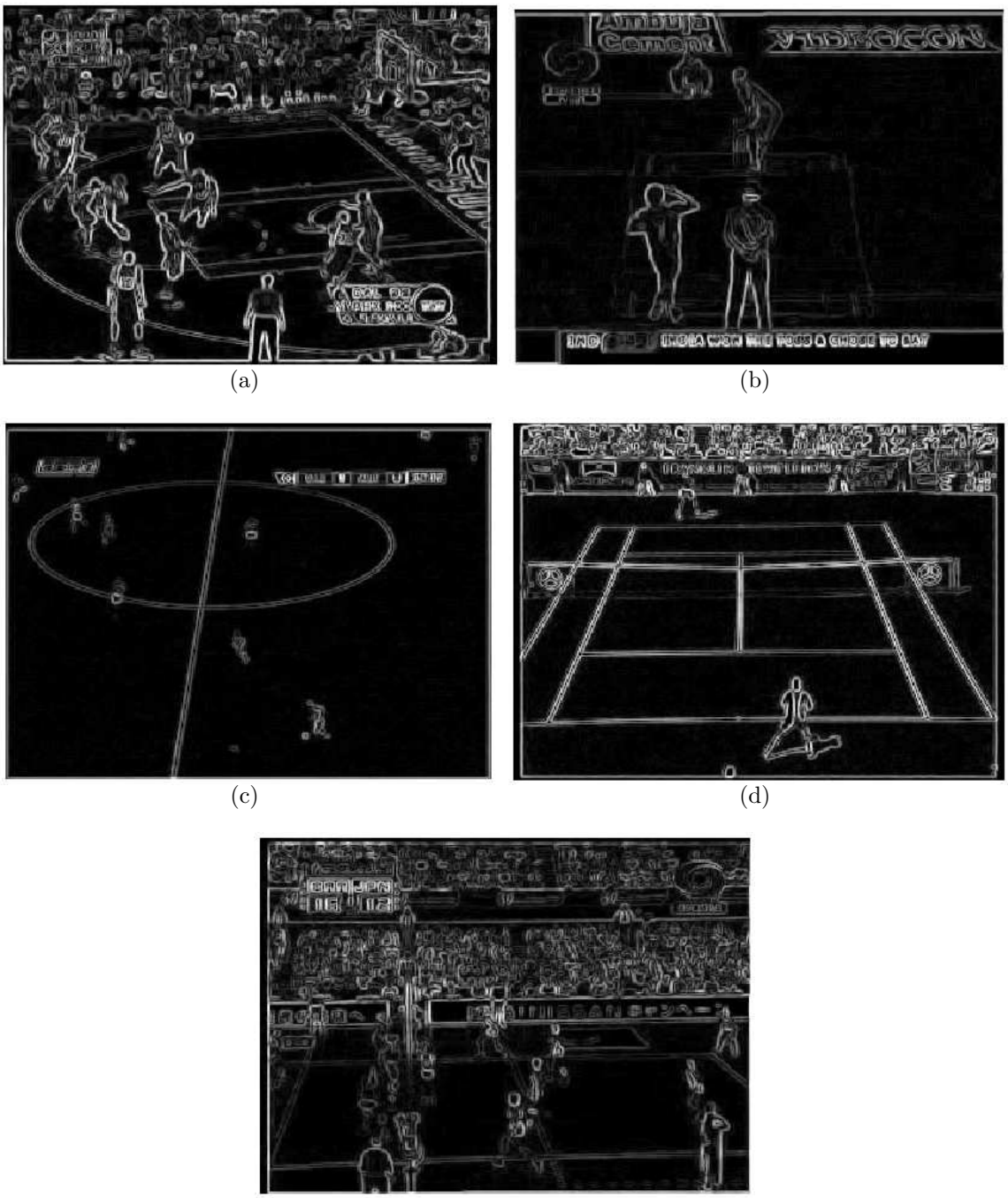


Fig. 5.3: Edge images corresponding to the five images shown in Fig. 5.2, for the sports categories: (a) Basketball, (b) cricket, (c) football, (d) tennis and (e) volleyball.

We have considered two features that can be derived to represent edge information, namely, edge direction histogram and edge intensity histogram. Edge direction histogram is one of the standard visual descriptors defined in MPEG-7 [63] for image and video, and provides a good representation of nonhomogeneous textured images. This descriptor captures the spatial distribution of edges. Our approach to compute the edge direction histogram is a modified version of the approach described in [63]. A given image is first segmented into four subimages. The edge information is then calculated for each subimage using Canny algorithm [64]. The range of the edge directions ($0^o - 180^o$) is quantized into 5 bins. Thus, an image partitioned into 4 subimages results in a 20-dimensional edge direction histogram feature vector for each frame of a video clip. Fig. 5.4 shows 20-dimensional edge direction histograms for five different categories. Each histogram is obtained by averaging the histograms obtained from individual frames of a clip. The clips were selected randomly from five different classes. The figure shows that the pattern of edge direction histogram is different for different classes and that the selected features carry discriminative information among different video classes.

We have also considered the distribution of edge intensities to evaluate the degree of uniformity of edge pixels. This feature is derived from the magnitude information of the edge pixels. The range of magnitudes ($0 - 255$) is quantized into 16 bins, and a 16-dimensional edge intensity histogram is derived from each frame of a video clip. Fig. 5.5 shows 16-dimensional edge intensity histogram for five different categories. Each histogram is obtained by averaging the histograms obtained from individual frames of a clip. The clips were selected randomly from five different classes. From Figs. 5.4 and 5.5, we observe that edge direction histogram carries more discriminative information among the classes than edge intensity histogram.

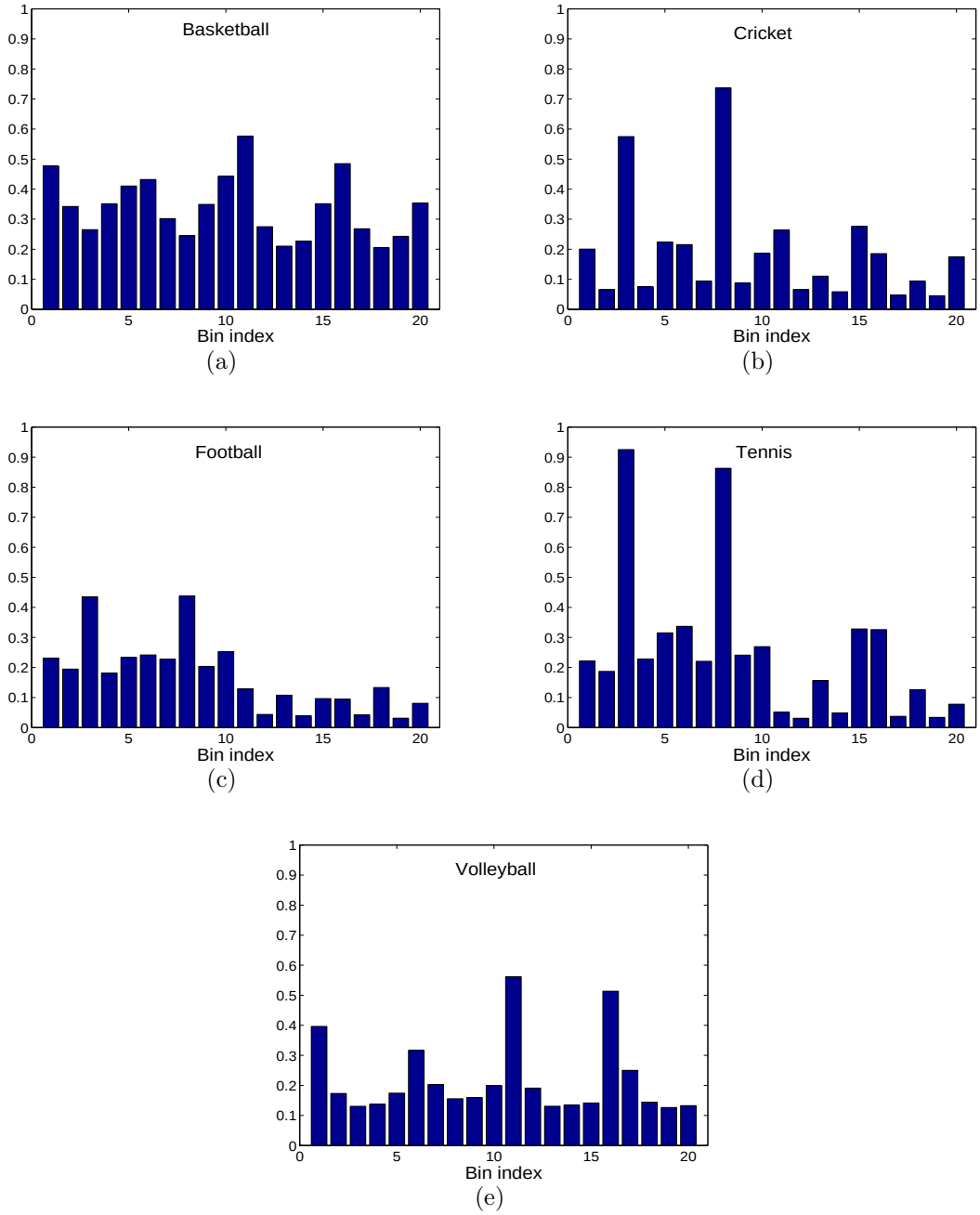


Fig. 5.4: Average edge direction histogram feature vectors of 20 dimension for sample clips selected randomly from the five different classes: (a) Basketball, (b) cricket, (c) football, (d) tennis and (e) volleyball.

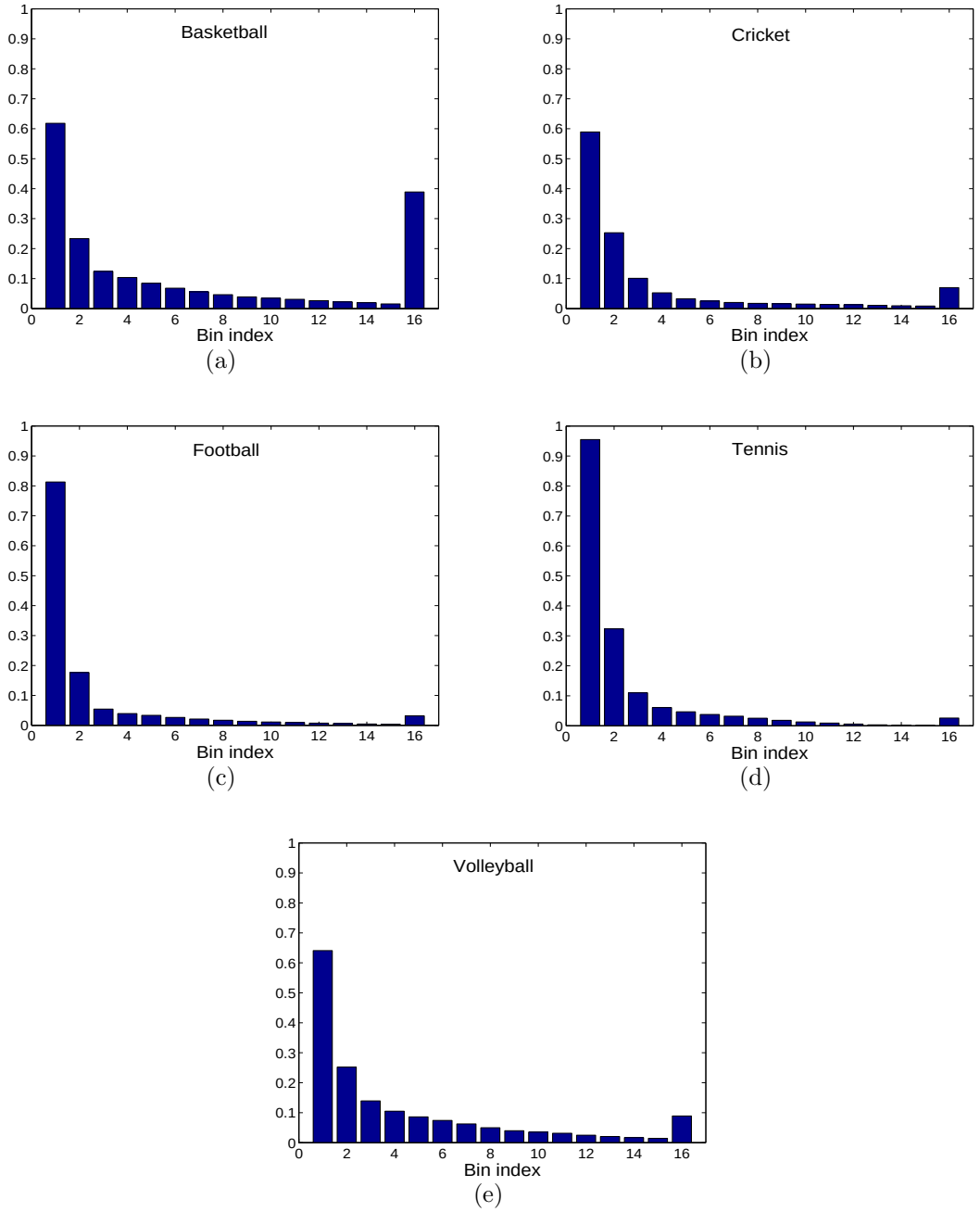


Fig. 5.5: Average edge intensity histogram feature vectors of 16 dimension for sample clips selected randomly from the five different classes: (a) Basketball, (b) cricket, (c) football, (d) tennis and (e) volleyball.

5.2 CLASSIFIER METHODOLOGIES

Once the features are extracted, the next step is to model the behaviour of features for performing classification. In this work, we have considered three classifier methodologies for our study, namely, autoassociative neural networks (AANN), hidden Markov models (HMMs), and support vector machines (SVMs). We have chosen autoassociative neural networks (AANN) to model the video content, due to their ability to capture distribution of feature vectors [65] based on the examples presented to the network. Given the temporal nature of video, and hidden Markov models (HMMs) [66] being effective tools for modeling time-varying patterns, we have chosen HMM as one of the classifier models for our study. We have also chosen support vector machines (SVMs) [67] for their inherent discriminative learning ability and good generalization performance. In the following subsections, a brief introduction to the three classifier methodologies is presented. Detailed description of the three classifier methodologies is given in Appendices B, C, and D.

5.2.1 AANN models for estimating the density of feature vectors

Autoassociative neural network (AANN) models are feedforward neural networks, performing an identity mapping of the input space [57] [52]. From a different perspective, AANN models can be used to capture the distribution of input data [65]. The distribution capturing ability of the AANN models is discussed in detail in Appendix B. In this study, separate AANN models are used to capture the distribution of feature vectors of each sports video category. A five layer AANN model is shown in Fig. 5.6. The structure of the AANN model used in the present studies is $20L\ 40N\ 6N\ 40N\ 20L$, where L denotes linear units and N denotes nonlinear units. This structure is arrived at experimentally to maximize the classification performance. The activation function of the nonlinear unit is a hyperbolic tangent function. The network is trained using error backpropagation learning algorithm for 500 epochs [57]. One epoch denotes the presentation of all the training examples (of a given class) to the neural network ex-

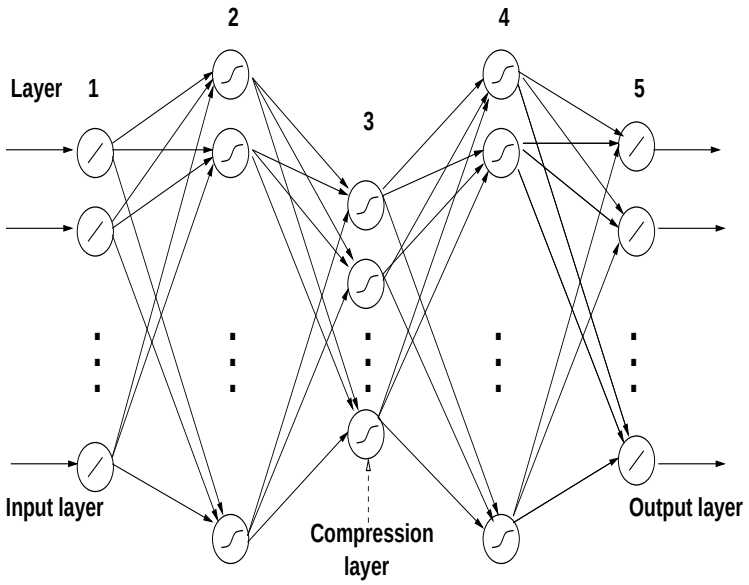


Fig. 5.6: Structure of five-layer AANN model used for video classification.

actly once. The number of epochs is chosen using cross-validation for verification, to obtain the best performance for experimental data.

The block diagram of the proposed sports video classification system based on edge direction histogram is shown in Fig. 5.7. For each video category, an AANN model based on edge direction histogram is developed. The category whose model provides the strongest evidence for a given test clip is hypothesized as the category of the test clip. A similar classification system is developed based on edge intensity histogram. Thus, the edge direction histogram and edge intensity histogram feature vectors extracted from the training data of a particular sports category are used to train two AANN models for that category, one model corresponding to each feature type. The AANN models are trained using backpropagation learning algorithm in the pattern mode [57] [52]. The learning algorithm adjusts weights of the network to minimize the mean squared error obtained for each feature vector. Once the two AANN models are trained, they are used as a model for that particular sports category.

A test video clip is processed to extract edge direction histogram and edge intensity

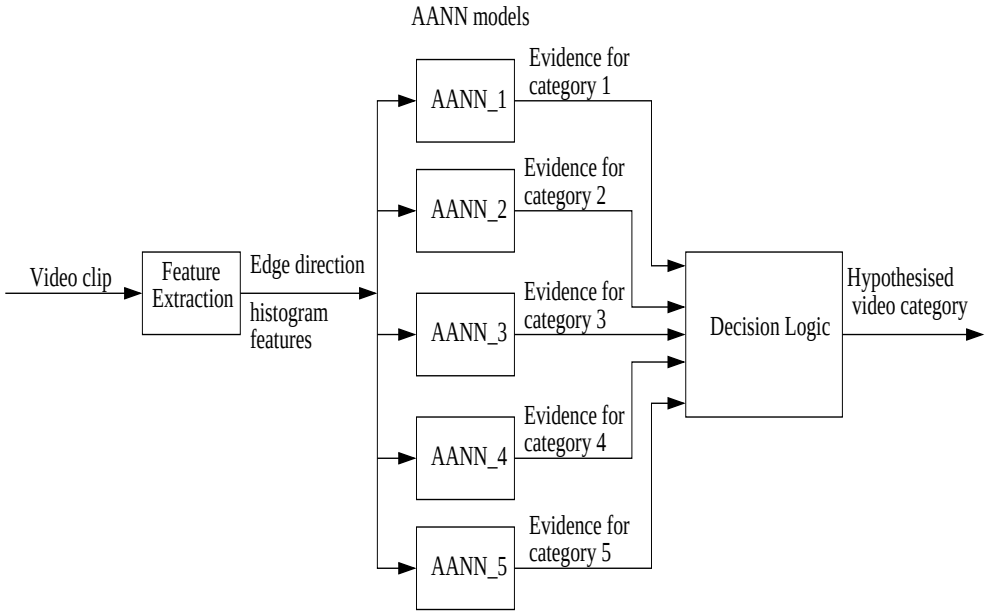


Fig. 5.7: Block diagram of the proposed video classification system using edge direction histogram features. Categories 1 to 5 are cricket, football, tennis, basketball and volleyball, respectively.

histogram features. These features are presented as input to AANN models of all the categories. The output of each model is compared with its input to calculate the squared error for each frame. The error E_k for k^{th} frame is transformed into a confidence value by using the relation $C_k = \exp(-E_k)$. A given test clip is presented to an AANN model to obtain a confidence value $C = \frac{1}{N}\sum_{k=1}^{N} C_k$ for that model, where N is the total number of frames in the test clip. For each category, two confidence values are obtained, one from each AANN model. These two scores are combined using linear weighted average rule to obtain a combined score $\hat{C}$ given by

$$\hat{C} = w \times C_d + (1 - w) \times C_i, \tag{5.1}$$

where C_d and C_i denote the confidence scores obtained from AANN models which are trained on edge direction histogram and edge intensity histogram respectively. The term w $(0 \leq w \leq 1)$ denotes the weight assigned to the score due to edge direction histogram. The value of w is chosen to maximize the classification performance for the given data set. Thus, for each test video clip, five scores are obtained. The category

whose model gives the highest confidence value is hypothesized as the sports category of the test clip. Experimental results are discussed in Section 5.4.

5.2.2 Hidden Markov models

The hidden Markov model (HMM) consists of finite number (N) of states. At each time step the system is at a given state and at the next time step the state is updated according to a probability distribution that depends only on the previous state. Additionally, at a given state a symbol is generated according to a probability distribution that depends on that state. The most likely parameters for the HMM that generate a given training set are estimated [68]. Given a model λ and an observation sequence $\mathbf{O}$, the probability $P(\mathbf{O}/\lambda)$ that this observation sequence is generated by the model λ is calculated as a sum over all possible state sequences. Efficient computation of $P(\mathbf{O}/\lambda)$ is described in Appendix C. The hidden Markov model toolkit (HTK) [69] was used for developing class-specific models. The choice of number of states ($N = 7$) and number of mixtures ($M = 1$) per state is made empirically corresponding to the best classification performance. During testing phase, given the features of a test video clip, the HMM outputs the log probability, representing the a posteriori probability that the given clip belongs to that particular class. The test methodology is similar to the block schematic shown in Fig. 5.7. Experimental results are discussed in Section 5.4.

5.2.3 Support vector machines for video classification

Support vector machines (SVMs) provide a new approach to pattern classification problems with underlying basis in statistical learning theory, in particular the principle of structural risk minimization [70]. The SVM models learn to separate the boundary regions between patterns belonging to two classes by mapping the input patterns onto a high dimensional space, and seeking a separating hyperplane in this space. The separating hyperplane is chosen in such a way as to maximize its distance (margin) from the closest training examples. More details about SVMs can be found in Appendix

D. We consider SVM models for classification due to their ability to generalize from limited amount of training data, and also due to their inherent discriminative learning [52]. The SVMTorch-II tool [71] was used for developing class-specific SVM models. When a given feature vector corresponding to a test clip is presented to an SVM model, the result is a measure of the distance of the feature vector from the hyperplane constructed as a decision boundary between a given class and the remaining classes.

The performance of pattern classification depends on the type of kernel function chosen. Possible choices of kernel function include polynomial, Gaussian and sigmoidal functions. In this work, we have used Gaussian kernel, since it was empirically observed to perform better than the other two. This class of SVMs involves two parameters, namely, the kernel width σ and the penalty parameter P. In our experiments, the value of the parameter σ is taken as the dynamic range of the features. The value of the parameter P is chosen corresponding to the best classification performance. SVMs are originally designed for two-class classification problems. In our work, multi-class ($M = 5$) 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. The test methodology is similar to the block schematic shown in Fig. 5.7. Experimental results are discussed in Section 5.4.

5.3 COMBINING EVIDENCE DUE TO MULTIPLE CLASSIFIERS

It has been shown in the literature [72–75] that combination of evidence obtained from several complementary classifiers can improve the performance of classification. There are a few reasons justifying the necessity of combining evidence from multiple classifiers/features:

1. For a pattern recognition application, there exist a number of classification algorithms developed from different theories and methodologies. For a specific problem, each of these classifiers could reach a certain degree of success, but none of them may be 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 to design one single classifier to make the decision.
3. Different features may represent complementary sources of information about a given class. Hence, combination of evidence due to different features may help in improving classification.

There are numerous types of features that can be extracted from the same raw data. Based on each of these features, a classifier or 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. Hence, we have considered the measurement level for our work. Firstly, the evidence due to two different features, namely, edge direction histogram and edge intensity histogram are combined using the rule of linear weighting, as described in Eq. 5.1. At the next level, evidence obtained from three different classifiers are combined using linear weighting. The outcome of such a combination of evidence is discussed in the next section.

5.4 RESULTS AND DISCUSSION

5.4.1 Data set

Experiments were carried out on about $5\frac{1}{2}$ hours of video data (1000 video clips, 200 clips per sports category, and each clip of 20 seconds duration) comprising of cricket, football, tennis, basketball and volleyball video categories. The video clips were captured at the rate of 25 frames per second, at 320×240 pixel resolution, and stored in AVI format. The data were collected from different TV channels in various sessions to ensure variety. For each sports video category, 100 clips were used for training, and the remaining 100 clips were used for testing.

5.4.2 Performance of different classifiers

The performance of AANN based classification system using edge direction histogram (EDH), edge intensity histogram (EIH), and combined evidence from EDH and EIH is given in Table 5.1. The performances of classification systems based on HMMs and SVMs are given in Tables 5.2 and 5.3, respectively. From the results, it can be observed that the classification performance is poorer for video clips of cricket and football categories, compared to those of tennis, basketball and volleyball categories. This is because, in the latter three categories, the playing fields have well defined lines and they appear in a majority of frames of a video clip. Moreover, a few specific camera views dominate the broadcast. For example, such a view may cover the full court in tennis or volleyball. Thus, a large area of an image frame comprises of the playing field. On the other hand, in cricket and football categories the camera view tends to change from one position to another depending on the action. Thus, continuous motion along with lack of well manifested edge-specific information results in poorer classification. It is also evident that edge direction is a stronger feature for discriminating between the classes, compared to edge intensity. One can visually perceive the content of an image from the binary edge image, which preserves only the edge directions but not

Table 5.1: Performance of AANN based sports video classification system using EDH, EIH, and combined evidence (correct classification in %). Entries in the last column denote the average performance of classification.

	Cricket	**Football**	**Tennis**	**Basketball**	**Volleyball**	**Avg. perf.**
EDH	81	84	95	94	95	89.8
EIH	54	57	93	93	92	77.8
Combined	84	88	100	100	100	94.4

Table 5.2: Performance of HMM based sports video classification system using EDH, EIH, and combined evidence (correct classification in %). Entries in the last column denote the average performance of classification.

	Cricket	**Football**	**Tennis**	**Basketball**	**Volleyball**	**Avg. perf.**
EDH	77	86	92	95	94	88.8
EIH	45	58	84	93	92	74.4
Combined	80	87	93	98	96	90.8

the magnitudes. The performance of SVM based classifier is particularly poor for EIH features compared to AANN and HMM based classifiers for the same feature. This is due to lack of discriminative information in EIH and the fact that SVMs are chosen for their discriminative ability. Since edge direction and edge intensity features can be viewed as complementary sources of information, the evidence due to these features can be combined. Tables 5.1, 5.2, and 5.3 also show the performance of classification that is obtained due to weighted combination of evidence from edge direction histogram and edge intensity histogram for different classifiers. We can observe that there is an improvement in the performance of classification due to the combination of evidence, for all the classifiers.

Table 5.3: Performance of SVM based classification system using EDH, EIH, and combined evidence (correct classification in %). Entries in the last column denote the average performance of classification.

	Cricket	**Football**	**Tennis**	**Basketball**	**Volleyball**	**Avg. perf.**
EDH	81	84	92	93	95	89.0
EIH	68	86	32	100	100	77.2
Combined	83	86	100	100	100	93.8

5.4.3 Effect of duration of test video sequence

The duration of test data (test video clip) has significant bearing on the classification performance. Several existing techniques for video classification typically use test clips whose durations vary from 60 seconds to 180 seconds [33, 34, 37, 41, 42, 76]. The classification performance in these cases is observed to improve as the duration of the test clip increases. In [34], average edge ratio used in conjunction with k-nearest neighbour algorithm requires 120 seconds of test data to yield a classification performance of 92.4% on a five-class problem. It is evident that the AANN based classifier has better generalizing ability than the k-nearest neighbour classifier, in this context. Similarly, a time-constrained clustering algorithm [41] using compressed colour features requires a minimum of 50 seconds of test data to yield a classification performance comparable to the proposed method. In contrast, the proposed method uses test clips of 20 seconds duration in all the experiments on video classification. The resulting performance, listed in Tables 5.1, 5.2 and 5.3 is comparable to that obtained due to the above methods which use a larger duration of test clip. Apart from the duration of test data, the quality of test data also influences classification performance. Some methods [41] retain only the class-specific frames in the test data by editing out images related to crowd/audience or off-field action. Such editing results in an improved performance. In our experiments, no such editing of the test data is performed.

5.4.4 Performance due to combining evidence from multiple classifiers

The normalized measurement values obtained from the three classifiers are combined using linear weighting. Table 5.5 shows classification performance obtained by combining evidence from different combinations of the three classifiers. It is observed that the combination of evidence from any two classifiers results in a performance better than those of the individual classifiers. The confusion matrix for the final classifier (combined AANN, HMM, and SVM) is given in Table 5.4. The improvement in classification due to combination of evidence can be attributed to the different classifier methodologies, which emphasize different types of information present in the features, such as their spatial distribution and temporal sequence.

Table 5.4: Confusion matrix of video classification results (in %) corresponding to the score obtained by combining evidence due to all the three classifiers (in %) (AANN, HMM, and SVM).

	Cricket	**Football**	**Tennis**	**Basketball**	**Volleyball**
Cricket	**96**	00	04	00	00
Football	02	**94**	04	00	00
Tennis	00	00	**100**	00	00
Basketball	00	00	00	**100**	00
Volleyball	00	00	00	00	**100**

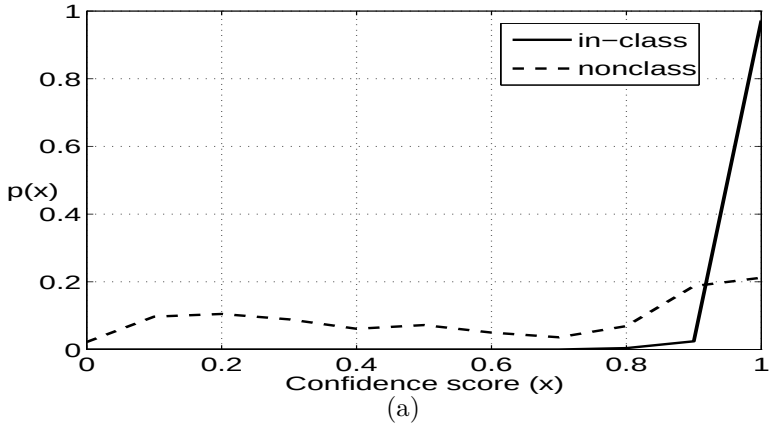
Table 5.5: Classification performance obtained by combining evidence from different classifiers (correct classification in %). Entries in the last column denote the average performance of classification.

	Cricket	**Football**	**Tennis**	**Basketball**	**Volleyball**	**Avg. perf.**
AANN	84	88	100	100	100	94.0
SVM	83	86	100	100	100	93.8
HMM	80	87	93	98	96	90.8
AANN+SVM	96	94	100	100	100	98.0
AANN+HMM	92	92	100	100	100	96.8
HMM+SVM	90	92	100	100	100	96.4
AANN+HMM+SVM	96	94	100	100	100	98.0

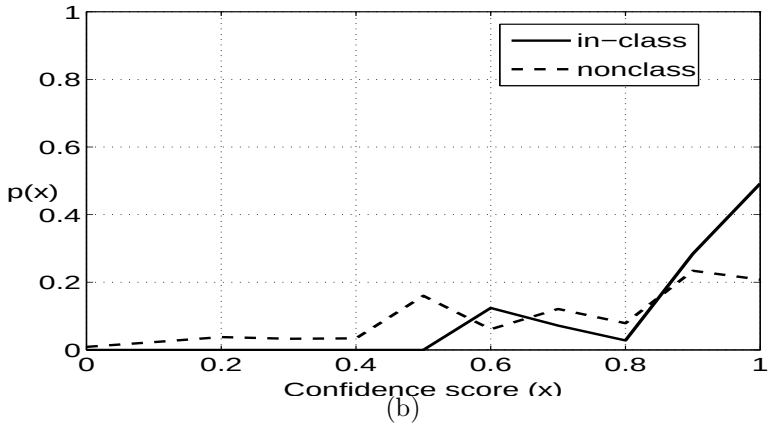
5.4.5 Verification of test video sequences using the classifiers

It is necessary to examine the response of a classifier for test inputs of a different class. More specifically, if a test video clip belongs to any class other than the above five classes, the system is expected not to assign it the label of any of the five classes. Instead, the system should assign a separate label to all such test cases. This, however, depends on two factors: (a) the nature of evidence/measurement output by a classifier and (b) the decision logic based on which a test video clip is assigned a class label. In SVM based classifiers, one-against-rest approach is used for decomposition of multi-class pattern classification problem into several two-class pattern classification problems. Hence, one should ideally get all negative confidence scores as output of the SVM model for a test clip which does not belong to any of the predefined categories. Thus, a natural threshold of zero helps in decision making in the case of SVM, although the decision could also be in error.

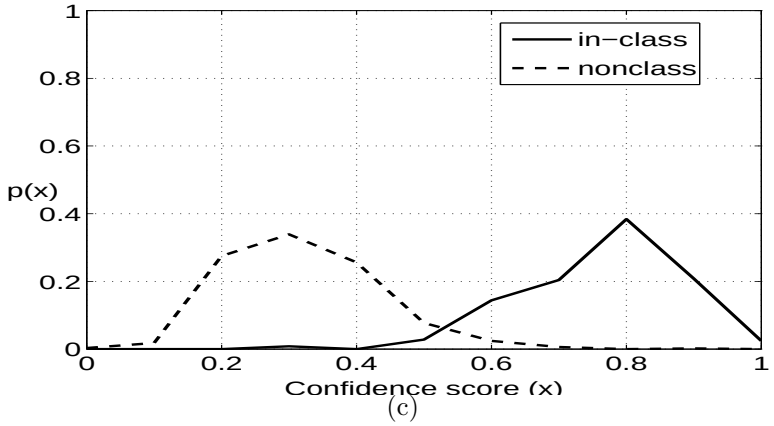
In the case of AANN models and HMMs, the training process attempts to capture only the within-class properties, and no specific attempt is made to distinguish a given class from others. Thus, a nonclass test input to these models still results in positive measurements, although small. Fig. 5.8 shows the histogram of in-class confidence scores along with that of nonclass confidence scores, for AANN models, SVMs and HMMs. The scores are normalized between 0 and 1. The in-class scores are obtained by presenting test video clips of a given category to the models of that category. The nonclass scores are obtained by presenting test video clips of a given category to the models of other categories. Hundred (100) test video clips of each class were used to obtain the in-class and nonclass confidence scores. The extent of separation of the histograms indicates the ability of the model to discriminate between in-class and nonclass examples. The area of overlap of the two histograms is a measure of minimum classification error. From Fig. 5.8, we observe that this area of overlap is least for SVM based classifier, followed by AANN based classifier. If the confidence score corresponding to the intersection of the two histograms is chosen as threshold



(a)



(b)



(c)

Fig. 5.8: Histograms of in-class confidence scores along with nonclass confidence scores for (a) AANN models (b) HMMs and (c) SVM models.

for decision, then such a choice results in minimum classification error on the training data. The same threshold is used for decision in the case of test data. Tables 5.6, 5.7, and 5.8 indicate the outcome of presenting test video clips of cartoon, commercial and news categories, to the models based on AANN, SVM, and HMM, respectively, trained on cricket, football, tennis, basketball, and volleyball. The entries in the tables denote the percentage of misclassification. For instance, if a test video clip of cartoon category, when presented to the model of cricket category, is labeled as cricket, then the test video clip is said to be misclassified. For verification, 100 test video clips of each of cartoon, commercial and news categories were used. The average misclassification is less than 15% for classifiers based on AANN and SVM. The classifier based on HMM does not seem to be very useful for discrimination. Misclassification error may be reduced further by extracting features specific to a given class.

Table 5.6: Performance of misclassification (in %) obtained from AANN models, for test clips which do not belong to any of the five sports categories. Entries in the last column denote the average performance of classification.

	Cricket	**Football**	**Tennis**	**Basketball**	**Volleyball**	**Avg. perf.**
Cartoon	08	06	02	01	01	3.60
Commercial	19	12	08	03	02	8.80
News	29	18	23	05	04	15.80

Table 5.7: Performance of misclassification (in %) obtained from SVM models, for test clips which do not belong to any of the five sports categories. Entries in the last column denote the average performance of classification.

	Cricket	**Football**	**Tennis**	**Basketball**	**Volleyball**	**Avg. perf.**
Cartoon	39	16	02	01	04	12.00
Commercial	34	03	02	01	01	8.40
News	55	12	01	02	01	14.00

Table 5.8: Performance of misclassification (in %) obtained from HMM models, for test clips which do not belong to any of the five sports categories. Entries in the last column denote the average performance of classification.

	Cricket	**Football**	**Tennis**	**Basketball**	**Volleyball**	**Avg. perf.**
Cartoon	47	16	27	01	25	22.20
Commercial	59	02	28	02	33	24.80
News	11	08	01	02	01	4.40

5.5 SUMMARY

We have presented an approach to sports video classification based on edge-specific features, namely, edge direction histogram and edge intensity histogram. We have also studied different classifier methodologies, namely, AANN, HMM and SVM. A video database of TV broadcast programs containing five sports video categories, namely, cricket, football, tennis, basketball and volleyball was used for training and testing the models. Experimental results indicate that the edge-based features can provide useful information for discriminating among the classes considered, and that edge direction histogram is a superior feature compared to edge intensity histogram. It was shown that combining evidence from complementary edge features and from different classifiers results in an improvement in the performance of classification. It was also observed that the classification system is able to decide, whether a given test video clip belongs to one of the five predefined video categories or not.

CHAPTER 6

EVENT-BASED CLASSIFICATION OF SPORTS VIDEOS USING HIDDEN MARKOV MODEL FRAMEWORK

In the previous chapter, we studied the use of edge-based features for classification of sports videos using different classifier methodologies. While the AANN based classifier attempts to model the probability distribution of edge-based features, the HMM based classifier attempts to model the information inherent in the temporal sequence. The latter, however, does not aim to model actions specific to a given sport. Each sport is uniquely characterized by certain events or actions, which are inherent in the sequence of frames. The challenge is to automatically detect these actions from the sequence of frames, so that the detected actions can be used to distinguish between sports categories. In this chapter, we propose a method based on hidden Markov model (HMM) to detect the actions in sports videos and thereby use them for classification.

We address the problem of classification of sports videos into different categories. Each sports video category can be identified using actions in that particular sport. For instance, the act of a bowler delivering the ball to a batsman is unique to the game of cricket. Similarly, a player serving the ball into the opponent's court is specific to the game of lawn tennis. These actions help a human viewer to readily identify a given sport. What is important here is the sequence of changes that are integral to an action and which qualify the action. For instance, when a bowler delivers the ball, his bowling run up, bowling action and speed of the delivery are not so much important as the act of delivering the ball. It is the act of delivering the ball that is significant, and is common across different bowlers. Such acts occur within a limited time duration. They help in uniquely characterizing a sport and can be useful for

classification, provided they can be detected automatically from the video data. In this context, an event can be defined as any significant change during the course of the game. An activity may be regarded as a sequence of certain semantic events. Automatic detection of events from raw data is a challenging task, since the variations in the raw data make it difficult interpret a change as an event. At the level of feature too, these changes cannot be observed by comparing low-level features across adjacent frames, due to variability and noise. Moreover, the changes may span over a sequence of image frames. In this context, an event can be viewed as a feature at a higher level, while an activity (sequence of events) can be viewed as a signature of a given class. This necessitates the need to model the information present in a sequence of frames.

The problem of automatic detection of events in sports videos has been addressed in literature, by modeling events that are defined a priori for a given sport (or a set of sports) [44, 45]. The main goal in such approaches is to classify the video of a given sport into different semantic events. In such approaches [44, 45], video sequences are presegmented into clips where each clip contains only one event. Another class of approaches performs automatic segmentation of the given video into shots. However, the detected events themselves are not used to distinguish between different categories of sports. In this chapter, the objective is to automatically detect events from the video sequences in a generic manner without a priori knowledge of the events and without predefining the events. Moreover, the hypothesized events are used to distinguish between classes, since the nature of events is likely to differ among the classes. We propose a probabilistic approach to detect the events, using the framework of hidden Markov model (HMM). Since the variations in the events are reflected only indirectly through the feature vectors derived from the data, HMM is a better choice to capture the hidden sequence from the observed sequence of feature vectors.

Hidden Markov models have been used in the literature for identifying activities from observation sequences [77]. Given an HMM denoted by λ and an observation sequence $\mathbf{O}$, the probability $P(\mathbf{O}/\lambda)$ that this observation sequence is generated by the model, is calculated as either the *sum* or *maximum* over all possible state sequences

to detect the activity [78]. The optimal state sequence is not driven by the events that occur during the activity, but by the likelihood of the observed data. No attempt has been made to examine the sequence of states to characterize any (hidden) activity in the sequence of events that may be present in the observed data. This study attempts to derive/interpret the sequence of events that may be present in a subset of (hidden) state sequences, and not from the raw data itself because the raw data may vary too much to interpret any change as an event [79]. In the case of sports video data, activities are characterized by certain events of interest that are embedded in the motion information, and occur within a limited time duration. These event probabilities obtained using the HMM framework are used to characterize the activity in a particular game.

The remainder of this chapter is organized as follows: In Section 6.1, we describe a method for detection of events in the framework of hidden Markov models. Section 6.2 describes the representation of motion based features for detection of events. Once the events are hypothesized, a measure of similarity is required for comparison of events obtained from reference and test video data. In Section 6.3, a method is proposed for the comparison of events. Section 6.4 discusses experiments on video classification using five sports categories, and the performance of the system. Section 6.5 summarizes the study.

6.1 DETECTION OF EVENTS USING HMM

Hidden Markov models are powerful tools for characterizing the temporal behavior of time sequences, and have been used in various ways for content-based video processing. The HMM is a Markov model in which the state is a probabilistic function of observation symbol. Typically, the number of states N is far less than the number T of observation symbols. The HMM can be described by the parameter set $\lambda = (A, B, \Pi)$ where

- $\Pi = \{\pi_1, \pi_2, \ldots, \pi_N\}$ denotes the initial state probability.

- $A = \{a_{ij}\}$ denotes the state transition matrix.
- $B = \{b_j(k)\}$ denotes the distribution of feature vectors in each state.

Given a large number of examples of an activity, the parameter set $\lambda = (A, B, \Pi)$ is estimated using Baum-Welch algorithm [66]. More details about HMMs are included in Appendix B. Once the parameter set λ is estimated, the probability of a test observation sequence $\mathbf{O} = (\mathbf{o}_1\mathbf{o}_2\mathbf{o}_3 \ldots \mathbf{o}_T)$ being generated by the model λ can be computed in two ways:

- Sum over all the possible state sequences
  $P(\mathbf{O}/\lambda) = \sum_{\{q_1,q_2,\ldots,q_T\}} P(q_1q_2 \ldots q_T, \mathbf{o}_1\mathbf{o}_2 \ldots \mathbf{o}_T/\lambda)$.

- Maximum of all the possible state sequences
  $P(\mathbf{O}/\lambda) = \max_{\{q_1,q_2,\ldots,q_T\}} P(q_1q_2 \ldots q_T, \mathbf{o}_1\mathbf{o}_2 \ldots \mathbf{o}_T/\lambda)$.

The key idea in this traditional HMM formulation is that the sum or the maximum over all possible state sequences is considered in evaluating the probabilities. But the optimal state sequence obtained using these methods is not driven by the events that occur during the activity, rather by the likelihood of the observed data. In this kind of formulation, there is no attempt to examine the sequence of states to characterize any hidden activity in the form of sequence of events that may be present in the observed data.

We propose a method to examine a subset of all the possible state sequences and explore the possibility of interpreting them as sequence of events. The hypothesis is that, though the state sequences themselves may look different across the examples of the same activity, certain (hidden) state transitions may be preserved in the subset of state sequences, and we call such transitions as events.

Exploring events in sequence of states

Let a given sports video clip be represented by an observation symbol sequence $\mathbf{O} = (\mathbf{o_1 o_2 o_3} \ldots \mathbf{o}_T)$. Suppose that the underlying activity responsible for the production of the observation symbol sequence can be described by K events occurring at times $\tau_1, \tau_2, \tau_3, \ldots, \tau_k$. So, to represent this activity, we need to detect the number K of events, the nature of events and the time instants at which they occur. As these events are localized in time, it is reasonable to expect that an event at time t is affected by the observations in its immediate neighbourhood. Hence, we define a variable $\eta_t^p(i,j)$, given by [79]

$$\eta_t^p(i,j) = P(q_{t-p} = i, q_{t-p+1} = i, \ldots, q_t = i, q_{t+1} = j, q_{t+2} = j, \ldots, q_{t+p} = j/\mathbf{O}, \lambda), \tag{6.1}$$

where $2p+1$ frames around t are considered. The superscript p refers to support of p frames used on either side of the time instant t in order to detect an event. The $\eta_t^p(i,j)$ can be written as

$$\begin{aligned} \eta_t^p(i,j) &= \frac{P(q_{t-p} = i, q_{t-p+1} = i, \ldots, q_t = i, q_{t+1} = j, q_{t+2} = j, \ldots, q_{t+p} = j, \mathbf{O}/\lambda)}{P(\mathbf{O}/\lambda)} \\ &= \frac{\alpha_{t-p}(i) a_{ii}^p b_i(\mathbf{o}_{t-p+1}) \ldots b_i(\mathbf{o}_t) a_{ij} b_j(\mathbf{o}_{t+1}) \ldots b_j(\mathbf{o}_{t+p}) a_{jj}^p \beta_{t+p}(j)}{P(\mathbf{O}/\lambda)} \end{aligned} \tag{6.2}$$

where α and β are the forward and backward variables [78]. We define one more variable e_t^p, similar to the one that is used in Viterbi maximization, as

$$e_t^p(k,l) = \max_{i,j} \eta_t^p(i,j) \qquad i \neq j \tag{6.3}$$

where

$$(k,l) = \arg\max_{i,j} \eta_t^p(i,j) \qquad i \neq j. \tag{6.4}$$

Here $e_t^p(k,l)$ represents the probability with which there can be a transition from stable state k to stable state l, with a support of p frames for stable states, at the time instant t. At every instant of time, we hypothesize an event, and evaluate the event probability $e_t^p(k,l)$. Large values of e_t^p indicate the presence of an event, and the corresponding

(k, l) pair indicate the state transition responsible for the event. The nature of the event is specified by state pair (k, l), and the intensity of the event is specified by the value of e_t^p.

For every symbol $\mathbf{o}_t$ in the given observation symbol sequence $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T\}$, an event probability value e_t^p and a state transition (k, l) are associated. The value of p determines the support given to the stable states on either side of the transition. A small value of p results in too many spurious events, whereas a large value of p might result in missing an event of interest totally. Hence, the value of p is determined by using the domain knowledge and some preliminary experimental studies. The event probability sequence e_t^p and the corresponding state transitions (k, l) form the signature for the activity in a particular sports category.

This idea of detection of events from the sequence of observation vectors has been examined for recognition of utterances of isolated digits [80]. It was observed in [80] that certain state transitions were preserved across different speakers for a given sound unit. In the present context, we observe whether a given state transition is common to different video clips of a particular sports category. We also observe whether different sets of state transitions are prominent for different sports.

6.2 FEATURES FOR DETECTION OF EVENTS

Motion is an important cue for understanding video and widely used in semantic video content analysis. Since features based on motion carry important information about the temporal sequence corresponding to a given sports category, we use motion-based features for event detection. The approach adopted here for extraction of motion information from the video is based on the work by Matthew et al. [81]. From the video sequence, we derive the binary maps as shown in Fig. 6.1. These binary maps are representative of moving and non-moving areas of the video sequence, where moving areas are highlighted in white. The binary maps are extracted by pixel-wise differencing of consecutive frames. We divide each frame into four sub-images of equal size in

Fig. 6.1: Examples of binary maps for different sports categories. Each row shows two consecutive frames and the corresponding binary map for five different sports, namely, (a) basketball (b) cricket (c) football (d) tennis and (e) volleyball.

order to capture the location specific information. The motion feature is computed as follows:

$$M(t) = \frac{1}{w * h} \sum_{x=1}^{w} \sum_{y=1}^{h} P_t(x, y), \quad 0 < t \leq N, \tag{6.5}$$

where

$$P_t(x, y) = \begin{cases} 1, & \text{if } |I_t(x, y) - I_{t-1}(x, y)| > \beta \\ 0, & \text{otherwise.} \end{cases} \tag{6.6}$$

In the above equation, N is the total number of frames in the video clip, $I_t(x, y)$ and $I_{t-1}(x, y)$ are the pixel values at location (x, y) in t^{th} and $(t-1)^{\text{th}}$ frames, respectively. Here β is the threshold, and w and h are width and height of the subimage, respectively. A 4-dimensional feature vector is derived from each pair of consecutive frames. Thus, the sequence of 4-dimensional feature vectors derived from a video clip of a particular sports category forms one observation symbol sequence for that category.

6.3 MATCHING OF EVENTS

We now describe a method for matching of events between a test video sequence and a reference video sequence. The block diagram of the proposed sports video classification system using HMM framework is shown in Fig. 6.2. Given L_1 observation symbols of a video category, a five state ergodic HMM model with one Gaussian per state is trained with motion features extracted from the video frames. The events (event probabilities and corresponding state transition) corresponding to the reference examples are obtained. These reference event sequences are used to create a dictionary of events for the given sports category. The distribution of the event probability values corresponding to a particular state transition (k, l) in the reference events is

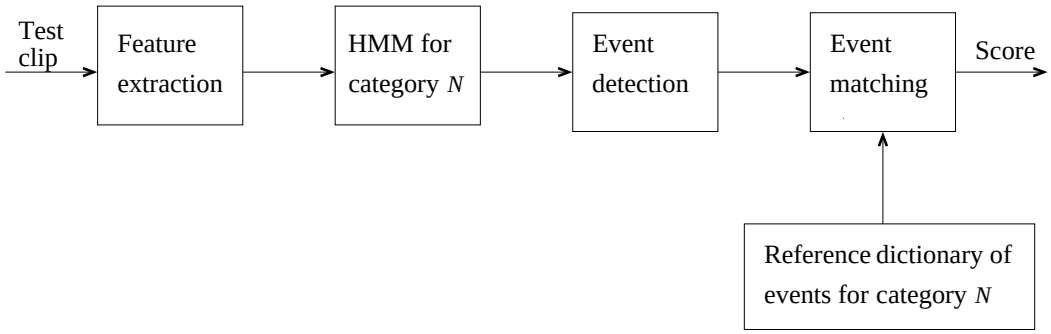


Fig. 6.2: Block diagram of the proposed video classification system using HMM framework.

approximated by a Gaussian density $\mathcal{N}(\mu_{kl}, \sigma_{kl})$, where μ_{kl} and σ_{kl} denote mean and variance of the density function, respectively, given by

$$\mu_{kl} = \frac{1}{L_1} \sum_{t=1}^{L_1} e_t^p(k,l) \tag{6.7}$$

and

$$\sigma_{kl} = \sqrt{\frac{1}{L_1} \sum_{t=1}^{L_1} (e_t^p(k,l) - \mu_{kl})^2}. \tag{6.8}$$

So, every state transition is assigned a mean and a variance which represents the probabilities with which the event $e_t^p(k,l)$ occurs in that category. For a test video clip not presented during training, the events are obtained using a reference HMM. Let us denote by $\hat{e}_t^p(k,l)$, the event probability corresponding to the state transition (k,l) at time t, when a test sequence of observation symbols is presented to a reference model. Let L_2 denote the number of observation symbols in the test sequence. A similarity score s between the test video clip and the reference model is given by

$$s = \frac{1}{L_2} \sum_{t=1}^{L_2} \frac{1}{\sqrt{2\pi}\sigma_{kl}} \exp\left[-\frac{(\hat{e}_t^p(k,l) - \mu_{kl})^2}{2\sigma_{kl}^2} \right]. \tag{6.9}$$

There exists a possibility that two different categories may have similar distributions of event probabilities. In such a case, it is necessary to examine the sequence of events, which may help in discriminating between the two categories.

6.4 EXPERIMENTAL RESULTS

Experiments were carried out on about $5\frac{1}{2}$ hours of video data (1000 video clips, 200 clips per sports category, and each clip of 20 seconds duration) comprising of cricket, football, tennis, basketball, and volleyball categories. The video clips were captured at a rate of 25 frames per second, at 320×240 pixel resolution, and stored in AVI format. The data were collected from different TV channels in various sessions to ensure variety. For each sports video category, 100 clips were used for training and the remaining 100 clips were used for testing.

The choice of the number of states (N) of HMM is critical. It is difficult to arrive at the number of states from the physical process, since the process is not explicit. Hence, N needs to be determined experimentally, by observing the classification performance. The classification performance was observed by varying N between 3 and 10. The values $N = 7$ and $N = 9$ were chosen based on the performance of classification. Also, the choice of the value of p is critical since it determines the support given to the stable states on either side of the transition. A small value of p results in too many spurious events, whereas a large value of p might miss an event of interest. Hence, the value of p is determined by using some preliminary experimental studies.

An ergodic HMM model is built for each class using 4-dimensional motion feature vectors. The performance of the event-based HMM classifier is given in Table 6.1 for the case with number of states $N = 7$ and number of mixtures $M = 1$. In Table 6.1, the entries in parenthesis denote the classification performance for $N = 7$ and $M = 2$. The classification performance for $N = 9$ and $M = 1, 2$ is given in Table 6.2. It is observed that the average performance for $N = 7$ is better for all sports except football. This can be attributed to the relatively low motion of the four sports categories basketball, cricket, tennis, and volleyball compared to football, where dynamic variations are better modeled using more number of states ($N = 9$). This greater variation also necessitates the choice of two mixtures per state ($M = 2$) for improved classification in the case of football category, as shown in Table 6.2.

Table 6.1: Performance of correct video classification for $N = 7$, and $M = 1$ (in %). The entries in parenthesis denote the performance for $N = 7$ and $M = 2$. The parameter p denotes the number of frames provided as support on either side of the time instant t. Entries in the last column denote the average performance of classification.

	Basketball	**Cricket**	**Football**	**Tennis**	**Volleyball**	**Avg. perf.**
p=5	92 (62)	68 (56)	76 (58)	78 (80)	96 (96)	**82.0** (70.4)
p=7	90 (66)	58 (52)	80 (62)	78 (84)	98 (96)	80.8 (**72.0**)
p=10	82 (66)	56 (32)	88 (72)	78 (80)	98 (80)	80.4 (66.0)
p=13	72 (46)	56 (20)	90 (62)	80 (88)	98 (58)	79.2 (54.8)
p=15	60 (34)	48 (20)	92 (60)	74 (90)	98 (62)	74.4 (53.2)
p=17	48 (20)	42 (18)	92 (56)	74 (92)	98 (70)	70.8 (51.2)

The confusion matrix for the best average classification performance ($N = 7$, $M = 1$, and $p = 5$) is given in Table 6.3. The relatively poorer performance for cricket and football categories can be attributed to the inability of the models to detect the events. Large playing fields in cricket and football categories result in significant camera motion within a given time, compared to other categories. Hence, the number of examples of a given event is lesser in the training data, leading to poor representation of events.

Using the above method, we could detect significant changes in each sports category using motion information. For example, some of the events detected in cricket category are bowler releasing the ball, batsman hitting the ball, fielder picking up the ball, and fielder throwing the ball. Two such cases events, of bowler releasing the ball and fielder picking the ball, are shown in Fig. 6.3. Similarly, some of the events detected in other sports categories are shown in Figs. 6.4, 6.5, 6.6, and 6.7.

Table 6.2: Performance of correct video classification for $N = 9$, and $M = 1$ (in %). The entries in parenthesis denote the performance for $N = 9$ and $M = 2$. The parameter p denotes the number of frames provided as support on either side of the time instant t. Entries in the last column denote the average performance of classification.

	Basketball	**Cricket**	**Football**	**Tennis**	**Volleyball**	**Avg. perf.**
p=5	94 (68)	56 (06)	96 (98)	60 (60)	82 (74)	**77.6** (**61.2**)
p=7	84 (58)	48 (06)	98 (98)	60 (62)	78 (48)	73.6 (54.4)
p=10	76 (22)	36 (08)	98 (98)	60 (64)	82 (10)	70.4 (40.4)
p=13	58 (04)	34 (10)	98 (98)	60 (62)	88 (06)	67.6 (36.0)
p=15	44 (04)	38 (08)	98 (98)	60 (62)	86 (09)	65.2 (35.6)
p=17	44 (04)	24 (10)	96 (98)	60 (62)	84 (02)	61.6 (34.8)

Table 6.3: The confusion matrix for the best classification performance (in %) ($N = 7$, $M = 1$, and $p = 5$).

	Basketball	**Cricket**	**Football**	**Tennis**	**Volleyball**
Basketball	**92**	00	00	02	06
Cricket	02	**68**	16	12	04
Football	06	14	**76**	04	00
Tennis	00	06	16	**78**	00
Volleyball	00	04	00	00	**96**

(a) (b)

Fig. 6.3: Sequence of image frames (from top to bottom) of two events of cricket category where the event of (a) bowler releasing the ball and (b) fielder picking up the ball are detected. The detected events are marked by a circle.

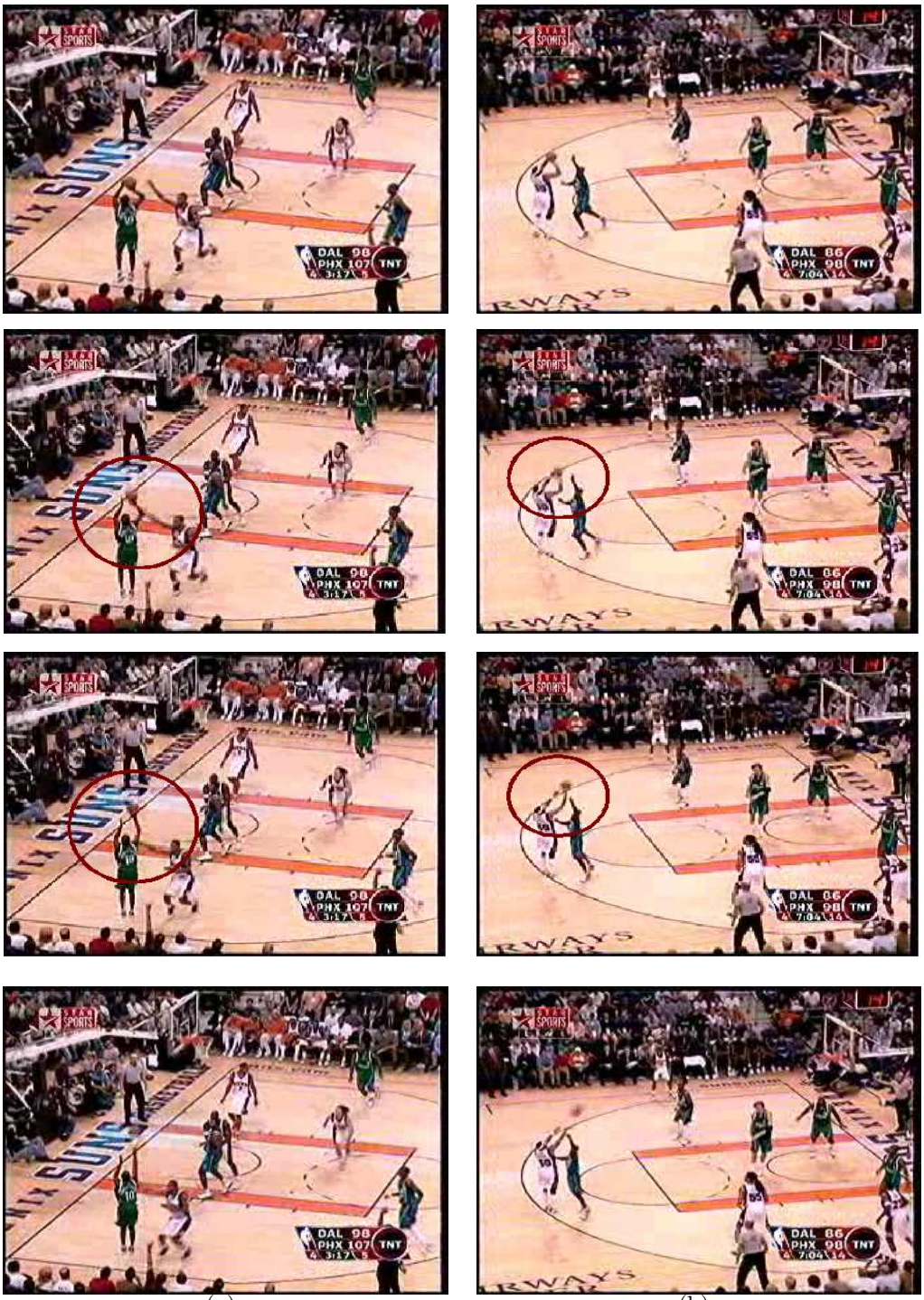
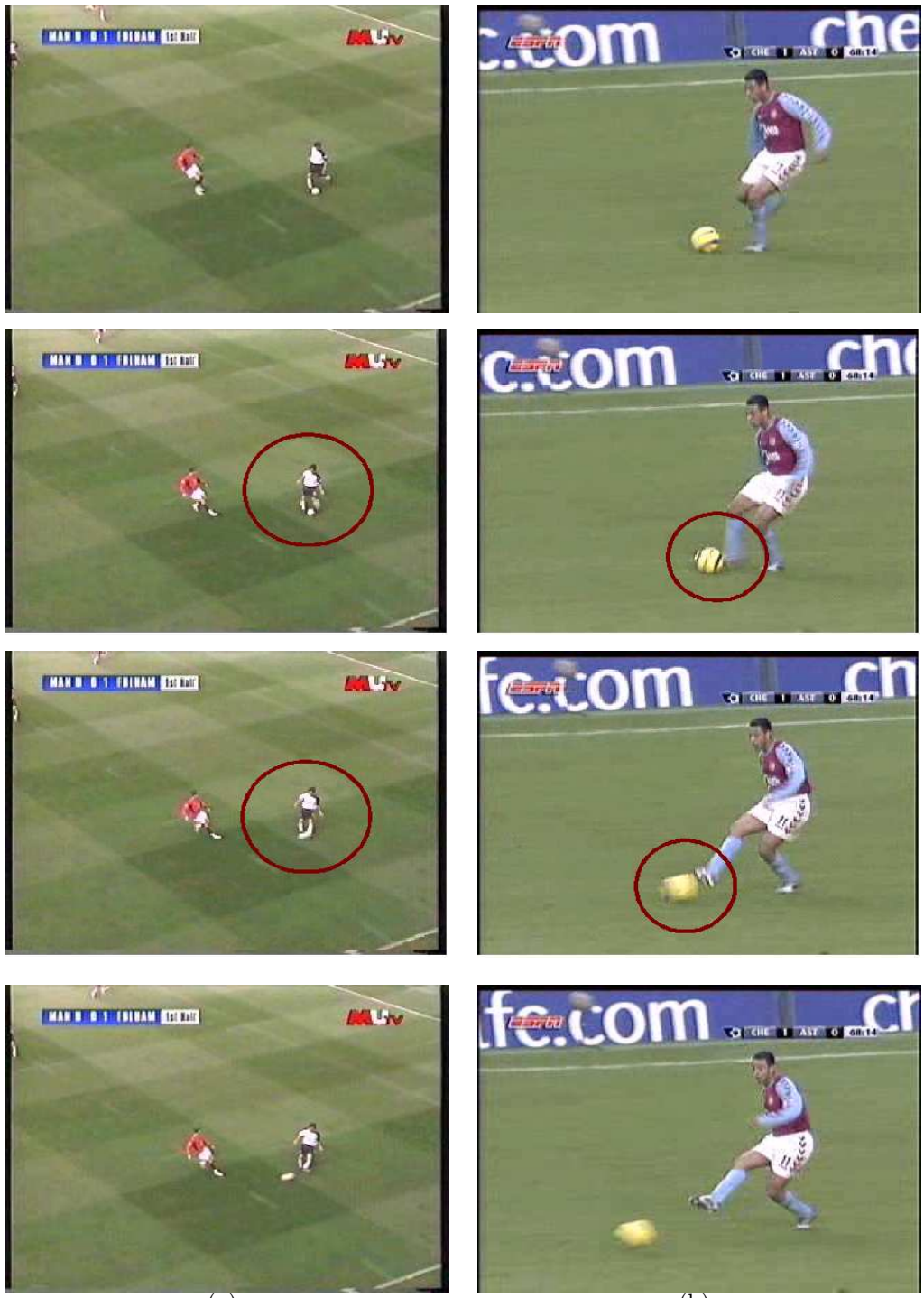


Fig. 6.4: Sequence of image frames (from top to bottom) of basketball category where the event of player throwing the ball is detected. Two examples of such an event are shown in (a) and (b). The detected events are marked by a circle.



(a) (b)

Fig. 6.5: Sequence of image frames (from top to bottom) of football category where the event of player passing the ball is detected. Two examples of such an event are shown in (a) and (b). The detected events are marked by a circle.

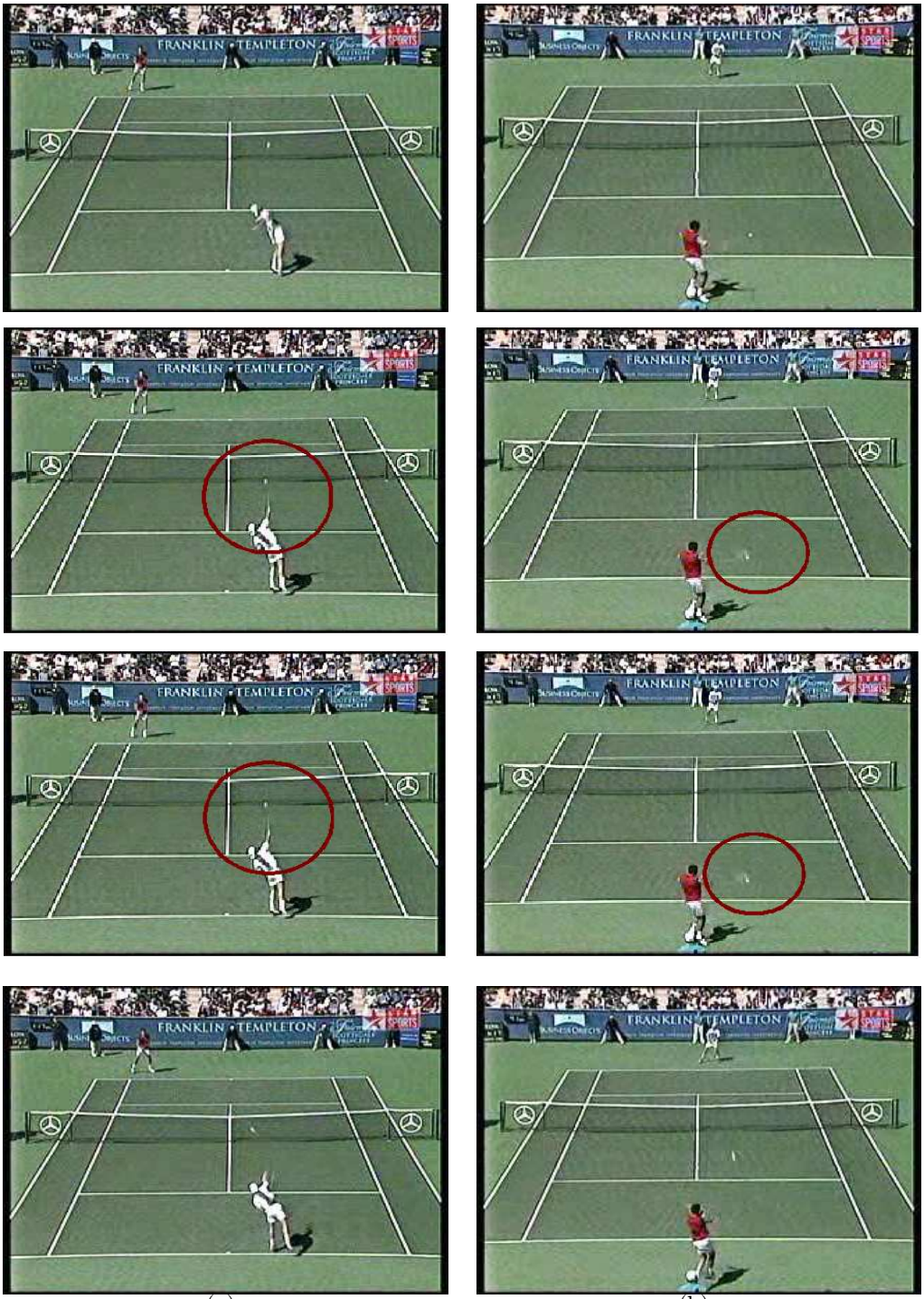


Fig. 6.6: Sequence of image frames (from top to bottom) of two events of tennis category where the events of (a) serving the ball and (b) playing a forehand shot are detected. The detected events are marked by a circle.

(a) (b)

Fig. 6.7: Sequence of image frames (from top to bottom) of two events of volleyball category where the events of (a) playing an underarm shot and (b) smashing the ball are detected. The detected events are marked by a circle.

6.5 SUMMARY

We have presented a technique for classification of sports videos using events to represent class-specific information. The events were detected using a framework based on hidden Markov models. Each sports video category can be identified using actions in that particular sport. Activities are characterized by a sequence of semantic events that are embedded in the motion, and occur within a limited time duration. The event probabilities were used to characterize the activity. Video classification was performed based on the similarity score obtained by matching the events. A video database of TV broadcast programs containing five popular sports categories, namely, basketball, cricket, football, tennis, and volleyball was used for training and testing the models. A correct classification of 82.0% has been achieved. Classification performance can be improved by using sequence knowledge during score computation.

CHAPTER 7

SUMMARY AND CONCLUSIONS

In this thesis, new approaches were proposed to address some issues in video segmentation and classification. These two tasks are important in the context of video content analysis, and present some challenging problems. Video segmentation involves the partitioning of a given video sequence into smaller and more meaningful units. In this thesis, issues specific to detection of shot boundaries in video sequences were addressed. The key issue is to derive features which can help detect a change in video sequence due to a shot boundary, and which are robust to illumination or camera/object motion. A novel method based on the late fusion of evidence was proposed for addressing this issue, which detects significant change in a few components of color histogram feature for hypothesis of shot boundaries. The decision due to the late fusion method was combined with that due to the existing approach of early fusion. Since early fusion depends on the extent of overall change in features and late fusion depends on a few significant changes, the combination improves the robustness of shot boundary detection. We also proposed modifications to the traditional early fusion algorithm for improving the performance of shot boundary detection. Firstly, a one-pass algorithm was proposed for simultaneous detection of abrupt and gradual transitions. The basis for this method is that, barring the region of gradual transitions, an abrupt and a gradual transition are essentially similar. Secondly, bidirectional processing of video was proposed in order to reduce the number of missed detections. Thirdly, the hypothesized shot boundaries were validated on the basis of majority logic. Finally, a method was proposed to classify a detected shot boundary as a cut or a gradual transition, using a measure of variance. These modifications, in conjunction with early and late fusion, were shown to reduce the criticality of the choice of threshold for hypothesizing

the presence of shot boundaries. The proposed methods yield robust performance over a range of threshold values.

Another important issue is the dimension of feature vector used for representation of images. In this thesis, color histogram of 512 dimension has been used for this purpose. Since color histograms do not represent spatial distribution of color, color coherence vector was used for representation of images. Both these feature vectors are sparse and can be represented using a much smaller dimension. Feature vectors of reduced dimension were obtained using linear compression schemes such as independent component analysis (ICA) and singular value decomposition (SVD), and nonlinear autoassociative neural network (AANN) models. It was shown that reduction in the dimension of feature vectors does not result in significant decrease in performance of shot boundary detection, due to the sparsity of distribution of color features. Feature vectors of reduced dimension obtained using AANN models perform better than those due to SVD and ICA for shot boundary detection, primarily due to the ability of AANN models to represent nonlinear basis functions from the given data. The use of linear and nonlinear compression techniques provides compact representation, better visualization and the option for multiple validations at low computational cost.

The key issues in video classification are representation of class-specific information using suitable features, and developing models to capture information present in the features. In this thesis, these issues were addressed in the context of classification of sports videos, using two different approaches. The first approach was based on the use of edge-based features to represent class-specific information, while the second approach was based on the hypothesis of events from video sequences. In the first approach, edge direction histogram and edge intensity histogram features were used in conjunction with three different classifier techniques, based on autoassociative neural networks (AANN), support vector machines (SVM) and hidden Markov models (HMM). The AANN models were used to capture class-specific distributions of edge-based features, while the HMMs were used to model the sequence information present in the features. Models based on SVM were also explored as they incorporate differen-

tial training to determine a hyperplane that separates the examples of a given category from those of other categories. Edge direction histogram and edge intensity histogram were shown to have complementary evidence. Also, combination of evidence due to the different classifiers resulted in an improvement in the performance of classification, illustrating the complementary nature of the modeling techniques. The ability of the classification system to decide whether a given test clip belongs to any one of the five sports categories or not, was studied.

The second approach to video classification is based on detection of events in video sequences. The events denote significant changes in video, and a sequence of events denotes an activity. The activities are characterized by a sequence of semantic events embedded in the motion, and occur within a limited time duration. The activities are intended to correspond to physical actions in different sports. The events and the activities are detected using a framework based on hidden Markov models. The physical events are not defined a priori. Instead, the model is trained to hypothesize the events, by presenting several example sequences of a given sport. Given a test video sequence, a similarity score is computed by matching the events in the test sequence with those obtained from reference data. A video database of TV broadcast programs containing five sports categories was used for training and testing the models. A correct classification of 82.0% has been achieved.

7.1 CONTRIBUTIONS OF THE WORK

The contributions of the research work carried out as part of this thesis can be summarized as follows:

1. A new method based on late fusion of evidence from individual color components was proposed for detecting shot boundaries in video sequences. This was based on the observation that most of the shot boundaries are characterized by significant changes in only a small number of components of color histogram feature.

2. Algorithms were proposed for simultaneous detection of abrupt and gradual transitions, and for categorization of the detected shot boundaries into the above two groups. These algorithms are based on effective usage of statistics derived from the neighbourhood of shot boundaries.

3. The combination of early fusion and late fusion, along with the proposed algorithmic modifications, were shown to improve the performance of shot boundary detection and reduce the criticality of threshold.

4. The ability of AANN models to perform nonlinear compression was exploited for reducing the dimension of color coherence feature vector. Such a representation allows for multiple validations without significant reduction in the performance of shot boundary detection.

5. The ability of AANN models for capturing distribution of feature vectors was exploited for classification of sports videos, using edge direction histogram and edge intensity histogram.

6. The combination of evidence due to complementary features (edge direction histogram and edge intensity histogram) and different classifiers (AANN models, SVM and HMM) was shown to improve the performance of classification. The classification system can also be used to verify whether a given video sequence belongs to one of the predefined classes or not.

7. A new method was proposed for classification of sports videos, by hypothesizing events in each sports category using the framework of hidden Markov models. These events are not defined a priori and are detected from the video sequences presented during training phase.

7.2 DIRECTIONS FOR FURTHER RESEARCH

In this thesis, color based features such as color histogram and color coherence vector were used for representation of images for shot boundary detection in video sequences. These are one-dimensional representations obtained from three-dimensional color histograms. An important issue is the mapping of the three-dimensional color space onto one dimension, followed by a dissimilarity measure that preserves the proximity information during the mapping. One solution to this problem is to provide a fuzzy border while quantizing the color space.

The problem of video classification was addressed on the basis of events detected in each sports category. Here, events obtained from a test sequence were compared with those obtained from training sequences, without exploiting information present in the sequence of events. Dynamic programming techniques can be explored to match the sequences of events, to obtain better classification performance. The proposed method can also be used to detect the events that are specific to a given sport. The basic idea is that a state transition that commonly occurs across different clips should model a particular event. This can also help in classifying a given sport into different conceptual categories. The performance of the video classification system can be improved by combining the evidence from other modalities, such as audio and text, with the visual evidence.

Automatic video content analysis to generate the video-table-of-contents is a natural application for the two tasks discussed in this thesis. This would require higher levels of segmentation of the video, by combining adjacent and relevant shots into larger units like scenes and stories. The classification studies presented in this thesis can help in this process of labeling and merging of individual shots.

APPENDIX A

EXISTING SYSTEMS FOR VIDEO BROWSING AND RETRIEVAL

Researchers have developed numerous schemes and tools for video indexing and query. A brief description of some of the existing systems is given below.

Informedia-The CMU digital Video Library: The Informedia Digital Video Library Project [82] at Carnegie Mellon University is an on-line digital video library, which allows for full-content and knowledge-based search and retrieval using desktop computer over local, metropolitan, and wide-area networks [83, 84]. The library contains news and documentary video. The system integrates speech recognition, natural language understanding, and image processing for multimedia content analysis. The system contains methods to create short synopsis of each video. Language understanding is applied to the audio track to extract meaningful keywords. Each video in the database is represented as a group of representative frames extracted from the video at points of significant activity. Caption text is also extracted from these frames, which adds to the set of indices for the video.

AT&T Pictorial Transcript System: Pictorial Transcript is an automated archiving and retrieval system for broadcast news program, developed at AT&T Labs [85]. Combined audio-visual analysis has been used to automatically generate the content hierarchy. At the first level news programs are separated into news reports and commercials. At the next level, news report is further segmented into anchorperson speech and others, which includes live report. At the highest level, text processing is used to generate a table of contents based on the boundary information extracted at lower levels and corresponding closed-caption information.

Movie Content Analysis (MoCA): MoCA is a project at the University of Mannheim, Germany, targeted mainly for understanding the semantic content of movies [86, 87]. The system segments movies into salient shots and generates a digital abstract of the movie. The text detection component tracks moving text and performs OCR on the text. The audio analysis component detects silence, human speech, music and noise. The latter is further analyzed to detect violence in the scenes.

CONtent-based Image and Video Access System (CONIVAS): CONIVAS is a client-server based system developed at Phillips Research [88]. The system employs cut detection for extraction of a storyboard used for browsing and retrieval from digital studio archive. Features extracted from the key frames are used for building an index of the content. Segmentation can be applied either in the compressed domain or the uncompressed domain. Feature extraction is performed either using low level visual features such as color, shape, and texture, or using full text retrieval. The extracted features are stored in a database. Image and video segments can be retrieved using example query segments.

Query By Image and Video Content (QBIC): QBIC system [89] developed at IBM's Almaden Research Center uses a variety of features for retrieving images from image/video database. The system allows a user to search, browse and retrieve image, graphic and video data from large on-line collections. Visual features such as color, layout and texture are extracted and stored in database. The system allows Query-by-Example (QBE) type queries, wherein the user can select any thumbnail from the list of images within the database or specify an image and request retrieval of similar images. The system segments the video into shots and generates storyboards consisting of representation frames extracted from the shots. The methods used for image retrieval can be applied to these representative frames to retrieve video clips by content.

VideoQ: VideoQ is a web enabled content based video search system [90, 91]. VideoQ expands the traditional search methods (e.g., keywords and subject navigation) with a novel search technique that allows users to search compressed video based

on a rich set of visual features and spatio-temporal relationships. Visual features include color, texture, shape and motion. Spatio-temporal video object query extends the principle of query by sketch in image databases to a temporal sketch that defines the object motion path.

WebSeek: WebSeek is a prototype image and video search engine which collects images and videos from the Web and catalogs them. It also provides tools for searching and browsing [92,93] using various content based retrieval techniques that incorporate the use of color, texture and other properties. Relevance feedback mechanisms are used to enhance performance.

Multimedia Analysis and Retrieval System (MARS): MARS is a system developed at the University of Illinois at Urbana Champaign [94]. The system supports content-based image retrieval based on color, texture, shape, and any Boolean combinations of them. The novel part in the system is the integration of database management techniques (query processing), information retrieval techniques (boolean retrieval model), and image processing techniques (image features). MARS supports image retrieval using relevance feedback.

Automatic News Summarization Extraction System (ANSES): ANSES is a system developed at Imperial College, London [95, 96]. This project combines a video scene change algorithm with current text segmentation and summarization techniques, to build an automatic news summarization and extraction system. Television broadcast news are captured both in video and audio format with the accompanying subtitles in text format. News stories are identified, extracted from the video, and summarized in a short paragraph which reduces the amount of information to a manageable size. Individual news video clips can be retrieved effectively by a combination of video and text, using a reverse indexed search engine to provide distilled information such as a summarized version of the original text and to highlight important keywords in the text.

Semantic Annotation of Sports Videos: In this system [97], an approach for semantic annotation of sports videos that include several different sports and even

nonsports content is implemented. Videos are automatically annotated according to elements of visual contents at different layers of semantic significance. The system primarily distinguishes studio and interview shots from sports action shots and further decompose sports videos into their main visual and graphic content elements, including sport type, foreground versus background and text captions. Relevant semantic elements from videos are extracted by combining several low-level visual primitives such as image edges, corners, segments, curves and color histograms, according to context-specific aggregation rules. The annotation task is organized into three distinct subtasks: Preclassifying sports shots, identifying graphic features and classifying visual shot features.

Name It-Naming and Detecting Faces in News Videos: The Name-It system [98] associate faces and names in news videos. It processes information from the videos and can infer possible name candidates for a given face or locate a face in news videos by name. To accomplish this task, the system takes a multimodal video analysis approach : Face sequence extraction and similarity evaluation from video, name extraction from transcript and video-caption recognition. Suppose that we are watching a TV news program. When unknown persons appear in the news video, we can eventually identify most of them by watching only the video. To do so, system detects faces from a news video, locates names in the sound track, and then associates each face with the correct name. For face-name association, as many hints as possible based on structure, context, and meaning of the news video are used. Name-It can associate faces in news videos with their right names without using an a prior face-name association set. In other words, Name-It extracts face-name correspondences only from news videos.

APPENDIX B

AUTOASSOCIATIVE NEURAL NETWORK MODELS

Autoassociative neural network models are feedforward neural networks performing an identity mapping of the input space, and are used to capture the distribution of the input data [65], [99]. The distribution capturing ability of the AANN model is described in this section. Let us consider the five layer AANN model shown in Fig. B.1, which has three hidden layers. In this network, the second and fourth layers have more units than the input layer. The third layer has fewer units than the first or fifth. The processing units in the first and third hidden layer are nonlinear, and the units in the second hidden layer (compression layer) can be linear or nonlinear. As the error between the actual and the desired output vectors is minimized, the cluster of points in the input space determines the shape of the hypersurface obtained by the projection onto the lower dimensional space. Fig. B.2(b) shows the space spanned by the one-dimensional compression layer for the two-dimensional data shown in Fig. B.2(a) for the network structure *2L 10N 1N 10N 2L*, where L denotes a linear unit and N denotes a nonlinear unit. The integer value indicates the number of units used in that layer. The nonlinear output function for each unit is $\tanh(s)$, where s is the activation value of the of the unit. The network is trained using backpropagation algorithm [57], [52]. The solid lines shown in Fig. B.2(b) indicate mapping of the given input points due to the one-dimensional compression layer. Thus, one can say that the AANN captures the distribution of the input data depending on the constraints imposed by the structure of the network, just as the number of mixtures and Gaussian functions do in the case of Gaussian mixture models (GMM).

In order to visualize the distribution better, one can plot the error for each input data point in the form of some probability surface as shown in Fig. B.2(c). The error

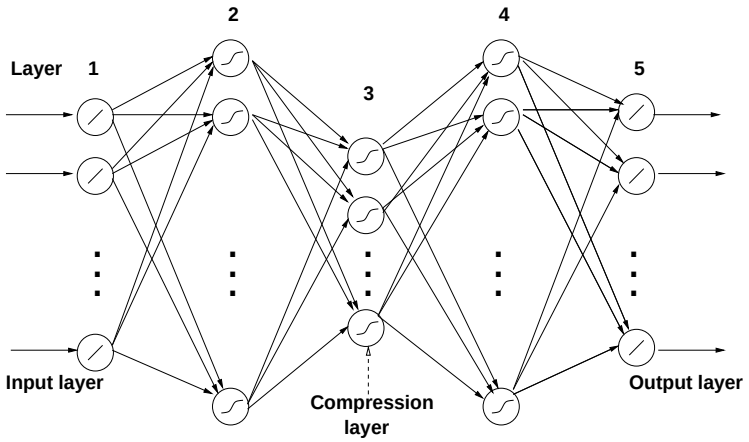


Fig. B.1: A five layer AANN model.

E_i for the data point i in the input space is plotted as $p_i = \exp^{(-E_i/\alpha)}$, where α is a constant. Note that p_i is not strictly a probability density function, but we call the resulting surface as probability surface. The plot of the probability surface shows a large amplitude for smaller error E_i, indicating better match of the network for that data point. The constraints imposed by the network can be seen by the shape the error surface takes in both the cases. One can use the probability surface to study the characteristics of the distribution of the input data captured by the network. Ideally, one would like to achieve the best probability surface, best defined in terms of some measure corresponding to a low average error.

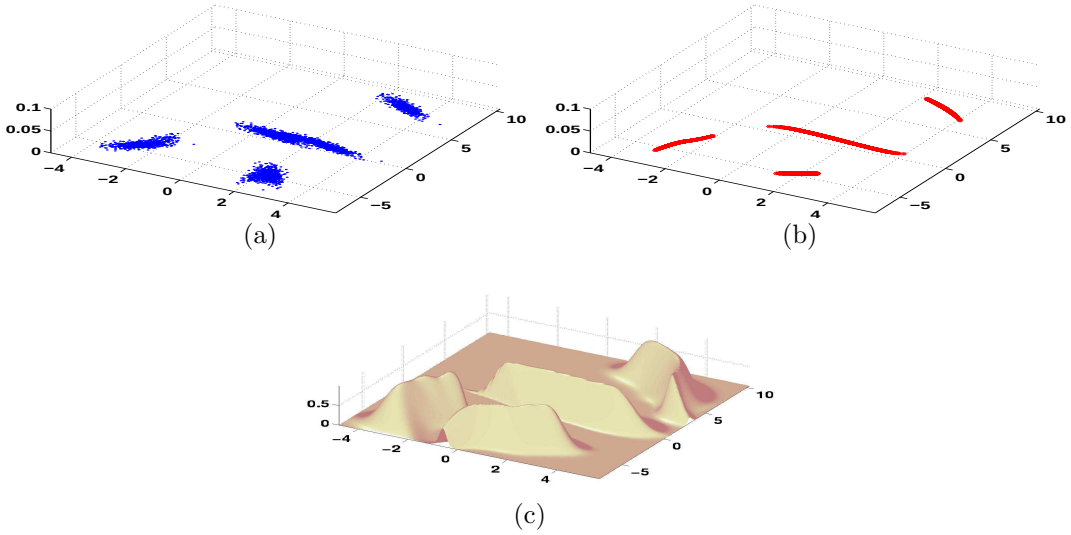


Fig. B.2: Distribution capturing ability of AANN model. (a) Artificial 2-dimensional data. (b) 2-dimensional output of AANN model with the structure $2L\ 10N\ 1N\ 10N\ 2L$. (c) Probability surfaces realized by the network structure $2L\ 10N\ 1N\ 10N\ 2L$.

APPENDIX C

HIDDEN MARKOV MODELS

A Markov model is a finite state machine that makes a transition of state once every time unit, governed by a probability law. The probability of occupying a state is determined solely by recent history. A hidden Markov model (HMM) is a doubly stochastic process that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observed symbols.

Elements of an HMM

An HMM is characterized by the following:

1. The number N of states in the model. Although the states are hidden, for many practical applications there is often some physical significance attached to the states of the model. Generally the states are interconnected in such a way that any state can be reached from any other state (an ergodic model); however, other possible interconnections of states are often of interest. The individual states are denoted as $S = \{q_1, q_2, \ldots, q_N\}$, and the state at time t as q_t.

2. The number M of distinct observation symbols per state, i.e., the discrete alphabet size. The observation symbols correspond to the physical output of the system being modeled. The individual symbols are denoted as $V = \{v_1, v_2, \ldots, v_M\}$.

3. The state transition probability distribution $A = \{a_{ij}\}$ where

$$a_{ij} = P\left[q_{t+1} = j | q_t = i\right], \quad 1 \leq i, j \leq N. \tag{C.1}$$

For the special case where any state can reach any other state in a single step,

we have $a_{ij} > 0$ for all i, j. For other types of HMMs, we would have $a_{ij} = 0$ for one or more (i, j) pairs.

4. The observation symbol probability distribution in state j, $B = \{b_j(k)\}$, where

$$b_j(k) = P\left[\mathbf{o}_t = v_k | q_t = j\right], \quad 1 \leq j \leq N, \quad 1 \leq k \leq M. \tag{C.2}$$

5. The initial state distribution $\Pi = \{\pi_i\}$ where

$$\pi_i = P\left[q_1 = i\right], \quad 1 \leq i \leq N. \tag{C.3}$$

Given appropriate values of N, M, A, B, and Π, the HMM can be used as a generator to give an observation sequence

$$\mathbf{O} = (\mathbf{o}_1 \mathbf{o}_2 \ldots \mathbf{o}_t \ldots \mathbf{o}_T) \tag{C.4}$$

(where each observation $\mathbf{o}_t$ is one of the symbols from V, and T is the number of observations in the sequence) as follows:

1. Choose an initial state $q_1 = i$ according to the initial state distribution Π.
2. Set $t = 1$.
3. Choose $\mathbf{o}_t = v_k$ according to the symbol probability distribution in state i, i.e., $b_i(k)$.
4. Transit to a new state $q_{t+1} = j$ according to the state transition probability distribution for state i, i.e., a_{ij}.
5. Set $t = t + 1$; return to step 3 if $t < T$; otherwise terminate the procedure.

The above procedure can also be used as a model for how a given observation sequence was generated by an appropriate HMM. It can be seen from the above discussion that the complete specification of an HMM requires specification of two model parameters (N and M), specification of observation symbols, and the specification of three probability measures A, B and Π. For convenience, the compact notation,

$$\lambda = (A, B, \Pi) \tag{C.5}$$

is used to indicate the complete parameter set of the model.

Given the observation sequence $\mathbf{O} = (\mathbf{o}_1\mathbf{o}_2 \ldots \mathbf{o}_T)$, and a model $\lambda = (A, B, \Pi)$, how to efficiently compute $P(\mathbf{O}|\lambda)$, the probability of the observation sequence, given the model? The problem can also be viewed as one of scoring how well a given model matches a given observation sequence. For example, if the case is considered in which it is tried to choose among several competing models, the solution to the above problem allows the choice of the model which best matches the observations.

The HMM parameters are estimated in a computationally efficient way using the following variables:

- Forward variable: $\alpha_t(i) = P(\mathbf{o}_1\mathbf{o}_2 \ldots \mathbf{o}_t, q_t = i/\lambda)$

  The probability of producing a partial observation sequence $\mathbf{o}_1\mathbf{o}_2 \ldots \mathbf{o}_t$ and ending in state i at time t, given the model λ.

$$\alpha_t(i) = \sum_{\{q_1, q_2, \ldots, q_t\}} P(q_1 q_2 \ldots q_{t-1}, q_t = i, \mathbf{o}_1\mathbf{o}_2 \ldots \mathbf{o}_t/\lambda)$$

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

- Backward variable: $\beta_t(j) = P(\mathbf{o}_{t+1}\mathbf{o}_{t+2} \ldots \mathbf{o}_T/q_t = i, \lambda)$

  The probability of producing a partial observation sequence $(\mathbf{o}_{t+1}\mathbf{o}_{t+2} \ldots \mathbf{o}_T)$, given the state i at time t and the model λ.

$$\beta_t(j) = \sum_{\{q_{t+1}, q_{t+2}, \ldots, q_T\}} P(q_{t+1} q_{t+2} \ldots q_T, \mathbf{o}_{t+1}\mathbf{o}_{t+2} \ldots \mathbf{o}_T/q_t = i, \lambda)$$

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i)$$

APPENDIX D

SUPPORT VECTOR MACHINES

The support vector machine (SVM) is a linear machine pioneered by Vapnik [100]. The main idea of an SVM is to construct a hyperplane as the decision surface in such a way that the margin of separation between positive and negative examples is maximized. The notion that is central to the construction of the support vector learning algorithm is the innerproduct kernel between a support vector $\mathbf{x}_i$ and a vector $\mathbf{x}$ drawn from the input space. The support vectors constitute a small subset of the training data extracted by the support vector learning algorithm. The separation between the hyperplane and the closest data point is called the margin of separation, denoted by ρ. The goal of a support vector machine is to find a particular hyperplane for which the margin of separation ρ is maximized. Under this condition, the decision surface is referred to as the optimal hyperplane. Fig. D.3 illustrates the geometric construction of a hyperplane for two dimensional input space. The support vectors play a prominent role in the operation of this class of learning machines. In conceptual terms, the support vectors are those data points that lie closest to the decision surface, and therefore the most difficult to classify. They have a direct bearing on the optimum location of the decision surface.

The idea of an SVM is based on the following two mathematical operations [100]:

1. Nonlinear mapping of an input pattern vector onto a higher dimensional feature space that is hidden from both the input and output.
2. Construction of an optimal hyperplane for separating the patterns in the higher dimensional space obtained from operation 1.

Operation 1 is performed in accordance with Cover's theorem on the separability of patterns [100]. Consider an input space made up of nonlinearly separable patterns.

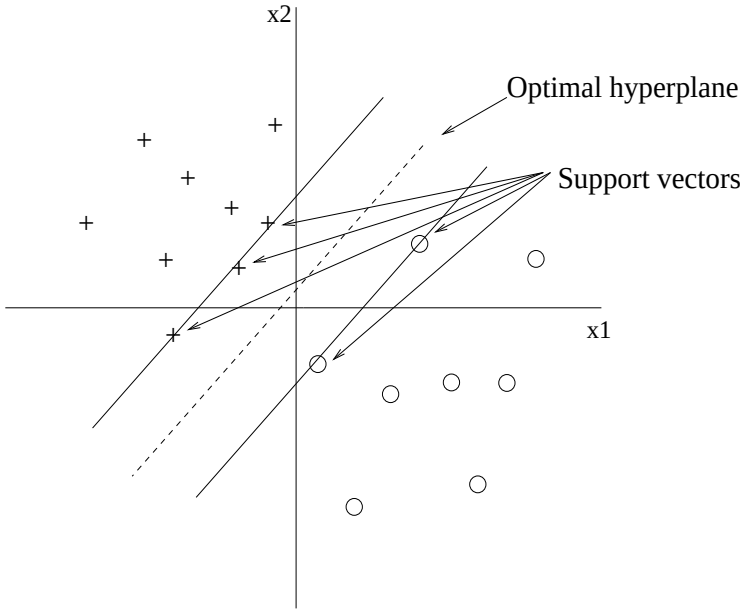


Fig. D.3: Illustration of the idea of support vectors and an optimal hyperplane for linearly separable patterns.

Cover's theorem states that such a multidimensional space may be transformed into a new feature space where the patterns are linearly separable with a high probability, provided the transformation is nonlinear, and the dimension of the feature space is high enough. These two conditions are embedded in operation 1. 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 [52]. 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 complex classification system, it is preferable to have those features which lead to lesser number of support vectors.

The optimal hyperplane is defined by:

$$\sum_{i=1}^{N_L} \alpha_i d_i K(\mathbf{x}, \mathbf{x}_i) = 0 \tag{D.6}$$

where $\{\alpha_i\}_{i=1}^{N_L}$ is the set of Lagrange multipliers, $\{d_i\}_{i=1}^{N_L}$ is the set of desired classes and $K(\mathbf{x}, \mathbf{x}_i)$ is the innerproduct kernel, and is defined by:

$$\begin{aligned} K(\mathbf{x}, \mathbf{x}_i) &= \varphi^T(\mathbf{x})\varphi(\mathbf{x}_i) \\ &= \sum_{j=0}^{m_1} \varphi_j(\mathbf{x})\varphi_j(\mathbf{x}_i), \qquad i = 1, 2, \ldots, N_L \end{aligned} \tag{D.7}$$

where $\mathbf{x}$ is a vector of dimension m 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 (D.6) it is seen that the construction of the optimal hyperplane is based on the evaluation of an innerproduct kernel. The innerproduct 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 $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N_L}$. Dimension of the feature space is determined by the number of support vectors extracted from the training data by the solution to the optimization problem (D.6).

REFERENCES

[1] M. A. Smith and T. Chen, "Image and video indexing and retrieval," *Hand Book on Image Processing*, pp. 687–704, 2000.

[2] C. Djeraba, "Content-based multimedia indexing and retrieval," *IEEE Multimedia*, vol. 9, pp. 18–22, Apr. 2002.

[3] D. Lelescu and D. Schonheld, "Statistical sequential analysis for real-time video scene change detection on compressed multimedia bitstream," *IEEE Multimedia*, vol. 5, pp. 106–117, Mar. 2003.

[4] A. Hanjalic, "Shot-boundary detection: Unraveled and resolved?," *IEEE Trans. Circuits, Systems, Video Technology*, vol. 12, pp. 90–104, Feb. 2002.

[5] J. Nam and A. Tewfik, "Detection of gradual transitions in video sequences using B-spline interpolation," *IEEE Multimedia*, vol. 7, pp. 667–679, Aug. 2005.

[6] H. Zhang and S. Kankanhalli, "Automatic partitioning of full-motion video," *ACM Journal of Multimedia Systems*, vol. 1, pp. 10–28, Jan. 1993.

[7] Z. Cernekova, C. Kotropoulos, and I. Pitas, "Video shot segmentation using singular value decomposition," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, (Hong Kong), pp. 181-184, Apr. 6-10, 2003.

[8] R. Zabih, J. Miller, and K. Mai, "A feature-based algorithm for detecting and classifying production effects," *ACM Journal of Multimedia Systems*, vol. 7, pp. 199–128, Mar. 1999.

[9] J. Yu and M. D. Srinath, "An efficient method for scene cut detection," *Pattern Recognition Letters*, vol. 22, pp. 1379–1391, Nov. 2001.

[10] R. Lienhart, "Reliable dissolve detection," *SPIE*, vol. 4315, pp. 219–230, Jan. 2001.

[11] J. Zhou and X.-P. Zhang, "Video shot boundary detection using independent component analysis," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, (Philadelphia, USA), pp. 541-544, Mar. 18-23, 2005.

[12] Z. Cernekova, C. Nikou, and I. Pitas, "Shot detection in video sequences using entropy-based metrics," in *Proc. IEEE Int. Conf. Image Processing*, pp. 421-424, 2002.

[13] Z.-N. Li, X. Zhong, and M. S. Drew, "Spatial temporal joint probability images for video segmentation," *Pattern Recognition*, vol. 35, pp. 1847–1867, Sep. 2002.

[14] W. Li and S. Lai, "Storage and retrieval for media databases," *SPIE*, vol. 5021, pp. 264–271, Jan. 2003.

[15] G. Boccignone, A. Chianese, V. Moscato, and A. Picariello, "Foveated shot detection for video segmentation," *IEEE Trans. Circuits, Systems, Video Technology*, vol. 15, pp. 365–377, Mar. 2005.

[16] R. Lienhart, "Comparison of automatic shot boundary detection algorithms," in *Proc. SPIE Storage and Retrieval for Image and Media DatabasesVII*, (San Jose, CA, USA), pp. 290-301, Jan. 26-29, 1999.

[17] R. Lienhart, "Reliable transition detection in videos: A survey and practitioner's guide," *Int. Journal of Image and Graphics*, vol. 1, pp. 469–486, Sept. 2001.

[18] I. Koprinska and S. Carrato, "Temporal video segmentation: A survey," *Signal Processing: Image Communication*, vol. 16, pp. 477–500, Jan. 2001.

[19] U. Gargi, R. Kasturi, and S. H. Strayer, "Performance characterization of video-shot-change detection methods," *IEEE Trans. Circuits, Systems, Video Technology*, vol. 10, pp. 1–13, Feb. 2000.

[20] M. Drew, Z.-N. Li, and X. Zhong, "Video dissolve and wipe detection via spatio-temporal images of chromatic histogram differences," in *Proc. IEEE Int. Conf. Image Processing*, pp. 929-932, 2000.

[21] Y. Wang, Z. Liu, and J.-C. Huang, "Multimedia content analysis using both audio and visual clues," *IEEE Signal Processing Magazine*, vol. 17, pp. 12–36, Nov. 2000.

[22] B. T. Truong, C. Dorai, and S. Venkatesh, "New enhancements to cut, fade and dissolve detection processes in video segmentation," in *ACM Int. Conf. on Multimedia*, pp. 219-227, Nov. 2000.

[23] S. Tsekeridou and I. Pitas, "Content-based video parsing and indexing based on audio-visual interaction," *IEEE Trans. Circuits, Systems, Video Technology*, vol. 11, no. 4, pp. 522–535, 2001.

[24] B. Truong, S. Venkatesh, and C. Dorai, "Automatic genre identification for content-based video categorization," *Proc. of Int. Conf. Pattern Recognition*, vol. 4, pp. 230–233, Sep. 2000.

[25] L. Perez-Freire and C. Garcia-Mateo, "A multimedia approach for audio segmentation in TV broadcast news," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, (Canada), pp. 369-372, May 17-21, 2004.

[26] C. Saraceno and R. Leonardi, "Audio as support to scene change detection and characterization of video sequences," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, (Munich, Germany), pp. 2597-2600, Apr. 21-24, 1997.

[27] J. S. Boreczky and L. D. Wilcox, "A hidden Markov model framework for video segmentation using audio and image features," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, (Seattle, WA), pp. 1228-1231, May 7-10, 1998.

[28] T. Vlachos, "Cut detection in video sequences using phase correlation," *IEEE Signal Processing Letters*, vol. 7, pp. 173–175, July 2000.

[29] J. Bescos, G. Cisneros, J. M. Martinez, J. M. Menendez, and J. Cabrera, "A unified model for techniques on video-shot transition detection," *IEEE Trans. Multimedia*, vol. 7, pp. 293–307, Apr. 2005.

[30] H. Feng, W. Fang, S. Liu, and Y. Fang, "A new general framework for shot boundary detection and key-frame extraction," in *ACM Int. workshop on Multimedia information retrieval*, (Singapore), pp. 121-126, Nov. 10-11, 2005.

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

[32] A. Kokaram, N. Rea, R. Dahyot, and A. M. Tekalp, "Browsing sports video," *IEEE Signal Processing Magazine*, vol. 5021, pp. 47–58, Mar. 2006.

[33] 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.

[34] Y. Yuan and C. Wan, "The application of edge features in automatic sports genre classification," in *Proc. IEEE Conf. Cybernetics and Intelligent Systems*, (Singapore), pp. 1133-1136, December 1-3, 2004.

[35] Y.-F. Ma and H.-J. Zhang, "Motion pattern based video classification using support vector machines," in *Proc. IEEE Int. Symposium on Circuit and Systems*, (Arizona, USA), pp. 69-72, May 26-29, 2002.

[36] R. Fablet and P. Bouthemy, "Statistical modeling for motion-based video classification and retrieval," in *Proc. Workshop Multimedia Content Based Indexing and Retrieval*, (France), pp. 69-72, Sep. 24-25, 2001.

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

[38] M. J. Roach, J. S. D. Mason, and M. Pawlewski, "Video genre classification using dynamics," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, (Utah, USA), pp. 1557-1560, May 07-11, 2001.

[39] A. Girgensohn and J. Foote, "Video classification using transform coefficients," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, (Arizona, USA), pp. 3045-3048, Mar. 15-19, 1999.

[40] L.-Q. Xu and Y. Li, "Video classification using spatial-temporal features and PCA," *Int. Conf. Multimedia and Expo*, pp. 345–348, July 2003.

[41] E. Sahouria and A. Zakhor, "Content analysis of video using principal components," *IEEE Trans. Circuits, Systems, Video Technology*, vol. 9, pp. 1290–1298, Dec. 1999.

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

[43] Y. Yuan, Q.-B. Song, and J.-Y. Shen, "Automatic video classification using decision tree method," in *Proc. IEEE Int. Conf. Machine Learning and Cybernetics*, (Beijing), pp. 1153-1156, Nov. 4-5, 2002.

[44] D. A. Sadlier and N. E. O'Connor, "Event detection in field sports video using audio-visual features and a support vector machine," *IEEE Trans. Circuits, Systems, Video Technology*, vol. 15, pp. 1225–1233, October 2005.

[45] G. Xu, Y.-F. Ma, H.-J. Zhang, and S.-Q. Yang, "An HMM-based framework for video semantic analysis," *IEEE Trans. Circuits, Systems, Video Technology*, vol. 15, pp. 1422–1433, November 2005.

[46] N. Rae, R. Dahyot, and A. Kokaram, "Semantic event detection in sports through motion understanding," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, (Dublin, Ireland), pp. 88-97, July 21-23, 2004.

[47] V. Tzouvaras, G. T. penakis, G. Stamou, and S. Kollias, "Adaptive rule-based recognition of events in video sequences," in *Proc. IEEE Int. Conf. Image Processing*, (Barcelona, Spain), pp. 607-610, Sep. 14-17, 2003.

[48] G. Xu, Y.-F. Ma, H.-J. Zhang, and S. Yang, "A HMM based semantic analysis framework for sports game event detection," in *Proc. IEEE Int. Conf. Image Processing*, (Barcelona, Spain), pp. 25-28, Sep. 14-17, 2003.

[49] B. Li and I. Sezan, "Semantic sports vidoe analysis: Approaches and new applications," in *Proc. IEEE Int. Conf. Image Processing*, (Barcelona, Spain), pp. 17-20, Sep. 14-17, 2003.

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

[51] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE Journal*, vol. 37, pp. 233–243, Feb. 1991.

[52] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New Jersey: Prentice-Hall International, 1999.

[53] The color images available at: `http://www.cs.cornell.edu/home/rdz/ccv.html`.

[54] M. Flickner, "Query by image and video content," *IEEE Computer*, vol. 28, pp. 23–32, Sep. 1995.

[55] V. Ogle and M. Stonebraker, "Chabot: Retrieval from a relational databases of images," *IEEE Computer*, vol. 28, pp. 40–48, Sep. 1995.

[56] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. USA: John Wiley & Sons. Inc., 2001.

[57] B. Yegnanarayana, *Artificial Neural Networks*. New Delhi: Prentice-Hall India, 1999.

[58] K. Diamantaras and S. Kung, *Principle Component Neural Networks: Theory and Applications*. New York: John Wiley and Sons, Inc., 1996.

[59] A. K. Jain, R. P. W. Dubin, and J. Mao, "Statistical pattern recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 4–37, Jan. 2000.

[60] R. Duda, P. Hart, and D. Stork, *Pattern Classification.* New York: John Wiley and Sons, Inc, 2001.

[61] S. V. Gangashetty, A. N. Khan, S. R. M. Prasanna, and B. Yegnanarayana, "Neural network models for preprocessing and discriminating utterances of consonant-vowel units," in *Proc. Int. Joint Conf. Neural Networks*, (Honolulu, Hawaii, USA), pp. 613-618, May, 2002.

[62] A. Jain, R. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 4–37, Jan. 2000.

[63] J. M. M. (UPM-GTI, ES), *MPEG-7 Overview (version 8).* Klangenfurt: ISO/IEC JTC1/SC29/WG11 N4980, July, 2002.

[64] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 679–698, Nov. 1986.

[65] B. Yegnanarayana and S. Kishore, "AANN: An alternative to GMM for pattern recognition," *Neural Networks*, vol. 15, pp. 459–469, Apr. 2002.

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

[67] 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.

[68] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Annals of Mathematical Statistics.*, vol. 41, no. 1, pp. 164–171, 1970.

[69] HMM Tool Kit available at: `http://htk.eng.cam.ac.uk/`.

[70] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[71] SVM Tool Kit available at: `http://www.idiap.ch/~bengio/projects/SVMTorch.html`.

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

[73] 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.

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

[75] 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.

[76] 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.

[77] Y. A. Ivanov and A. F. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 852–872, Aug. 2000.

[78] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *IEEE*, vol. 77, pp. 257–286, Feb. 1989.

[79] N. P. Cuntoor, B. Yegnanarayana, and R. Chellappa, "Interpretation of state sequences in HMM for activity representation," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, (Philadelphia, USA), pp. 709-712, Mar. 18-23, 2005.

[80] K. S. R. Murty and B. Yegnanarayana, "Event based analysis of speech using HMM framework," *Communicated to Speech Communication*, Sep. 2006.

[81] M. Roach, J. S. Mason, and M. Pawlewski, "Motion-based classification of cartoons," *Proc. Int. Symposium on Intelligent Multimedia*, pp. 146–149, May 2001.

[82] Informedia demo available at: *http://www.informedia.cs.cmu.edu*.

[83] H. D. Wactlar, T. Kanade, and M. A. Smith, "Intelligent access to digital video: Informedia project," *IEEE Comput. Mag.*, vol. 29, pp. 45–52, May 1996.

[84] H. D. Wactlar, "Lessons learned from building a terabyte digital video library," *IEEE Comput. Mag.*, vol. 32, pp. 66–73, Feb. 1999.

[85] B. Shahraray and D. C. Gibbon, "Pictorial transcripts: Multimedia processing applied to digital library creation," (Princeton, NJ), pp. 581-586, June 23-25, 1997.

[86] S. Pfeiffer, "Abstracting digital movies automatically," *Journal of Visual Communication and Image Representation*, vol. 7, pp. 345–353, Dec. 1996.

[87] MOCA demo available at: `http://www.informatik.uni-mannheim.de/informatik/pi4/projects/MoCA/`.

[88] M. Abdel-Mottaleb, N. Dimitrova, R. Desai, and J. Martino, "CONIVAS: Content-based image and video access system," in *ACM Conf. Multimedia*, (Boston, Massachusetts, US), pp. 427-428, Nov. 18-22, 1996.

[89] J. Ashley, "The query by image content (QBIC) system," in *Proc. ACMSIGMOD Int. Conf. Management of Data*, (Sanjose, California, US), pp. 475, May 22-25, 1995.

[90] S.-F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong, "VIDEOQ: An automated content based video search system using visual cues," in *ACM Conf. Multimedia*, (Seattle, Washington, US), pp. 313-3248, Nov. 09-13, 1997.

[91] VideoQ demo available at: http://www.ctr.columbia.edu/VideoQ.

[92] J. R. Smith and S.-F. Chang, *Searching for images and videos on the world wide web.* Technical Report: 459-96-25: Dept. of Electrical Engineering, Columbia University, Aug. 19, 1996.

[93] Webseek demo available at: http://www.ctr.columbia.edu/webseek.

[94] T. Huang, S. Mehrotra, and K. Ramchandran, "Multimedia analysis and retrieval system (MARS) project," (University of Illinois at Urbana-Champaign), Mar. 24-26, 1996.

[95] M. J. Pickering, L. Wong, and S. M. Ruger, "Anses: Summarization of news video," in *Int. Conf. Image and Video Retrieval*, (Urbana-Champaign, IL, USA), pp. 425-434, July 24-25, 2003.

[96] ANSES demo available at: http://www.doc.ic.ac.uk/∼mjp3/anses/.

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

[98] S. Satoh, Y. Nakamura, and T. Kanade, "Name-it : Naming and detecting faces in news videos," *IEEE Multimedia*, vol. 6, pp. 22–35, Jan.-Mar 1999.

[99] B. Yegnanarayana, S. V. Gangashetty, and S. Palanivel, "Autoassociative neural network models for pattern recognition tasks in speech and image," in *Soft Computing Approach to Pattern Recognition and Image Processing*, (World Scientific publishing Co. Pte. Ltd, Singapore), pp. 283–305, December 2002.

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

List of Publications

REFERRED JOURNALS

1. C.Krishna Mohan and B.Yegnanarayana, "Sports video classification using edge-based features," Communicated to *Signal, Image and Video Processing*, November 2006.

2. C. Krishna Mohan, N. Dhananjaya, and B. Yegnanarayana, "Video shot boundary detection using autoassociative neural network model," Communicated to *Signal Processing: Image Communication*, June 2006.

3. C. Krishna Mohan, K. Srirama Murthy, and B. Yegnanarayana, "Event-based sports video classification using HMM framework," Communicated to *Pattern Recognition*, June 2006.

4. C. Krishna Mohan, N. Dhananjaya, and B. Yegnanarayana, "Video shot boundary detection by combining evidence from early and late fusion techniques," Communicated to *Pattern Recognition Letters*, May 2006.

5. Vakkalanka Suresh, C. Krishna Mohan, and B. Yegnanarayana, "Performance-based classifier fusion for video classification," Communicated to *Computer Vision and Image Understanding*, March 2006.

CONFERENCES

1. C.Krishna Mohan, N. Dhananjaya, Suryakanth V. Gangashetty and B.Yegnanarayana, "Sports video classification using autoassociative neural network models," in *Tenth International Conference on Cognitive and Neural Systems (ICCNS-06)*, Boston, USA, May 17-20, 2006, pp. 30.

2. Vakkalanka Suresh, C.Krishna Mohan, R. Kumaraswamy and B.Yegnanarayana, "Combining multiple evidence for video classification," in *IEEE Int. Conf. Intelligent Sensing and Information Processing (ICISIP-05)*, Chennai, India, Jan. 4-7, 2005, pp. 187-192.

3. Vakkalanka Suresh, C.Krishna Mohan, R. Kumaraswamy and B.Yegnanarayana, "Content-Based Video Classification using SVMs," in *Int. Conf. Neural Information Processing (ICONIP-04)*, Calcutta, India, Nov. 22-25, 2004, pp. 726-731.

CURRICULUM VITAE

1. **NAME:** C. Krishna Mohan

2. **DATE OF BIRTH:** 12^{th} April 1967

3. **PERMANENT ADDRESS:**

C. Krishna Mohan
Senior Lecturer
Dept. of M.A.C.S
N.I.T.K. Surathkal
P.O. Srinivasnagar - 575 025
Mangalore, Karnataka, India

4. **EDUCATIONAL QUALIFICATIONS:**

- June 1988: Bachelor of Science Education (B.Sc.Ed., University of Mysore)
- August 1991: Master of Computer Applications (M.C.A., University of Mysore)
- July 2000: Master of Technology (M.Tech., S.A.C.A., Mangalore University)
- November 2006: Doctor of Philosophy (Ph.D., Dept. of CSE, IIT Madras)

DOCTORAL COMMITTEE

1. **CHAIRPERSON:** Prof. Timothy A. Gonsalves

2. **GUIDES:** Prof. B. Yegnanarayana and Dr. C. Chandra Sekhar

3. **MEMBERS:**

   - Dr. V. Kamakoti (Dept. of CSE)
   - Dr. Deepak Khemani (Dept. of CSE)
   - Dr. S. Mohan (Dept. of CE)
   - Dr. V. Srinivas Chakravarthy (Dept. of Bio-Tech)