Understanding and Describing Tennis Videos

Thesis submitted in partial fulfillment
of the requirements for the degree of

MS by Research
in
Computer Science and Engineering

by

Mohak Kumar Sukhwani
201307583
mohak.sukhwani@research.iiit.ac.in

Center for Visual Information Technology
International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2016

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Understanding and Describing Tennis Videos" by Mohak Kumar Sukhwani, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Adviser: Dr. C.V. Jawahar

To

Family and friends

Acknowledgements

This work is the manifestation of shared vision, determination, commitment, patience, encouragement and synergistic contribution of many people. This report would be incomplete without my expression of indebtedness towards them. I have great pleasure in expressing my deep sense of gratitude to my supervisor, Dr. C.V. Jawahar, for providing his guidance and encouragement at every phase of the project. His vast experience in this field was of invaluable help to me.

Special mention to Dr. Ravindra Babu Tallamraju, Principal Researcher - Infosys Labs, for his constant support and belief. His contribution as a manager, teacher and guide will always remain invaluable. I'm thankful to him for always taking time from his busy schedule to proofread all my research papers. I also take this opportunity to thank all the faculty members at CVIT – Dr. P. J. Narayanan, Dr. Anoop Namboodiri and Dr. Jayanthi Sivaswamy for creating a wonderful environment in CVIT, and CVIT administration Satya, Phani, Rajan and Nandini for helping me on numerous occasions.

I am grateful to Yashaswi Verma for taking keen interest in my work and for his invaluable contributions in form of suggestions and guidance. Thanks to my lab mates and friends at IIIT for keeping me sane and providing feedbacks all throughout my research term – Pritish, Viresh, Aniket, Anand, Suriya, Devendra(Dau), Ajeet(ji), Udit, Praveen, Saurabh, Vijay, Pramod, Koustav, Vishal, Vishakh, Bipin, Anirudh, Arwa, Anurag, Yash and many others. I profusely thank Kumar Vaibhav, Anubhav Agarwal and Tushar Verma for always being there through thick and thin. Last but not the least, I would also like to wholeheartedly thank my family, for their love and unwavering support in all my endeavours.

Abstract

‘Our most advanced machines are like toddlers when it comes to sight.’ [1] When shown a tennis video to kid, he mostly probably would blabber words like ‘tennis’, ‘racquet’, ‘ball’ etc. Similar is the case with present day state-of-art video understanding algorithms. We in this work try to solve one such multimedia content analysis problem – ‘How to get machines go beyond object and action recognition and make them understand lawn tennis video content in a holistic manner ?’. We propose a multi-facet approach to understand the video content as a whole - (a) Low level Analysis: Identify and isolate court regions and players (b) Mid Level Understanding: Recognize players actions and activities (c) High Level Annotations: Generate detailed summary of event comprising of information from full game play.

Annotating visual content with text has attracted significant attention in recent years. While the focus has been mostly on images, of late few methods have also been proposed for describing videos. The descriptions produced by such methods capture the video content at certain level of semantics. However, richer and more meaningful descriptions may be required for such techniques to be useful in real-life applications. We make an attempt towards this goal by focusing on a domain specific setting – lawn tennis videos. Given a video shot from a tennis match, we intend to predict detailed (commentary-like) descriptions rather than small captions. Rich descriptions are generated by leveraging a large corpus of human created descriptions harvested from Internet. We evaluate our method on a newly created tennis video data set comprising of broadcast video recordings of matches from London Olympics 2012. Extensive analysis demonstrate that our approach addresses both semantic correctness as well as readability aspects involved in the task.

Given a test video, we predict a set of action/verb phrases individually for each frame using the features computed from its neighbourhood. The identified phrases along with additional meta-data are used to find the best matching description from the commentary corpus. We begin by identifying two players on the tennis court. Regions obtained after isolating playing court regions assist us in segmenting out the candidate player regions through background subtraction using thresholding and connected component analysis. Each candidate foreground region thus obtained is represented using HOG descriptors over which a SVM classifier is trained to discard non-player foreground regions. The candidate player regions thus obtained are used to recognize players using using CEDD descriptors and Tanimoto distance. Verb phrases are recognized, by extracting features from each frame of input video using sliding window. Since this typically results into multiple firings, non-maximal suppression (NMS) is applied.

[1] Fei-Fei Li

This removes low-scored responses that are in the neighbourhood of responses with locally maximal confidence scores. Once we get potential phrases for all windows along with their scores, we remove the independence assumption and smooth the predictions using an energy minimization framework. For this, a Markov Random Field (MRF) based model is used which captures dependencies among nearby phrases. We formulate the task of predicting the final description, as an optimization problem of selecting the best sentence among the set of commentary sentences in corpus which covers most number of unique words in obtained phrase set. We even employ Latent Semantic Indexing (LSI) technique while matching predicted phrases with descriptions and demonstrate its effectiveness over naïve lexical matching. The proposed pipeline is benchmarked against state-of-the-art methods. We compare our performance with recent methods. Caption generation based approaches achieve significantly low score owing to their generic nature. Compared to all the competing methods, our approach consistently provides better performance. We validate that in domain specific settings, rich descriptions can be produced even with small corpus size.

The thesis introduces a method to understand and describe the contents of lawn tennis videos. Our approach illustrates the utility of simultaneous use of vision, language and machine learning techniques in a domain specific environment to produce human-like descriptions. The method has direct extensions to other sports and various other domain specific scenarios. With deep learning based approaches becoming a de-facto standard for any modern machine learning task, we wish to explore them for present task in future augmentations. The flexibility and power of such structures have made them outperform other methods in solving some really complex vision problems. Large scale deployments of combination of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have already surpassed other comparable methods for real time image summarization. We intend to exploit the power of such combined structures in VIDEO TO TEXT regime and generate real time commentaries for the game-videos as one of the proposed future extensions.

Contents

Chapter Page

List of Figures

Figure Page

List of Tables

Chapter 1

Introduction

> ‘If we want our machines to think, we need to teach them to see and understand.’*Fei-Fei Li, Stanford AI Lab*
>
> WIRED, April’15

‘IN, Winner: Djokovic!!! Monaco goes for a out-wide serve, Djokovic works a backhand return, hard fought rally, Monaco forehand goes outside the court.’ The ease with which a human describes an ongoing game scene (lawn tennis in this case) still remains unmatched with present day state-of-art vision and machine learning algorithms. Multitude of algorithms running behind the scene still find it difficult to achieve such ‘human like’ details in the text they generate describing the given scene. This work takes a step towards smart vision for ‘videos’ and propose a pipeline comprising of vision, learning and language algorithms to generate descriptions for ‘lawn-tennis’ videos. It is indeed challenging to develop a generic approach for such tasks. We hence are confined to a specific domain and focus on generating fine grained and rich descriptions for tennis videos. We address the problem of automatically generating human-like descriptions for unseen videos, given a collection of videos and their corresponding human-generated descriptions. Here, we present a method which simultaneously utilizes the visual clues, corpus statistics and the available text to construct final commentary like textual descriptions.

Videos convey visual information of scene bounded by temporal ordering. They tend to hold abundance of information which needs to be catered by underlying algorithm to generate natural text. Computer vision for long has strived to design such algorithms, with an ultimate goal of generic multimedia (image/video) understanding. To date much of the video understanding has been tackled by object and subject detection, but it does not cater to holistic needs for full video content. Only a limited content of videos are summarized by such approach, leading to ‘machine like’ or ‘robotic’ description generation for the scene. We tackle the challenge of natural description generation for videos by adding a new dimension of fine-grained action recognition over object and subject recognitions. We present a comprehensive set of experiments and results to prove the effectiveness of constrained setting on quality of

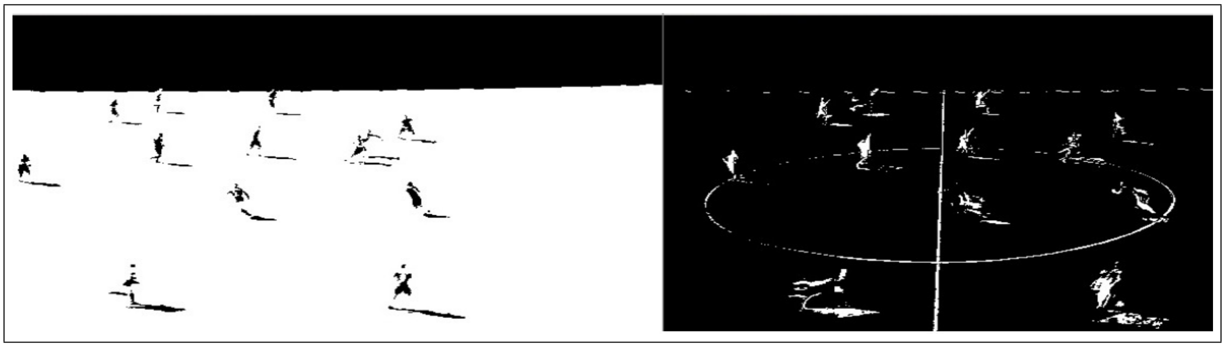


Figure 1.1: Sport Analysis in Soccer: Real-time football analysis include automatic game summarization, player tracking, highlight extraction etc.

descriptions. Ours is a supervised learning method that considers video-text associations to generate the commentary. We generate richer description for videos through a retrieval based approach.

With the amount of video data growing rapidly on the web, one has to resort to the techniques that analyse these videos and index them for human consumption. We need to develop methods that assist us in fine grained video analysis. Such methods should impart semantic meanings and associations among identified events in the video and generate better insights for the overall video content. Talking in terms of text based video understanding, the final descriptions generated for a given video should be a culmination of smaller but related texts (each smaller text describing an event). This project looks at the ways to identify the salient events in a given video, describe them and finally generate a holistic text combining all the event information to describe the full video. We analyse and study various video features and text features to learn the associating models and exploit the complimentary nature of the 'pixels' in videos and the 'characters' in text to generate our final description. The challenges associated with our present work can be summarized as – (a) generate a rich vocabulary to create 'human like' descriptions. (b) use appropriate video features to capture maximum video information. (c) exploit domain information to reproduce most effective description.

This project has direct significance to various tasks that access videos and their textual descriptions simultaneously – video search, browsing, question-answer system, commercial applications such as tennis coaching ('virtual assistance') and societally important applications such as assistance for the blind. Additionally, the outputs of this work, have potential for cross impact on both computer vision and natural language communities.

1.1 Computer Vision and Language Processing

With recent technological advancements we have witnessed a sudden surge in image and video content on-line. Millions of images and more than hundred hours of video get uploaded on daily basis.

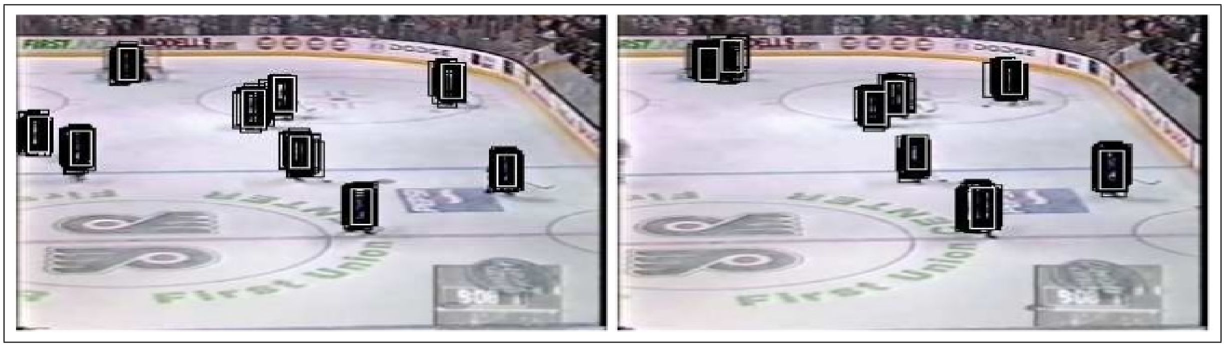


Figure 1.2: Sport Analysis in Ice-Hockey: Player recognition and tracking on field.

This huge chunk of data is mostly left untagged and unlabelled on web space with no semantic associations. This gives us huge opportunity to discover the automated methods to add semantic descriptions to available multimedia content. Although there has been much progress in developing theories, models and systems in the areas of Natural Language Processing (NLP) and Vision Processing (VP), there has heretofore been little progress on integrating these two subareas of Artificial Intelligence (AI). We focus on one such integration – we generate commentary like description for lawn tennis videos. Describing the visual content with natural language labels have attracted large number of people in vision community. While majority of the work has been done in image domain, works related to video descriptions are still in nascent stages. However, even for simple videos automatically generating such descriptions may be quite complex, thus suggesting the hardness of the problem.

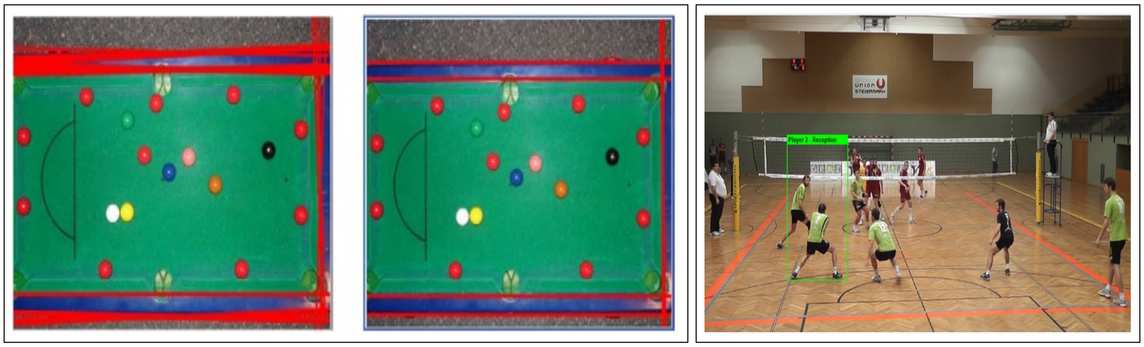


Figure 1.3: Sport Analysis in snooker and volley ball: (Left) Analysis of shot trajectories and stroke analysis in snooker. (Right) Player identification and action recognition in volley-ball.

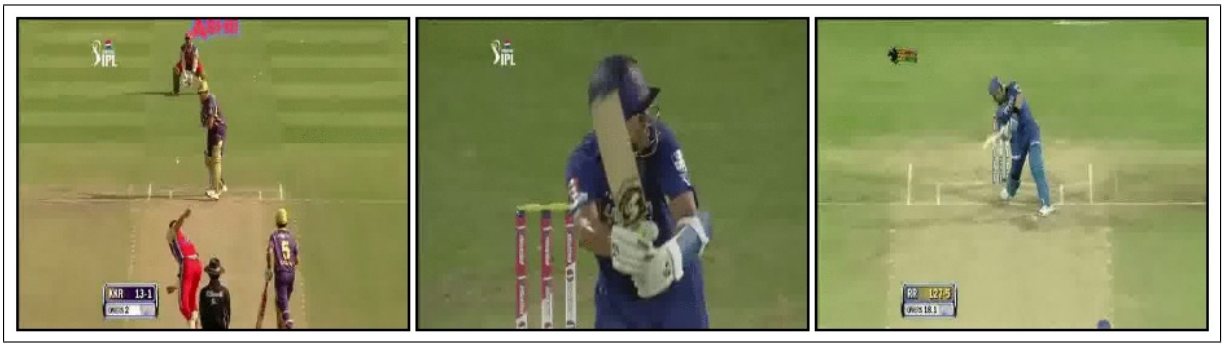


Figure 1.4: Sport Analysis in Cricket: Temporal segmentation and annotation of actions with semantic descriptions.

1.2 Sports Video Analysis

Sports content is easily one of the most popular new age infotainment content available on web. Due to its mass appeal and massive popularity across the globe there has been a tremendous growth in the areas of sports video analysis. Volleyball [78, 79], basketball [68, 48] ,cricket [66], handball [28], snooker [67],ice-hockey [53], football [89] etc. all major sporting events are being analysed and studied in great depth. High amenability to automatic video analysis processing has made sport analyses a popular genre in vision and machine learning community. Areas like performance assessments, which were previously mainly of interest to coaches and sports scientists are now finding applications in broadcast and other mainstream media applications. Driven by the increase in use of on-line sports viewing tools detailed performance statistics in every sports is just a click away from viewers. Enormous commercial investments with humongous viewer-ship involved sport analyses has poured in tons of opportunities for researchers and data experts.

A significant amount of resources are been channelled into sports in order to improve understanding, performance and presentation. Computer vision and Machine Learning have recently began playing an important and indispensable role in sports – real-time football analysis [Figure 1.1], annotations in cricket [Figure 1.4], player recognition and tracking in handball, game analysis of ice-hockey[Figure 1.2], shot detections in volleyball [Figure 1.3] and much more. Computer vision algorithms have a huge potential ranging from automatic annotation of broadcast footage, through to better understanding of sport injuries, and enhanced viewing etc. So far the use of computer vision in sports has been scattered between different disciplines.

From viewers point of view, only few portions in game are interesting. These parts of video often have associated high-level concepts, such as goals in soccer, point in tennis, homerun in baseball games etc. . The detection and extraction of such events are termed as 'semantic analysis' of sports video. It aims at detecting and extracting information that describes facts in a video. In contrast, we also have 'tactic analysis' of sports video which aims to recognize and discover tactic patterns and match strategies

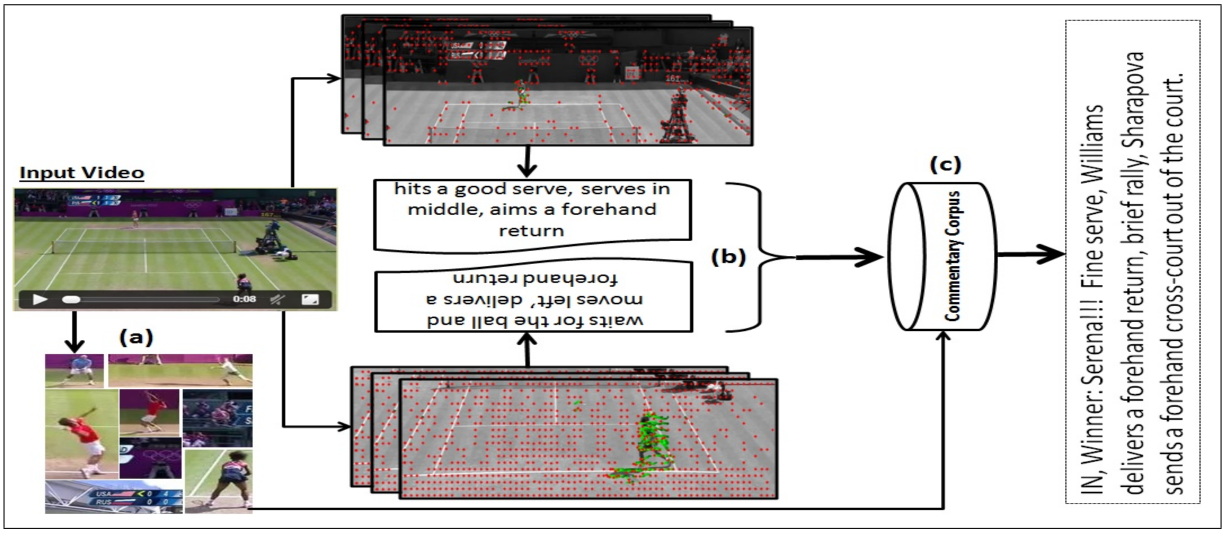

Input Video
(a)
hits a good serve, serves in middle, aims a forehand return
waits for the ball and moves left, delivers a forehand return
(b)
(c)
Commentary Corpus
IN, Winner: Serena!!! Fine serve, Williams delivers a forehand return, brief rally, Sharapova sends a forehand cross-court out of the court.

Figure 1.5: For a test video, the *description* predicted using our approach is dense and human-like. Above figure demonstrates our approach – a) Player Identification b) Verb Phrase Prediction c) Description Generation

in sport videos. This kind of analysis is more focused on coaches and team staff. Computer vision and machine learning in sports is gradually attracting the attention of large sports associations. Baseball team 'Boston Red Sox' and football club 'AC Milan' were among the first few organizations that started to apply such methodologies for sports analysis. Appropriate use of large amounts of video and other data available to sports organizations led to growth of this field. Application of such analysis could lead to better overall team performance by analysing player behaviors in varied situations, determining their individual impact, revealing the opponent's tactics and pointing possible weaknesses in play etc. . While the use of advance machine learning, statistics and data mining in decision making is certainly an improvement over present day semi-automated works but it comes with its own set of challenges and inhibitions. We in our present work focus our attention to understand the game of lawn tennis using vision based approaches and generate a text 'commentary' that best describes the input scene.

1.3 Contributions

We propose an approach which explicitly splits the problem of analysis and understanding lawn tennis videos into first 'action localization' for player identification followed by 'verb phrase recognition' for fine grained action classification. A retrieval based approach is thereafter used for final description generation. To this end we make following notable contributions:

(i) **New Dataset:** release of a new dataset comprising of over 1000 lawn tennis game and commentary aligned videos for public consumption [1]. The dataset even comprises of over $400K$ unaligned commentary text lines.

(ii) **Content Analysis – Player and Court Recognition:** use of domain information for action localization and player identification.

(iii) **Fine grained action recognition:** use of action trajectories for recognizing lawn tennis actions and representing them with 'verb phrases'. Our main contributions in this direction are – (a) joint model to assign video frames into appropriate phrase bins under weak label supervision. (b) probabilistic label consistent dictionary learning for sparse modelling and classification.

(iv) **Commentary Generation for lawn tennis:** use of action phrases for fine grained retrieval which assists in creation of florid and verbose commentary text.

We demonstrate that in a domain restrictive setting, limited number of labelled video data is sufficient enough to generate rich descriptions for a new input video. We exploit the untapped potential of group of pre-trained weak-classifiers and leverage the hidden information of parallel (but related) text to produce human like descriptions for video inputs.

1.4 Thesis Outline

In our present work, we address the problem of analysing and understanding the contents of an input lawn tennis video. Chapter 2 introduces the technical background needed for the later sections of the thesis. We begin by formally defining our problem statement in Chapter 3 and introduce a new 'lawn tennis' dataset. The new dataset has been publicly released for the community. Chapter 4 deals with identification and localization of active action zones (players), Figure 1.5(a), in the video which act like a precursor for fine grained action recognition in subsequent chapters. Chapter 5 describes a novel 'verb phrase' recognition module, Figure 1.5(b). Action verb phrases thus identified are used for final description generation using a retrieval based approach, Figure 1.5(c), in Chapter 6. Chapter 6 also deals in comparisons with other present state of art methods and demonstrates the effectiveness of our method over others. We conclude with a short discussion and future directions of our work in Chapter 7

[1] http://cvit.iiit.ac.in/research/projects/cvit-projects/fine-grained-descriptions-for-domain-specific-videos

Chapter 2

Background

In this chapter, we briefly discuss the machine learning, vision and text analysis tools which have been used in the thesis. In Section 2.1, we look at the popular linear as well as non-linear feature extraction strategies used for most of the upcoming tasks. Section 2.2, describes various classification techniques used for both phrase detections and player identification. The penultimate section describes the optimization techniques to compute the best commentary text for an input video clip. We end the chapter by describing evaluation techniques used for both quantitative and qualitative analysis.

2.1 Feature Extraction

Feature extraction begins from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the learning and generalization, and in some cases leading to better human interpretations. In (majority of) cases it is related to dimensionality reduction and involves reducing the amount of resources required to describe a large set of data. Feature extraction is a general term of constructing combinations of the variables to create a succinct representation and to get around the problems of large memory and power requirements while still describing the data with sufficient accuracy. The extracted features are expected to contain the relevant information from the input data, so that the desired task can be performed by using this reduced representation instead of the complete initial data.

2.1.1 HOG - Histograms of Oriented Gradients

Detecting humans in tennis videos is a challenging task owing to the variability in appearances and poses. We need a robust feature set that allows the human form to be discriminated cleanly, even in cluttered backgrounds under difficult illumination. Histogram of oriented gradients HOG is a feature descriptor [11] used to detect objects in computer vision and image processing. The HOG descriptor technique counts occurrences of gradient orientation in localized portions of an image - detection window, or region of interest. The HOG descriptors are reminiscent of edge orientation histograms SIFT

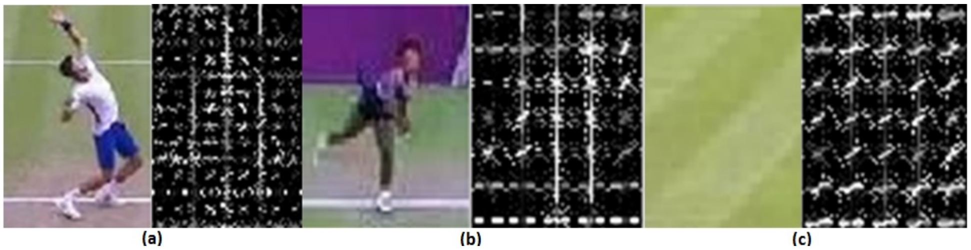


Figure 2.1: Visualization of HOG descriptors: (a) Down Player (b)Up Player (c) No Player

descriptors and shape contexts, but they are computed on a dense grid of uniformly spaced cells and they use overlapping local contrast normalizations for improved performance.

HOG divides the input image into square cells of size ‘cellSize’, fitting as many cells as possible, filling the image domain from the upper-left corner down to the right one. For each row and column, the last cell is at least half contained in the image. Then the image gradient is computed by using central difference (for colour image the channel with the largest gradient at that pixel is used). The gradient is assigned to one of $2 \times numOrientations$ orientation in the range $[0, 2\pi)$. Contributions are then accumulated by using bilinear interpolation to four neigbhour cells, as in Scale Invariant Feature Transform SIFT. This results in an histogram h_d (of dimension $2 \times numOrientations$) of directed orientations. It accounts for the direction as well as the orientation of the gradient. A second histogram h_u of undirected orientations of half the size is obtained by folding h_d into two.

Let a block of cell be a 2×2 sub-array of cells. Let the norm of a block be the l_2 norm of the stacking of the respective unoriented histogram. Given a HOG cell, four normalisation factors are then obtained as the inverse of the norm of the four blocks that contain the cell. Each histogram h_d is copied four times, normalised using the four different normalisation factors, the four vectors are stacked, saturated at 0.2, and finally stored as the descriptor of the cell. This results in a $numOrientations \times 4$ dimensional cell descriptor. Blocks are visited from left to right and top to bottom when forming the final descriptor.

2.1.2 CEDD - Color and Edge Directivity Descriptor

This feature is called ‘Color and Edge Directivity Descriptor’ and incorporates color and texture information in a histogram. CEDD size is limited to 54 bytes per image, rendering this descriptor suitable for use in large image databases. One of the most important attribute of the CEDD is the low computational power needed for its extraction.

To make detection resilient to lighting, occlusion and direction our framework extracts monocular cues, viz. color and texture features, from each candidate object and stores in the form of histogram- Color and Edge Directivity Descriptor (CEDD) [6]. The size of histogram is 144 bins with each bin represented by 3 bits ($144 \times 3 = 432bits$). The histogram is divided into 6 regions, each determined by the extracted texture information. Each region is further divided into 24 individual sub regions,

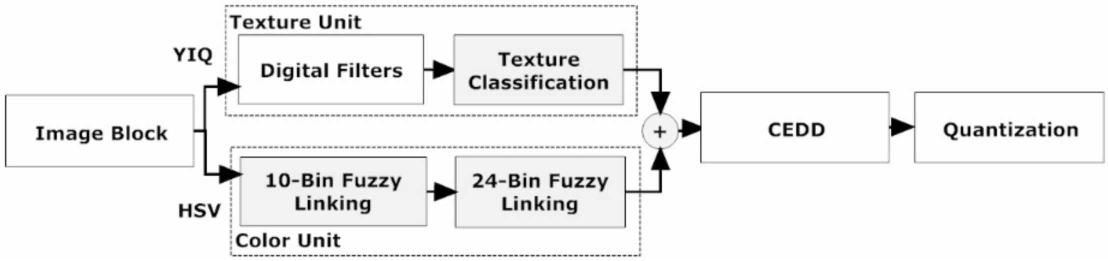


Figure 2.2: CEDD Histogram: Each Image Block feeds successively all the units. Color unit extracts color information and Texture unit extracts texture information. The CEDD histogram is constituted by 6 regions, determined by the Texture Unit. Each region is constituted by 24 individual regions, emanating from the Color Unit.

with each sub region containing color information $6 \times 24 = 144$. The HSV color channel is provided as input to fuzzy system to obtain color information. The texture information is composed of edges characterized as vertical, horizontal, 45^o, 135^o and non-directional, more details in [6]. This histogram is used as feature vector to describe the candidate object (player candidate regions in present case) and provided as input to the learning framework, figure 2.2.

2.1.3 Dense Trajectories

In our present work we employ dense trajectories for action recognition [80].The trajectories are obtained by tracking densely sampled points using optical flow fields. The number of tracked points can be scaled up easily, as dense flow fields are already computed. Furthermore, global smoothness constraints are imposed among the points in dense optical flow fields, which results in more robust trajectories than tracking or matching points separately. Dense trajectories [80, 81] are extracted for multiple spatial scales, figure 2.3. Feature points are sampled on a grid spaced by W pixels and tracked in each scale separately. Each point in a frame is tracked to the next frame by median filtering in a dense optical flow field. Once the dense optical flow field is computed, points can be tracked very densely without additional cost. Points of subsequent frames are concatenated to form a trajectory. The length of a trajectory is limited to L frames to avoid problem of drift. As soon as a trajectory exceeds L, it is removed from the tracking process. The shape of a trajectory encodes local motion patterns and thus assists in identifying actions.

Motion is the most informative cue for action recognition. It can be due to the action of interest, but also be caused by background or the camera motion. This is inevitable when dealing with realistic actions in uncontrolled settings. To overcome the problem of camera motion, [80] introduced a local descriptor that focuses on foreground motion. The descriptor extends the motion coding scheme based on motion boundaries developed in the context of human detection to dense trajectories. Moreover, in game of 'lawn tennis' the camera motion is negligible after the onset of 'serve'.

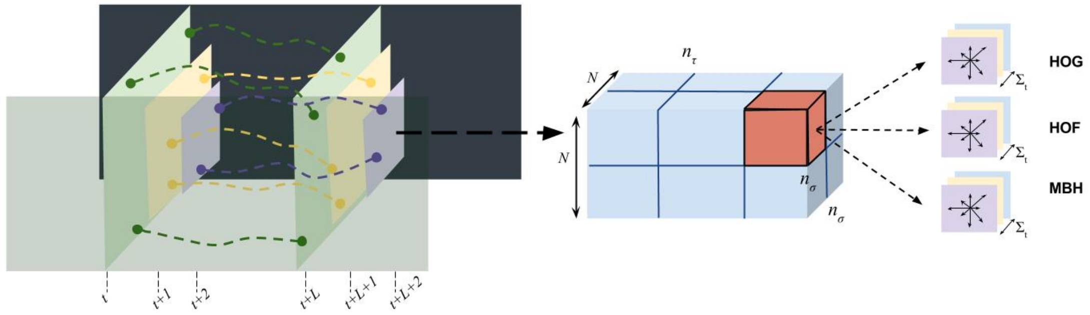


Figure 2.3: Dense Trajectories: Feature points are sampled densely for multiple spatial scales and tracking is performed in the corresponding spatial scale over L frames. Trajectory descriptors are based on its shape represented by relative point coordinates as well as appearance and motion information over a local neighborhood of $N \times N$ pixels along the trajectory. The trajectory neighborhood is divided into a spatio-temporal grid of size $n_\sigma \times n_\sigma \times n_\tau$

The motion information in dense trajectories are included by computing descriptors within a space-time volume around the trajectory. The size of the volume is $N \times N$ pixels and L frames. To embed structure information in the representation, the volume is subdivided into a spatio-temporal grid of size

For each trajectory descriptors comprise of Trajetory, HOG (Histogram of Oriented gradients), HOF (histograms of optical flow) and MBH (motion boundary histogram). HOG captures static appearance information and HOF, MBH measure motion information based on optical flow. The final dimension of descriptors are 30 for Trajectory, 96 for HOG, 108 for HOF and 192 for MBH (quantizes derivatives into horizontal and vertical component - MBHx and MBHy). For both HOF and MBH descriptors, the dense optical flow that is already computed to extract dense trajectories is reused. This makes feature computation process very efficient.

2.1.4 Text Features

Text Analysis is a one of the major application field for machine learning algorithms. Most of the machine learning algorithms expect numerical feature vectors with a fixed size rather than the raw text documents with variable length. So an essential task to generate such features is core of text processing. Vectorization is the term of the general process of turning a collection of text documents into numerical feature vectors. This strategy comprising of tokenization, counting and normalization is called the Bag of Words or ‘Bag of n-grams’ representation in text domain. Documents (in our case commentaries and phrases) are described by word occurrences while completely ignoring the relative position information of the words in the document. **Tokenizing** refers to giving an integer id for each possible token, for instance by using white-spaces and punctuation as token separators. **Counting** refers to frequency count of the occurrences of tokens in each document. **Normalizing** and weighting with diminishing importance tokens that occur in the majority of samples and documents is key for text feature computation.

In a large text corpus, frequently occurring words (e.g. 'the', 'a', 'is' in English) hence carry very little meaningful information about the actual contents of the document. If we were to feed the direct count data directly to a classifier those very frequent terms would shadow the frequencies of rarer yet more interesting terms. In order to re-weight the count features into floating point values suitable for usage by a classifier it is very common to use the TFIDF transform. Tf means term-frequency while TFIDF means term-frequency times inverse document-frequency. This was originally a term weighting scheme developed for information retrieval (as a ranking function for search engines results), that has also found good use in document classification and clustering.

The *term frequency* is the term count within a document divided by the number of words in that document):

$$tf(t,d) = \frac{count(t), t \in d}{|d|} \qquad \text{(Term frequency of term } t\text{, document } d\text{)}$$

A term's *document frequency* is the count of documents containing that term divided by the total number of documents (probability of seeing t in a document):

$$df(t,N) = \frac{|\{d_i : t \in d_i,\ i = 1..N\}|}{N} \qquad \text{(Document frequency of } t \text{ in } N \text{ documents)}$$

In order to attenuate the TFIDF scores for terms with high document frequencies, we need the document frequency in the denominator:

$$tfidf(t,d,N) = \frac{tf(t,d)}{df(t,N)} \qquad \text{(First approximation to TFIDF)}$$

This formula is meaningful but gives a poor term score because the document frequency tends to engulf the term frequency in the numerator so we take the log of the denominator first.

$$tfidf(t,d,N) = tf(t,d) \times log(\frac{1}{df(t,N)}) \qquad \text{(TFIDF with attenuated document frequency)}$$

To prevent division by 0 errors when a term does not exist in a corpus (e.g., $df(t,N) = 0$ when we pass unknown term(s) t), we add 1 to the denominator. This is similar to *additive smoothing* and pretends that there is an imaginary document with every unknown word. To keep document frequencies in $[0..1]$, we can to bump the document count, N, as well.

$$df(t,N) = \frac{|\{d_i : t \in d_i,\ i = 1..N\}| + 1}{N + 1} \qquad \text{(}df\text{ with smoothing)}$$

2.2 Classification

The term 'classification' in machine learning is the problem of identifying the category (or class) to which a new observation belongs, on the basis of a training set of data containing observations whose category membership is known. It is considered an instance of supervised learning, i.e. learning where a training set of correctly identified observations is available. The individual observations are converted into a set feature vectors and thereafter classified into respective category using pre-trained models. An algorithm that implements classification is known as a classifier. Its a mathematical function that maps input data to a category.

2.2.1 SVM - Support Vector Machines

Given a set K of training samples from two lineally separable classes P and N: $\{(\mathbf{x}_k, y_k), k = 1, \cdots, K\}$, where $y_k \in \{1, -1\}$ are class labels. We find a hyper-plane in terms of $\mathbf{w}$ and b, that linearly separates the two classes. For a decision hyper-plane $\mathbf{x}^T\mathbf{w} + b = 0$ to separate the two classes P $(\mathbf{x}_i, 1)$ and N $(\mathbf{x}_i, -1)$, it should satisfy

$$y_i(\mathbf{x}_i^T\mathbf{w} + b) \geq 0$$

for both $\mathbf{x}_i \in P$ and $\mathbf{x}_i \in N$. Among all such planes satisfying this condition, we find the optimal one that separates the two classes with the maximal margin (the distance between the decision plane and the closest sample points).

The optimal plane should be in the middle of the two classes, so that the distance from the plane to the closest point on either side is the same. We define two additional planes H_+ and H_- that are parallel to H_0 and go through the point closest to the plane on either side:

$$\mathbf{x}^T\mathbf{w} + b = 1, \quad \text{and} \quad \mathbf{x}^T\mathbf{w} + b = -1$$

All points $\mathbf{x}_i \in P$ on the positive side satisfy $\mathbf{x}_i^T\mathbf{w} + b \geq 1,\ y_i = 1$ and all points $\mathbf{x}_i \in N$ on the negative side satisfy $\mathbf{x}_i^T\mathbf{w} + b \leq -1,\ y_i = -1$. These can be combined into one inequality:

$$y_i(\mathbf{x}_i^T\mathbf{w} + b) \geq 1, \quad (i = 1, \cdots, m)$$

The equality holds for those points on the planes H_+ or H_-. Such points are called *support vectors*, for which $\mathbf{x}_i^T\mathbf{w} + b = y_i$ i.e., the following holds for all support vectors:

$$b = y_i - \mathbf{x}_i^T\mathbf{w} = y_i - \sum_{j=1}^{m} \alpha_j y_j (\mathbf{x}_i^T\mathbf{x}_j)$$

Moreover, the distances from the origin to the three planes H_-, H_0 and H_+ are, respectively, $|b - 1|/||\mathbf{w}||$, $|b|/||\mathbf{w}||$, and $|b + 1|/||\mathbf{w}||$, and the distances between planes H_- and H_+ is $2/||\mathbf{w}||$, which is to be maximized. Now the problem of finding the optimal decision plane in terms of $\mathbf{w}$ and b can be formulated as:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\mathbf{w}^T\mathbf{w} = \frac{1}{2}||\mathbf{w}||^2 \qquad \text{(objective function)} \\ \text{subject to} & y_i(\mathbf{x}_i^T\mathbf{w} + b) \geq 1, \ \text{ or } \ 1 - y_i(\mathbf{x}_i^T\mathbf{w} + b) \leq 0, \quad (i = 1, \cdots, m) \end{array}$$

This QP problem is solved by Lagrange multipliers method to minimize

$$L_p(\mathbf{w}, b, \alpha) = \frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^m \alpha_i (1 - y_i(\mathbf{x}_i^T \mathbf{w} + b))$$

with respect to $\mathbf{w}$, b and the Lagrange coefficients $\alpha_i \geq 0$ $(i = 1, \cdots, \alpha_m)$. We let

$$\frac{\partial}{\partial W} L_p(\mathbf{w}, b) = 0, \quad \frac{\partial}{\partial b} L_p(\mathbf{w}, b) = 0$$

These leads, respectively, to

$$\mathbf{w} = \sum_{j=1}^m \alpha_j y_j \mathbf{x}_j, \quad \text{and} \quad \sum_{i=1}^m \alpha_i y_i = 0$$

Substituting these two equations back into the expression of $L(\mathbf{w}, b)$, we get the *dual problem* (with respect to α_i) of the above *primal problem*:

$$\begin{array}{ll} \text{maximize} & L_d(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T, \mathbf{x}_j \\ \text{subject to} & \alpha_i \geq 0, \quad \sum_{i=1}^m \alpha_i y_i = 0 \end{array}$$

Solving this dual problem (an easier problem than the primal one), we get α_i, from which $\mathbf{w}$ of the optimal plane can be found. Those points $\mathbf{x}_i$ on either of the two planes H_+ and H_- (for which the equality $y_i(\mathbf{w}^T \mathbf{x}_i + b) = 1$ holds) are called *support vectors* and they correspond to positive Lagrange multipliers $\alpha_i > 0$. The training depends only on the support vectors, while all other samples away from the planes H_+ and H_- are not important.

For a support vector $\mathbf{x}_i$ (on the H_- or H_+ plane), the constrained condition is $y_i \left(\mathbf{x}_i^T \mathbf{w} + b \right) = 1$, $(i \in sv)$. Here, sv is a set of all indices of support vectors $\mathbf{x}_i$ (corresponding to $\alpha_i > 0$). Substituting

$$\mathbf{w} = \sum_{j=1}^m \alpha_j y_j \mathbf{x}_j = \sum_{j \in sv} \alpha_j y_j \mathbf{x}_j$$

we get

$$y_i(\sum_{j \in sv} \alpha_j y_j \mathbf{x}_i^T \mathbf{x}_j + b) = 1$$

For the optimal weight vector $\mathbf{w}$ and optimal b, we have:

$$\begin{aligned} ||\mathbf{w}||^2 &= \mathbf{w}^T \mathbf{w} = \sum_{i \in sv} \alpha_i y_i \mathbf{x}_i^T \sum_{j \in sv} \alpha_j y_j \mathbf{x}_j = \sum_{i \in sv} \alpha_i y_i \sum_{j \in sv} \alpha_j y_j \mathbf{x}_i^T \mathbf{x}_j \\ &= \sum_{i \in sv} \alpha_i (1 - y_i b) = \sum_{i \in sv} \alpha_i - b \sum_{i \in sv} \alpha_i y_i \\ &= \sum_{i \in sv} \alpha_i \end{aligned}$$

The last equality is due to $\sum_{i=1}^{m} \alpha_i y_i = 0$ shown above. The distance between the two margin planes H_+ and H_- is $2/||\mathbf{w}||$, and the margin, the distance between H_+ (or H_-) and the optimal decision plane H_0, is

$$\frac{1}{||\mathbf{w}||} = \left(\sum_{i \in sv} \alpha_i \right)^{-1/2}$$

2.2.1.1 Kernel Mapping: Non-Linear Case

The algorithm above converges only for linearly separable data. If the data set is not linearly separable, we can map the samples $\mathbf{x}$ into a feature space of higher dimensions:

$$\mathbf{x} \longrightarrow \phi(\mathbf{x})$$

in which the classes can be linearly separated. The decision function in the new space becomes:

$$f(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{w} + b = \sum_{j=1}^{m} \alpha_j y_j (\phi(\mathbf{x})^T \phi(\mathbf{x}_j)) + b$$

where $\mathbf{w} = \sum_{j=1}^{m} \alpha_j y_j \phi(\mathbf{x}_j)$ and b are the parameters of the decision plane in the new space. As the vectors $\mathbf{x}_i$ appear only in inner products in both the decision function and the learning law, the mapping function $\phi(\mathbf{x})$ does not need to be explicitly specified. Instead, all we need is the inner product of the vectors in the new space. The function $\phi(\mathbf{x})$ is a kernel-induced *implicit* mapping. A kernel is a function that takes two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ as arguments and returns the value of the inner product of their images $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$:

$$K(\mathbf{x}_1, \mathbf{x}_2) = \phi(\mathbf{x}_1)^T \phi(\mathbf{x}_2)$$

As only the inner product of the two vectors in the new space is returned, the dimensionality of the new space is not important. The learning algorithm in the kernel space can be obtained by replacing all inner products in the learning algorithm in the original space with the kernels:

$$f(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{w} + b = \sum_{j=1}^{m} \alpha_j y_j K(\mathbf{x}, \mathbf{x}_j) + b$$

The parameter b can be found from any support vectors $\mathbf{x}_i$:

$$b = y_i - \phi(\mathbf{x}_i)^T \mathbf{w} = y_i - \sum_{j=1}^{m} \alpha_j y_j (\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)) = y_i - \sum_{j=1}^{m} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

2.2.2 SSVM - Structured Support Vector Machines

Structured SVM is a Support Vector Machine (SVM) learning algorithm for predicting multivariate or structured outputs. It performs supervised learning by approximating a mapping using labelled training samples. Unlike the regular SVMs which consider only univariate predictions like in classification and regression, SSVM can predict complex objects like trees, sequences or sets.

In the SSVM model, the initial learning model parameters are set and the pattern-label pairs are read with specific functions. The user defined special constraints are then initialised and then the learning model is initialised. After that, a cache of combined feature vectors is created and then the learning process begins. The learning process repeatedly iterates over all the examples. For each example, the label associated with most violated constraint for the pattern is found. Then, the feature vector describing the relationship between the pattern and the label is computed and the loss is also computed with loss function. The program determines from feature vector and loss whether the constraint is violated enough to add it to the model. The program moves on to the next example. At various times (which depend on options set) the program retrains whereupon the iteration results are displayed. In the event that no constraints were added in iteration, the algorithm either lowers its tolerance or, if minimum tolerance has been reached, ends the learning process.

2.2.3 Nearest Neighbors

Nearest neighbors are one of the most classifiers, possibly because it does not involve any training. This technique simply consists of storing all the labeled training examples and given a test images, the label of closest training sample(or majority label of k-neighbors) is assigned to the test image. Nearest neighbors can be easily kernelized or coupled with metric learning. For classification techniques such as LDA or linear SVM, only a single weight vector needs to be stored per class(in a one-vs-rest setting), however, a nearest neighbor classification strategy typically consists of storing all the training samples. This is one of the major demerits of nearest neighbors.

2.3 Graphical Models - Optimization

A graphical model (in probability theory, statisticsparticularly Bayesian statisticsand machine learning) is a probabilistic model for which a graph expresses the conditional dependence structure between random variables. PGMs use a graph-based representation as the foundation for encoding a complete distribution over a multi-dimensional space and a graph that is a compact or factorized representation of a set of independences that hold in the specific distribution.

2.3.1 MRF - Markov Random Fields

A Markov Random Field (MRF) is a graphical model of a joint probability distributions that encode spatial dependencies. It consists of an undirected graph $G = (N, E)$ in which the nodes N represent random variables. Let X_S be the set of random variables associated with the set of nodes S. Then, the edges E encode conditional independence relationships via the following rule: given disjoint subsets of nodes A, B, and C, is conditionally independent of given if there is no path from any node in A to any node in B that doesn't pass through a node of C. The neighbour set N_n of a node n is defined to be the set of nodes that are connected to n via edges in the graph. Given its neighbour set, a node n is

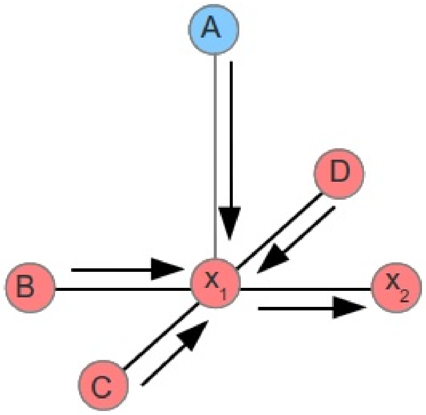


Figure 2.4: Belief Propagation : Figure shows an example of a message passing from x_1 to x_2. A node passes a message to an adjacent node only when it has received all incoming messages, excluding the message from the destination node to itself. x_1 waits for messages from nodes A, B, C, D before sending its message to x_2.

independent of all other nodes in the graph (Markov Property). MRF has plentiful applications in both vision and language community [36].

We solve an optimization problem over a MRF network to smoothen our phrase prediction result. An optimisation problem is one that involves finding the extremum of a quantity or function. Such problems often arise as a result of a source of uncertainty that precludes the possibility of an exact solution. Optimisation in an MRF problem involves finding the maximum of the joint probability over the graph, usually with some of the variables given by some observed data. Equivalently, as can be seen from the equations above, this can be done by minimising the total energy, which in turn requires the simultaneous minimisation of all the clique potentials. Techniques for minimisation of the MRF potentials are plentiful. Many of them are also applicable to optimisation problems other than MRF. Gradient descent methods are well-known techniques for finding local minima, while the closely-related method of simulated annealing attempts to find a global minimum.

2.3.2 BP - Belief propagation for Inference

Belief propagation (sum-product message passing), is a message passing algorithm for performing 'inference' on graphical models – Bayesian networks and Markov random fields. It computes the marginal distribution for each unobserved node, conditional on any observed nodes [86]. BP are presented as message update equations on a factor graph, involving messages between variable nodes and their neighboring factor nodes and vice versa. Considering messages between regions in a graph is one way of generalizing the belief propagation algorithm. There are several ways of defining the set of regions in a graph that can exchange messages, e.g. Kikuchi's cluster variation method. Improvements in the performance of belief propagation algorithms are achievable by breaking the replicas symmetry in the distributions of the fields (messages).

The original belief propagation algorithm was proposed by Pearl in 1988 for finding exact marginals on trees. Trees are graphs that contain no loops. It turns out the same algorithm can be applied to general graphs, those that contain loops, hence the 'loopy BP' [51]. LBP is a message passing algorithm. A node passes a message to an adjacent node only when it has received all incoming messages, excluding the message from the destination node to itself. The choice of using cost/penalty or probabilities is dependent on the choice of the MRF energy formulation. LBP is an iterative method. It runs for a fixed number of iterations or terminate when the change in energy drops below a threshold.

2.4 Retrieval

Information retrieval in general is done by naively matching terms in documents with those of a query. However, such lexical matching methods can be inaccurate. Since there are usually many ways to express a given concept (synonymy), most words have multiple meanings (polysemy) etc. the end task doesn't remain that simple. A robust approach should allow users to retrieve information on the basis of a conceptual topic or meaning of a document. LSI is one such approach.

2.4.1 LSI: Latent Semantic Indexing

LSI uses linear algebra techniques to learn the conceptual correlations in text collection. It involves constructing a weighted term-document matrix, performing a Singular Value Decomposition on the matrix, and using the matrix to identify the concepts contained in the text Latent Semantic Indexing [60] overcomes the problems of lexical matching by using statistically derived conceptual indices instead of individual words for retrieval. It assumes underlying or latent structure in word usage that is partially obscured by variability in word choice. A truncated singular value decomposition SVD is used to estimate the structure in word usage across documents. Retrieval is then performed using the database of singular values and vectors obtained from the truncated SVD.

This technique projects queries and documents into a space with 'latent' semantic dimensions. In such space, a query and a document can have high cosine similarity even if they do not share any terms - as long as their terms are semantically similar in a sense. The latent semantic space has fewer dimensions than the original space (which has as many dimensions as terms). LSI is thus a method for dimensionality reduction. Latent semantic indexing is the application of Singular Value Decomposition to a word-by-document matrix. SVD (and hence LSI) is a least-squares method. The projection into the latent semantic space is chosen such that the representations in the original space are changed as little as possible when measured by the sum of the squares of the differences.

Textual documents are represented as vectors in a vector space. Each position in a vector represents a term (typically a word), with the value of a position i equal to 0 if the term does not appear in the document, and having a positive value otherwise. Positive values are represented as the log of the total frequency in that document weighted by the entropy of the term. As a result, the corpus can be looked at as a large term-by-document ($t \times d$) matrix X, with each position x_{ij} corresponding to the presence or absence of a term (a row i) in a document (a column j). This matrix is typically very sparse, as most documents contain only a small percentage of the total number of terms seen in the full collection of documents.

The SVD of the $t \times d$ matrix, X, is the product of: TSD^T, where T and D are the matrices of the left and right singular vectors and S is the diagonal matrix of singular values. The diagonal elements of S are ordered by magnitude, and therefore these matrices can be simplified by setting the smallest k values in S to zero. The columns of T and D that correspond to the values of S that were set to zero are deleted. The new product of these simplified three matrices is a matrix $\hat{X}$ that is an approximation of the term-by-document matrix. This new matrix represents the original relationships as a set of orthogonal factors. When used for retrieval, a query is represented in the same new small space that the document collection is represented in. This is done by multiplying the transpose of the term vector of the query with matrices T and S^{-1}. Once the query is represented this way, the distance between the query and documents can be computed using the cosine metric, which represents a numerical similarity measurement between documents. LSI returns the distance between the query and all documents in the collection. Those documents that have higher cosine distance value than some cutoff point are returned as relevant to the query.

2.5 Evaluation Measures

This section describes various terms/keywords used to measure the effectiveness of our methods. It talks about importance of ‘robust’ evaluations schemes and justifies the harnessing of well known evaluations techniques for our experiments.

2.5.1 Classification Accuracy

We use classification accuracy for quantifying performance for the phrase recognition tasks. For defining classification accuracy, we first define the following quantities

- **True Positive(tp):**number of correctly classified positive samples.
- **True Positive(tn):**number of correctly classified negative samples.
- **False Positive(fp):**number of positive samples misclassified as belonging to the negative class.
- **False Negative(fn):**number of negative samples misclassified as belonging to the positive class.

Accuracy is defined as

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{2.1}$$

2.5.2 BLUE scores

Human evaluations of machine translation are extensive but expensive. Human evaluations can take months to finish and involve human labor that can not be reused. BLEU [55] is a method of automatic machine translation evaluation that is quick, inexpensive, and language-independent, that correlates highly with human evaluation, and that has little marginal cost per run.

It is a score for evaluating the quality of text which has been generated by any underlying algorithm. Quality is considered to be the correspondence between a machineś output and that of a human: 'the closer a machine translation is to a professional human translation, the better it is' – this is the central idea behind BLEU. Scores are computed for individual sentences comparing them with a set of good quality reference sentences (ground truths). Scores are then averaged over the whole corpus to reach an estimate of the overall quality of system. Intelligibility or grammatical correctness are not taken into account. BLEU output is a number between 0 and 1. This value indicates how similar the candidate and reference texts are, with values closer to 1 representing more similar texts. BLEU correlates well with human judgement, and remains a benchmark for the assessment of many evaluation metric [55]. BLEU cannot in its present form deal with languages lacking word boundaries.

2.5.3 Human Evaluation

There are several venues where development of machine algorithms can benefit from human involvement. Humans can be involved in the task of collecting, labelling and evaluating data. Especially with the increasing popularity of tools like LabelMe [62] and Amazon Mechanical Turk [38], one finds it very easy to involve human 'experts' in large scale studies. From labelling task in vision to machine translations in NLP, micro-task markets, such as Amazon's Mechanical Turk, offer a potential paradigm for engaging a large number of users for low time and monetary costs.

Experiments have shown high agreement between Mechanical Turk non-expert annotations and existing gold standard labels provided by expert labels [69]. Using non-expert labels for training machine learning algorithms can be as effective as using gold standard annotations from experts. Many large labelling tasks can be effectively designed and carried out in this method at a fraction of the usual expense. The need for efficient, reliable human evaluation of text output has led to the creation of several judgement tasks [14]. Evaluations investigating the quality of text output often elicit absolute quality judgements such as adequacy or fluency ratings. Several problems are encountered with this approach – annotators have difficulty agreeing on what factors constitute 'good' or 'bad' quality and are often unable to reproduce their own absolute scores. Relative judgement tasks such as ranking address these problems by eliminating notions of objective 'good' or 'bad' of translations in favor of simpler comparisons. While these tasks show greater agreement between judges, ranking can prove difficult and confusing when translation hypotheses are nearly identical or contain

Since we did not have control over evaluators involved in AMT, we in our present work use in house developed human evaluation schema and use 'experts' for evaluations. 'Experts' in our case comprise of people with decent lawn-tennis exposure. Experts were asked to give relative scores to generated descriptions and the final score was obtained by averaging the scores obtained over top five retrieval. Utmost care was taken while deciding the experts evaluators involved and we believe this has strengthened our evaluations.

Chapter 3

Problem Statement and Dataset

We make an attempt towards the goal of creating 'richer and human-like descriptions' by focusing on a domain specific setting – lawn tennis videos. For such videos, we aim to predict detailed (commentary-like) descriptions rather than small captions. Figure 3.1 depicts the problem of interest and an example result of our method. It even depicts the difference between a caption and a description. Rich description generation demands deep understanding of visual content and their associations with natural text. This makes our problem challenging.

3.1 Introduction

The majority of previous work in multimedia recognition has focused on image labeling with a fixed set of visual categories [85, 20]. While closed vocabularies constitute a convenient modeling assumption, they are vastly restrictive when compared to the enormous amount of variability that a human can compose. Other approaches that focus on generating much more dense image descriptions [40, 19] often rely on hard-coded visual concepts and sentence templates, which imposes limits on their variety. Number of approaches have even posed the task as a retrieval problem, and transfer the most most

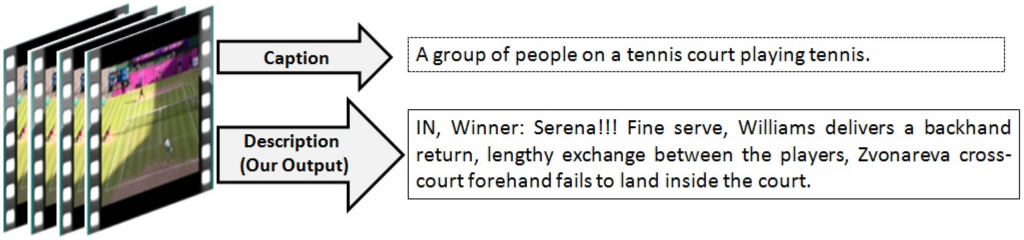

Caption
A group of people on a tennis court playing tennis.
Description
(Our Output)
IN, Winner: Serena!!! Fine serve, Williams delivers a backhand return, lengthy exchange between the players, Zvonareva cross-court forehand fails to land inside the court.

Figure 3.1: For a test video, the *caption* generated using an approximation of a state-of- the-art method [34], and the *description* predicted using our approach. Unlike recent methods that focus on short and generic captions, we aim at detailed and specific descriptions.

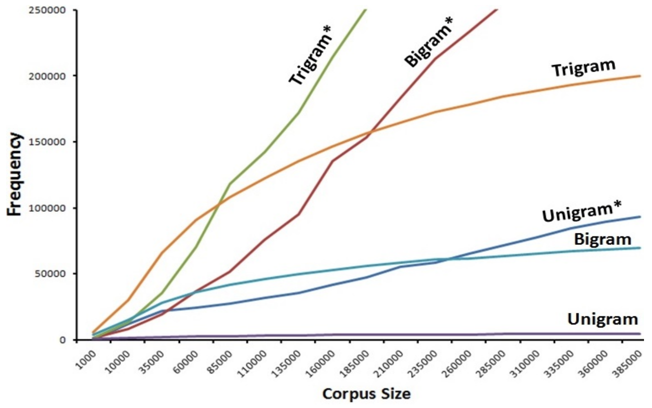
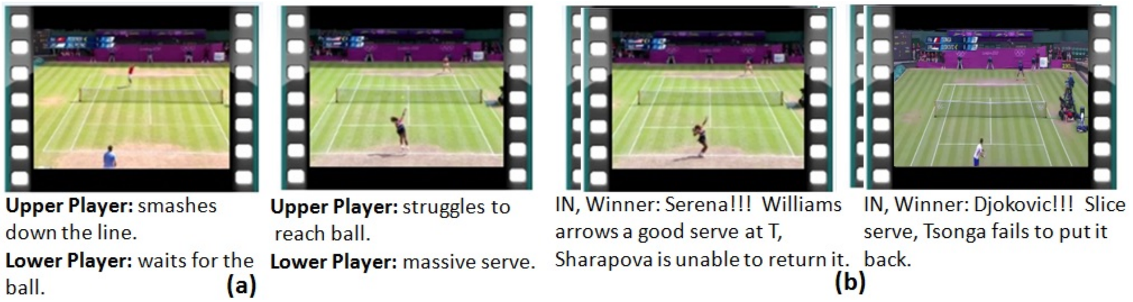


Figure 3.2: Trend illustrating saturation of word-level trigram, bigram and unigram counts in domain specific settings. Emphasized(*) labels indicate corpus of unrestricted tennis text (blogs/news) and remaining labels indicate tennis commentary corpus. Bigram* and Trigram* frequencies are scaled down by a factor of 10 to show the comparison.

compatible annotation in the training set to a test image [29, 19, 70, 54] or where training annotations are fragmented and stitched together [41, 44]. Several other image caption generating approaches are based on fixed templates [24, 40, 19, 84] or generative grammars [85, 49]. Much more recent works use combination of combination of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks to generate image descriptions [34, 47, 76, 37, 16].

One of the earliest works in tennis video analytics [72, 50] focuses on detecting court area and players in significant frames. The relative positions of both the players with respective to court lines and net are used to determine action types. Using special set-ups to detect people, ball, court lines with utmost precision [56] have been explored by many. Use of audiovisual and textual cues [33] to summarise the game by set of static representative frames (shot boundaries) has also received considerable attention in past. None of these works have still focused on creation of 'text based' commentaries for lawn-tennis in realistic settings. Although soccer 'commentary' generation has been explored in past by few [77, 73] but the methods described by them are not generic and can not be extended for our case. While Rocco uses simulation to track players (robots) and ball, Mike uses soccer server information for same task. Rocco uses template based method to create descriptions. We are keen on generating systems like above and generate final output as detailed as [87]. Authors in [87] use discriminative model for action attribute detection and use contextual constraints to produce detailed attributes labels. They detect major action attributes so as to generate extremely detailed video description.

For the game of tennis, which has a pair of players hitting the ball, actions play the central role. Here, actions are not just simple verbs like 'running', 'walking', 'jumping' etc. as in the early days of action recognition but complex and compound *phrases* like 'hits a forehand volley', 'delivers a backhand return' etc. Although learning such activities add to the complexities of the task, yet they make our descriptions diverse and vivid. To further integrate finer details into descriptions, we consider constructs



Upper Player: smashes down the line.
Lower Player: waits for the ball.

Upper Player: struggles to reach ball.
Lower Player: massive serve.

IN, Winner: Serena!!! Williams arrows a good serve at T, Sharapova is unable to return it.

IN, Winner: Djokovic!!! Slice serve, Tsonga fails to put it back.

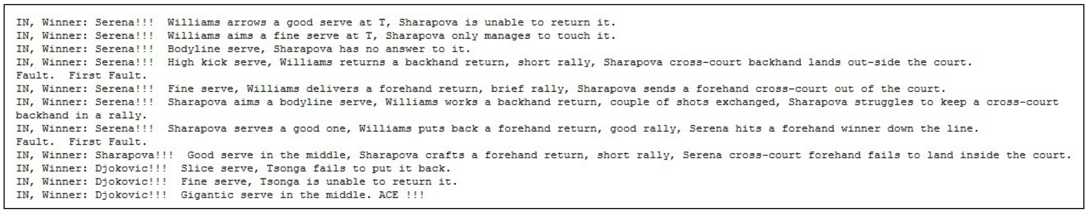
Figure 3.3: Dataset contents: (a) Annotated-action dataset: short videos aligned with verb phrases. (b) Video commentary dataset: game videos aligned with commentary.

that modify the effectiveness of nouns and verbs. Though phrases like 'hits a nice serve' ,'hits a good serve' and 'sizzling serve' describe similar action, '*nice*', '*good*' and '*sizzling*' add to the intensity of that action. We develop a model that learns the effectiveness of such phrases, and builds upon these to predict florid descriptions. Empirical evidences demonstrate that our approach predicts descriptions that match the videos.

Lawn tennis is a racquet sport played either individually against a single opponent (*singles*), or between two teams of two players each (*doubles*). We restrict our attention to singles. Videos of such matches have two players – one in the upper half and the other in the lower half of a video frame. A complete tennis match is an amalgamation of sequence of 'tennis-sets', each comprising of a sequence of 'tennis-games' played with service alternating between consecutive sets. A 'tennis-point' is the smallest sub-division of match that begins with the start of the service and ends when a scoring criteria is met. We work at this granularity.

3.2 Motivation

We seek to analyse how focusing on a specific domain confines the output space. We compute the count of unique (word-level) unigrams, bigrams and trigrams in tennis commentaries. Each commentary sentence in the Tennis-text corpus is processed individually using standard Natural Language ToolKit (NLTK) library, and word-level n-gram frequencies of corpus are computed. The 'frequency' (count) trends of unigrams, bigrams and trigrams plotted over 'corpus size' (number of lines in corpus) are depicted in Figure 3.2. We compare these with the corresponding frequencies in unrestricted tennis text mined from on-line tennis news, blogs, etc. (denoted by '*' in the figure). It can be observed that in case of tennis commentary, the frequency of each n-gram saturates well within a small corpus as compared to corresponding frequencies of unrestricted text. The frequency plots reveal that the vocabulary specific to tennis *commentary* is indeed small, and sentences are often very similar. Hence, in a domain specific environment, we can create rich descriptions even with a limited corpus size.



```
IN, Winner: Serena!!!  Williams arrows a good serve at T, Sharapova is unable to return it.
IN, Winner: Serena!!!  Williams aims a fine serve at T, Sharapova only manages to touch it.
IN, Winner: Serena!!!  Bodyline serve, Sharapova has no answer to it.
IN, Winner: Serena!!!  High kick serve, Williams returns a backhand return, short rally, Sharapova cross-court backhand lands out-side the court.
Fault.  First Fault.
IN, Winner: Serena!!!  Fine serve, Williams delivers a forehand return, brief rally, Sharapova sends a forehand cross-court out of the court.
IN, Winner: Serena!!!  Sharapova aims a bodyline serve, Williams works a backhand return, couple of shots exchanged, Sharapova struggles to keep a cross-court
backhand in a rally.
IN, Winner: Serena!!!  Sharapova serves a good one, Williams puts back a forehand return, good rally, Serena hits a forehand winner down the line.
Fault.  First Fault.
IN, Winner: Sharapova!!!  Good serve in the middle, Sharapova crafts a forehand return, short rally, Serena cross-court forehand fails to land inside the court.
IN, Winner: Djokovic!!!  Slice serve, Tsonga fails to put it back.
IN, Winner: Djokovic!!!  Fine serve, Tsonga is unable to return it.
IN, Winner: Djokovic!!!  Gigantic serve in the middle. ACE !!!
```

Figure 3.4: Text Corpus: Glimpse of online scraped commentary text. Each commentary line is comprised of 'player names', 'dominant shots' and other details.

3.3 Dataset

We use broadcast video recordings for five matches from *London Olympics* 2012 for our experiments. The videos used are of resolution 640×360 at 30 fps. Each video is manually segmented into shots corresponding to 'tennis-points', and is described with a textual commentary obtained from [1]. This gives a collection of video segments aligned with corresponding commentaries. In total, there are 710 'tennis-points' of average frame length 155. We refer to this collection as 'Video-commentary' dataset. This serves as our test dataset and is used for the final evaluation. In addition to this, we create an independent 'Annotated-action' data set comprising 250 short videos (average length of 30 frames) describing player actions with verb phrases. Examples of the verb phrases include *'serves in the middle'*, *'punches a volley'*, *'rushes towards net'*, etc. In total, we have 76 action phrases. We use this collection to train our action classifiers. Figure 3.3 and 3.4 show samples from our dataset. 'Annotated action' dataset comprises of video shots linked with corresponding action phrases. Figure 6.3 showcases some of the contents of 'Annotated action' dataset.

As an additional linguistic resource for creating human readable descriptions, we crawl tennis commentaries (with no corresponding videos). This text corpus is built using (human-written) commentary of 2689 lawn tennis matches played between 2009-14 from [1]. A typical commentary describes the players names, prominent shots and the winner of the game. We refer this collection as 'Tennis-text'. Table 3.1 summarizes the three datasets. Note that all the datasets are independent, with no overlap among them.

Name	Contents	Role
Annotated-action	250 action videos and phrases	Classification and Training
Video-commentary	710 game videos and commentaries.	Testing
Tennis Text	$435K$ commentary lines	Dictionary Learning, Evaluation and Retrieval

Table 3.1: Dataset statistics: Our dataset is a culmination of three standalone datasets. Table describes them in detail, along with the roles they play in the experiments.

Chapter 4

Action Localization

Recognition of human actions and activities is one of the most sought out topics in automatic video analysis. The two most relevant and related problems in this area are localization and classification of human actions. Much of the learning has been successfully applied to object localization in past, where the mapping between an image and an object bounding box can be well captured. Extension of such approaches action localization in videos is much more challenging. One needs to predict the locations of the action patterns both spatially and temporally, i.e., identifying a sequence of bounding boxes that track the action in video. The problem becomes intractable due to the exponentially large size of the video space enriched with actions. In this chapter we deal with preliminaries and explain how does domain restrictions eases the the task of action localization.

4.1 Introduction

While the idea of action localization is not new, it has been applied to video restricted to a static cameras [30, 88] or videos with simple background with limited clutter [18, 45]. It is easy to apply simpler methods such as background subtraction (to localize actors) in such restricted cases which is nearly an impossible case with moving cameras. Figure 4.1 explains one such scenario. Localizing natural actions in realistic cluttered videos has been challenging – localizing actions by training an action-pose specific human detector [43] and use visual words [82] to discriminate actions are slow and data dependent. We instead use 'domain knowledge' to generate a real time solution in our case.

Early work on action recognition [15, 31, 65] in video used sequences with prominent actions, mainly static cameras, simple backgrounds and full bodies visible, as in the KTH [9] and Weizmann [21] datasets. This enabled action classifiers to be explored with variation in the actors and actions, but without the added complexity of change of viewpoint, scale, lighting, partial occlusion, complex background etc. Predicting the locations of the action patterns is far more demanding than classification. In the case of action classification, which is often the aim in action analysis, sequences with pre-defined temporal extent are labeled as belonging to one of the action classes. However, for video search and annotation applications, action localization is far more important and acts like a precursor to classifica-

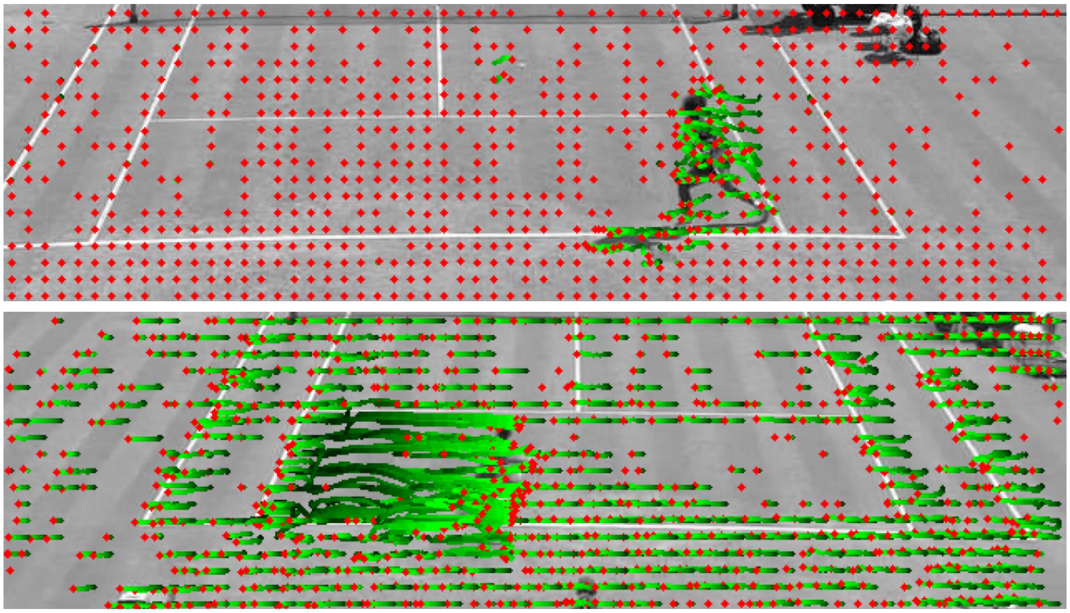


Figure 4.1: Camera motion makes localization a difficult task: (UP) Trajectories with no camera motion (BELOW) False trajectories due to camera motion

tion. It enables temporal sequence containing the action to be delimited and also the actor carrying out the action to be identified, when there are multiple actors in a shot.

We take a similar approach and focus our attention on identifying active action zones. We use an approach which splits the spatio-temporal action localization into first detecting court area and subsequently tracking the players. This determines the spatial localization of the action, followed by a temporal action classification of the computed trajectories, and hence detect and localize the actions in time.

4.2 Action Localization

4.2.1 Court Detection

We begin by identifying/tracking both players on the tennis court. The playing court in lawn tennis has a set of prominent straight white lines, each of which adds meaning to the game in a unique way. We detect these lines using the Hough transform and consider only the most prominent lines (by keeping a threshold on length). A bounding-box is created encompassing the set of identified lines, which is then considered as foreground seed for the GrabCut algorithm [61]. This in turn returns the playing field as shown in Figure 4.2. We tested our algorithm on the extended dataset of others tournaments

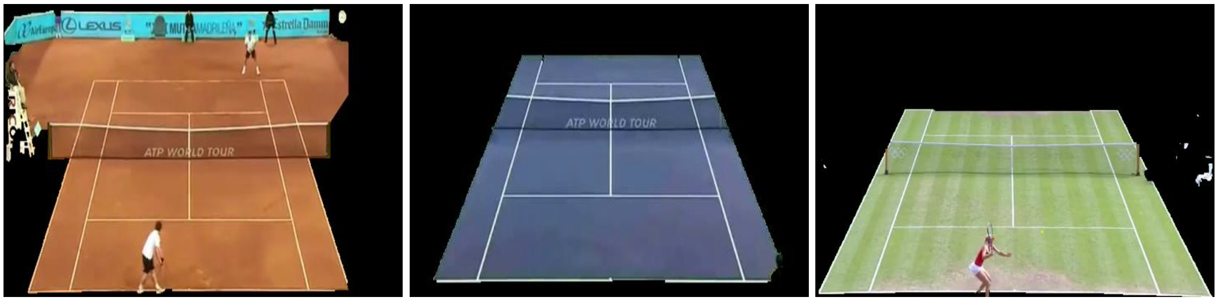


Figure 4.2: Court detection for various cases: Overlapping structures, lack of prominent field lines (due to lack of contrast) and other occlusions

(random videos from `youtube.com`) to test the robustness of our court detection solutions. Figure 4.2 showcases some of success and failure cases.

4.2.2 Player Detection

In a tennis broadcast video background is (nearly) static after the serve begins, and remains the same till a player wins a point. Based on this, the candidate regions for players are segmented through background subtraction using thresholding and connected component analysis. Each candidate foreground region thus obtained is represented using HOG descriptor [10]. To prune away false detections (i.e., non-player regions), multi-class SVM with RBF-kernel is employed distinguishing between ‘upper-player’, ‘lower-player’ and ‘no player’ regions. Figure 4.4(b) visualizes HOG features for few examples. As done in the case of court detection we tested our algorithm on extended dataset of others tournaments (random videos from `youtube.com`). Figure 4.3 depicts few unique cases.

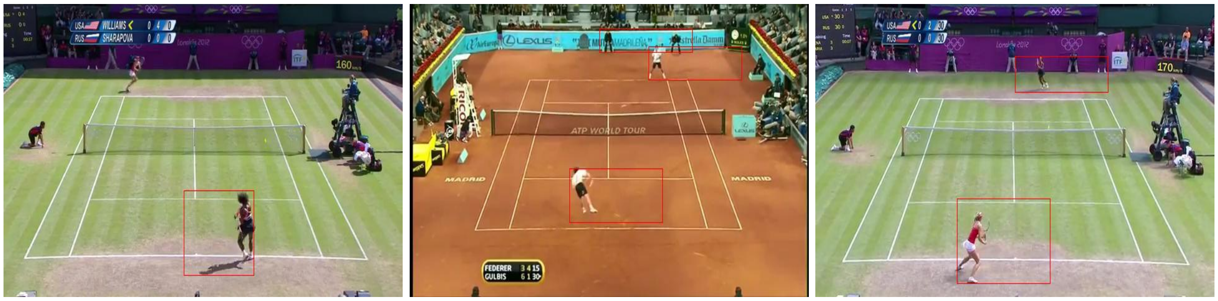


Figure 4.3: Player detection for various success and failure cases.

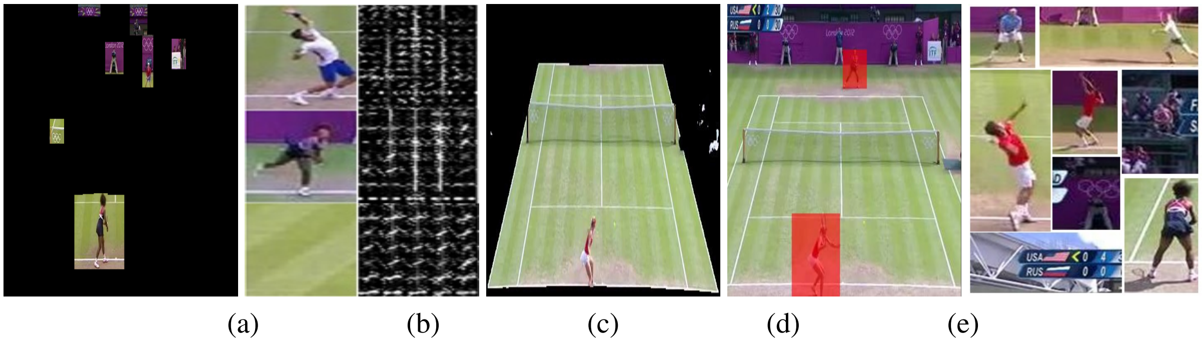


Figure 4.4: Retrieving player details: (a) Extracted foreground regions. (b) Visualization of HOG features for players and non-player regions. (c) Court detection. (d) Player Detection. (d) Some examples of successful and failed player detections.

4.2.3 Player Recognition

The detected windows thus obtained are used to recognize players. In any particular tournament (and in general) players often wear similar colored jerseys and depict unique stance during game play, we use these cues to recognize them. We perceive both color and stance information using CEDD [6] descriptor. This descriptor captures both colour and edge (for stance) information of each detected candidate player region. We use Tanimoto distance [6] to build our classifier. Classifier scores averaged over initial ten frames are used to recognize both the players. Figure 4.4(d) highlights the players detected in a frame, and Figure 4.4(e) shows some true and false detections.

4.3 Experiments and Results

There are two players in the game and they are typically in the top and bottom half of the video when they play. To process the videos, we begin by tracking both the players on the field. We begin by isolating the playing area from rest of the field using hough lines and grabCut algorithm [61]. We create a bounding box encompassing the set of lines identified. Area inside this box is considered as foreground seed for grabCut algorithm which in turn returns playing field as shown in Figure 4.4(c). We extract HOG features for each of the candidate foreground regions (identified player regions) so obtained after background subtraction and prune away false detections (non-player regions) by using classifier built over HOG features. Thereafter the players are identified using CEDD features. Some of the true and false positives detections are depicted in and Figure 4.4(e).

Since major activities during game take place on each side of the net we analyse videos by dividing them across net. In almost all tennis broadcasts net is at the center of the frame. The upper and lower videos thus formed are analysed separately in subsequent sections. Note that the activities taking place in these two halts are correlated. We use this larger game structure when we refine the action labels using MRFs. We begin by foreground extraction to identify the players in input video. This step results in many potential candidate regions. To prune out the false candidate regions we learn SVM classifier to differentiate between regions that contain a player from regions that do not contain them. We train this classifier using hand labelled examples. Player bounding boxes (for both upper and lower players) were identified by randomly drawing bounding boxes across players in video frames of training set. Training set comprised of 2432 'lower player', 2421 'upper player' and 13050 'no players' windows. We extract HOG features for each of these candidate bounding boxes and model it as a multi-class classification problem for three identified categories.

For recognition purpose we learn classifiers over CEDD features built using 2432 'lower players' and 2419 'upper players' windows. Each window is identified by one of the eight unique player names in our training set. We run our player recognition classifier over selected candidate regions generated by the previous module and predict the player name. Since in any frame we can have only two players we use this information to suppress left over false positives (if any) by considering only top two classifier scores. Table 4.1 summarize both court detections and player detections results. Our recognition accuracies are 0.98 and 0.978 for lower and upper players respectively.

	Correct #	Actual #
Court	621	710
Player	746	1420

Table 4.1: Court and Player Detection: Detections on test videos each with two players on single court. In all we have 710 videos in our test dataset and hence 710 × 2 = 1420 players in all. The players thus detected are then identified using recognition module.

Chapter 5

Weak Learners for Action recognition

Field of human activity recognition has grown dramatically over last few years. Much of the credit of such growth goes to usage of activity recognition in many high-impact applications such as smart surveillance, query based video retrieval etc. With advent of new devices and technological advancements this growth is destined to grow even more in coming days. The field is no more restricted to limited set of actions and is gradually moving towards more structured interpretation of complex human activities. We describe one such case in this chapter and use recently proposed 'improved dense trajectories' [81] for action classification. We learn fine grained action classifiers to generate 'action phrases'. The trajectory features used for classification encode the optical flow based motion information and are robust to fast irregular motions as well as camera motions. [80, 81]

5.1 Introduction

Action classification has been studied in a variety of settings [9, 18]. Initial attempts used the laboratory videos of few well defined actions. Of late, interest has been shifted to the actions captured in the natural settings like movies [42, 17]. They have even taken help of contextual information [23, 63, 83] and movie scripts [17, 42] as means of weak supervision to describe human actions in videos. In all these cases, the actions of interest are visually and semantically very different. Some of the recent methods have also started looking into the fine grain classification of human actions. At times when there is low inter-action and high intra-action variability in human actions people have used such approaches to distinguish between subtle variations in actions [59].

Videos were annotated with natural sentences by aligning the parallel scripts available elsewhere [64]. However, this was applicable only for a very limited class of videos. In recent years, there has been many successful attempts that created natural language sentences for images and videos without any parallel textual descriptions. Focus has now shifted from recognizing nouns, verbs to more complex task of observing *actor-action-object* relationships. Niveda *et al.* [39] generate descriptions for short videos by identifying the best SVO *(subject-verb-object)* triplet. They determine the likelihood of various SVO combinations from large corpus and select best among all to generate the sentences. Sergio *et al.* [22]

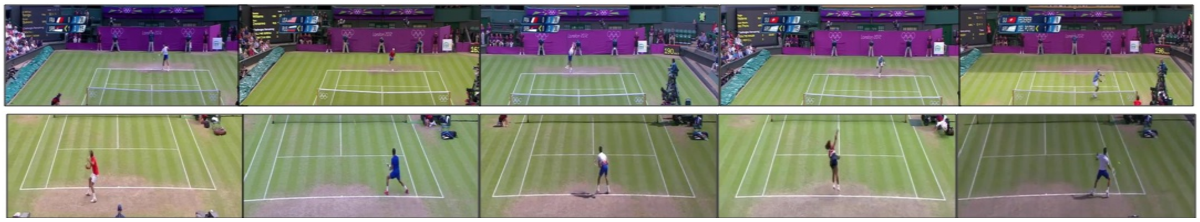


Figure 5.1: Example frames depicting varied actions. Upper frames are shown at the top, and lower frames at the bottom. Here, upper and lower frames do not correspond to same video.

extend this for out-of domain actions (not in training set) by generating brief sentences that sum up main activity in the video with broader verb and object coverage. Barbu *et al.* [4] produce rich sequential descriptions of video using detection based tracking and body-posture codebook to determine the subject, verb and object.

Our goal is to automatically describe video segments of *tennis points* in the Video-commentary dataset. We begin by learning phrase classifiers using Annotated-action dataset. Given a test video, we predict a set of action/verb phrases individually for each frame using the features computed from its neighbourhood. Since this sequence of labels could be noisy, these are smoothed by introducing priors in an energy minimization framework. The identified phrases along with additional meta-data (such as player details) are used to find the best matching description from the Tennis-text dataset. Major activities during any tennis game take place on each side of the net, we analyse videos by dividing them across the net in the subsequent sections (referred as 'upper' and 'lower' video/frame). In almost all tennis broadcast videos, this net is around the center of a frame, and thus can be easily approximated.

For the game of tennis, which has a pair of players hitting the ball each other, actions are the central component. Here, actions are not simple verbs like 'running', 'walking' and 'jumping'as in the early days of action recognition. Many recent methods [39] use verbs in interchangeable fashion to describe activities in videos ('hits a forehand return', 'delivers a forehand return'). In our case, such richer action labels are obtained by mining a large text data and learning the semantic relationships that are able to differentiate between lesser and more frequent action words [22]. We also add finer details to our descriptions by placing constructs that modifies the effectiveness of nouns and verbs in our sentences. 'hits a nice serve','hits a good serve'and 'sizzling serve'etc. describe similar task yet *nice, good and sizzling* add to the intensity of similar action. We develop a model that learns effectiveness of phrases and build upon these phrases to generate descriptions that are florid and verbose.

5.2 Learning Action Phrases

5.2.1 Supervised Approach

We learn phrase classifiers using 'Annotated-action' dataset. For representation, we use descriptors as described in [80], and extract dense trajectory features over space-time volumes (using default pa-

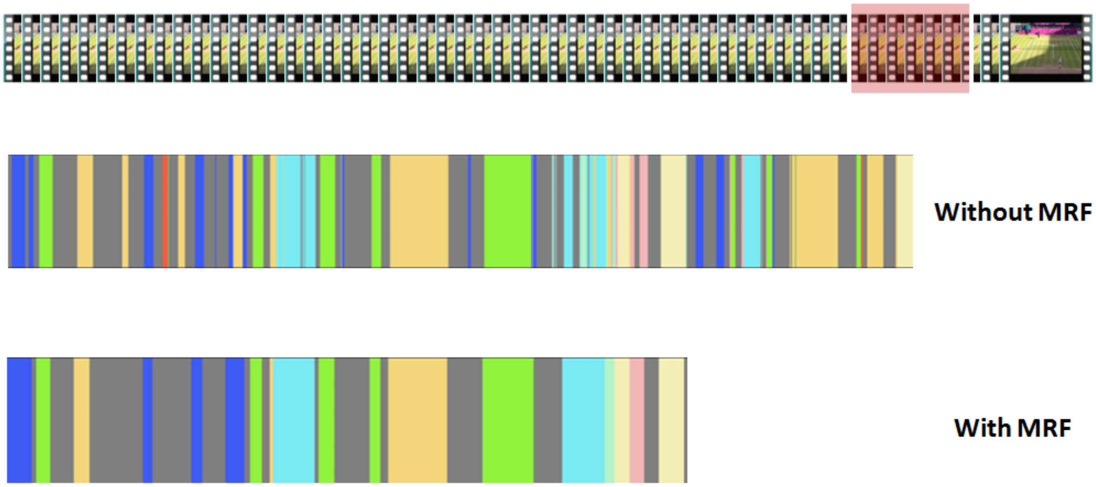


Figure 5.2: Effect of using MRF based Temporal Smoothing. Coloured segments represent various 'action phrases'.

rameters). For each trajectory, we use Trajetory, HOG, HOF (histograms of optical flow) [7] and MBH (motion boundary histogram) [11] descriptors. While HOG captures static appearance information, HOF and MBH measure motion information based on optical flow. The dimensions of each of these descriptors are: 30 for Trajectory, 96 for HOG, 108 for HOF and 192 for MBH. For each descriptor, bag-of-words (BOW) representation is adopted (with vocabulary size 2000). We take square root of each element in a feature vector before computing the codebook (similar to RootSIFT [3]). The final representation is concatenation of BOW histograms of all the descriptors. Using this, a 1-vs-rest SVM classifier (with RBF kernel and χ^2 distance) is learned for each phrase. In all, we have 76 verb phrases - 39 for upper and 37 for lower player. Figure 6.3 illustrates some examples of player actions.

5.2.1.1 Verb Phrase Prediction and Temporal Smoothing

Given a (test) video, we recognize verb phrase for each frame by extracting features from neighbouring frames using sliding window (neighbourhood of size 30 frames). Since this typically results into multiple firings, non-maximal suppression (NMS) is applied. This removes low-scored responses that are in the neighbourhood of responses with locally maximal confidence scores. Once we get potential phrases for all windows along with their scores, we remove the independence assumption and smooth the predictions using an energy minimization framework. For this, a Markov Random Field (MRF) based model is used which captures dependencies among nearby phrases. We add one node for each window sequentially from left to right and connect these by edges. Each node takes a label from the set of action phrases. The energy function for nodes ν, neighbourhood $\mathcal{N}$ and labels $\mathcal{L}$ is:

$$E = \sum_{p \in \nu} D_p(f_p) + \sum_{p,q \in \mathcal{N}} V_{pq}(f_p, f_q) \tag{5.1}$$

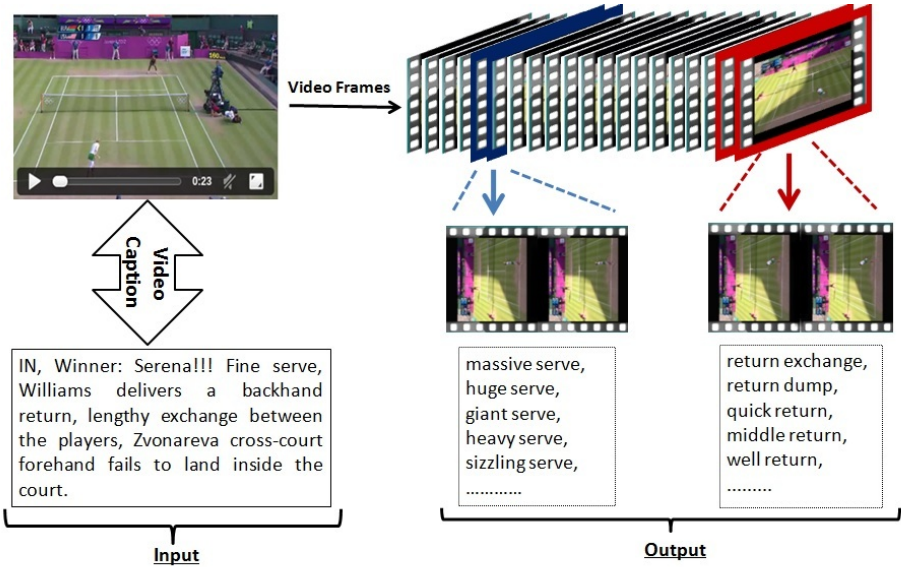
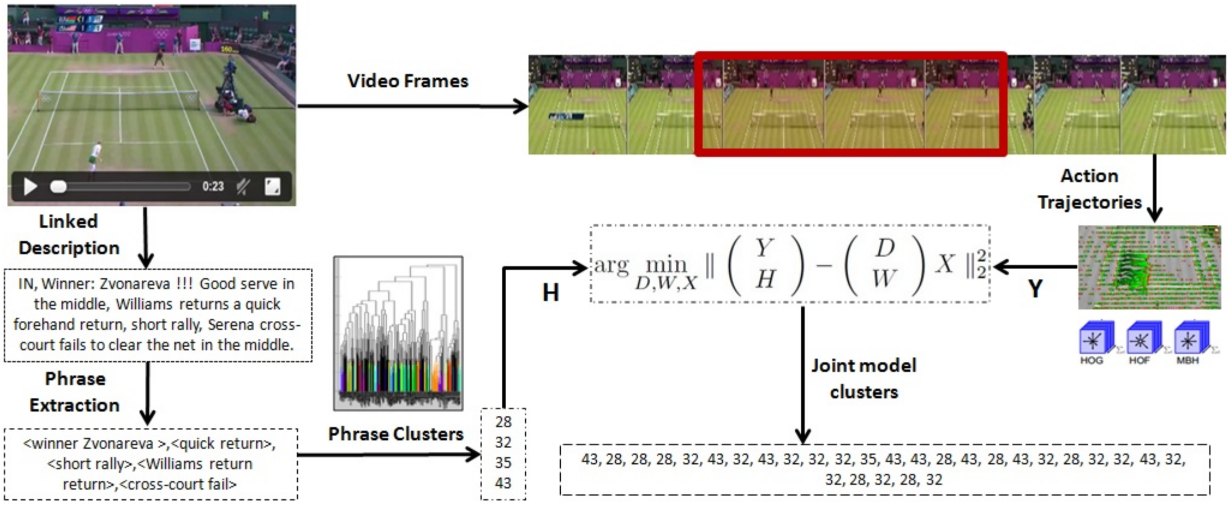


Figure 5.3: Semi-Supervised approach: (*Input:*) Given a collection of tennis videos along with linked captions, (*Output:*) Our approach generates annotations for constituent frames of each input video. The approach aligns video frames with corresponding 'action phrases'. Window size of 'two' assigns similar labels to adjacent two frames.

Here, $D_p(f_p)$ denotes *unary phrase selection cost*. This is set to $1 - p(l_j|x_p)$, where $p(l_j|x_p)$ is the SVM score of the phrase label l_j for node x_p, normalized using the Platt's method [57]. The term $V_{pq}(f_p, f_q)$ denotes *pairwise phrase cohesion cost* associated with two neighbouring nodes x_i and x_j taking some phrase label. For each pair of phrases, this is determined by their probability of occurring together in the game play, and is computed using frame-wise transition probability. In our case, since there are two players, and each player's shot depends on the other player's shot and his own previous shot, we consider four probability scores: $p(l_{iP_1}, l_{jP_1})$, $p(l_{iP_1}, l_{jP_2})$, $p(l_{iP_2}, l_{jP_2})$ and $p(l_{iP_2}, l_{jP_1})$. Here, $p(l_{iP_1}, l_{jP_2})$ refers to the probability of phrase l_i of $player1$ and l_j of $player2$ occurring together during game play. We compute pairwise cost, $1-p$, for each of the four probabilities and solve the minimization problem using a loopy belief propagation (BP) algorithm [35].

5.2.2 Weakly Labelled Semi-Supervised Approach

In this section we propose a that method encodes local temporal structure in tennis videos and aligns frames with appropriate find grained annotations ,i.e. 'phrases' that best describe the constituent video frames. Our solution bears resemblance to method [2] that learns sequence of steps to complete certain tasks – discriminative joint clustering scheme is proposed by authors to identify prominent steps from verbal descriptions and align them to appropriate video frames. We take a step towards weakly labelled semi-supervised 'action phrase' recognition for tennis videos and suggest a unified objective function



IN, Winner: Zvonareva !!! Good serve in the middle, Williams returns a quick forehand return, short rally, Serena cross-court fails to clear the net in the middle.

<winner Zvonareva >,<quick return>, <short rally>,<Williams return return>,<cross-court fail>

28
32
35
43

$$\arg\min_{D,W,X} \| \begin{pmatrix} Y \\ H \end{pmatrix} - \begin{pmatrix} D \\ W \end{pmatrix} X \|_2^2$$

43, 28, 28, 28, 32, 43, 32, 43, 32, 32, 32, 35, 43, 43, 28, 43, 28, 43, 32, 28, 32, 32, 43, 32, 32, 28, 32, 28, 32

Figure 5.4: The proposed approach: Every frame is associated with a corresponding tag which is a collection of similar 'action phrases'. Both trajectory matrix (Y) and phrase-cluster matrix (H) are stacked together to compute the dictionary. The dictionary is used to generate frame level annotations for the input videos. We represent each group of 'phrases' by a cluster number.

comprising of probabilistic label consistent constraint and classification error. The proposed objective function solved using KSVD yields dictionaries such that local temporal structures of videos bearing similar actions have similar action phrases. We use probabilistic label constrained KSVD for learning sparse dictionaries for recognition [32]. The suggested approach utilizes video descriptions at hand to learn different varied constructs and associates them with actions features extracted from videos. It extracts various linguistic phrases [26] from available sentences and clusters them into group of semantically similar [27] 'action phrases', for e.g. phrases like 'huge rally' and 'contested rally' are part of one cluster and 'backhand catches net' and 'Serena catches net' belong to same cluster. The phrase clusters are thereafter used for classifying action features – similar looking actions are labelled with similar phrase clusters. A group of semantically similar phrases belong to same 'phrase clusters' and group of frames encoding similar actions are assigned similar phrase clusters as corresponding labels.

5.2.2.1 Phrase Extraction and Clustering

We automate the process of phrase extraction from given video descriptions using CoreNLP toolkit [1]. We use 'collapsed-coprocessed-dependencies' [12], i.e. dependencies involving prepositions and conjuncts are collapsed to reflect direct relation between content words. Each sentence is mapped to list of nine distinct types of phrases encoding order preference information – (subject), (object), (subject;verb), (object;verb), (subject;prep;object), (object;prep;object), (attribute;subject), (attribute;object)

[1] http://nlp.stanford.edu/software/corenlp.shtml

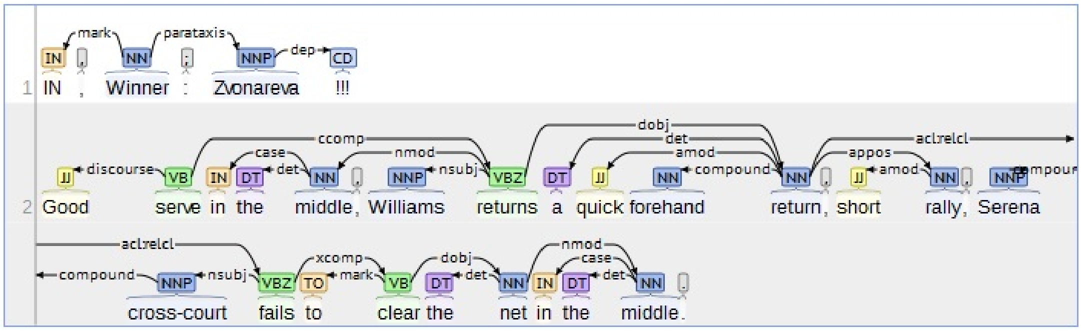

1 IN , Winner : Zvonareva !!!
IN NN NNP CD
mark
parataxis
dep
2 Good serve in the middle, Williams returns a quick forehand return, short rally, Serena
JJ VB IN DT NN NNP VBZ DT JJ NN NN JJ NN NNP
discourse
case
det
ccomp
nmod
nsubj
dobj
det
amod
compound
appos
amod
aclrelcl
compour
cross-court fails to clear the net in the middle.
NNP VBZ TO VB DT NN IN DT NN
aclrelcl
compound
nsubj
xcomp
mark
dobj
det
nmod
case
det

Figure 5.5: Parsed dependencies (collapsed and propagated) for an input commentary text. We only keep nine selected encodings and discard others to assimilate all possible phrase information in the linked sentence.

and (verb;prep;object), Figure 5.5. We believe these nine encodings are sufficient to assimilate all possible information in linked sentence. Since generated phrases are targeted to increase overall retrieval efficiency we kept player names en-tact in the extracted phrases.

The extracted phrases are clustered using hierarchical agglomerative clustering using Semantic Textual Similarity (STS) measure described in [27]. In an agglomerative setting each phrase starts with its own cluster and subsequently the pairs of text clusters are merged as one climbs up the hierarchy. The similarity scores describe the degree of equivalence in the underlying semantics of paired snippets of text (phrases). The distance metric uses a word similarity feature combining both LSA word similarity and WORDNET knowledge. The text clusters so obtained are visualized using a dendrogram as shown in Figure 5.6

5.2.2.2 Dictionary Learning for Action-Phrase Alignment: Probabilistic Label Consistent KSVD

For action representation, we extract dense trajectory features to describe actions and process them similar to previous sections. Given set of videos, we extract action features from both upper and lower part of frames (capturing actions of both upper and lower player respectively). Features are computed from neighbouring frames using sliding window (neighbourhood of size 30 frames with no overlap). Computed action features (from both halves) when stacked over each other represent the feature vector of frames in sliding window. We aim to classify stacked action features depended on phrase clusters computed in section 5.2.2.1. A compact and discriminative dictionary is learnt for sparse coding which is thereafter used for classification, 5.6

Let, Y be a matrix of stacked video features (from upper and lower halves) of training videos (collection of N sliding windows). A single training video is represented by consecutive columns of Y and number of such columns equate to count of sliding windows encompassing whole video. Every column represents feature vector of frames in one sliding window, each of n dimensions, i.e.

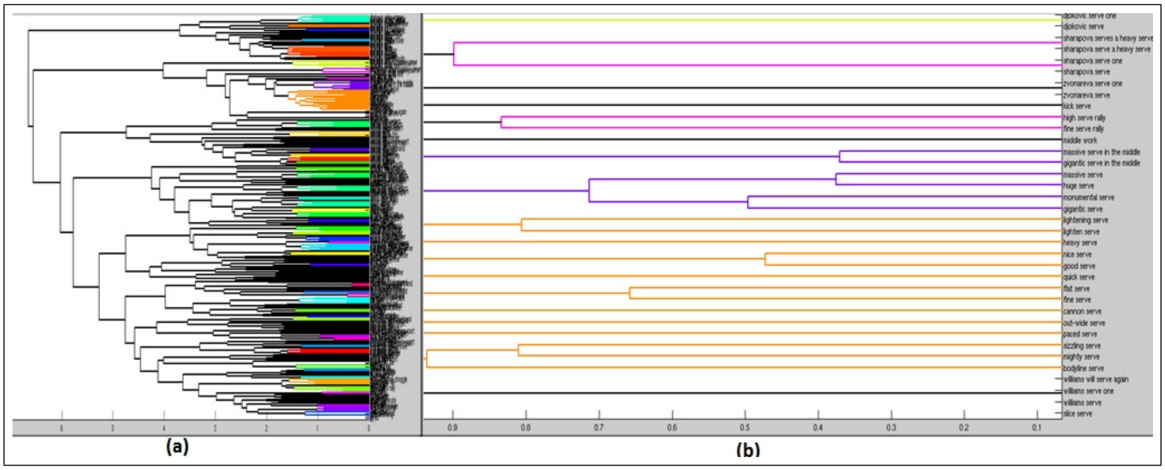


Figure 5.6: Qualitative Result for phrase clustering in semi-supervised regime: (a) Dendrogram of clusters of all generated phrases (b) Dendrogram of prominent phrases.

$Y = \{y_1, y_2 \dots y_N\} \in R^{n \times N}$. We learn a single re-constructive dictionary, D, with K items for sparse representation of Y:

$$< D, X >= \arg\min_{D,X} \parallel Y - DX \parallel_2^2, s.t.\forall i, \parallel x_i \parallel_0 \leq T \tag{5.2}$$

Here, $D = \{d_1, d_2 \dots d_K\} \in R^{n \times K}$ is the learnt dictionary, $X = \{x_1, x_2 \dots x_N \in R^{K \times N}$ are sparse codes of Y and T is sparsity constraint factor (each sparse code has fewer than T items). We leverage the supervised information (i.e. phrase labels obtained after text clustering) of feature vectors to learn a more efficient dictionary. In order to make dictionary optimal for classification we include the classification error term in the objective function for dictionary learning. Similar to [32] we use a linear predictive classifier $f(x; W) = Wx$ and learn weights,W. Following objective function for learning a dictionary, encompasses both reconstruction and classification errors:

$$< D, W, X >= \arg\min_{D,W,X} \parallel Y - DX \parallel_2^2 + \parallel H - WX \parallel_2^2, s.t.\forall i, \parallel x_i \parallel_0 \leq T \tag{5.3}$$

There is an explicit correspondence between dictionary items and the phrase labels. The term $\parallel H - WX \parallel_2^2$ represents classification error, with W being the classifier (weights) parameters. $H = \{h_1, h_2 \dots h_K\} \in R^{m \times N}$ are the class labels of Y. A video (represented by consecutive columns of Y) can have multiple (extracted) phrases, so we assign equal probabilities to corresponding phrases to each column of H. Number of columns depends on number of features columns in Y for each video.

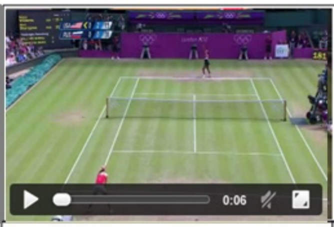
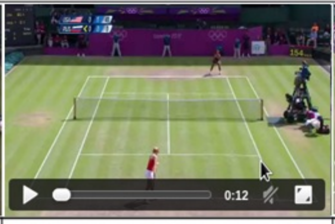
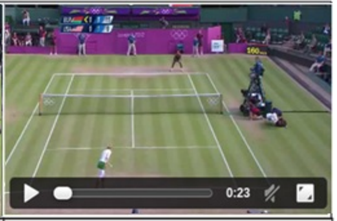
Input Video (Lawn Tennis)			
Linked Commentary	IN, Winner: Serena!!! Huge serve. Ace !!!	IN, Winner: Sharapova!!! Quick serve, Sharapova crafts a forehand return, Serena goes for a forehand down the line but catches the net	IN, Winner: Zvonareva !!! Good serve in the middle, Williams returns a quick forehand return, short rally, Serena cross-court fails to clear the net in the middle.
Phrases Generated	<winner Serena>, <huge serve> <ace>	<winner Sharapova>,<quick serve>, <Sharapova craft return>, <Serena catch net>,<Serena go>	<winner Zvonareva >,<quick return>, <short rally>,<Williams return return>,<cross-court fail>
Phrase Clusters	4, 23	23, 32, 36, 42, 46	28, 32, 35, 43
Joint Model (output)	4, 23, 23, 4	42, 23, 32, 42, 42,42	43, 28, 28, 28, 32, 43, 32, 43, 32, 32, 32, 35, 43, 43, 28, 43, 28, 43, 32, 28, 32, 32, 43, 32, 32, 28, 32, 28, 32

Figure 5.7: Illustration of semi-supervised approach: First two rows correspond to videos and the linked description. The extracted phrases and assigned phrase clusters are shown in next two rows. Last row demonstrates output phrase clusters obtained using proposed approach (Number of such clusters depend on duration of video input)

A video with 90 frames will constitute $(90/30)$=3 columns (sliding window size: 30 frames) in matrix Y and H; y_i corresponds to stacked dense trajectory feature of 30 frames and h_i represents equal probabilities of phrase labels (all three corresponding columns in H will be identical). $h_i = [0, 0.33, 0.33, 0, 0.33, \ldots, 0, 0]$ would mean that phrase identified from linked description belong to phrase clusters $2, 3, 5$ (non-zero entries in h_i), Figure 5.8. In such case we assign each phrase cluster an equal probability $(1/3 = 0.33)$. Intuitively this would mean this video is comprised of these three cluster labels and hence classification error should be minimized with respect to all three. We use KSVD to find optimal solution for all parameters simultaneously. The objective function can be re-written as:

$$< D, W, X >= \arg\min_{D,W,X} \| \begin{pmatrix} Y \\ H \end{pmatrix} - \begin{pmatrix} D \\ W \end{pmatrix} X \|_2^2, s.t. \forall i, \| x_i \|_0 \leq T \tag{5.4}$$

$$< D, W, X >= \arg\min_{D,W,X} \| Y^* - D^* X \|_2^2, s.t. \forall i, \| x_i \|_0 \leq T \tag{5.5}$$

D^*, is the final dictionary with ‘optimal’ reconstruction and classification error. The learned dictionary makes sure that the features from same class have similar sparse codes and those from different classes have dissimilar sparse codes; hence can be used for action classification.

Initialization: To begin with dictionary learning using KSVD, we initialize the parameters D_0 and W_0 similar to [32]. We use the multivariate ridge regression, with the quadratic loss and L_2 norm regularization. Parameters used are similar to what suggested in [32].

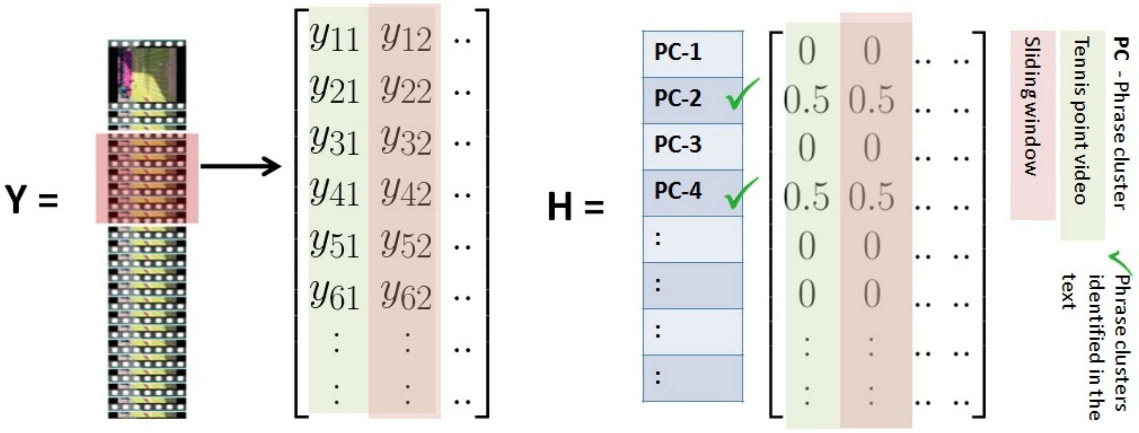


Figure 5.8: Matrix structure: Considering the sliding window size of 'n' frames, a video with m frames $(n < m)$ will constitute (m/n) columns in matrix Y and H; Each column of Y corresponds to the dense trajectory feature of frames in sliding window and each column of H represents equal probabilities of phrase cluster labels detected in the input linked text (m/n adjacent columns in H are identical).

Classification: Both D and W are transformed and normalized using $\dfrac{d_k}{\| d_k \|_2}$ and $\dfrac{w_k}{\| w_k \|_2}$ (for every kth column) before we begin classification. For a test video, we compute sparse representation (x_i) of dense trajectories action features (y_i) using modified dictionary, $\hat{D}$:

$$x_i = \arg\min_{x_i} \| y_i - \hat{D}x_i \|_2^2, s.t.\forall i, \| x_i \|_0 \leq T \tag{5.6}$$

We use the linear predictive classifier $\hat{W}$ to estimate the label j (where $l \in R^m$ is the class label):

$$j = \arg\max_j (l = \hat{W}x_i) \tag{5.7}$$

5.3 Experiments and Results

5.3.1 Supervised Action Learners

For classification experiments we randomly partition our annotated-action dataset into 150 training videos and 100 test videos. We repeat this process for five trials and report average classification accuracy. Labels like *'hits a forehand return'*, *'returns a forehand return'*, *'works a forehand return'* etc. are significantly similar in our case (high correlation between the classes); as a result looking for exact matches could be a strict constraint. We relax these conditions by looking for label matches which are considerably near rather than exact. Phrase pairs

Approach	Upper	Lower
Relaxed	0.869	0.705
Exact	0.770	0.613

Table 5.1: Verb phrase recognition accuracy averaged over top five retrieval.

Class	Upper Player		Lower Player	
	Freq.	Acc.	Freq.	Acc.
waits for ball	64	0.87	44	0.64
player focused	30	0.93	30	0.91
audience seems enjoying	13	0.33	13	0.25
serves a good one	5	0.5	7	0.33
crafts a forehand return	8	0.33	8	0.53
hits a backhand return	3	0.60	7	0.14

Table 5.2: Class classification accuracy for frequent phrases. Freq. refers to number of times these phrases occur in annotated-action dataset and Acc. refers to classification accuracy.

having overlapping subtopics have high similarity values when measured by standard cosine similarity. For exact matches the cosine similarity score is 1 while for relaxed case we consider pairs with cosine similarity score of more than 0.50. We report our results in Table 5.1, averaged over top five retrievals for both exact and relaxed matches. We also report average class wise accuracy of frequent classes in Table 5.2. Figure 6.3 illustrates some actions classes for both upper and lower players. In all we have 76 verb phrases - 39 for upper and 37 for lower player.

5.3.2 Semisupervised Action Learners

For experiments, we partition dataset into 60% train and 40% test data. The training set is used to learn dictionary and other model parameters. At time of text based phrase clustering we use all available linked descriptions.

Clusters (Only Description)	**Aligned Clusters (Joint Model)**
4,23	4,23,23,23
2,16,26,32,36,43	16,43
4,9,16,23,40	23,4,23,16,4,40,4,16,4,9
23,29,48	23,29
4,15,23,26,29,32	23,32,29,23,15
4,23	23,4,23,23,23,4,23,4,4,23,23,2,2,23,18,23,18,49

Table 5.3: Qualitative Result for assigned clusters: Clusters computed using proposed solution bind temporal information and depend on the length of the video.

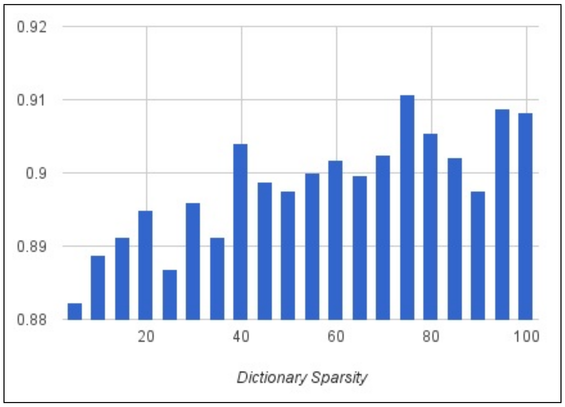


Figure 5.9: Correct match v.s. Dictionary sparsity: Initial phrase clusters (computed using only linked description) act as ground truth and are compared to clusters computed using proposed approach.

While extracting phrases from available descriptions we replace all possible words with respective synonyms determined using WordNet synsets [26]. Overall we had 50 phrase clusters during text based phrase clustering – these clusters determine groups and bin to which a descriptions would belong. Figure 5.7 summarises the steps involved in our method. We compute action phrases and assign clusters to them as described in section 5.2.2.1. The assigned clusters in turn act like a ground truth for comparison purpose. One should note that count of phrases generated and phrase clusters may differ – In video 3 (Fig. 5.7), 'quick return' and 'Williams return return' fall into same clusters. We determine output of the proposed system by using learnt classifiers described in section 5.2.2.2. Number of such assigned phrase clusters depend on the size of the (test) video, which is evident from all examples shown in Figure 5.7.

To evaluate the temporal localization, we need to have a one-to-one mapping between the identified phrase clusters in the video and the ground truth alignment. Owing to lack of such data we demonstrate the effectiveness of proposed system by comparing it with only text based model. Table 5.3 illustrates subtleness involved in the phrase alignments using a joint model. The clusters obtained using join modelling bind temporal information and depend on the length of input video which is not the case with text based modelling which depends only on size of description. We consider clusters computed using only descriptions as ground truth and compare them to clusters obtained using proposed solution. Figure 5.7 illustrates the match between ground truth and computed clusters – an overlap of one cluster is considered as a match.

Approach	Without Frame Division		With Frame Division	
	Upper	Lower	Upper	Lower
Relaxed	0.713	0.636	0.869	0.705
Exact	0.672	0.516	0.770	0.614

Table 5.4: Impact of harnessing domain relevant cues on verb phrase recognition accuracy.

5.4 Baselines and comparisons

Using Domain Specific cues: Compared to a naive use of phrase classifier, our verb phrase classifier performs better by dividing the frame into upper and lower halves. Table 5.4 summarizes the effects of domain specific cues, i.e. division across the net.

Chapter 6

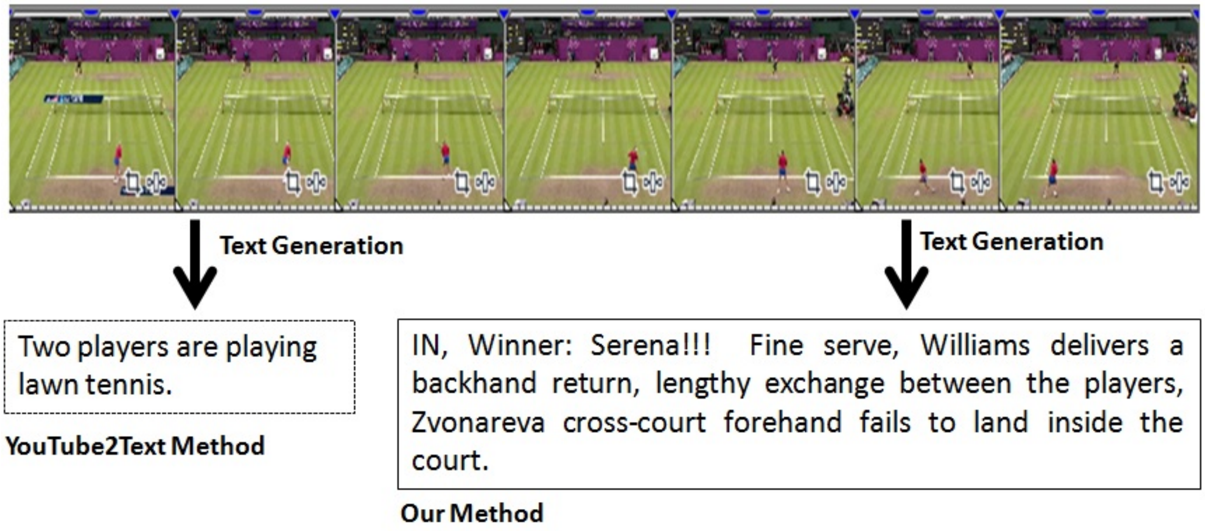
Commentary Generation

Mere a cursory glance of a video content is sufficient for users to describe it in great depth. However, such remarkable ability has proven to be an elusive task for present day state of art visual recognition systems and models. We aim to mimic this unique ‘human ability’ and describe the lawn-tennis video contents using fine-grained descriptions. The chapter describes the final stage of the proposed pipeline and demonstrates the overall efficacy of our approach.

6.1 Introduction

With reincarnation of ‘deep learning’ and other neural network based approaches, caption generation for multimedia content has taken central stage in both vision and language community [34]. Both video [22, 39] and image [75, 46] annotation generation are present day active field of research – given a multimedia content the gaol is to create most descriptive and relevant annotation. Some of these methods find most appropriate phrases/sentence from a database by formulating the problem as a structured output prediction [75]. Another successful paradigm has been to identify objects (nouns) and actions (verbs), and thereafter construct natural language sentences using computational linguistic techniques. In case of images [40], objects localized using precomputed detectors and their spatial arrangements helped in creating meaningful sentences. Videos provide more reliable information about the verbs in those sentences. For example, one can detect “Human” and “running” in a video clip, and then form a sentence like “A person is running towards the left”. While such sentences capture the visual content at certain level of semantics, richer and more meaningful descriptions are required for these techniques to be useful in diverse applications. We aim of describing the video content similar to the way a human *expert* does. For example, “Serena rushes towards left and punches a backhand volley” could have been far more apt for the same video clip. A commentary is actually more elaborate than this.

We attempt to generate descriptions (rather than captions) for broadcast videos of lawn tennis. Figure 6.1 depicts the problem of interest and an example result of our method. We also depict the difference between the captions and descriptions for the same video. Richer descriptions demand deeper understanding of the visual



Two players are playing lawn tennis.

IN, Winner: Serena!!! Fine serve, Williams delivers a backhand return, lengthy exchange between the players, Zvonareva cross-court forehand fails to land inside the court.

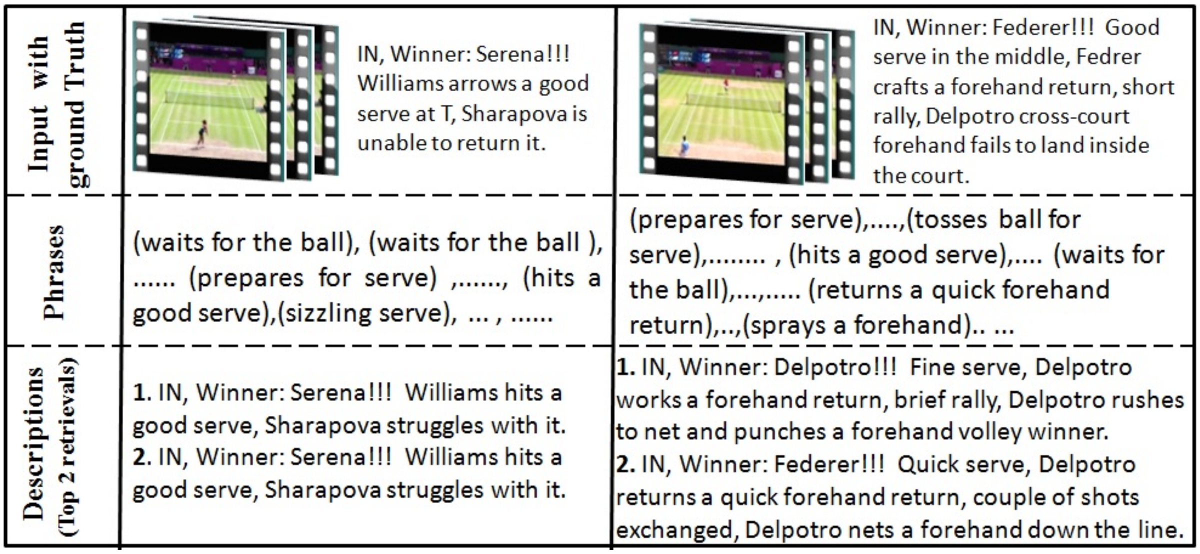
Figure 6.1: Distinction between captions and descriptions. We generate more descriptive sentences when compared to present state of art

content. We address this by limiting our attention to a specific domain. Though these videos were captured in natural setting for human use, they are not diverse like those harvested from youtube [22]. However, the fine variations in the data and the granularity of the description, make the problem challenging. The actions that we need to recognise are of finer granularity compared to the popular action recognition tasks. Richer natural descriptions also demand the sentences to be of higher linguistic quality. We address this by exploiting a large corpus of human commentaries crawled from a public web site [1].

6.2 Description Prediction

Let, $W = \{w_1, w_2, \ldots, w_n\}$ be set of unique words present in the group of phrases along with player names, and $S = \{s_1, s_2, \ldots, s_m\}$ be the set of all the sentences in the Tennis-text corpus. Here, each sentence is a separate and full commentary description. We formulate the task of predicting the final description as an optimization problem of selecting the best commentary from S that covers as many words as possible. Let, x_i be a variable which is 1 if sentence s_i is selected, and 0 otherwise. Similarly, let a_{ij} be a variable which is 1 if sentence s_i contains word w_j, and 0 otherwise. A word w_j is said to be covered if it is present in the selected sentence ($\sum_i a_{ij}x_i = 1$). Hence, our objective is to find a sentence that covers as many words as possible:

$$\max \sum_{i \in \{1,2,\ldots,m\}, j \in \{1,2,\ldots,n\}} a_{ij}x_i, \quad s.t. \sum_{i=1}^{m} x_i = 1, \;\; \forall a_{ij}, x_i \in \{0,1\} \tag{6.1}$$



Input with ground Truth	IN, Winner: Serena!!! Williams arrows a good serve at T, Sharapova is unable to return it.	IN, Winner: Federer!!! Good serve in the middle, Fedrer crafts a forehand return, short rally, Delpotro cross-court forehand fails to land inside the court.
Phrases	(waits for the ball), (waits for the ball), ... (prepares for serve) ,......, (hits a good serve),(sizzling serve), ... , ...	(prepares for serve),.....,(tosses ball for serve),........ , (hits a good serve),.... (waits for the ball),...,..... (returns a quick forehand return),..,(sprays a forehand).. ...
Descriptions (Top 2 retrievals)	**1.** IN, Winner: Serena!!! Williams hits a good serve, Sharapova struggles with it. **2.** IN, Winner: Serena!!! Williams hits a good serve, Sharapova struggles with it.	**1.** IN, Winner: Delpotro!!! Fine serve, Delpotro works a forehand return, brief rally, Delpotro rushes to net and punches a forehand volley winner. **2.** IN, Winner: Federer!!! Quick serve, Delpotro returns a quick forehand return, couple of shots exchanged, Delpotro nets a forehand down the line.

Figure 6.2: Illustration of our approach. Input sequence of videos is first translated into a set of phrases, which are then used to produce the final description.

In the above formulation, doing naïve lexical matching can be inaccurate as it would consider just the presence/absence of words, and fail to capture the overall semantics of the text. To address this, we adopt Latent Semantic Indexing (LSI) [13], and use statistically derived conceptual indices rather than individual words. LSI assumes an underlying structure in word usage that is partially obscured by variability in word choice. It projects derived phrases and corpus sentences into a lower dimensional subspace, and addresses the problems of synonymy (similar meaning words). Figure 6.2 illustrates the steps involved in our method by taking two examples. The verb phrase prediction and smoothing steps provide a set of relevant phrases. Number of such phrases depend on the size of the (test) video. This is evident from the second example (right), which is of longer duration and thus has more phrases predictions. These phrases are used to select the best matching commentary from the Tennis-text corpus. Since similar events are described by identical descriptions in text corpus, there could be instances where the retrieved descriptions are same – first example (left) in Figure 6.2.

6.3 Experiments and Results

We use an action classifier similar to that presented in chapter 5. During test time the classifier is evaluated on every frame by extracting features from neighbourhood of 30 frames. To encode the contextual information into the classifier they are trained on annotated-action dataset by jittering the clip boundaries. A frame-wise transition probability is computed from the annotated frames that constitute the videos of Annotated-action classifiers and is used as pair wise costs to solve MRF using loopy BP implementation of [35]. Using these phrases we build retrieval system over our corpus using both naive lexical and LSI approach.

Corpus #	Vector #	B-1	B-2	B-3	B-4
100	85	0.3789	0.2348	0.1545	0.0954
500	118	0.4279	0.2510	0.1676	0.1071
1K	126	0.4312	0.2554	0.1689	0.1084
5K	128	0.4578	0.2647	0.1785	0.1111
10K	134	0.4607	0.2754	0.1818	0.1115
30K	140	0.4601	0.2766	0.1821	0.1128
50K	144	0.4612	0.2765	0.1826	0.1136

Table 6.1: Comparison between corpus size and BLEU scores averaged over top five retrieval. B-n stands for n-gram BLEU score. Vector size represent the dimension of tf-idf vector.

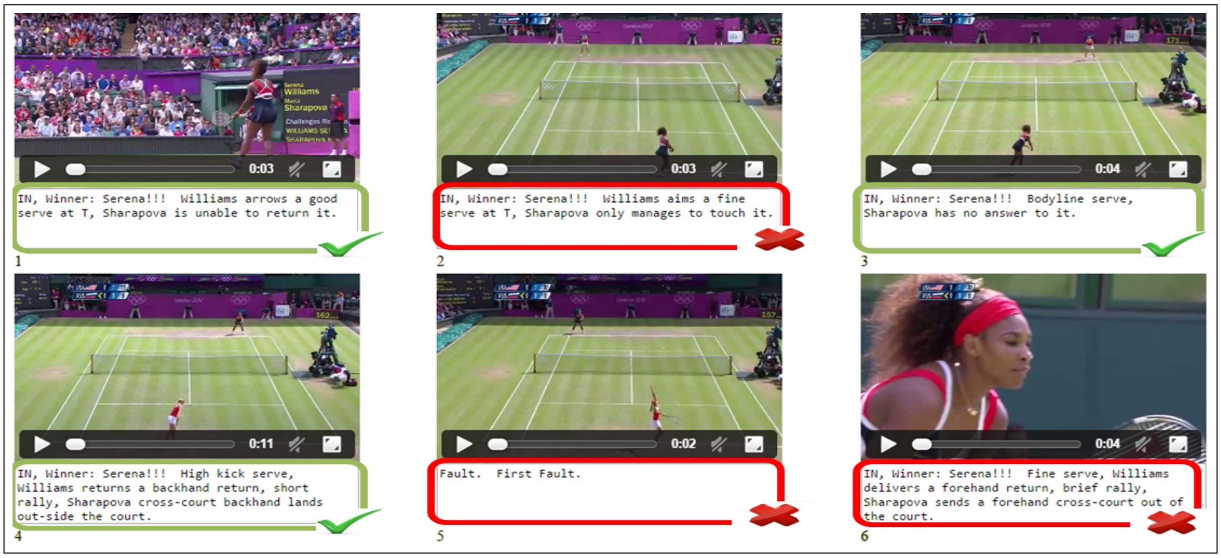
Table 6.3 (left) demonstrates the effect of variations in corpus size on BLEU scores. It can be observed that the scores saturate soon, which validates our initial premise that in domain specific settings, rich descriptions can be produced even with small corpus size. In Table 6.3 (right), we compare our performance with some of the recent methods. Here we observe that caption generation based approaches [22, 34] achieve very low BLEU score. [1] This attributes to their generic nature, and their current inability to produce detailed descriptions. On the other hand, cross-modal retrieval approaches [58, 75] perform much better than [22, 34]. Compared to all the competing methods, our approach consistently provides better performance. The performance improvements increase as we move towards higher n-grams, with an improvement of around 50% over [75] for 4-gram. These results confirm the efficacy of our approach in retrieving descriptions that match the semantics of the data much better than cross-modal retrieval approaches.

6.4 Baseline and Comparisons

The proposed system is benchmarked against state-of-the-art methods from two streams that are popular in predicting textual descriptions for visual data: description/caption generation and cross-modal description retrieval.

(1) Description generation: Since the approaches in this domain are either too generic [22, 39, 34], or designed for images [34], we evaluate by adapting them to our setting. In both [22, 39], a template-based caption is generated by considering a triplet of form (SVO). To compare with this setting, we align the best predicted verb phrase into a template 'player1 $-$ verbPhrase1 , player2 $-$ verbPhrase2'. Since a verb phrase is a combination of an action verb and an object, this template resembles the 'SVO' selection of [22, 39]. To compare with [34], we use the publicly available pretrained model and generate captions for key frames in a video. Since this is a generic approach, the captions generated are nearly similar, and the set of distinct captions is far less than the total

[1] Recall that [22, 34] work for generic videos and images, we approximate them for comparisons.



1 IN, Winner: Serena!!! Williams arrows a good serve at T, Sharapova is unable to return it. ✓

2 IN, Winner: Serena!!! Williams aims a fine serve at T, Sharapova only manages to touch it. ✗

3 IN, Winner: Serena!!! Bodyline serve, Sharapova has no answer to it. ✓

4 IN, Winner: Serena!!! High kick serve, Williams returns a backhand return, short rally, Sharapova cross-court backhand lands out-side the court. ✓

5 Fault. First Fault. ✗

6 IN, Winner: Serena!!! Fine serve, Williams delivers a forehand return, brief rally, Sharapova sends a forehand cross-court out of the court. ✗

Figure 6.3: Success and failure cases: Example videos along with their descriptions. The 'ticked' descriptions match with the ground truth, while the 'crossed' ones do not.

number of key-frames. To associate a caption with a video, we pick the one with the highest frequency. Figure, 6.4 showcases the qualitative comparisons with present state of the art methods.

(2) Cross-modal description retrieval: Cross-modal retrieval approaches [58, 75] perform retrieval by matching samples from the input modality with those in the output modality (both of which are represented as feature vectors). While comparing with [75], we consider the best performing variant of their approach; i.e., the one that uses projected features with Euclidean distance as loss function. Note that our approach is also based on retrieving a description; however, it makes explicit use of low-level visual and textual cues unlike cross-modal retrieval approaches. This difference is also evident from the experimental results, where our approach is shown to retrieve better descriptions than [58, 75].

(2.1) Extension and detailed comparison [75]:

Method [75]	Vector#	B-1	B-2	B-3	B-4
CCA-Cosine	144	0.4299	0.2414	0.1493	0.0796
CCA-L2	144	0.4219	0.2334	0.1417	0.0747
CCA-L1	144	0.4132	0.2279	0.1356	0.0723

Table 6.2: Detailed comparison with cross-modal retrieval method of [75]. Various distance functions are tried and final scores are reported in above table.

Each video V_{Si} is represented using p-dimensional feature vector x_i in space $X = \Re^p$ and each text (commentary) T_{Si} is represented using q-dimensional feature vector y_i in space $Y = \Re^q$. Our objective is to learn a discriminant function $F : X \times Y \longrightarrow \Re$ that can be used to predict optimal output using input output pairs.

$$y^* = \underset{y \in Y}{\operatorname{argmax}} F(x, y; w) \tag{6.2}$$

We make an assumption of $F = w.\Psi(x, y)$; i.e. F is linear function of joint feature representation $\Psi(.)$ of input-output pair.The task of learning w is formulated as following:

$$\min_{w,\xi \geq 0} \frac{1}{2}||w||^2 + \frac{C}{N}\sum_{i=1}^{N} \xi_i \tag{6.3}$$

s.t. $w.\Psi(x_i, y_i) \geq w.\Psi(x_i, y) + \Delta(y_i, y) - \xi_i \forall i, y \epsilon Y \backslash \{y_i\}$ where $||.||^2$ denotes L_2norm, $C > 0$ controls trade off between regularization term and loss term,ξ_i denotes slack variable and $\Delta(y_i, y)$ is loss function.

Both $\Psi(.)$ and $\Delta(.)$ are problem specific. Similar to [75] we define $\Psi(x, y) = x \otimes y \epsilon \Re^r, r = p \times q$. Since phrases are represented by tfidf *(term frequency-inverse document frequency)* vectors one of the obvious choice of loss function would be cosine distance. As cosine distance is not a distance metric, we instead use angular distance, $\Delta(y_i, y) = \frac{2}{\Pi}\cos^{-1}(\cos Similarity)$ as our loss function. We solve the above equation using cutting plane algorithm [74] which requires efficient computation of the most violated constraint during each iteration. The following equation describes most violated constraint corresponding to incorrect prediction $\hat{y}$:

$$\underset{y \in Y \backslash \{y_i\}}{\operatorname{argmax}} \Delta(y_i, y) + w.\Psi(x_i, y) - w.\Psi(x_i, y_i) \tag{6.4}$$

Since, $w.\Psi(x_i, y_i)$ is constant with respect to y, we have

$$\underset{y \in Y \backslash \{y_i\}}{\operatorname{argmax}} \Delta(y_i, y) + w.\Psi(x_i, y) \tag{6.5}$$

Under the assumption that a video and its corresponding text are two heterogeneous representation of similar information we map each representation into maximally correlated vector subspaces using CCA. The results are reported in Table 6.2.

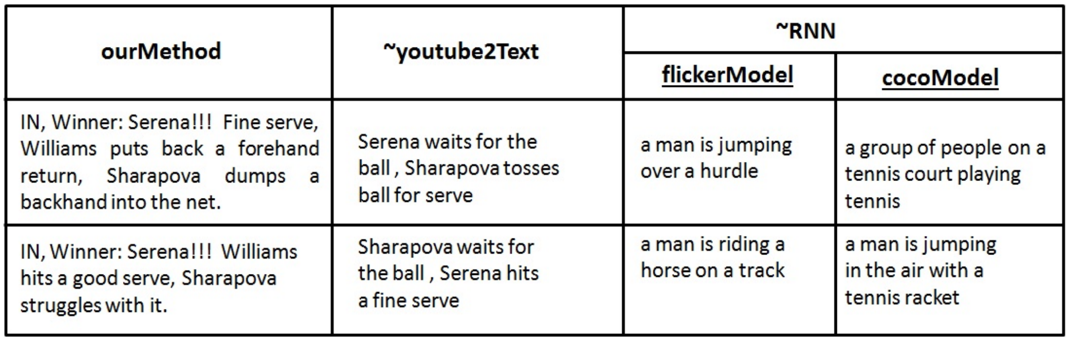
6.4.1 Performance of Individual Modules

To validate the utility of our design choices, we discuss performance of various modules in this section.

(1) Verb Phrase Recognition: Compared to a straightforward use of state-of-the-art technique [80] to recognize verbs, our verb phrase recognition module performs better by around 15%. This suggests the utility of harnessing relevant cues (dividing a frame into two halves) while working in a domain specific environment.

(2) Smoothing Vs. No Smoothing of Verb Phrase Predictions: In practice, using MRF for smoothing phrase predictions improved average BLEU score from 0.204 to 0.235.

(3) LSI-based Matching Vs. Lexical Matching: Employing LSI technique while matching predicted phrases with descriptions achieves an average BLEU score of 0.235, whereas naïve lexical matching achieves 0.198.



ourMethod	~youtube2Text	~RNN	
		flickerModel	cocoModel
IN, Winner: Serena!!! Fine serve, Williams puts back a forehand return, Sharapova dumps a backhand into the net.	Serena waits for the ball , Sharapova tosses ball for serve	a man is jumping over a hurdle	a group of people on a tennis court playing tennis
IN, Winner: Serena!!! Williams hits a good serve, Sharapova struggles with it.	Sharapova waits for the ball , Serena hits a fine serve	a man is riding a horse on a track	a man is jumping in the air with a tennis racket

Figure 6.4: Qualitative comparison of output of present state of the art methods: Youtube2Text [22] and RNN [34]. Both [22, 34] work for generic videos and images, we approximate them for comparisons.

Method	B-1	B-2	B-3	B-4
Guadarrama [22]	0.119	0.021	0.009	0.002
Karpathy [34]	0.135	0.009	0.001	0.001
Rasiwasia [58]	0.409	0.222	0.132	0.070
Verma [75]	0.422	0.233	0.142	0.075
This work	0.461	0.276	0.183	0.114

Table 6.3: Performance comparison with previous methods using the best performing dictionary. 'Corpus#' denotes the number of commentary lines, 'Vocab#' denotes the dimensionality of the textual vocabulary/dictionary and 'B-n' means n-gram BLEU score.

6.5 Human Evaluation

We created an online testing portal and asked twenty people with decent tennis exposure to evaluate our annotations (commentary). Videos were presented in form fifteen sets comprising of six videos each. Every evaluator was randomly presented (atleast) two sets and asked to rate the text on subjective scale of *'Perfect ', 'Good ', 'OK ', 'Poor ', 'Flawed '*. Participants were asked to give scores for both structure and semantics of commentary. We converted this on scale of 1-5 with 'Perfect 'being 5 and 'Flawed 'being 1 and took average over both scores. On scale of 5 we scored 2.8 on semantics and 4.7 on structure of the generated descriptions.

Chapter 7

Conclusions and Future Directions

We have introduced a novel method for predicting commentary-like descriptions for lawn tennis videos. Our approach demonstrates the utility of the simultaneous use of vision, language and machine learning techniques in a domain specific environment to produce semantically rich and human-like descriptions. Though we demonstrate our method on tennis videos, we believe it is fairly generic and has direct extensions to similar situations where activities are in a limited context, the linguistic diversity is confined and the description has to be semantically rich. Similar approaches can be developed for driver assistance systems (for human interface), entertainment (creating narrations for classical dance performances), commentaries for wide variety of sports (e.g. baseball, soccer) and many more. Each of these applications need domain specific tracking, recognition modules and a corpus of examples to generate novel rich descriptions. Our method generates semantically richer descriptions, and not just captions. This is possibly the first time such a task is being attempted, as far as we are aware of. Previous methods [22, 39, 46, 54] were interested in annotating images and videos with words, phrases or captions (simple sentences). Their objective has been primarily to make images and videos searchable, while our descriptions are aimed at direct human consumption. Our method is novel in the sense that we take advantages of two complementary approaches that match and retrieve [54, 75] and recognize and synthesize [22, 39, 40].

7.1 Applications

The proposed method is generic and can be adopted to similar situations where activities are in a limited context and the linguistic diversity is confined, however the output description can be semantically rich. Applications of our solution could range from content based retrieval to real life tennis coaching. In this section we describe some of the explored applications of the pipeline.

7.1.1 Search and Retrieval

Following the solution in Chapter 6 each video is associated with a set of descriptive tags. We can build a text-based search engine over these annotated scenes. We perform a standard stop-word detection before indexing the scenes. Our system accepts text queries given by the user and searches through the index to retrieve the scenes matching the query. The user could thus query for all scenes involving a particular player, or with a certain outcome. For e.g. a user query of 'Sharapova all forehand shots' would result in collection of videos where Sharapova has hit forehand shot. To retrieve the scenes most suitable to user query, we use standard vector space models popular in text retrieval. Each annotated scene is considered as a bag-of-words of its corresponding annotations. The query too is considered as a bag-of-words. Both the scenes and queries are assumed to be points in a space whose dimensions are given by all unique words (sans stop words) in the annotations. The similarity between the query and annotation is given by the cosine similarity measure.

7.1.2 Automatic match Summarization

An extension in form of automatic full-match summary is also one of the probable extension. Match summaries or highlights generated by our proposed method have potential to be descriptive and less machine like. An additional module of automated transitions detection [5, 71] in tennis videos, i.e. shot transition detection to localize game scenes, would be sufficient to achieve such a goal. Every identified game scene passes through our 'proposed solution' in a time-serial manner and the descriptions thus generated summarize the match in totality.

7.1.3 Mine rare-shots

Action trajectories computed for every localized action in Chapter 5 could assist in mining rare and unique shots from the set of inputs. A joint clustering model [8] learnt over multimedia (video) features and text features could be designed to isolate rare and unique shots. Sparse clusters generated using co-clustering framework correspond to rare and unique shots with action phrases linked to them.

7.2 Future directions

Future research extensions of the proposed technique would be application of present work in other video-text pairs. There are many other video collections, figure 7.1 ,that have parallel text information that are yet to be exploited for both annotation and retrieval, for example, (a) Smart theatrics: Narration generation for dance dramas have huge potential of such applications. Any classical dance form narrating a story has a common story line and plot and hence this domain restriction makes our method apt. (b) Commentary generation [25] for other sports events like cricket, baseball etc. could be well be generalized using described procedure. (c) Cooking Instructions [52]: A cooking recipe is one more example of restricted domain. Generally a recipe of a dish tends to pretty similar hence the problem can be framed as two step process as in our case – generate phrases for actions like 'sprinkle salt', 'pick pan' etc. could be generated and later used for full recipe generation. We believe that annotation and retrieval based pipeline is a stepping stone for multimedia understanding. We showcase the

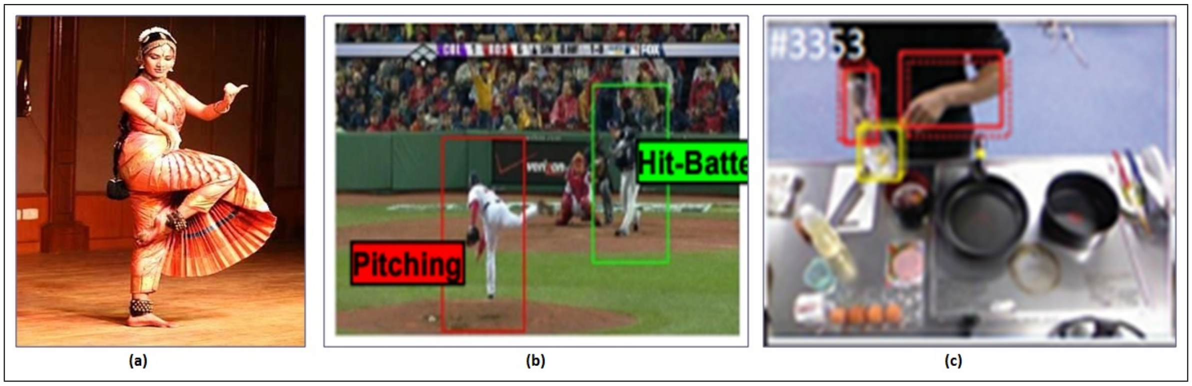
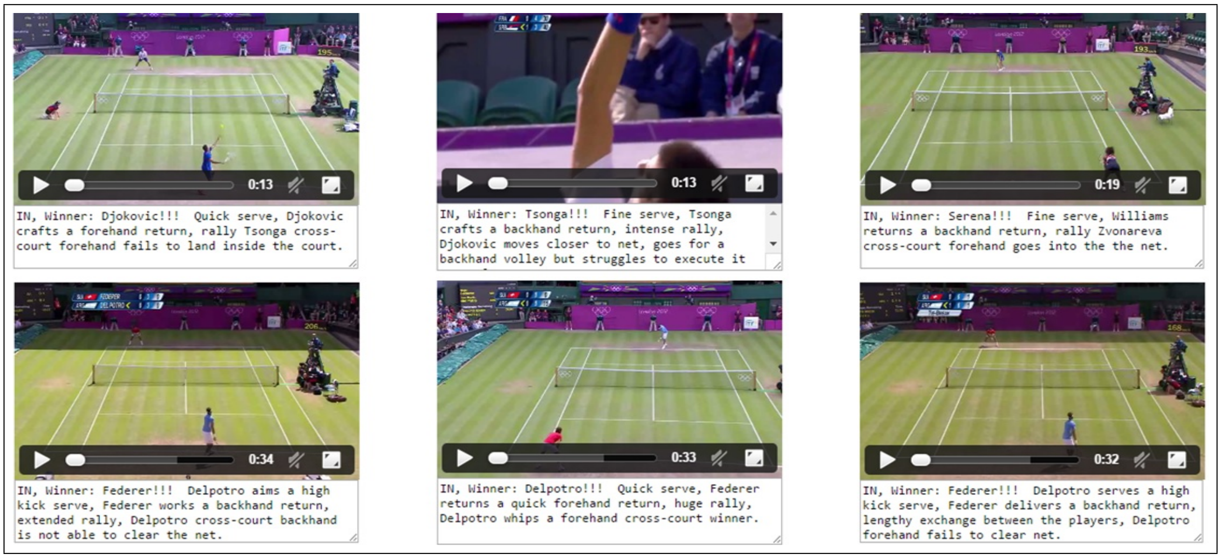


Figure 7.1: Future directions: (a) Smart theatrics: Narration generation for dance dramas. (b) Sports [25]: Commentary generation for various sporting events. (c) Cooking Instructions [52]: Automated generation of cooking instructions.

effectiveness of leveraging parallel text for multimedia understanding and hence take a step towards what Fei-Fei Li said ‘If we want our machines to think, we need to teach them to **see and understand**.’

Appendix A

Qualitative auxiliary



IN, Winner: Djokovic!!! Quick serve, Djokovic crafts a forehand return, rally Tsonga cross-court forehand fails to land inside the court.

IN, Winner: Tsonga!!! Fine serve, Tsonga crafts a backhand return, intense rally, Djokovic moves closer to net, goes for a backhand volley but struggles to execute it

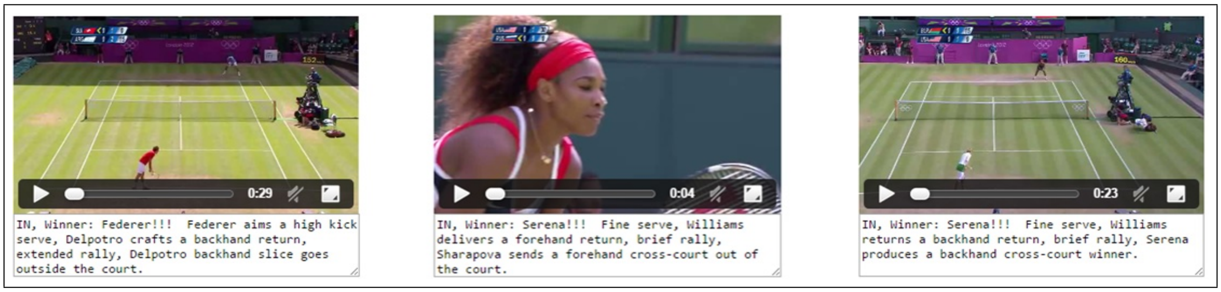
IN, Winner: Serena!!! Fine serve, Williams returns a backhand return, rally Zvonareva cross-court forehand goes into the the net.

IN, Winner: Federer!!! Delpotro aims a high kick serve, Federer works a backhand return, extended rally, Delpotro cross-court backhand is not able to clear the net.

IN, Winner: Delpotro!!! Quick serve, Federer returns a quick forehand return, huge rally, Delpotro whips a forehand cross-court winner.

IN, Winner: Federer!!! Delpotro serves a high kick serve, Federer delivers a backhand return, lengthy exchange between the players, Delpotro forehand fails to clear net.

Figure A.1: Success Cases: Example videos along with their generated descriptions that are similar to ground truth descriptions

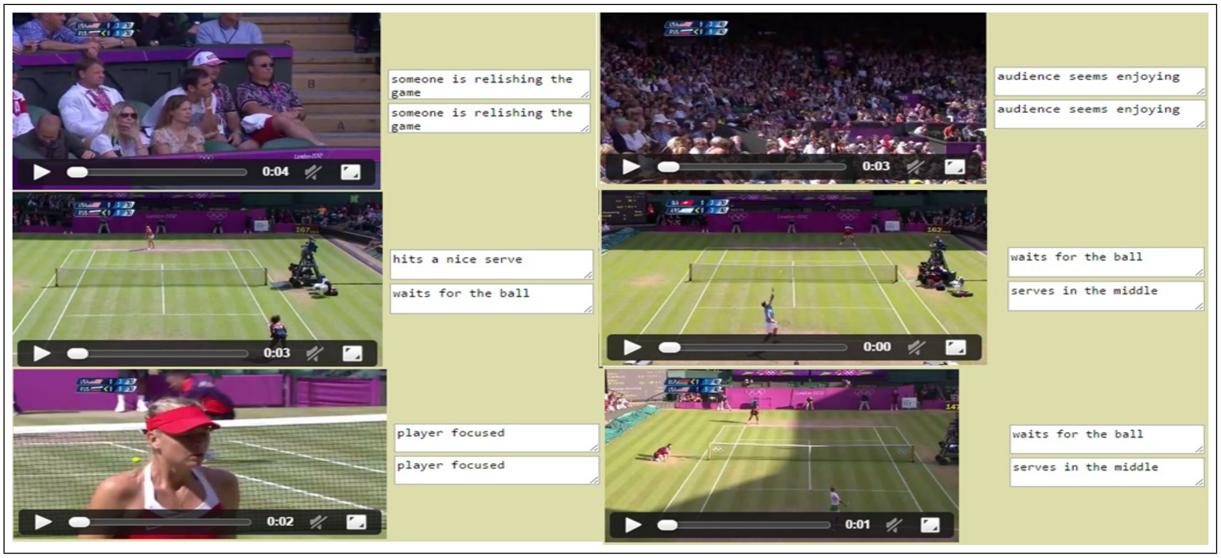


IN, Winner: Federer!!! Federer aims a high kick serve, Delpotro crafts a backhand return, extended rally, Delpotro backhand slice goes outside the court.

IN, Winner: Serena!!! Fine serve, Williams delivers a forehand return, brief rally, Sharapova sends a forehand cross-court out of the court.

IN, Winner: Serena!!! Fine serve, Williams returns a backhand return, brief rally, Serena produces a backhand cross-court winner.

Figure A.2: Failure Cases: Example videos along with their generated descriptions which fail to match with ground truth descriptions. Descriptions in failed category involve wrong action sequences in descriptions.



someone is relishing the game

someone is relishing the game

audience seems enjoying

audience seems enjoying

hits a nice serve

waits for the ball

waits for the ball

serves in the middle

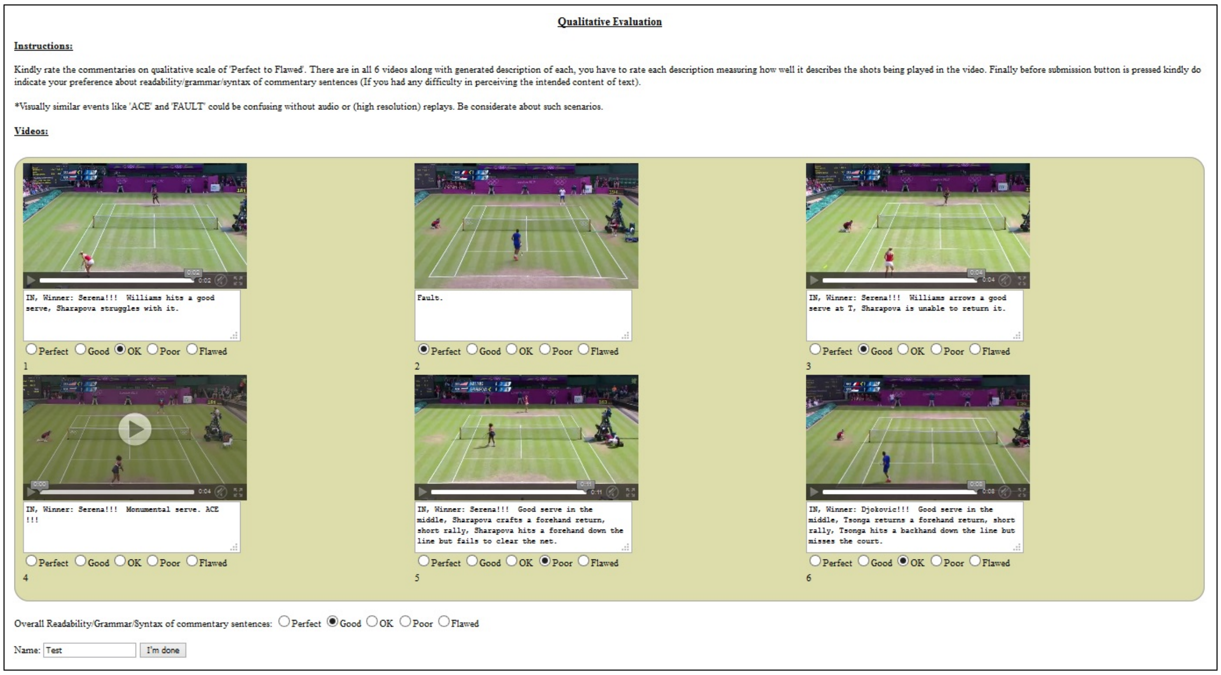
player focused

player focused

waits for the ball

serves in the middle

Figure A.3: Annotated-action dataset: Video shots aligned with verb phrases. Every video shot is labelled with two phrases, the upper text correspond to actions of upper player and lower text to actions of lower player.



Qualitative Evaluation

Instructions:

Kindly rate the commentaries on qualitative scale of 'Perfect to Flawed'. There are in all 6 videos along with generated description of each, you have to rate each description measuring how well it describes the shots being played in the video. Finally before submission button is pressed kindly do indicate your preference about readability/grammar/syntax of commentary sentences (If you had any difficulty in perceiving the intended content of text).

*Visually similar events like 'ACE' and 'FAULT' could be confusing without audio or (high resolution) replays. Be considerate about such scenarios.

Videos:

IN, Winner: Serena!!! Williams hits a good serve, Sharapova struggles with it.

○ Perfect ○ Good ● OK ○ Poor ○ Flawed

1

Fault.

● Perfect ○ Good ○ OK ○ Poor ○ Flawed

2

IN, Winner: Serena!!! Williams arrows a good serve at T, Sharapova is unable to return it.

○ Perfect ● Good ○ OK ○ Poor ○ Flawed

3

IN, Winner: Serena!!! Monumental serve. ACE !!!

○ Perfect ○ Good ○ OK ○ Poor ○ Flawed

4

IN, Winner: Serena!!! Good serve in the middle, Sharapova crafts a forehand return, short rally, Sharapova hits a forehand down the line but fails to clear the net.

○ Perfect ○ Good ○ OK ● Poor ○ Flawed

5

IN, Winner: Djokovic!!! Good serve in the middle, Tsonga returns a forehand return, short rally, Tsonga hits a backhand down the line but misses the court.

○ Perfect ○ Good ● OK ○ Poor ○ Flawed

6

Overall Readability/Grammar/Syntax of commentary sentences: ○ Perfect ● Good ○ OK ○ Poor ○ Flawed

Name: Test

Figure A.4: Human Evaluations: Testing portal used for human evaluation. Fifteen sets comprising of six videos each was presented randomly to twenty people with exposure to game of tennis.

Related Publications

Mohak Sukhwani, C.V. Jawahar Tennis Vid2Text : Fine-Grained Descriptions for Domain Specific Videos, Proceedings of the 26th British Machine Vision Conference (BMVC), 07-10 Sep 2015, Swansea, UK.

Bibliography

[1] Tennis Earth - webpage. `http://www.tennisearth.com/`.

[2] J.-B. Alayrac, P. Bojanowski, N. Agrawal, I. Laptev, J. Sivic, and S. Lacoste-Julien. Learning from narrated instruction videos. In *Arxiv 1506.09215*, 2015.

[3] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012.

[4] A. Barbu, A. Bridge, Z. Burchill, D. Coroian, S. J. Dickinson, S. Fidler, A. Michaux, S. Mussman, S. Narayanaswamy, D. Salvi, L. Schmidt, J. Shangguan, J. M. Siskind, J. W. Waggoner, S. Wang, J. Wei, Y. Yin, and Z. Zhang. Video in sentences out. In *CoRR*, 2012.

[5] J. Bescós, G. Cisneros, J. M. Martínez, J. M. Menéndez, and J. Cabrera. A unified model for techniques on video-shot transition detection. In *IEEE Transactions on Multimedia*, 2005.

[6] S. A. Chatzichristofis and Y. S. Boutalis. Cedd: Color and edge directivity descriptor: A compact descriptor for image indexing and retrieval. In *ICVS*, 2008.

[7] R. Chaudhry, A. Ravich, G. Hager, and R. Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *CVPR*, 2009.

[8] X. Chen, A. Ritter, A. Gupta, and T. Mitchell. Sense discovery via co-clustering on images and text. In *CVPR*, 2015.

[9] I. L. Christian Schuldt and B. Caputo. Recognizing human actions: A local svm approach. In *CVPR*, 2004.

[10] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[11] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006.

[12] M.-C. De Marneffe and C. D. Manning. The stanford typed dependencies representation. In *COLING Workshop*, 2008.

[13] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. In *JASIST*, 1990.

[14] M. Denkowski and A. Lavie. Choosing the right evaluation for machine translation: an examination of annotator and automatic metric performance on human judgment tasks. In *AMTA*, 2010.

[15] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005.

[16] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *arXiv preprint arXiv:1411.4389*, 2014.

[17] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *ICCV*, 2009.

[18] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, 2003.

[19] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In *ECCV*. 2010.

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[21] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, 2005.

[22] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*, 2013.

[23] A. Gupta and L. S. Davis. Objects in action: An approach for combining action understanding and object perception. In *CVPR*, 2007.

[24] A. Gupta and P. Mannem. From image annotation to image description. In *NIPS*, 2012.

[25] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In *CVPR*, 2009.

[26] A. Gupta, Y. Verma, and C. Jawahar. Choosing linguistics over vision to describe images. In *AAAI*, 2012.

[27] L. Han, A. Kashyap, T. Finin, J. Mayfield, and J. Weese. Umbc ebiquity-core: Semantic textual similarity systems. In **SEM*, 2013.

[28] A. Hervieu, P. Bouthemy, and J.-P. L. Cadre. Trajectory-based handball video understanding. In *ICIV*, 2009.

[29] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. In *JAIR*, 2013.

[30] Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T. S. Huang. Action detection in complex scenes with spatial and temporal ambiguities. In *ICCV*, 2009.

[31] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, 2007.

[32] Z. Jiang, Z. Lin, and L. S. Davis. Learning a discriminative dictionary for sparse coding via label consistent K-SVD. In *CVPR*, 2011.

[33] S. S. Kanade and P. M. Patil. Lawn tennis video summarization based on audiovisual and text feature analysis. In *IJCA*, 2012.

[34] A. Karpathy and L. Fei-Fei. Deep visual-vemantic alignments for generating image descriptions. In *CVPR*, 2015.

[35] J. Kim and K. Grauman. Observe locally, infer globally: A space-time mrf for detecting abnormal activities with incremental updates. In *CVPR*, 2009.

[36] R. Kindermann, J. L. Snell, et al. Markov random fields and their applications. In *American Mathematical Society*, 1980.

[37] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. In *arXiv preprint arXiv:1411.2539*, 2014.

[38] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *SIGCHI*, 2008.

[39] N. Krishnamoorthy, G. Malkarnenkar, R. J. Mooney, K. Saenko, and S. Guadarrama. Generating natural-language video descriptions using text-mined knowledge. In *AAAI*, 2013.

[40] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Baby talk: Understanding and generating image descriptions. In *CVPR*, 2011.

[41] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi. Collective generation of natural image descriptions. In *ACL*, 2012.

[42] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[43] I. Laptev and P. Pérez. Retrieving actions in movies. In *ICCV*, 2007.

[44] S. Li, G. Kulkarni, T. L. Berg, A. C. Berg, and Y. Choi. Composing simple image descriptions using web-scale n-grams. In *ACL*, 2011.

[45] W.-L. Lu and J. J. Little. Simultaneous tracking and action recognition using the pca-hog descriptor. In *CRV*, 2006.

[46] A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In *ECCV*, 2008.

[47] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Explain images with multimodal recurrent neural networks. In *arXiv preprint arXiv:1410.1090*, 2014.

[48] B. Markoski, Z. Ivanković, L. Ratgeber, P. Pecev, and D. Glušac. Application of adaboost algorithm in basketball player detection. In *Acta Polytechnica Hungarica*, 2015.

[49] M. Mitchell, X. Han, J. Dodge, A. Mensch, A. Goyal, A. Berg, K. Yamaguchi, T. Berg, K. Stratos, and H. Daumé III. Midge: Generating image descriptions from computer vision detections. In *ECACL*, 2012.

[50] H. Miyamori and S. Iisaku. Video annotation for content-based retrieval using human behavior analysis and domain knowledge. In *FG*, 2000.

[51] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *UAI*, 2013.

[52] B. Ni, V. R. Paramathayalan, and P. Moulin. Multiple granularity analysis for fine-grained action detection. In *CVPR*, 2014.

[53] K. Okuma, A. Taleghani, N. De Freitas, J. J. Little, and D. G. Lowe. A boosted pinproceedings filter: Multitarget detection and tracking. In *ECCV*, 2004.

[54] V. Ordonez, G. Kulkarni, and T. L. Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, 2011.

[55] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: A method for automatic evaluation of machine translation. In *ACL*, 2002.

[56] G. Pingali, A. Opalach, and Y. Jean. Ball tracking and virtual replays for innovative tennis broadcasts. In *ICPR*, 2000.

[57] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, 2000.

[58] N. Rasiwasia, J. C. Pereira, E. Coviello, G. Doyle, G. R. G. Lanckriet, R. Levy, and N. Vasconcelos. A new approach to cross-modal multimedia retrieval. In *ACM MM*, 2010.

[59] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *CVPR*, 2012.

[60] B. Rosario. Latent semantic indexing: An overview. In *Techn. rep. INFOSYS*, 2000.

[61] C. Rother, V. Kolmogorov, and A. Blake. ”grabcut”: Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004.

[62] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. In *IJCV*, 2008.

[63] M. S. Ryoo and J. Aggarwal. Hierarchical recognition of human activities interacting with objects. In *CVPR*, 2007.

[64] K. P. Sankar, C. V. Jawahar, and A. Zisserman. Subtitle-free movie to script alignment. In *BMVC*, 2009.

[65] K. Schindler and L. Van Gool. Action snippets: How many frames does human action recognition require? In *CVPR*, 2008.

[66] R. A. Sharma, K. P. Sankar, and C. V. Jawahar. Fine-grain annotation of cricket videos. In *ACPR*, 2015.

[67] W. Shen and L. Wu. A method of billiard objects detection based on snooker game video. In *ICFCC*, 2010.

[68] H. B. Shitrit, J. Berclaz, F. Fleuret, , and P. Fua. Tracking multiple people under global appearance constraints. In *ICCV*, 2011.

[69] R. Snow, B. O’Connor, D. Jurafsky, and A. Y. Ng. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. In *EMNLP*, 2008.

[70] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng. Grounded compositional semantics for finding and describing images with sentences. In *TACL*, 2014.

[71] P. S. Sowjanya and R. Mishrab. Comparison of video shot boundary detection. In *Digital Image Processing*, 2011.

[72] G. Sudhir, J. Lee, and A. Jain. Automatic classification of tennis video for high-level content-based retrieval. In *International Workshop on content-based access of image and video database*, 1998.

[73] K. Tanaka, H. Nakashima, I. Noda, K. Hasida, I. Frank, and H. Matsubara. Mike: an automatic commentary system for soccer. In *ICMS*, 1998.

[74] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.

[75] Y. Verma and C. V. Jawahar. Im2text and Text2Im: Associating images and texts for cross-modal retrieval. In *BMVC*, 2014.

[76] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *arXiv preprint arXiv:1411.4555*, 2014.

[77] D. Voelz, E. Andre, G. Herzog, and T. Rist. Rocco: A robocup soccer commentator system. In *Proceedings of RoboCup*, 1999.

[78] G. Waltner, T. Mauthner, and H. Bischof. Improved sport activity recognition using spatio-temporal context. In *DVS/GSSS*, 2014.

[79] G. Waltner, T. Mauthner, and H. Bischof. Indoor activity detection and recognition for automated sport games analysis. In *AAPR/OAGM*, 2014.

[80] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action Recognition by Dense Trajectories. In *CVPR*, 2011.

[81] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[82] G. Willems, J. H. Becker, T. Tuytelaars, and L. J. Van Gool. Exemplar-based action recognition in video. In *BMVC*, 2009.

[83] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. Rehg. A scalable approach to activity recognition based on object use. In *ICCV*, 2007.

[84] Y. Yang, C. L. Teo, H. Daumé III, and Y. Aloimonos. Corpus-guided sentence generation of natural images. In *EMNLP*, 2011.

[85] M. Yatskar, L. Vanderwende, and L. Zettlemoyer. See no evil, say no evil: Description generation from densely labeled images. In ** SEM 2014*, 2014.

[86] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In *Exploring artificial intelligence in the new millennium*, 2003.

[87] X. Yu, C. L. Teo, Y. Yang, C. Fermüller, and Y. Aloimonos. Action attribute detection from sports videos with contextual constraints. In *BMVC*, 2013.

[88] J. Yuan, Z. Liu, and Y. Wu. Discriminative subvolume search for efficient action detection. In *CVPR*, 2009.

[89] G. Zhu, C. Xu, Q. Huang, Y. Rui, S. Jiang, W. Gao, and H. Yao. Event tactic analysis based on broadcast sports video. In *IEEE Transactions on Multimedia*, 2009.