Analyzing Racket Sports From Broadcast Videos

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

Masters of Science
in
Computer Science and Engineering by Research

by

Anurag Ghosh
201302179
anurag.ghosh@research.iiit.ac.in

International Institute of Information Technology, Hyderabad
(Deemed to be University)
Hyderabad - 500 032, India
June 2019

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Analyzing Racket Sports From Broadcast Videos" by Anurag Ghosh, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Prof. C.V. Jawahar

To Neelabh

Acknowledgments

I would like to express my deepest gratitude to my adviser, Prof. C.V. Jawahar for his constant support and encouragement to explore and chart my own research path. The freedom he has afforded has been liberating and the guidance I have got from him has been the most cherished part of my research journey. He has always been the primary source of inspiration to learn work ethics in research. I hope I have learnt a minuscule fraction from him and imbibed some his research style and methodology, apart from his rigorous yet interesting classes. I would also like to specially thank Dr. Kartheek Alahari for accepting to work with me and listening to my ideas very patiently, and I hope I have learnt some aspects of a researcher in his guidance. I'm really fond of my stints as a TA under Prof. Naresh Manwani and his encouragement and support has been unparalleled, and I surely wish to become a mentor like him some day. I'm also grateful to Prof. Madhava Krishna, Prof. Harjinder Singh, Prof. Kavita Vemuri and Prof. Aniket Alam for their very nice classes and insights during my conversations with them and I have surely learnt a lot from them.

It has been a really amazing undergraduate and graduate experience at IIIT-H and the last five years have really molded me and transformed me. One of my most memorable moments was my first birthday at IIIT. I felt at home while being away for the first time and I think that has been a defining picture of IIIT which I'll always cherish along with the person etched in my mind. I would thank my friends, Romil for being a great roommate, Parth for being the fellow tinkerer, Shrenik for his constant support, Abhijeet for being the CVIT *bro*, Yash for his candid *convos*, Saksham for being himself, Arpit for his research philosophy, Abhirath for being the constant source of entertainment, Tushant for his *gyaan*, Shreedhar for being a fellow traveler, Anuj for his UP ka dhang, Vignesh for just being the pleasure he is, Abrar for being my favorite person, Nair, Paul and Shukla for our amazing *dulla* talks. I'd also like to thank Shantanu, Motwani, Monu, Shubham, Dhruva, Shraddhan, Sikander, Meha, Lasya, Amitha, Vaishnavi, and many more for always being there for me when I needed them. I would also thank Rohit bhaiya for giving me candid advice when I needed it. And Saumya for entering very late into my journey here, yet leaving an indelible mark on it. Lastly, I'm grateful to Ankita, who has constantly been there for years, unfailingly, despite being thousands of miles away, despite her own struggles. She has been a beacon of light, as she made sure I finish my journey here and her presence is something I feel I hardly deserve.

I would really like to thank my lab mates for their constant support and discussion, Mohak sir for being a great mentor, Suriya sir for being a bouncing board for ideas, Harish for being such a sport,

Bharat for pushing me forward, Swetha for her encouragement, Isha for sitting beside me and not getting irritated, Tejaswi for listening to me, Abhishek for the great *saké*, Sahil for our storytelling discussions, Sourabh sir for being a towering yet humble presence, Aditya sir for listening to my ridiculous ideas and patiently debugging them, and Mohit for being there at a low moment, Ameya, Abbhinav, Gaurav, Praveen sir, Pritish sir, Jobin sir and many more CVITians. I would also like to specially thank Rakesh for working with me on related problems and helping me out when he could. Special thanks to Siva for always helping me out with lab things I still can't wrap my head around.

My journey wouldn't have been as successful as it is without my family. My parents have been very supportive towards all my endeavors and have let me make my own mistakes. I'm specially thankful towards my father for being very supportive even when I took decisions which seemed counter-intuitive. Finally, I'm thankful to my brother, Neelabh, for making me a more conscious and better individual.

Abstract

Sports video data is recorded for nearly every major tournament but remains archived and inaccessible to large scale data mining and analytics. However, Sports videos have a inherent temporal structure, due to the nature of sports themselves. For instance, tennis, comprises of points, games and sets played between the two players/teams. Recent attempts in sports analytics are not fully automatic for finer details or have a human in the loop for high level understanding of the game and, therefore, have limited practical applications to large scale data analysis. Many of these applications depend on specialized camera setups and wearable devices which are costly and unwieldy for players and coaches, specially in a resource constrained environments like India. Utilizing very simple and non-intrusive sensor(s) (like a single camera) along with computer vision models is necessary to build indexing and analytics systems. Such systems can be used to sort through huge swathes of data, help coaches look at interesting video segments quickly, mine player data and even generate automatic reports and insights for a coach to monitor.

Firstly, we demonstrate a score based indexing approach for broadcast video data. Given a broadcast sport video, we index all the video segments with their scores to create a navigable and searchable match. Even though our method is extensible to any sport with scores, we evaluate our approach on broadcast tennis videos. Our approach temporally segments the rallies in the video and then recognizes the scores from each of the segments, before refining the scores using the knowledge of the tennis scoring system. We finally build an interface to effortlessly retrieve and view the relevant video segments by also automatically tagging the segmented rallies with human accessible tags such as 'fault' and 'deuce'. The efficiency of our approach is demonstrated on broadcast tennis videos from two major tennis tournaments.

Secondly, we propose an end-to-end framework for automatic attributes tagging and analysis of broadcast sport videos. We use commonly available broadcast videos of badminton matches and, unlike previous approaches, we do not rely on special camera setups or additional sensors. We propose a method to analyze a large corpus of broadcast videos by segmenting the points played, tracking and recognizing the players in each point and annotating their respective strokes. We evaluate the performance on 10 Olympic badminton matches with 20 players and achieved 95.44% point segmentation accuracy, 97.38% player detection score (mAP@0.5), 97.98% player identification accuracy, and stroke segmentation edit scores of 80.48%. We further show that the automatically annotated videos alone could enable

the gameplay analysis and inference by computing understandable metrics such as player's reaction time, speed, and footwork around the court, etc.

Lastly, we adapt our proposed framework for tennis games to mine spatiotemporal and event data from large set of broadcast videos. Our broadcast videos include all Grand Slam matches played between Roger Federer, Rafael Nadal and Novac Djokovic. Using this data, we demonstrate that we can infer the playing styles and strategies of tennis players. Specifically, we study the evolution of famous rivalries of Federer, Nadal, and Djokovic across time. We compare and validate our inferences with expert opinions of their playing styles.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The marriage of technology and sports has been observed with both interest and skepticism in the recent years. With the advent of Hawkeye and similar proprietary systems made for umpires, players and coaches, the commercial impact on the sports themselves has been immense. New technological innovations have permeated into the rules of the sports themselves, such as "HawkEye challenges" in tennis, "3rd Umpire" in cricket, or "Goal-Line Technology" in football, and these innovations have permanently changed how the sport is both played and enjoyed.

However, the use of technology in sports has been restricted to certain specific applications and have been inaccessible in nature. For instance, goal line technology is so expensive that only top European Leagues have adopted it. [1] Another problem is that the applications have been restricted, with deployed systems heavily geared towards referee-decision making. However, Sports analytics is one space that has seen some progress but due to the proprietary nature of the systems and data, high cost, manual or semi-automated analyses requiring expertise, penetration has been low and out of the reach from most amateur players. Similarly, even with the advent of the interactive web, it's very surprising to note that our sports media streaming websites lack any "smart" features like semantic indexing, and they stream videos exactly as our televisions did.

Injecting intelligence to real world applications through the use of machine learning and AI is thus of growing importance. To democratize sports intelligence, utilizing very simple and non-intrusive sensor(s) (like a single camera) along with computer vision models is necessary to build sports indexing and analytics systems. Such systems can be used to sort through huge swathes of data, help coaches look at interesting video segments quickly, mine player data and even generate automatic reports and insights for a coach to monitor. In this thesis, we focus on leveraging easily available broadcast video data to generate annotations that can be used to mine such insights automatically. Such systems also make it possible to analyze players and matches from an era when more advanced setups did not exist, leveraging thousands of hours of archived video data.

[1] Goal-line technology 'unaffordable' for Scottish Premiership, BBC (https://www.bbc.co.uk/sport/football/42504610)

1.1 Problem Scope

We define the scope of the problems tackled in this thesis. It is important to note that there are many problems tackled in the space of sports analytics, from generating highlights, to forecasting players outcomes, to building hardware and wearable devices for use in sports, however, we realized that there has been little work in constructing systems which automatically provide rich semantic annotations to broadcast videos and thus enrich them for later analysis by players, coaches and enable richer viewing experiences for viewers of these broadcasts.

Automatic semantic indexing and retrieval in videos is a huge problem in Computer Vision, however, we narrow our focus to semantic labeling of domain-specific videos like sports videos. Sports videos have a inherent temporal structure, due to the nature of sports themselves. For instance, cricket is structured as a sequence of overs and each over is in itself a sequence of balls bowled while the opponent bats. Or tennis, which comprises of points, games and sets played between the two players/teams. Moreover, the sports in themselves are embedded with terminologies corresponding to semantic information in itself, like 'off drive' or 'short pitch delivery' etc in cricket and 'backhand shot' or 'duece' in tennis. These features of sports videos make it easy for us to imagine semantic labeling and indexing systems, either through scores and other forms of events, which can be very useful for various applications.

Moreover, such semantic information is also useful if the semantics are associated with the gameplay itself. For instance, it would be very informative to know if a batsmen has a particular inclination for 'off drives' over and above an average player. Player analytics is boosted if we are able to extract finer level semantics such as action and movement information. Such data is also amenable to automated analysis and strategy planning techniques.

1.2 An Overview of Racket Sports Rules

In this section, we explain some of the basic mechanism and terminology used in racket sports (with specific terms mentioned for either tennis or badminton, as necessary), which would be used throughout the thesis.

- The game is played between two or four players on a rectangular court with a net in between. The surface type, dimensions and positioning of the net depends on the sport, see Fig 1.1 for the schematic in tennis.

- A match is composed of sets and points. A set comprises of multiple points. In tennis, a points are also bunched into games, which are further bunched into a set. When a player wins majority of the sets, they win the match.

- A badminton match is usually comprised of maximum three sets of with the first player who scores either eleven or twenty one points with a two point difference wins that set. A tie-breaker is played out in case the set is tied.

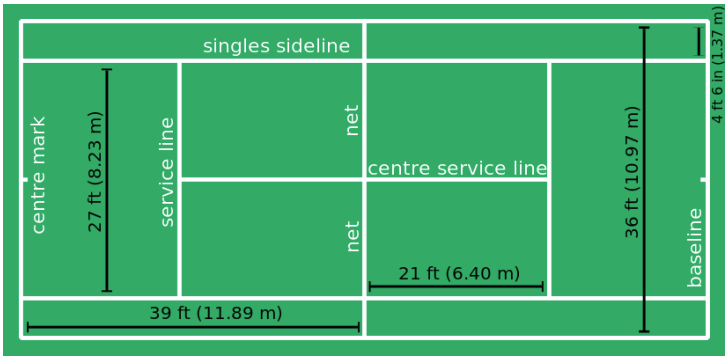

Figure 1.1: Schematic of a Tennis court with associated dimensions. (Source: Wikimedia)

- A tennis match comprises of maximum three or five sets, with a player who wins at least 6 games with a two game advantage wins the set. A player wins the game, if they win four points with a two point advantage. In the absence of two point advantage, the player must win two points in succession to win the game. A tie-breaker is played out in case the set is tied.

- A point is the most granular level of score keeping. A point consists of a sequence of back and forth shots (called strokes in badminton) between players, also known as rally.

- In a rally, a serve is the opening shot (stroke in badminton). The rally is continued through returns until one of the player commits an error. The serving side and opponent are bound to certain parts of the court depending on the sport (See Fig. 1.2 for the case in Badminton).

- In tennis, the server continues serving the ball to the receiver until the set has ended. After the set has ended, the receiver will become the server and serve the ball until the next set has ended. In badminton, serve is won along with the point, i.e. if the opponent was serving in the last rally, the serve is passed; if the server won, they keep on serving.

1.3 Challenges

Sports data and particularly broadcast sports data present a lot of challenges, namely,

- **Temporal Cluttering in Broadcast Videos** Broadcast videos are curated in such a way to make the game exciting to watch for the viewer to satiate commercial interests. For instance, a cool tennis shot is replayed from many angles, focusing on the players and their expressions. Also, consider advertisements played during the rest periods of a match. All these temporal sections of the video are not very important for analysis as it's very hard to mine player information from and thus need to be automatically discarded.

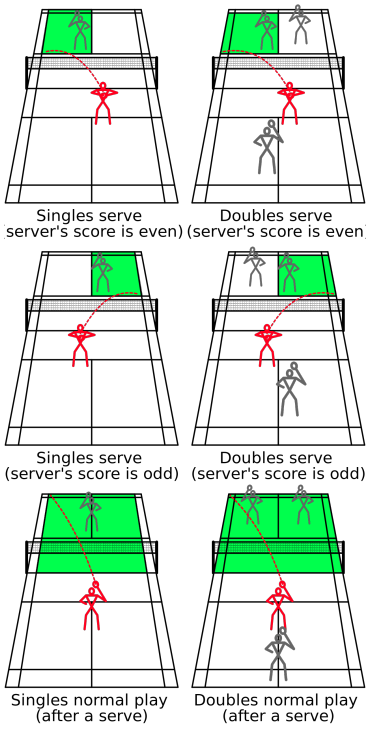

Figure 1.2: The legal bounds of a badminton court during various stages of a rally for singles and doubles games. Red player indicates active player (making a shot), while players in gray are the other players (Source: Wikimedia)

- **Speed and Motion Blur** Both Badminton and Tennis are very fast paced sports with intermittent and complex actions. Also, the tennis balls and shuttlecock are exchanged at a very rapid pace, the shuttlecock touching highest speeds of 400km/h and tennis ball of 250km/h respectively [2]. The players thus move very very quickly to play catch up against each other. All these factors introduce a lot of motion blur in broadcast video, even when the video is recorded at a higher FPS.

- **Shape Variation** For the players to play a certain stroke, like forehand or smash, the associated motion can be very difficult and awkward looking due to the very fast paced nature of the game. Players push themselves to the last extent to make sure they hit the shuttlecock/ball and that results in huge shape variations, even in actions that belong to the same class.

- **Occlusion** The high speed nature and the small court size of badminton means both the players sometimes play very near to the net, introducing a lot of occlusion in the process, covering the far player's actions. Occlusion is less common in Tennis, however, the camera viewpoint due to the larger court size introduces other issues.

[2] Shuttlecock and balls: The fastest moving objects in sport, Olympic Canada, https://olympic.ca/2014/09/11/shuttlecock-and-balls-the-fastest-moving-objects-in-sport/

- **Camera Viewpoint and Movement** The camera viewpoint of the game play segments in both tennis and badminton are from behind the baseline of the court. This viewpoint results in two issues - the far off player is much smaller than the near player and the near player's back faces the camera. Thus, recognizing the actions of the far off player is difficult due to the lower resolution while the near player's actions are difficult to decipher due of the occlusion. Moreover, the camera viewpoint change across tournaments, panning and other movements need to be accounted for when registering the court lines.

- **Temporal Ambiguity of Actions** Badminton and Tennis are very fast paced sports. The back and forth strokes are a blink and miss for most viewers, however, more importantly, it's very difficult to define the temporal extent of each stroke because of the pace of the sport itself. These leads to inherent ambiguity while modeling and evaluating the action segmentation tasks. This is consistent with recent research in action segmentation [79] that suggests there's an inherent ambiguity in precisely localizing activities and that evaluation metrics must account for this.

1.4 Contributions

The major contributions of this thesis are,

1. We propose frameworks aimed at macro-understanding and micro-understanding of racket sports. The first is a score based indexing system, to navigate and retrieve segments from large volumes of video data with considerable ease. The second is an end-to-end framework to automatically annotate videos with player information. Unlike previous approaches, our methods do not rely on special camera setup or additional sensors and work with broadcast video data.

2. We introduce two datasets for evaluating our proposed frameworks. The first is a dataset of broadcast tennis videos sourced from two tournaments, London Olympics and French Open, annotated with the point segments and the scores for each points. The second dataset is a large collection of badminton broadcast videos from London Olympics with match level point segments and scores/outcomes as well as frame level players' tracks and their strokes.

3. We propose an effective score recognition algorithm using domain knowledge which can be adapted for different games. Here, we do our experiments on tennis videos by leveraging the specifics of the tennis scoring system. We demonstrate one such application of the indexing system, through human accessible event tagging.

4. For understanding finer nuances of the players from sports videos, we predict game point segments and the outcome of each segment, players' movement tracks as well as annotate each of their strokes. We demonstrate the utility of recent advancements in object detection, action recognition and temporal segmentation, for our tasks.

5. We identify various understandable metrics, computed using our framework, for match and player analysis as well as qualitative understanding of badminton games. Further, we utilize our framework to study three tennis players (on a large set of tennis match videos) and obtain inferences which are comparable to a large crowd-sourced dataset.

1.5 Thesis Layout

The organization of the thesis is as follows. In the following chapter, we discuss computer vision in the context of sports and discuss some of the machine learning methods we utilize in the further chapters. In chapter 3, we propose a novel framework for semantically indexing broadcast sports videos by exploiting domain specific information about the sports. In chapter 4, a novel data mining and extraction framework is proposed for mining player information such as feet location and strokes solely from broadcast videos without any additional cameras or sensors.

Chapter 2

Background

In this chapter, we take a closer look at some of the computer vision and machine learning techniques and strategies we have utilized throughout this thesis. Section 2.1 describes the proliferation of technology into different sports and focuses more on computer vision and associated technological advances. Section 2.2 describes deep learning architectures and strategies for computer vision tasks for scene text recognition, object detection and action segmentation, which have been used to detect scorecards, players and strokes respectively.

2.1 Computer Vision in Sports

Computer Vision community has been interested in problems pertaining to sports for a long while. There have been numerous applications in various sports, like cricket [76], soccer [14, 75], volleyball [36], basketball [13, 66], tennis [60, 70, 90, 91, 104], badminton [16] and hockey [94] among other sports. Moreover, sports has emerged as a popular proxy task for evaluating computer vision algorithms due to complex nature of most sports. In this thesis, we restrict our focus on tennis and badminton as the sports of interest.

Some of the major applications of computer vision in the context of sports are usually in the space of video summarization, highlight generation [25], automatic broadcasting tools from multiple camera feeds [13, 14], as coaching aids [60, 70], player's fitness, weaknesses and strengths assessment [4], player data mining and analysis, among other applications. Problems involving sports forecasting is another upcoming area of interest, with works like Felsen et al. [24] who proposed a method for forecasting future events in team sports videos directly from visual inputs.

Despite these advancements, recent attempts in sports analytics are not fully automatic for finer details [16, 104] or have a human in the loop for high level understanding of the game [1, 16] and, therefore, have limited practical applications to large scale data analysis. Many of these applications depend on specialized camera setups and wearable devices which are costly and unwieldy for players and coaches, specially in a resource constrained environments like India. Though previous work has

elaborated on generic representations for broadcast sports video analysis [21], such attempts have not focused on automating player analyses.

The diversity of actions and situations along with the fast paced nature of sports make them very good benchmarks for a variety of computer vision tasks. Video classification [40], activity and event recognition [36, 66, 94], pose estimation [39], multi object tracking [50, 53, 54, 103] are a few of the numerous tasks which have been benchmarked on sports images and videos.

2.1.1 Football

Event detection and summarization has been a popular task for video analysis of football. Jiang et al. [38] utilized a CNN to extract global features from football videos and an RNN temporally traversed a video to detect actions in football. Chen et al. [15] propose a method to detect events in amateur videos of American football games (e.g. offense, defense, kickoff, punt, etc). It should be noted that American Football is not the same as Football (which is known as Soccer in that country). To mitigate the issue of small and nonstandard datasets used for football analytics, Giancola et al. [26] proposed a large dataset for action spotting in long videos. The dataset consists of 500 complete football games with 6637 temporal annotations sourced from online match reports at a one minute resolution for three main classes of events (Goal, Yellow/Red Card, and Substitution).

There has been few other tasks that are starting to be explored in the space of Football, due to advent of deep learning methods and improvement in availability of large scale datasets. Homayounfar et al. [34] and Sharma et al. [75] proposed methods for automatic localization and registration of the football field from broadcast video sequences. Le el al. [45] present a data driven "ghosting" method trained on soccer tracking data which can provide insights about defensive plays between different teams. Chen et al. [14] propose a method to select the best camera from multiple feeds to create a football video broadcast.

2.1.2 Cricket

Hawk-Eye has become a mainstream part of cricket analytics by utilizing a multi camera setup to track the players and ball in the ground with applications like LBW prediction and player profiling. Along with these developments, there have been some strides in automated video understanding in the context of cricket. Sharma et al. [76] align textual commentaries to broadcast cricket videos which enables retrieval of specific actions associated with batsmen and bowler. This work builds up on Sankar et al. [72] who performed temporal segmentation of video shots and highlight generation from broadcast cricket videos using aligned textual commentaries. Kolekar et al [41] present a event detection and classification method that operates cricket video sequences.

2.1.3 Basketball

Bertasius et al. [4] assessed a basketball player's performance using first person videos from wearable devices. Ramanathan et al. [66] detected key actors and special events in basketball games by tracking players and classifying events using RNNs with attention mechanism. Bettadapura et al. [5] leveraged multimodal data to generate highlights for basketball by handcrafting features and then learning the excitement score for play sequences. Lucey et al. [51] utilize proprietary tracking data by representing a team play as "roles" to look at the spatiotemporal changes in a teams formation. The role representation further allows for large-scale retrieval of plays by using the tracking data.

2.1.4 Racket Sports

In this thesis, we focus on outdoor racket sports, namely tennis and badminton. Racket sports have received a lot of attention in this area with strides made in video summarization and highlight generation. For instance, Hanjalic et al. [28] was one of the first attempts at generating highlights automatically which have been superseded by other technqiues [25, 35]. We will now focus on works more specific to each of the sports themselves.

2.1.4.1 Tennis

Sukhwani et al. [90] generated descriptions for broadcast tennis shots automatically and also proposed [91] a dictionary learning method for frame level fine grained annotations for a given video clip, but their annotations are also computed at the level of actions, useful in the context of computing player statistics. Liu et al. [49] performed mutlimodal analysis to generate tennis video highlights while Connaghan et al. [17] attempted to segment out game into point, game and set, however, performed no score keeping and used multiple cameras to perform the task. Yan et al. [104] and Mentzelopoulos et al [56] worked on creating frameworks for annotating players while Xu et al. [102] and Miyamori et al. [59] focused on semantic annotations exploiting tennis domain knowledge to build retrieval systems based on positions and actions. Mlakar et al. [60] performed shot classification into forehand, backhand and serve using a wearable device for players.

Ball tracking has been another space with lots of prior work. For instance, Yan et al. [103] proposed a ball tracking method by proposing an all pairs shortest path formulation over spatiotemporal ball candidate tracks while Zhou et al. [107] generate a set of short trajectories using a shift token transfer method and then apply a dynamic programming based splice method to a directed acyclic graph consisting of these short trajectories. Reno et al. [70] proposed a platform for tennis which extract 3D ball trajectories using a specialized camera setup. In the space of spatiotemporal analytics, Wei et al. [100] performed an in-point analysis of rallies, predicted serve trajectory class [98] by leveraging player style priors, and, finally, predicted shot outcomes [99] by modeling player information through the style priors.

2.1.4.2 Badminton

Yoshikawa et al. [105] performed serve scene detection for badminton games with a specialized overhead camera setup. Chu et al. [16] performed semi-automatic badminton video analysis by detecting the court and players, classifying strokes and clustering player strategy into offensive or defensive. Careelmont [11] performed shot classification using a baseline camera similar to our scenario, however, worked with trimmed videos (unlike unconstrained broadcast videos in our scenario) and followed an approach wherein they tried to extract shuttlecock trajectories by following a two step procedure, firstly, they extracted 2d shuttle trajectories, then they assumed a theoretical model for modelling shuttle trajectories (defined by a set of initial parameters P), then found the parameter set closest to the extracted shuttle trajectory through an exhaustive search. Similar to Careelmont's work, Dierickx [19] extracted shuttlecock trajectories from badminton videos improving on the model Careelmont [11] proposed. Further, they identified shot type based on extrapolating the shuttle trajectory to its expected landing point.

2.2 Machine Learning for Computer Vision

2.2.1 Transfer Learning using CNN's

Consider a neural network ϕ parameterized by θ which takes an input sample x and categorizes the sample into one of C classes, or $y_i \in C$. This network is trained using gradient descent by optimizing the loss function over a large set of samples (x_i, y_i),

$$L(\theta) = \sum_{i=1}^{N} \mathcal{L}(\phi_\theta(x_i), y_i)$$

Now, say, the network is a sequential composition of functions, $\phi = \{\phi_1, \phi_2..\phi_L\}$ and say the weights are $\theta = \{\theta_1, \theta_2..\theta_L\}$. Given that we have a slightly different task, we can reuse the weights in a couple of ways,

1. **Fixed Feature Extractor:** Consider the first K layers of the CNN ϕ, we forward propagate each image x_i to obtain feature $\phi_{1:K}(x_i)$. Now, this feature can be used to learn a classifier (or essentially any other task, like clustering) and it has been observed that this strategy is very potent for most visual recognition tasks [74]. In our work, we use the SpatialCNN network from Two Stream Convolutional Network [81] trained on UCF101 dataset [86] to extract stroke features for the action segmentation task, described in detail later.

2. **Fine Tuning:** Consider a network $\phi' = \{\phi_1, \phi_2..\phi_K, \phi'_{K+1}, \phi'_{K+2}..\phi'_{L'}\}$, i.e first K layers correspond to ϕ. Now, let's say we initialize the weights $\theta' = \{\theta_1, \theta_2..\theta_K, \xi(\phi'_{K+1}), \xi(\phi'_{K+2})..\xi(\phi'_{L'})\}$ where ξ is an appropriate random weights initialization function. Now, we train the network using gradient descent for the new task. In our work, for instance, we have fine tuned Faster R-CNN [68]

for player detection by fine tuning the model with weights initialized from a VGG classification network [82] trained on the ImageNet dataset, described in detail later.

2.2.2 Person Detection and Tracking

The object detection problem is a widely studied problem in computer vision, with a lot of interest in the space of building trackers and detectors for people [63]. The general problem is simply, given an image, annotate a tight bounding box over each object of interest. Usually the objects of interest are a predefined set of classes. Tracking involves tracking an object over a sequence of frames.

In the context of sports analysis, many specific player detection and tracking methods have been proposed [56, 78, 104]. We adapt a popular object detection, called Faster R-CNN for this task.

2.2.2.1 Faster R-CNN

The Faster R-CNN [68] network comprises of three main submodules, as explained below,

1. A base convolutional neural network (such as VGG [82]) is used to extract a convolutional activation map from the input image.

2. Anchor points are defined for each spatial position in the convolution map (thus, separated by r pixels in the original image), and fixed sized boxes of different sizes and scales are generated for each anchor point, let's call them each of them an anchor proposal. The convolutional activation map is given as input to the Region Proposal Network (RPN), which has two outputs, the first is the objectness score per anchor proposal and the other is the adjustment to be applied to the anchor proposal. After that, non maximal suppression is applied to the anchor proposals and top N (=2000) proposals are kept.

3. Finally, a Region CNN (R-CNN) is defined to classify each proposal into an object class. However, the input to the Region CNN needs to be fixed sized. An ROI Pooling Layer is defined, this layer takes in the section of the input feature map that corresponds to the proposal and scales it to some pre-defined size by applying max-pool to nearly equal sized sections.

In the original implementation [68], the Base Network, the RPN and R-CNN were trained separately and then fine-tuned, however, recent implementations jointly train the model[1].

2.2.3 Scene Text Detection and Recognition

Text recognition has traditionally focused on document images, where OCR techniques are well suited to digitize documents. However, when applied to natural images, OCR techniques fail as they are

[1]In our work, we have utilized the following implementation: https://github.com/longcw/faster_rcnn_pytorch.

tuned for line spaced black-and-white documents. Scene text images have huge variation in appearance and layout, ranging from fonts and styles, inconsistent lighting, occlusions, orientations, noise. Moreover, background object may cause spurious false-positive detections. In this section, we discuss the two approaches we experimented with for the scene text recognition task in the context of identifying scores from scorecards in Chapter 3.

2.2.3.1 Textspot

Gupta et al [27] proposed a new method for text detection in natural images. They created a synthetic dataset by overlaying text to existing background images in a natural way.

They construct a fixed field of predictors centered at (u, v) for different values of u and v, say ϕ_{uv} and is tasked to prediction objects within a ball $(x, y) \in B_p(u, v)$. Each predictor, ϕ_{uv} predicts the presence of the object $c \in \mathbb{R}$ and the pose $p = (x - u, y - v, w, h)$ where (x, y) is the location and (w, h) denotes the size of the bounding box. Each detector ϕ_{uv} predicts directly object occurrences. However, unlike YOLO detector [67], detectors ϕ_{uv} are local and translation invariant, sharing parameters. They implement this field of predictors as the output of the last layer of a deep CNN, obtaining a fully-convolutional regression network (FCRN). The recognition is performed by the intermediary stage of the pipeline based on the lexicon-encoding CNN from Jaderberg et al [37].

2.2.3.2 CRNN

The CRNN, proposed by Shi et al [77], is designed to recognize text in images. For an input image, the convolutional layers are used to extract the feature sequence. Then a deep-bidirectional LSTM is fed the feature sequence to recognize the word in the image. As the length of the input sequence needn't be equal to the size of the word (predictions are made for each feature input), Connectionist Temporal Classification loss is used to train the model end-to-end. More details are present in Chapter 3.

2.2.4 Action Recognition and Segmentation

Action recognition, specially in the context of recognizing human actions and activities has been a widely studied problem by the computer vision community. The temporal nature of the videos provides an additional clue over the static image representations for recognizing actions. Many variants of the problems have received immense interest, however, we focus on two very popular tasks in this space. The first task is action classification, given a video containing one action, we wish to classify the video into one of the many categories. The second task is known as action segmentation, where, given a video consisting of a sequence of actions (overlapping or non-overlapping), we wish to recognize the category and the temporal extent of each action. In our work, we model the stroke recognition task as an action segmentation task with non-overlapping actions.

2.2.4.1 Two Stream Convolutional Networks

Two stream convolutional networks [81] divide the video into two streams, the spatial stream and temporal stream. Spatial stream Convolutional Network operates on individual video frames, performing action recognition from still images. However, the input of the temporal stream Convolutional Network is formed by stacking optical flow displacement fields between several consecutive frames. Softmax predictions of both the streams are then aggregated at the end, which significantly improves the accuracy. We utilize the Spatial stream CNN as a fixed feature extractor in our work and specific details are discussed in Chapter 3.

2.2.4.2 Temporal Convolutional Networks

Action segmentation can be thought of as a sequence labelling problem, considering the video frames/features as the input sequence and the action label for each frame as the output sequence. Thus, popular approaches to the general problem involve utilizing recurrent architectures such as LSTM. However, it has been noted that simple convolutional architectures outperform these recurrent architectures in certain tasks [3]. Temporal Convolutional Networks are a family of architectures, which can take a sequence of any length and map it to an output sequence of the same length, just as with an RNN with simple modifications to work in both causal and acausal modes. In our work, we utilize two variants described by Lea et al [47] for the action segmentation tasks, the ED-TCN and the Dil-TCN. These architectures have been described in detail in Chapter 4.

2.2.5 SVM's for Classification

Consider the binary classification problem, and K training samples, $\{(\mathbf{x_i}, y_i); i = 1, 2...K\}$ where $\mathbf{x_i} \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$. Let's assume that the two classes are linearly separable. We assume that there's a hyperplane ($\mathbf{w}$, b) that divides the two classes.

Now, there are many hyperplanes which may satisfy the above condition, however, an optimal hyperplane is defined as the one which maximizes the margin between the two classes. Consider two parallel hyperplanes, parallel to ($\mathbf{w}$, b), that go through the point closest to the hyperplane on the either side,

$$\mathbf{x_p^T w} + b = 1 \qquad \& \qquad \mathbf{x_n^T w} + b = -1$$

Points for which this equality holds are called the support vectors. Thus, every sample will have to satisfy $y_i(\mathbf{x_i^T w} + b) \geq 1$ and the distance between two parallel hyperplanes will be $\frac{2}{||\mathbf{w}||}$, which we need to maximize. The problem can be written as,

$$\begin{aligned} \underset{w,b}{\text{minimize}} \quad & \frac{1}{2}\mathbf{w^T w} \\ \text{subject to} \quad & y_i(\mathbf{x_i^T w} + b) \geq 1 \quad \forall i \end{aligned}$$

We can convert this problem to an unconstrained optimization problem by using lagrangian multipliers,

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w^T w} + \sum_i \alpha_i (1 - y_i(\mathbf{x_i^T w} + b))$$

Using KKT conditions, we can convert this to the dual problem,

$$\begin{aligned} \text{maximize} \quad & L(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \mathbf{x_i^T x_j} \\ \text{subject to} \quad & \alpha_i \geq 0, \quad \sum_i \alpha_i y_i = 0 \quad \forall i \end{aligned}$$

where our hyperplane is $w = \sum_j \alpha_j y_j \mathbf{x_j}$. The important point to note here is that the optimization doesn't depend on $\mathbf{x_i}$ and $\mathbf{x_j}$ and only depends on the inner product between any of the two samples. Thus, we can generalize the formulation by introducing a kernel function K and a mapping ϕ. Inner product in the earlier case was defined as $< \mathbf{x_i}, \mathbf{x_j} >= \mathbf{x_i^T x_j}$. Instead apply a kernel function K,

$$K(\mathbf{x_i}, \mathbf{x_j}) = \phi(\mathbf{x_i})^T \phi(\mathbf{x_j})$$

where ϕ is the kernel's implicit mapping and need not be explicitly defined. Now, the optimization can be rewritten as,

$$\begin{aligned} \text{maximize} \quad & L(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) \\ \text{subject to} \quad & \alpha_i \geq 0, \quad \sum_i \alpha_i y_i = 0 \quad \forall i \end{aligned}$$

and our classifier is now,

$$f(\mathbf{x}) = \phi(\mathbf{x})^T w + b = \sum_i \alpha_i y_i K(\mathbf{x}, \mathbf{x_i}) + b$$

This kernelizing procedure thus introduces non-linearity into the sample space which may help in improving the separability of the classes. In our work, we deal with polynomial kernels ($K_d(\mathbf{x}, \mathbf{y}) = (\mathbf{x^T y} + c)^d$) and χ-squared kernels ($K_\gamma(\mathbf{x}, \mathbf{y}) = exp(-\gamma \sum_i \frac{(x_i - y_i)^2}{x_i + y_i})$).

Further, to remove the assumption of linear separability, for each sample ($\mathbf{x_i}$, y_i), consider a slack variable ξ_i. Instead of assuming that every sample is situated beyond the margins, we penalize the distance they are within the margin. Mathematically,

$$\begin{aligned} \underset{w,b}{\text{minimize}} \quad & \frac{1}{2}\mathbf{w^T w} + C\sum_i \xi_i \\ \text{subject to} \quad & y_i(\mathbf{x_i^T w} + b) \geq 1 - \xi_i \quad \forall i \end{aligned}$$

where C is a hyperparameter that needs to be tuned. This is called a C-SVM and the resulting dual is very similar to the dual problem detailed earlier.

2.2.6 Gaussian Mixture Model for Clustering

Clustering is the task of grouping a set of instances in such a way that instances in the same group (called a cluster) are more similar to each other than to those in other clusters. The notion of similarity and the number of clusters usually need to be specified. In our work, we use a soft assignment based clustering method popularly known as a Gaussian Mixture Model which is described below.

Let's assume that the set of samples $\{x_j \in \mathbb{R}^d, j = 1, 2...N\}$ have to be grouped into K clusters. In a gaussian mixture model, we assume that there are K gaussian distributions (or components, w_i) with a mean vector μ_i and covariance Σ_i and samples are generated from these distributions. Thus, it can be said that each sample x_i was generated by randomly choosing w_i from $\{1, ..., K\}$, and then x_i was drawn from one of K Gaussians depending on w_i. To estimate the parameters here, we will be using the Expectation Maximization algorithm. For brevity we assume that parameters can be written as (where $p_i(t)$ denotes $p_{w_i}(t)$),

$$\lambda_t = \{\mu_1(t)..\mu_K(t), \Sigma_1(t)..\Sigma_K(t), p_1(t)..p_K(t)\}$$

There are two steps to the EM algorithm, the first is called the E-step, which involves computing the expected component for all samples of each component. Consider at timestep t,

$$P(w_i|x_j, \lambda_t) = \frac{G(x_j|w_i, \mu_i(t), \Sigma_i(t))p_i(t)}{\sum_{l=1}^{K} G(x_j|w_l, \mu_l(t), \Sigma_l(t))p_l(t)}$$

Now, in the M step, we re-estimate the parameter set from the expected values,

$$\mu_i(t+1) = \frac{\sum_j P(w_i|x_j, \lambda_t)x_j}{\sum_j P(w_i|x_j, \lambda_t)}$$

$$\Sigma_i(t+1) = \frac{\sum_j P(w_i|x_j, \lambda_t)[x_j - \mu_i(t+1)][x_j - \mu_i(t+1)]^T}{\sum_j P(w_i|x_j, \lambda_t)}$$

$$p_i(t+1) = \frac{\sum_j P(w_i|x_j, \lambda_t)}{N}$$

The EM algorithm is guaranteed to converge to a local optima under mild continuity conditions [101], however, the proof is outside the scope of the thesis.

Chapter 3

SmartTennisTV: Automatically Indexing Tennis Videos

3.1 Introduction

Sports streaming websites are very popular with many services like TennisTV and WatchESPN offering full game replays on demand. Millions of users use these services for entertainment, education and other purposes. However, tennis matches are usually very long, often running into hours. It's very hard to infer playing styles and patterns of players without investing hundreds of hours of viewing time. Thus, it's cumbersome to find "useful" parts. Streaming websites provide the video as-is, i.e. it's only possible to access the video stream sequentially. However, in case of sports and other event-rich video streams, an useful extension is to provide random access (like accessing an array) grounded in events along with sequential access, so that extensions like skipping to next event, filtering events etc can be provided.

In this chapter, we focus on constructing a point wise index of a tennis match and thus providing random access to the match. We propose a method to segment out the match into a set of rallies, then automatically extract the scorecard and the scores.Using tennis domain knowledge, we construct a novel algorithm to refine our extracted scores. We then demonstrate the utility of the automatically constructed index by building an interface to quickly and effortlessly retrieve and view the relevant point, game and set segments along with providing human accessible tags.

There are multiple challenges in this scenario. The tennis match videos are recorded from multiple camera angles and edited to have different kind of shots, to capture various emotions and drama along with the game play. With respect to extracting scores, the score board is never at a fixed position or in a specific format and the score digits are not constrained by font, size, and color.

The major contributions of this chapter are,

1. An effective score recognition algorithm using domain knowledge which can be adapted for different games. Here, we do our experiments on tennis videos by using the tennis domain knowledge.

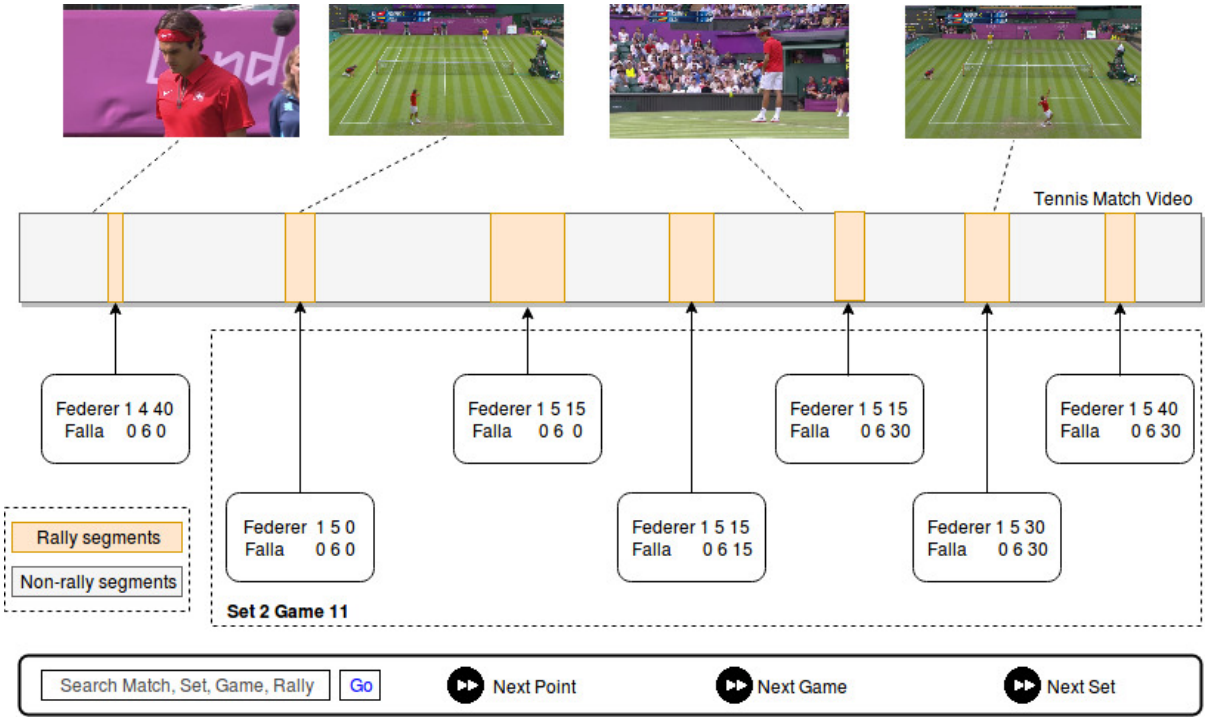

Figure 3.1: We aim to provide random access to tennis match videos and construct a point wise index of a tennis match so that a user can access, jump and skip "points", "games" and "sets".

2. We propose a score based indexing system, to navigate and retrieve segments from large volumes of video data with considerable ease.

3. Our method also enables many applications of indexing, we demonstrate one such application, human accessible event tagging.

Section 3.2 discusses advances and related work in literature. Section 3.3 forms the core of the paper, describing our core approach. Lastly, Section 3.4 provides a brief background of tennis and a high level description of our dataset(s), describes the implementation details and the experiments we performed along with obtained results.

3.2 Related Work

Sports Understanding: Using domain specific cues, several researchers have previously worked on improving sports understanding (specially tennis), with strides made in video summarization and automatically generating highlights [25, 28, 35], generating descriptions [90] and automatically segmenting coarse temporal scenes [106], annotating players [56, 104] and tracking the ball [103, 107].

Sports Video Indexing and Applications: Xu et al. [102] and Miyamori et al. [59] focus on semantic annotations exploiting tennis domain knowledge to build retrieval systems based on positions

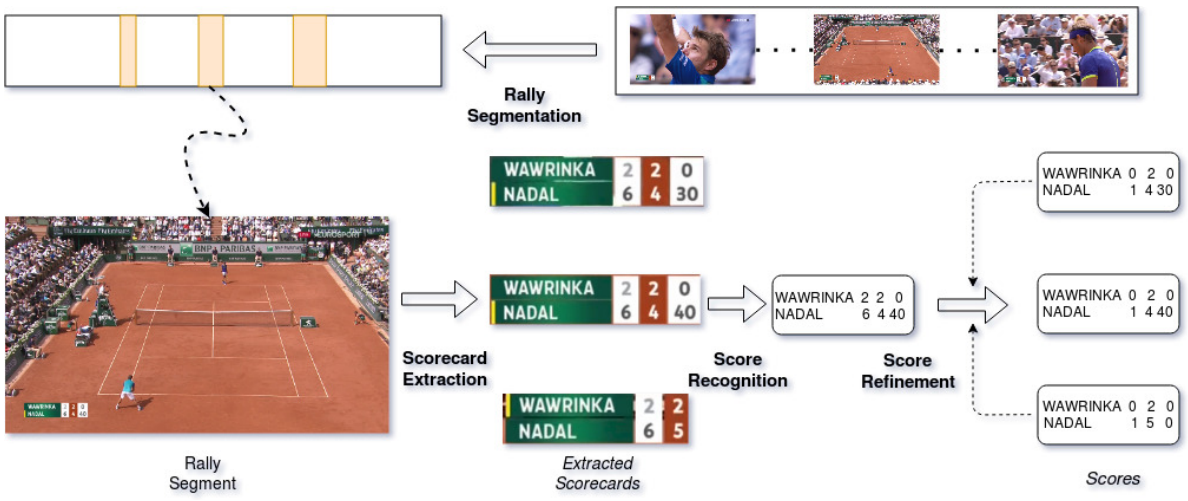

Figure 3.2: Our approach is illustrated in this figure. We start by temporally segmenting out the rallies, extracting the scoreboard and then recognizing the scores where we use contextual and domain knowledge to refine the recognized scores.

and actions. Sukhwani et al. [91] proposed a dictionary learning method for frame level fine grained annotations for a given video clip, but their annotations are also computed at the level of actions, useful in the context of computing player statistics. Kolekar et al. [42] use audio features to detect events in soccer scenes and generate highlights. Liu et al. [49] perform mutlimodal analysis to generate tennis video highlights while Connaghan et al. [17] attempt to segment out game into point, game and set, however, perform no score keeping and use multiple cameras to perform the task. However, these methods do not attempt to robustly index point level information to enable retrieval from the point of view of a viewer. Our work differs from all of these as we attempt to annotate point level information for a match.

Scorecard and Score Extraction: Liao et al. [48] focus only on detecting the scorecard while Miao et al. [58] focuses on both detection and extraction of scores, however the algorithm is specific for Basketball. Tesseract [85] is the commonly used OCR pipeline to detect text from images and documents which have a plain background. Convolutional Recurrent Neural Network (CRNN) [77] is applicable for performing end-to-end scene text recognition while Textspot [27] introduces a Fully Convolutional Regression Network (FCRN) which performs end-to-end scene text detection and for recognition, uses the intermediary stage of the pipeline based on the lexicon-encoding CNN from Jaderberg et al. [37].

3.3 Approach

Our goal is to automatically create an index for tennis videos. We begin by describing a method to automatically segment rallies. Then we detect and localize the scorecard in each of these rallies and

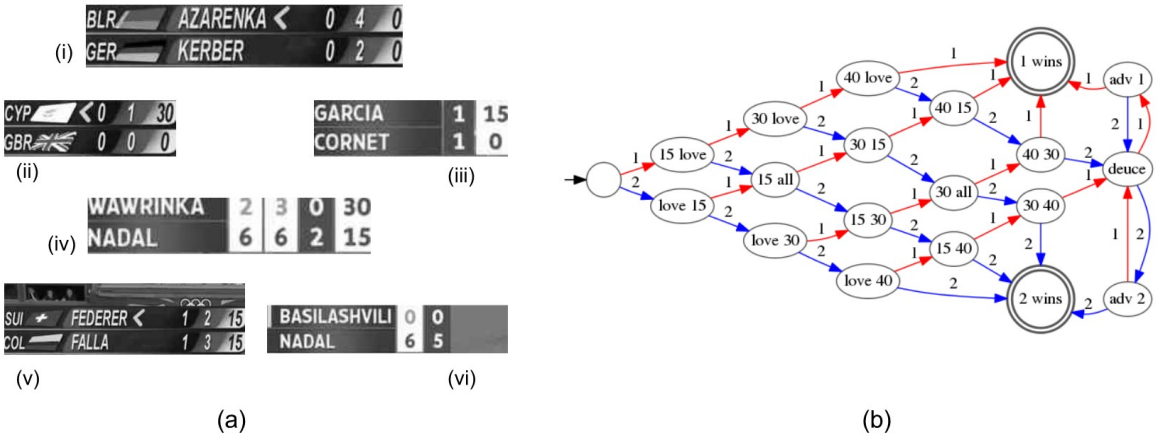

Figure 3.3: (a) depicts some of the extracted scorecards from different matches from our dataset. As one can see, the scorecards detected are of different sizes and formats, and differences across tournaments is noticeable. We have also included some of our failure cases, (v) and (vi) have extra regions that have been detected. (b) depicts the tennis point automaton that can be constructed from the tennis scoring system which is used to refine our extracted scores.

recognize the text to abstract out the game score state to annotate the video with the accessible tags. An overview of our pipeline can be seen in Fig. 5.1.

3.3.1 Rally Segmentation

Our method of segmenting out rallies stems from the observation that in BTV's, the camera is only overhead when the rally is in play and nowhere else. The background is mostly static after the serve begins, and remains the same till a player wins a point. HOG features are appropriate in such a scenario, so we extract frames from the Segment Dataset, downscale them, and extract HOG features. We then learn a χ-squared kernel SVM to label a frame either as a rally frame or a non-rally frame. Then, we use this learned classifier to label each frame of the BTV as part of a rally or otherwise and smoothen this sequence using Kalman filter to remove any false positives/negatives to obtain the segmented rallies.

3.3.2 Scorecard Extraction

We utilize the observation that the scorecard position is stationary in a rally, while the camera pans and moves around to cover the game. However, the scorecard may disappear and is not necessarily of the same size across the game as opposed to the assumptions in [58]. So, to overcome these issues, we extract the scorecard independently from each rally segment instead of assuming a single scorecard template.

We adapt the method described in [48]. We start by finding the gradient for each frame (say $I_x(i,j,t)$) using the sobel filter, and then calculate the normalized temporal sum for each frame using, $I_{norm}(i,j,n) = \frac{1}{n}\sum_{t=1}^{n} I_x(i,j,t)$. Then, we subtract I_x and I_{norm} to obtain the temporally correlated regions I_g. Further, we binarize the image using the following equation,

$$I_r(i,j,t) = (1 - \frac{I_x(i,j,t)}{max_{t,i,j}(I_x)})I_{norm}(i,j,t) \tag{3.1}$$

Empirically, the scorecard is found in one of the corners of the frame, we identify the four regions of size $(h/5, w/2)$ in the corners as the regions to search for the scorecard. Note, w and h are the width and height of the frame respectively. We identify the coarse scorecard region by selecting the region with the maximum number of white pixels in the specified regions in $I_r(i,j,t)$ summed over time. Further, after we have identified the coarse region, we apply morphological operators to remove small aberrations present and fit a rectangle which encloses the scorecard area. Our qualitative results can be seen in Fig. 3.3 (a).

3.3.3 Score Recognition

Traditional OCR based methods like Tesseract [85] can recognize text printed on a clear background however need the image to be preprocessed if the background is textured and shaded, and the contrast in the text fragments varies widely. However, with the advent of deep learning based OCR and scene text detection methods, a more general approach can be formulated.

To recognize scores, we experiment with three different methods, Tesseract, CRNN and Textspot. Textspot combines FCRN [27] which is an end to end text detection network, which constructs a field of predictors where each predictor is responsible for detecting a word if the word centre falls within the corresponding cell, akin to the YOLO network architecture. The recognition is performed by the intermediary stage of the pipeline based on the lexicon-encoding CNN from Jaderberg et al [37]. CRNN [77] is a scene text recognition network which treats the image as a sequence of strips. It proceeds by treating a CNN as a feature extractor to extract feature maps and construct a sequence of feature vectors. The sequence is fed into a bi-directional LSTM to obtain label sequence probabilities and CTC loss is employed to obtain labels. We adapt and perform a comparison of the various score recognition baselines in Section 3.4.

3.3.4 Score Refinement

To further refine our recognized scores, we use the knowledge of the tennis scoring system. As any structured game, score keeping in tennis is governed by a set of rules and thus, can be modeled as a finite automaton. Tennis in specific can be modeled as 3 automatons, one each for tracking the point, game and set score (See Fig. 3.3 (b)). Also, the vocabularies for point, game and set are restricted, so, we find errors by checking if the value belongs to the vocabulary or not. For instance, the the vocabulary for a point score is restricted to $\{0, 15, 30, 40, AD\}$.

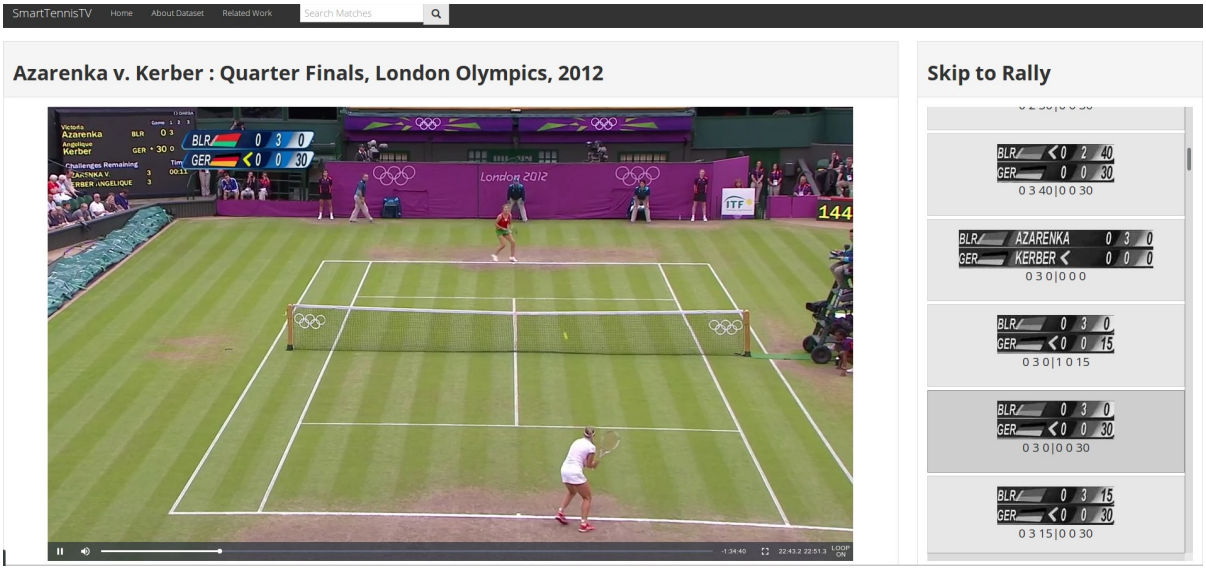

Figure 3.4: The developed interface supports the indexing and retrieval of a match as a point, game and set.

Let $J = (game_1, set_1, point_1, game_2, set_2, point_2)$ be the score state where game, set and point have the same meanings as in tennis. Firstly, we exploit the fact that the game and set scores are usually remain constant in a window, and thus replace errors with the mode of the value in the temporal window (with exceptions for score change within the window).

Consider the tennis scoring automaton T which is composed of score states and the transition function is constructed using the tennis scoring rules. Then we define a function $nextStates(s)$ which returns all possible states for the next game state. Likewise, $previousStates(s)$ provides the set of originating states for the current state s. For instance, from Fig. 3.3 (b), if we assume that we are at state $s = (0, 0, 30, 0, 0, 30)$ (referred to as 30 all in the figure), the function $previousStates(s)$ will return $\{(0, 0, 30, 0, 0, 15), (0, 0, 15, 0, 0, 30)\}$ and $nextStates(s)$ would return $\{(0, 0, 40, 0, 0, 30), (0, 0, 30, 0, 0, 40)\}$.

Assuming that the set of scores is $S = \{s_1, s_2...s_n\}$, and that s_i is erroneous (using vocabulary constraints), we compute the set $P = nextStates(s_{i-1}) \cap previousStates(s_{i+1})$, then we find the corrected score using,

$$s'_i = \arg\max_{p \in P} \frac{1}{|J|} \sum_{j \in J} \delta(s_i(j), p_i(j)) \tag{3.2}$$

where J is the set of game score states and δ is the Kronecker delta function. This equation is only needed if there are more than one possible score. It is to be noted that this method is extensible to any game which follows a structured scoring system like tennis.

Table 3.1: Averaged Edit Distance for score recognition (Lower is better)

Match	Textspot	CRNN	Tesseract-P
Match 1 (186 rallies)	0.2070	0.4272	0.2612
Match 2 (218 rallies)	0.2178	0.4476	0.3780

3.4 Experiments and Results

3.4.1 Dataset

A tennis match is divided into sets, each set is divided into games and each game has certain number of points or rallies. We restrict ourselves to "singles" matches and work with broadcast tennis video (BTV) recordings at 720p for 10 matches. 5 matches are taken from the French Open 2017 and remaining matches are from London Olympics 2012 for all our experiments. For performing rally segmentation, we created a "Rally Dataset" by manually annotating 2 matches into rally and non rally segments. The training and test set images are derived by dividing all the images in a 50-50 split. For evaluating score extraction, we further annotated 4 matches with score of each segment using the automated segmented rallies from our described algorithm. All together, we have annotated 1011 rallies to create the "Match Scores Dataset".

3.4.2 Rally Segmentation

For learning the rally segmentation classifier, we extracted every 10th frame from Rally Dataset and cross validated using a 0.8 split to find the optimal values of the hyper-parameters C and the period of the χ-squared kernel. The optimal value of C is 0.05 and the period of the χ-squared kernel SVM is found to be 3.

The mean $F1$ score on the test set for the task was found to be 97.46%, the precision for the non-rally segments was 98.94% and the rally segments was 95.41%.

3.4.3 Score Recognition

For employing Tesseract, we carefully preprocess the scorecard image and threshold the image manually. For each tournament such a preprocessing step needs to be manually defined. To train the CRNN, which is constrained to recognize words as sequences, we divided the scorecard to two parts horizontally. For employing Textspot, we don't train the network and use the model trained on "SynthText in the Wild" dataset as [27] note state-of-the-art performance on standard benchmarks. However, we post-process the text detection boxes and sort them to extract the scores. We used edit distance instead of the usual text recognition metrics because the "spaces" between scores (in the recognized string) are relevant in our case. For instance, CRNN removes repetitions of numbers, which causes the decrease in

Table 3.2: Averaged Score Accuracy AC(R) for our method and the defined baseline, FCRN (Higher is better)

Match	Textspot	Ours
Match 1 (186 rallies)	79.30%	91.66%
Match 2 (218 rallies)	77.90%	80.58%
Match 3 (201 rallies)	92.45%	95.19%
Match 4 (194 rallies)	85.22%	92.18%

accuracy. Table 3.1 here presents our experimental results on a subset of the matches and as we can see, Textspot performed the best and thus, for the next set of experiments we use that as our baseline.

3.4.4 Score Refinement

It is important to reiterate that our aim is not to recognize the text in the scorecard, but rather capture the game score state. To evaluate our results, we formulate a new metric, which inputs computed game state C_i and the actual game state G_i, and computes the following (for a set of rallies say R),

$$AC(R) = \sum_{i \in R} \frac{1}{|J|} \sum_{j \in J} \delta(C_i(j), G_i(j)) \tag{3.3}$$

where J and δ as defined earlier.

As can be seen from Table 3.2, our refinement algorithm shows a consistent improvement in the averaged score accuracy across matches over the best performing baseline method, Textspot [27]. However, as it is apparent, the performance of our method is dependent on the performance of the baseline score recognition and that is possibly the reason in the relatively meager improvements in score accuracy in the second match.

3.4.5 Event Tagging

Further, we automatically tagged common tennis events of importance to viewers such as “fault”, “deuce” and “advantage” using simple rules which define these tennis terms and our extracted scores. We compare our accuracy with and without score refinement and can observe that there is an improvement corresponding to improvement in the score accuracy. Accuracy for each tag per match (the matches are same as Table 3.2) can be seen in Table 3.3.

Table 3.3: Averaged Accuracy score of automatic event tagging (in percentage)

	Match 1		Match 2		Match 3		Match 4	
	Textspot	Ours	Textspot	Ours	Textspot	Ours	Textspot	Ours
Fault	66.66	70.83	52.24	56.71	87.87	90.90	84.44	84.44
Deuce	100.0	100.0	73.68	78.94	100.0	100.0	94.73	94.73
Advantage	100.0	100.0	77.77	77.77	100.0	100.0	95.65	95.65
Overall	75.00	79.41	60.58	64.43	92.45	94.33	89.65	89.65

3.5 Conclusion

In this chapter, we have presented an approach to create a tennis match index based on recognizing rallies and scores, supporting random access of "points" (Fig. 3.4) tagged with common tennis events. Further extensions to this work are numerous, such as providing point based semantic search and performing player analytics using videos instead of expensive sensor-based technologies.

Chapter 4

A Framework for Analyzing Broadcast Badminton Videos

4.1 Introduction

Sports analytics has been a major interest of computer vision community for a long time. Applications of sport analytic system include video summarization, highlight generation [25], aid in coaching [60, 70], player's fitness, weaknesses and strengths assessment, etc. Sports videos, intended for live viewing, are commonly available for consumption in the form of broadcast videos. Today, there are several thousand hours worth of broadcast videos available on the web. Sport broadcast videos are often long and captured *in the wild* setting from multiple viewpoints. Additionally, these videos are usually edited and overlayed with animations or graphics. Automatic understanding of broadcast videos is difficult due to its 'unstructured' nature coupled with the fast changing appearance and complex human pose and motion. These challenges have limited the scope of various existing sports analytics methods.

Even today the analysis of sport videos is mostly done by human sports experts [1] which is expensive and time consuming. Other techniques rely on special camera setup [70] or additional sensors [60] which adds to the cost as well as limits their utility. Deep learning based techniques have enabled a significant rise in the performance of various tasks such as object detection and recognition [30, 68, 82], action recognition [97], and temporal segmentation [47, 83]. Despite these advancements, recent attempts in sports analytics are not fully automatic for finer details [16, 104] or have a human in the loop for high level understanding of the game [1, 16] and, therefore, have limited practical applications to large scale data analysis.

In this work, we aim to perform automatic annotation and provide informative analytics of sports broadcast videos, in particular, badminton games (refer to Fig. 4.1). We detect players, points, and strokes for each frame in a match to enable fast indexing and efficient retrieval. We, further, use these fine annotations to compute understandable metrics (e.g., player's reaction time, dominance, positioning and footwork around the court, etc.) for higher level analytics. Similar to many other sports, badminton has specific game grammar (turn-based strokes, winning points, etc.), well separated playing areas (courts), structured as a series of events (points, rallies, and winning points), and therefore, are suited well for performing analytics at a very large scale. There are several benefits of such systems.

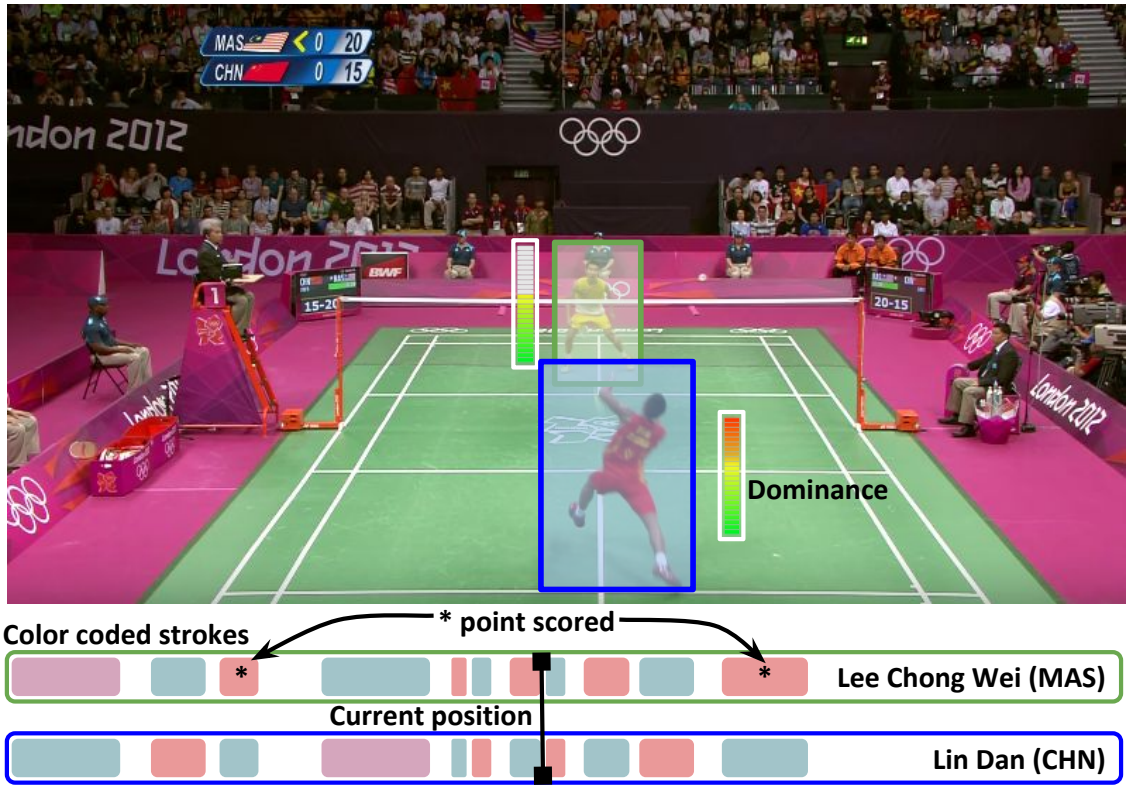

Figure 4.1: We aim to automatically detect players, their tracks, points and strokes in broadcast videos of badminton games. This enables rich and informative analysis (reaction time, dominance, positioning, etc.) of each player at point as well as match level.

Quantitative scores summarizes player's performance while qualitative game analysis enriches viewing experience. Player's strategy, strengths, and weaknesses could be mined and easily highlighted for training. It automates several aspects of analysis traditionally done manually by experts and coaches.

Badminton poses different difficulties for its automatic analysis. The actions (or strokes) are intermittent, fast paced, have complex movements, and sometimes occluded by the other player. Further, the best players employ various subtle deception strategies to fool the human opponent. The task becomes even more difficult with unstructured broadcast videos. The cameras have an oblique or overhead view of players and certain crucial aspects such as wrist and leg movements of both players may not be visible in the same frame. Tracking players across different views makes the problem even more complicated. We discard the highlights and process only clips from *behind the baseline* views which focus on both players and have minimal camera movements. For players detection, we rely on robust deep learning detection techniques. Our frame level stroke recognition module makes use of deep learned discriminative features within each player's spatio-temporal cuboid.

The major contributions of this chapter are

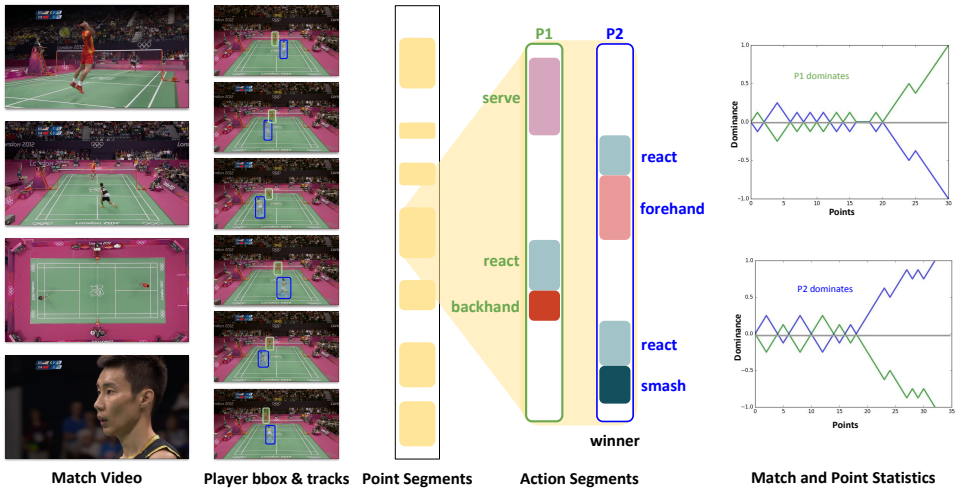

Figure 4.2: We propose to perform automatic annotation of various gameplay statistics in a badminton game to provide informative analytics. We model this task as players' detection and identification followed by temporal segmentation of each scored points. To enable deeper analytics, we perform dense temporal segmentation of player's strokes for each player independently.

1. We propose an end-to-end framework to automatically annotate badminton broadcast videos. Unlike previous approaches, our method does not rely on special camera setup or additional sensors.

2. Leveraging recent advancements in object detection, action recognition and temporal segmentaion, we predict game points and its outcome, players' tracks as well as their strokes.

3. We identify various understandable metrics, computed using our framework, for match and player analysis as well as qualitative understanding of badminton games.

4. We introduce a large collection of badminton broadcast videos with match level point segments and outcomes as well as frame level players' tracks and their strokes. We use the official broadcast videos of matches played in London Olympics 2012.

4.2 Related Work

Sports Understanding and Applications: Several researchers have worked on improving sports understanding using domain specific cues in the past [15, 73]. Racket sports have received a lot of attention in this area with strides made in video summarization and highlight generation [25, 28] and generating text descriptions [90]. Reno et al. [70] proposed a platform for tennis which extract 3D ball trajectories using a specialized camera setup. Yoshikawa et al. [105] performed serve scene detection for badminton games with a specialized overhead camera setup. Chu et al. [16] performed semi-automatic badminton video analysis by detecting the court and players, classifying strokes and clustering player strategy into

offensive or defensive. Mlakar et al. [60] performed shot classification while Bertasius et al. [4] assessed a basketball player's performance using videos from wearable devices. Unlike these approaches, our method does not rely on human inputs, special camera setup or additional sensors. Similar to our case, Sukhwani et al. [91] computed frame level annotations in broadcast tennis videos, however, they used a dictionary learning method to co-cluster available textual descriptions.

Action Recognition and Segmentation: Deep neural network based approaches such as Two Stream CNN [81], C3D [95], and it's derivatives [83, 97] have been instrumental in elevating the benchmark results in action recognition and segmentation. RNNs and LSTMs [33] have also been explored extensively [6, 22, 46] for this task owing to its representation power of long sequence data. Recently, Lea et al. [47] proposed temporal 1D convolution networks variants which are fast to train and perform competitively to other approaches on standard benchmarks for various temporal segmentation tasks.

In the context of sports activity recognition, Ramanathan et al. [66] detected key actors and special events in basketball games by tracking players and classifying events using RNNs with attention mechanism. Ibrahim et al. [36] proposed to recognize multi-person actions in volleyball games by using LSTMs to understand the dynamics of players as well as to aggregate information from various players. Action recognition in extreme sports with trajectory aligned features have also been studied by Singh et al. [84].

Person Detection and Tracking: An exhaustive survey of this area can be found in [63]. Specific methods for sports videos [56, 78, 104] and especially for handling occlusions [32] have also been proposed in the past. In the context of applications involving player tracking data, Wang et al. [96] used tracking data of basketball matches to perform offensive playcall classification while Cervone et al. [12] did point-wise predictions and discussed defensive metrics.

4.3 Badminton Olympic Dataset

We work on a collection of 27 badminton match videos taken from the official Olympic channel on YouTube[1]. We focus on "singles" matches played between two players for two or three sets and are typically around an hour long. Statistics of the proposed dataset used in our experiments are provided in Table 4.1. Please refer to the supplementary materials for the full list of matches and the corresponding broadcast videos. We plan to release our dataset and annotations publicly post acceptance of the work.

Matches: To train and validate our approach, we manually annotate a subset of 10 matches. For this, we select only one match per player which means no player plays more than one match against any other player. We choose this criteria to incorporate maximum gameplay variations in our dataset as well as to avoid overfitting to any specific player for any of the tasks. We divide the 10 matches into training

[1] https://www.youtube.com/user/olympic/

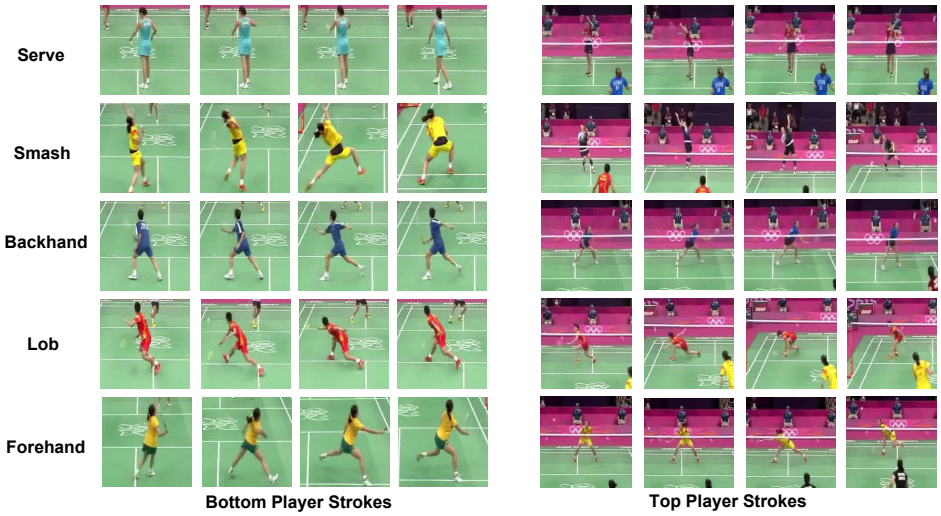

Figure 4.3: Representative "strokes" of bottom and top players for each class taken from our Badminton Olympic Dataset. The images have been automatically cropped using bounding boxes obtained from the player detection model. Top player appear smaller and have more complex background than the bottom player, therefore, are more difficult to detect and recognize strokes.

set of 7 matches and a test set of 3 matches. Note that this setup is identical to leave-N-subjects-out criteria which is followed in various temporal segmentation tasks [23, 83, 88]. Evaluation across pairs of unseen players also emphasize the generality of our approach.

Points: In order to localize the temporal locations of when points are scored in a match, we annotate 751 points and obtain sections that are corresponding to point and non-point segments. We annotate the current score, and the identity of the bottom player (to indicate the court switch after sets/between final set). Apart from this, we also annotate the serving and the winner of all the points in each set for validating outcome prediction.

Player bounding boxes: We focus on "singles" badminton matches of two players. The players switch court after each set and midway between the final set. In a common broadcast viewpoint one player plays in the court near to the camera while the other player in the distant court (see Fig. 4.1), which we refer to as bottom and top player respectively. We randomly sample and annotate 150 frames with bounding boxes for both players in each match (total around 3000 boxes) and use this for the player detection task. The players are occasionally mired by occlusion and the large playing area induces sudden fast player movements. As the game is very fast-paced, large pose and scale variations exist along with severe motion blur for both players.

Component	Classes	Total	Train	Test
Matches	NA	10	7	3
Players	NA	20	14	6
Player bboxes	2	2988	2094	894
Point segments	2	751	495	256
Strokes	12	15327	9904	5423

Table 4.1: Various statistics of our Badminton Olympic Dataset. Each match is typically one hour long. Train and test columns represents number of annotations used in respective split for experiments. Note that there is no overlap of players between train and test splits.

Strokes: The badminton strokes can be broadly categorized as "serve", "forehand", "backhand", "lob", and "smash" (refer to Figure 4.3 for representative images). Apart from this we identify one more class, "react" for the purpose of player's gameplay analysis. A player can only perform one of five standard strokes when the shuttle is in his/her court while the opponent player waits and prepare for response stroke. After each stroke the time gap for response from other player is labeled as "react". Also, we differentiate between the stroke classes of the top player and the bottom player to identify two classes per stroke (say, "smash-top" and "smash-bottom"). We also add a "none" class for segments when there is no specific action occurring. We manually annotate all strokes of 10 matches for both players as one of the mentioned 12 ($5 \cdot 2 + 2$) classes.

The "react" class is an important and unique aspect of our dataset. When a player plays aggressively, that allows very short duration for the opponent to decide and react. It is considered to be advantageous for the player as the opponent often fails to react in time or make a mistake in this short critical time. To the best of our knowledge, ours is the only temporal segmentation dataset with such property due to the rules of the game. This aspect is evident in racket sports as a player plays only a single stroke (in a well separated playing space) at a time.

Sports videos have been an excellent benchmarks for action recognition techniques and many datasets have been proposed in the past [36, 40, 66, 86]. However, these datasets are either trimmed [40, 86] or focused on team based sports [36, 66] (multi-person actions with high occlusions). On the contrary, for racket sports (multi-person actions with relatively less occlusion) there is no publicly available dataset. Our dataset is also significant for evaluation of temporal segmentation techniques since precise boundary of each action and processing each frame is of equal importance unlike existing temporal segmentation datasets [23, 84, 88] which often have long non-informative background class. Sports videos exhibit uncertain sequence of actions (depending on player's strategy, deceptions, injury etc.), in contrast to sequences in other domains (e.g. cooking activity follows a fixed recipe). Therefore, sports videos are excellent benchmarks for forecasting tasks.

4.4 Extracting Player Data

We start by finding the video segments that correspond to the play in badminton, discarding replays and other non-relevant sections. We then proceed to detect, track and identify players across these play segments. Lastly, we recognize the strokes played by the players in each play segment. We use these predictions to generate a set of statistics for effective analysis of game play.

4.4.1 Point Segmentation

We segment out badminton "points" from the match by observing that usually the camera is behind the baseline during the play and involves minimal camera panning. Other camera views are discarded by our setup. The replays are usually recorded from a closer angle and focus more on the drama of the stroke rather than the game in itself (however, rarely, points are also recorded from this view), and thus adds little or no extra information for further analysis. We extract HOG features from every 10^{th} frame of the video and learn a χ^2 kernel SVM to label the frames either as a "point frame" or a "non-point frame". We use this learned classifier to label each frame of the dataset as a "point frame" or otherwise and smoothen this sequence using a Kalman filter.

Evaluation The average F1 score for the two classes for optimal parameters (C and $order$) is 95.44%. The precision and recall for the point class are 97.83% and 91.02% respectively.

4.4.2 Player Tracking and Identification

We finetune a FasterRCNN [68] network for two classes, "PlayerTop" and "PlayerBottom" with manually annotated players bounding boxes. The "top player" corresponds to the player on the far side of the court and while the "bottom player" corresponds to the player on the near side of the court w.r.t to the viewpoint of the camera, and we use this notation for brevity. For the rest of the frames, we obtain the bounding boxes for both the players using the trained model. This approach absolves us from explicitly tracking the players with more complex multi-object trackers.

We further find the players' correspondences across points, as the players change court sides after each set (and the middle of the third set). For performing player level analysis it is important to know the identity of the player across points. The players wear the same colored jersey across a match and it is dissimilar from the opponent's jersey. We segment the background from the foreground regions using moving average background subtraction method [31]. We then extract color histogram features from the detected bounding box after applying the foreground mask, and take it as our feature. Now, for each point, we randomly average 10 player features corresponding to the point segments to create 2 player features per point. We cluster the features using a Gaussian Mixture Model into 2 clusters for each match. We then label one cluster as the first player and the other cluster as the second player. This approach is not extensible to tournaments where both the players are wearing a standard tournament kit.

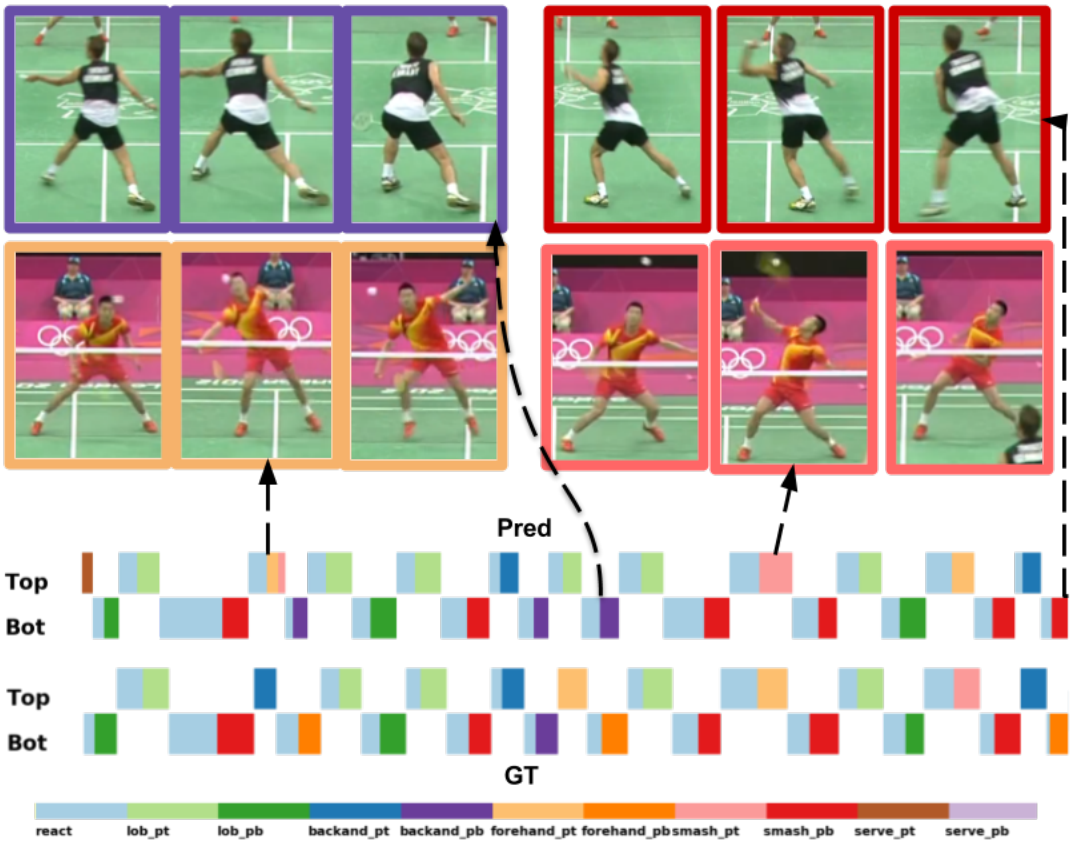

Figure 4.4: For a representative point, segment level strokes obtained from experiments are shown. Each action label has been color coded. *(Best viewed in color)*

Evaluation For evaluating the efficacy of the learned player detection model, we compute the $mAP@0.5$ values on the test set and obtain 97.85% for the bottom player, and 96.90% for the top player.

For evaluating the player correspondences, we compare the identity assignments obtained from the clusters with the manual annotations of player identity for each point. As our method is unsupervised, we evaluate the assignments for all the points in 10 matches. The method described above yield an average accuracy of 97.98%.

4.4.3 Player Stroke Segmentation

We employ and adapt the Temporal Convolutional Network (TCN) variants described by Lea et al. [47] for this task. The first kind, Encoder Decoder TCN (ED-TCN), is similar to the SegNet [2] architecture (used for semantic segmentation tasks), the encoder layers consist of, in order, temporal convolutional filters, non linear activation function and temporal max-pooling. The decoder is analogous to the encoder instead it employs upsampling rather than pooling, and the order of operations is reversed. The filter count of each encoder-decoder layer is maintained to achieve symmetry w.r.t. archi-

ED-TCN	**Metric**	**d=5**	**d=10**	**d=15**	**d=20**
HOG	Acc	71.02	72.56	71.98	71.61
	Edit	76.10	80.52	80.12	79.66
SpatialCNN	Acc	69.19	68.92	71.31	71.49
	Edit	77.63	80.48	80.45	80.40
Dilated TCN	**Metric**	**s=1**	**s=2**	**s=4**	**s=8**
HOG	Acc	70.24	68.08	68.25	67.31
	Edit	70.11	70.72	73.68	73.29
SpatialCNN	Acc	69.59	69.75	69.37	67.03
	Edit	59.98	69.46	74.17	71.86

Table 4.2: We evaluate the player stroke segmentation by experimenting with filter size 'd' and sample rate 's' respectively. *Acc* corresponds to per time step accuracy while *Edit* corresponds to edit score.

tecture. The prediction of the network is the probability of each class per time-step obtained by applying the softmax function.

The second kind, Dilated TCN is analogous to the WaveNet [64] architecture (used in speech synthesis tasks). A series of blocks are defined (say B), each containing L convolutional layers, with the same number of filters F_w. Each layer has a set of dilated convolutions with rate parameter s, activation and residual connection that combines the input and the convolution signal, with the activation in the l^{th} layer and the j^{th} block is denoted as $S^{(j,l)}$. Assuming that the filters are parameterized by $W^{(1)}$, $W^{(2)}$, b along with residual weight and bias parameters V and e,

$$\hat{S}_t^{(j,l)} = f(W^{(1)} S_{t-s}^{j,l-1} + W^{(2)} S_t^{j,l-1} + b)$$
$$S_t^{(j,l)} = S_t^{(j,l-1)} + V \hat{S}_t^{(j,l)} + e$$

The output of each block is summed using a set of skipped connections by adding up the activations and applying the $ReLU$ activation, say $Z_t^0 = ReLU(\sum_{j=1}^{B} S_(t)^{(j,L)}$. A latent state is defined as $Z_t^{(1)} = ReLU(V_r Z_t^{(0)} + e_r)$ where V_r and e_r are learned weight and bias parameters. The predictions are then given by applying the softmax function on $Z_t^{(1)}$.

To learn the parameters, the loss employed is categorical cross-entropy with SGD updates. We use balanced class weighting for both the models in the cross entropy loss to reduce the effect of class imbalance.

We experiment with two different feature types, HOG and a deep learned. Inspired by the use of HOG features by [16] for performing stroke recognition, we study the performance of HOG features on our dataset. The HOG features are extracted with a cell size of 64. As [47] benchmark use trained Spatial CNN features and other recent benchmarks use convolutional neural networks, we employ the Spatial

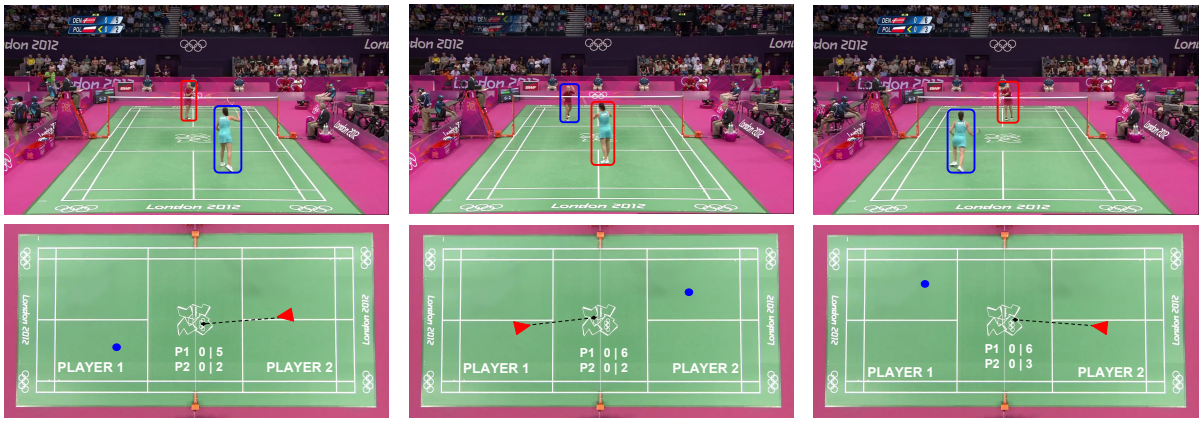

Figure 4.5: The serving player is indicated in red. It can be observed that the serving player is usually closer to the mid-line than the receiver who centers himself in the opposite court. Also, the positions of the players in the second point w.r.t. to the first point indicate that the bottom player has won the last point (as the serve is switched). *(Best viewed in color)*

CNN from the two stream convolution network model [81] to extract video features. However, instead of extracting features globally, we instead utilize the earlier obtained player tracks and extract the image region of input scale (454×340) centered at player track centroid for each time step. The players detections in each frame is then resized to 224×224. We then independently extract features for both the players and concatenate the obtained features per frame. The Spatial CNN used is trained on the UCF101 dataset, and we experiment with the output of FC7 layer as our features.

We use the default parameters for training the TCN variants as reported in [47]. We experiment the effect of dilation by setting the sample rate (s) at 1, 2, 4, and 8 fps for Dilated TCN. For ED-TCN we vary the convolutional filter size (say d) of 5, 10, 15 and 20 (setting $s = 2$). We employ acausal convolution for ED-TCN by convolving from $X_{t-\frac{d}{2}}$ to $X_{t+\frac{d}{2}}$. For the Dilated TCN case, we add the term $W^{(2)}S^{j,l-1}_{t+s}$ to the update equation mentioned earlier [47]. Here, X is the set of features per point and t is the time step.

Evaluation We employ the per frame accuracy and edit score metric used commonly for segmental tasks [47], and the results can be seen in Table. 4.2. We also experimented with causal models by convolving from X_{t-d} to X_t but observed that the performance of those models is not comparable to acausal models and thus did not report those results. The low performance of causal models can be attributed to the fact that badminton is fast-paced and unpredictable in nature. ED-TCN outperforms Dilated TCN which is consistent with benchmarking on other datasets [47]. We can observe that the filter size of 10 is most appropriate for the ED-TCN while the sample rate of 4 is most appropriate for Dilated TCN.

From Fig 4.4, it can be seen that the backhand and forehand strokes are prone to confusion, also smash and forehand strokes. In Fig. 4.4 note that the right shots do not look like a classical forehand shot and thus get confused with smashes. Similarly, the top and bottom right shots are hard to classify as backhand since first player is left-handed (most players are right handed), while the other's shot is visually hard to decipher. Please refer to the supplementary materials for more exhaustive and detailed results.

4.5 Detecting Point Outcome

The badminton scoring system is simple to follow and incorporate into our system. At the beginning of the game (score 0 — 0) or when the serving player's score is even, the serving player serves from the right service court, otherwise, from the left service court. If the serving player wins a rally, they score a point and continues to serve from the alternate service court. If the receiving player wins a rally, the receiving player scores a point and becomes the next serving player.

We exploit this game rule and its relationship to players' spatial positions on the court for automatically predicting point outcomes. Consider point video segments obtained from point segmentation (Section 4.4.1), we record predicted players' positions, and strokes played in each frame. At the start of next rally segment, the player performing the "serve" stroke is inferred as the winner of previous rally and point is awarded accordingly. We, therefore, know the point assignment history of each point by following this procedure from the start and until the end of the set. Also, it should be noted that for a badminton game the spatial position of both the serving and the receiving players is intrinsically linked to their positions on the court (See Fig. 4.5 for a detailed explanation). While similar observations have been used earlier by [105] to detect serve scenes, we detect both the serving player and the winning player (by exploiting the game rules) without any specialized setup.

We formulate the outcome detection problem as a binary classification task by classifying who is the serving player, i.e. either the top player is serving or the bottom player is serving. Thus, we employ a kernel SVM as our classifier, experimenting with polynomial kernel. The input features are simply the concatenated player tracks extracted earlier i.e. for the first k frames in a point segment, we extract the player tracks for both the players to construct a vector of length $8k$.

We varied the number of frames and the degree of polynomial kernel for our experiments and tested on the 3 test matches as described earlier. We observed that the averaged accuracy was found to be 94.14% when the player bounding boxes of the first 50 frames are taken and the degree of the kernel is six.

4.6 Analyzing Points

The player tracks and stroke segments can be utilized in various ways for data analysis. For instance, the simplest method would be the creation of a pictorial **point summary**. For a given point

(see Fig. 4.6), we plot the "center-bottom" bounding box positions of the players in the top court coordinates by computing the homography. We then color code the position markers depending on the "action"/"reaction" they were performing then. From Fig. 4.6, it is evident that the bottom player definitely had an upper hand in this point as the top player's positions are scattered around the court. These kind of visualizations are useful to quickly review a match and gain insight into player tactics.

We attempt to extract some meaningful statistics from our data. The temporal structure of Badminton as a sport is characterized by short intermittent actions and high intensity [65]. The pace of badminton is swift and the court situation is always continuously evolving, and difficulty of the game is bolstered by the complexity and precision of player movements. The decisive factor for the games is found to be speed [65], and it's constituents,

- Speed of an individual movement
- Frequency of movements
- Reaction Time

In light of such analysis of the badminton game, we define and automatically compute relevant measures that can be extracted to quantitatively and qualitatively analyze player performance in a point and characterize match segments. We use the statistics presented in Fig. 4.8 for a match as an example.

1. **Set Dominance** We utilize the detected outcomes and the player identification details to define dominance of a player. We start the set with no player dominating over the other and we define a player as dominating if they have won consecutive points in a set and add one mark to the dominator and subtract one mark from the opponent likewise. We then plot the time sequence to find both "close" and "dominating" sections of a match.

   For instance, in Fig. 4.8, which are statistics computed for a match, it's apparent that the initial half of the first set was not dominated by either players and afterwards one of the players took the lead. The same player continued to dominate in the second set and win the game.

2. **Number of strokes in a point** A good proxy for aggressive play is the number of strokes being played by the two players. Aggressive and interesting play usually results in long rallies and multiple back-and-forth plays before culminating in a point scored for one or the other players. To approximate, we count the number of strokes in a point. Interestingly, it can be observed in Fig. 4.8 that the stroke count is higher during the points none of the players are dominating in the match we have taken as example.

3. **Average speed in a point** To find the average speed of the players in a point, we utilize our player tracks. However, displacement of both the players would manifest differently in the camera coordinates. Thus we detect the court lines in the video frame and find the homography with the camera view (i.e. behind the baseline view) of the court. We then use the bottom of the player bounding boxes as proxy for feet and track that point in the camera view. Using a Kalman

Filter, we compute displacement and thus speed in the overhead view (taking velocity into account through the observations matrix) and normalize the values. This would act as a proxy for intensity within a point.

4. **Average Reaction Time** We approximate reaction time by averaging the time for react class separately for both the players and then normalizing the values. We assume that the reaction time for the next stroke corresponds to the player who is performing it (See Fig. 4.4) to disambiguate between the reactions. This measure could be seen as the leeway the opponent provides the player.

4.7 Discussions

Rare short term strategies (e.g., deception) and long term strategies (e.g., footwork around the court) can be inferred with varying degree of confidence but not automatically detected in our current approach. To detect deception strategy, which could fool humans (players as well as annotator), a robust fine-grained action recognition technique would be needed. Whereas, predicting footwork requires long term memory of game states. These aspects of analysis are out of scope of this work. Another challenging task is forecasting player's reaction or position. It's specially challenging for sports videos due to the fast paced nature of game play, complex strategies as well as unique playing styles.

Figure 4.6: We show the frame level players' positions and footwork around the court corresponding to game play of a single point won by the bottom player. The color index correspond to the stroke being played. Note that, footwork of bottom player is more dense compared to that of top player indicating the dominance of bottom player. *(Best viewed in color)*

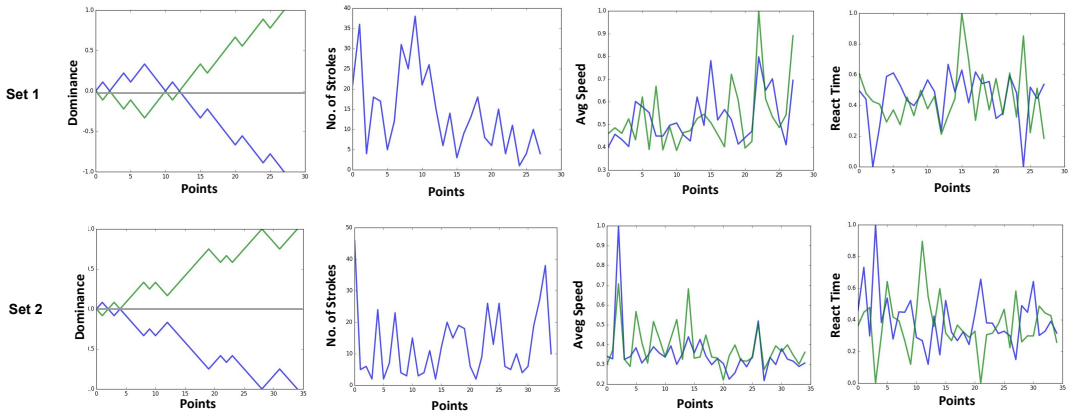

Figure 4.7: The computed statistics for a match, where each row corresponds to a set. It should be noted that green corresponds to the first player, while blue corresponds to second player. The first player won the match. *(Best viewed in color)*

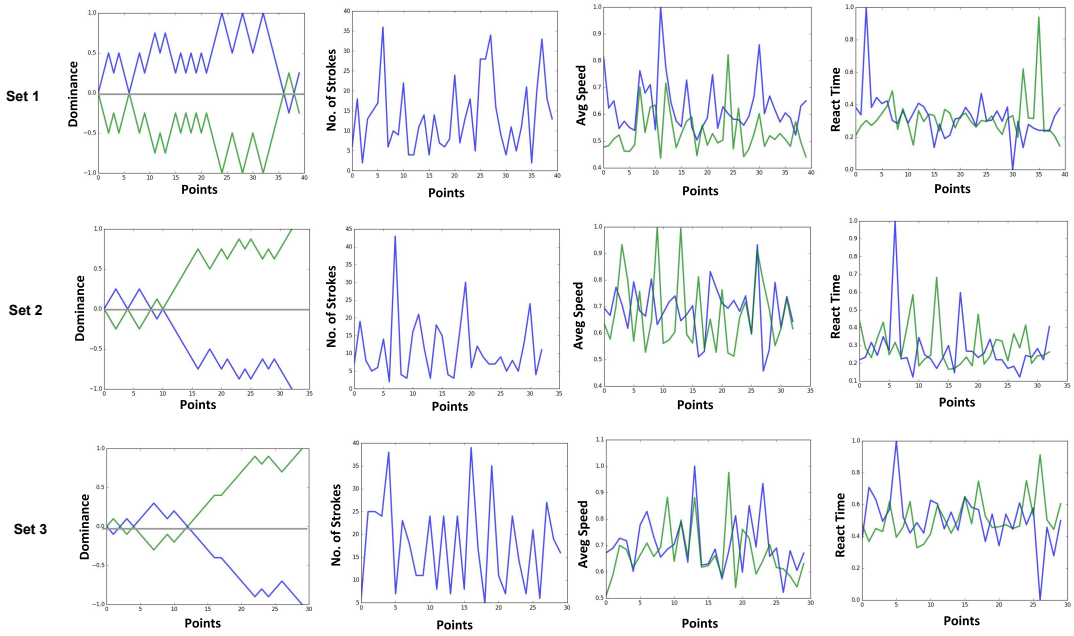

Figure 4.8: The computed statistics for a match, where each row corresponds to a set. It should be noted that green corresponds to the first player, while blue corresponds to second player. The first player won the match. *(Best viewed in color)*

Chapter 5

Mining Tennis Strategies from Broadcast Videos

5.1 Introduction

In recent years, there has been a great interest in utilizing machine learning and artificial intelligent techniques for applications in the physical world. Examples include building autonomous vehicles and health monitoring systems where perception modules have enabled reasoning. In the domain of sports and games, there has been great advances in building autonomous agents to play simple and structured games like Chess, Go [80] and multi-agent systems to play robot-soccer [9, 55]. Moreover, several multimedia systems have been deployed to archive and capture gameplay for both mainstream [61] and non-traditional sports [18].

However, that success has not translated to injecting machine intelligence into real world sports cheaply and ubiquitously. This is primarily due to lack of multimedia tools to extract sports data needed for reasoning, and planning. Though previous work has elaborated on generic representations for broadcast sports video analysis [21], such attempts have not focused on automating player analyses. Similarly, there have been previous attempts at developing analytics and data mining tools for sports and coaching [73], however, most of these systems or the generated data are proprietary [100] and [96]. Moreover, the data acquisition methods involve special camera setups further increasing the cost and complexity of the systems, making them out of reach from most players. Intrusion is another challenge in this space, with players asked to wear wearable devices for automated analyses [60]. For many sports, hours of game play time is analyzed by experts. These factors make sports analysis largely inaccessible.

To democratize sports intelligence, utilizing very simple and non-intrusive sensor(s) (like a single camera) is necessary to build a multimedia analytics system. Such a system can be used to mine sports or player data and generate automatic reports and insights for a coach to monitor. In this chapter, we focus on leveraging easily available broadcast video data to generate such insights automatically. Such a method also makes it possible to analyze players and matches from an era when more advanced setups did not exist, leveraging the thousand of hours of archived video data. We restrict our focus to tennis and investigate methods to mine tennis gameplay data from the broadcast video of the match. Finally,

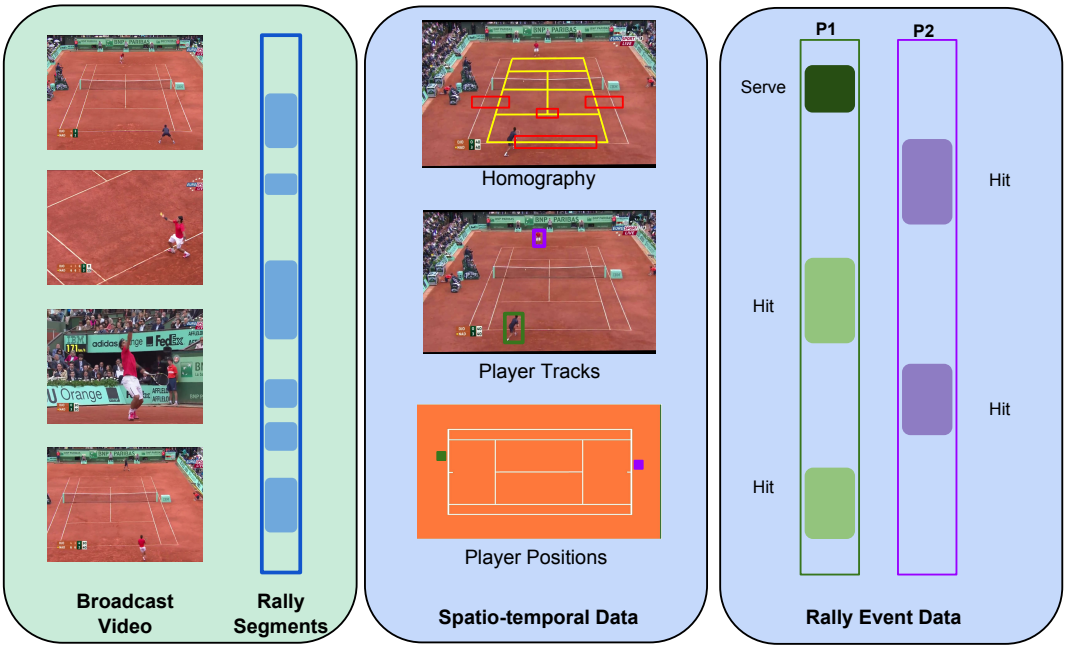

Figure 5.1: Data Mining Approach: The match video is segmented into rallies. Within a rally, we detect player locations and transform the observations wrt standard tennis court. We also detect the in-rally strokes from the visual data.

we analyze tennis players and their rivalries through a series of quantitative and qualitative experiments to prove the efficacy of the data in analyzing players.

Tennis as a sport is fast paced. It has opponent dependent strategic elements. For instance, some players show a preference to certain shot types and positions, which may be exploited by opponents. Likewise, players morph their game depending on this information and play to their advantage. Numerous variables, along with aforementioned issues affect play, such as court surface type, humidity and wind. These factors make the game challenging for analysis. However, as every player has a distinctive style of play, makes it ripe for the spatiotemporal data to be discernible.

We demonstrate that the gameplay data extracted from the broadcast videos can be used for further analysis through our extensive qualitative experiments to discover insights about the chosen players. We also demonstrate that the inferences made from the data extracted is comparable to a crowd-sourced (and manual) sports data collection project, Tennis Abstract [71], previously leveraged for tennis analytics [44].

5.2 Related Work

Multimedia Systems and Sports: Various multimedia applications have been proposed utilizing sports videos. Duan et al. [21] proposed a mid-level Representation Framework for Semantic Sports

Video Analysis. Bettadapura et al [5] and Merler et al [57] have utilized multimodal data to generate highlights for basketball, and golf respectively while there have been attempts to generate commentary from broadcast sports videos [90]. Destelle et al [18] present a multimodal 3D capturing plarform for preserving traditional sports and games like Gaelic and Basque sports. Monaghan et al [61] propose a low cost motion-capture system to understand player performance. For tennis analytics, many multiple-camera [54, 62, 69, 104] systems have been proposed. The constraint of only utilizing broadcast videos also differentiates this chapter from previous work on sports analytics using computer vision. However, tracking [20], recognition [87] and classification [40] problems in Computer Vision are some examples of the extensive use of sports videos for the proxy tasks.

Computer Vision and Sports: Recent advances in computer vision have utilized large scale datasets and deep learning approaches for many tasks, like video classification [40], scene understanding [30] and object detection [68]. In sports vision, there have been attempts to perform multi-person action recognition and event detection in volleyball [36] and basketball [66]. Sha et al. [73] perform analysis of swimmers from recorded video. Zhu et al [108] and Teachabarikiti et al [93] perform tracking, ball detection and stroke recognition (for near player) in broadcast videos. However, their videos are manually trimmed (for each point and action), and the corpus used is not very large and does not account for variation of players and tournaments. In this work, we benefit from breakthroughs in object detection by Faster R-CNN [68] and in action segmentation by Temporal Convolutional Networks [47] (recognizing and segmenting multiple actions in untrimmed videos).

Spatiotemporal Analytics in Sports: There has been previous work in many sports on spatiotemporal analytics, with the assumption of an available data source, usually from a proprietary system. In the space of Tennis, Wei et al. [100] performed an in-point analysis of rallies and predicted serve trajectory class [98] leveraging player style priors. Other sports like Football [8, 45, 52] and Basketball [4, 51] have also recently seen developments in player and team level analysis.

5.3 Preliminaries

5.3.1 Primer on Tennis

Here, we explain a few terms derived from the tennis vocabulary, which would be used throughout the chapter. These terms are in *italics* whenever mentioned in the context of tennis.

1. A tennis match is composed of *points*, *games* and *sets*. A *set* comprises of *games* which in turn consists of *points*.

2. When a player wins majority of the *sets*, they win the match. A *set* is won when the player has won a minimum of six *games* with at least a two *game* advantage. A tie-breaker is played out in case the *set* is tied at six *games* per player.

3. A player wins the *game*, if they win four *points* with a two *point* advantage. In the absence of two *point* advantage, the player must win two *points* in succession to win the *game*.

4. A *point* is the most granular level of scoring in tennis. A *point* consists of a sequence of back and forth *shots* between players, also known as *rally*.

5. In a *rally*, a *serve* is the opening *shot*. The *rally* is continued through *returns* until one of the player commits an error. The *return shot* of the non-serving player to the *serve* can also be termed as the *serve-return*.

6. While hitting a *return*, a right handed player is said to be in their *forehand position* or a left handed player is said to be in their *backhand position* if they are positioned at the right side of lower half of the court or left side of the upper half of the court respectively.

5.3.2 Scope

Tennis Players have distinctive styles of play and the gameplay analysis is usually performed by experts who have years of experience. They are able to pinpoint the player characteristics responsible for the player's dominance. For example, Rafael Nadal is considered to have a strong *forehand* and is thought to be the best player in the world on clay. For our analysis, we focus on the "Big Three" players [7], considered among the best in the world, Novak Djokovic, Roger Federer and Rafael Nadal. For brevity, we will use shorthands to represent Federer (**F**), Nadal (**N**) and Djokovic (**D**) in our figures. We wish to ask and answer the following relevant questions,

- Can broadcast videos be leveraged to mine data and characterize player style?
- How does player style manifests in the positioning and strategy of each player?
- Does the court type provide an advantage or a disadvantage to a certain player? Can we pinpoint specific characteristics?
- How does the strategy of a player evolve over time? Are there any striking differences that can be measured?

5.3.3 Dataset

Our dataset consists of 35 broadcast match videos, corresponding to Grand Slam matches (40 matches) played amongst Federer, Nadal and Djokovic from 2005 to 2017. We excluded 2 walkovers and for 3 matches, we were unable to find an appropriate source. These matches were played on all the three surface types: "grass", "hard" and "clay"; based on the Grand Slam. The resolution of the videos ranges from $360p$ to $720p$ with a minimum frame-rate of 25 frames per second.

We automatically extract the start and end timestamps of the whole *rally/point* from the match video. We do so by extracting HOG features for each frame and classifying as a "rally-frame" using a χ-squared SVM. Further, we annotate player bounding boxes which are classified as 'PlayerTop' and 'PlayerBottom' depending on the player position in the frame. Similarly, we annotate 4 stroke classes, 'serveTop', 'hitTop', 'serveBottom' and 'hitBottom' by providing the start and end timestamps for each action in the *rally*. We plan to release the mined spatiotemporal data and code post publication.

5.4 Mining Data from Videos

Our approach (see Figure 5.1) starts by stabilizing each *rally* segment and estimating the plane homography of the tennis court from the video. We then detect and track the positions of the players. Finally, we detect strokes in-*rallies* by classifying player actions to obtain spatio-temporal data.

5.4.1 Rally Stabilization and Homography Estimation

We first stabilize each *rally* segment. Then we estimate homography to transform points in camera coordinates to top-view standard coordinates. We stabilize the segment by finding a rigid transformation from previous to current frame using optical flow for all frames. The accumulated trajectories are then smoothened out using a sliding window.

Now, for the homography, we assume that the tennis court is a symmetrically located 2D object in a frontal plane which is tilted by an unknown angle θ. We find the gradient of each frame and use Hough transform to detect court lines and find the intersecting corner points (see Figure 5.2) [10]. Once the four correspondences are found, we interpolate the rest of the court points [89] and refine the homography using RANSAC [29]. We apply this method on all the frames of the *rally* after modifying the region of interest for the next frame by taking into account the locations of the points in the current frame (see Figure 5.1).

Evaluation: We successfully estimated the homography in 96.3% of the *rallies*. Due to peculiar camera views, rarely, the estimator fails to detect the required lines and points, thus resulting in an error. We discard such failure cases from our further analyses.

5.4.2 Player Detection and Re-Identification

To detect and track the two players in our obtained point segments, we utilize a Faster R-CNN network [68] and train it to detect two-player classes, 'PlayerTop' and 'PlayerBottom'.We run the detector on every frame, and thus did not require to explicitly use a multi-object tracker. We consider the "bottom-center" point of the bounding box as a proxy to feet position. We assume that this feet proxy is in the plane of the tennis court. Then we use the learnt homography estimate to transform the point to standard coordinates which is suitable for analysis.

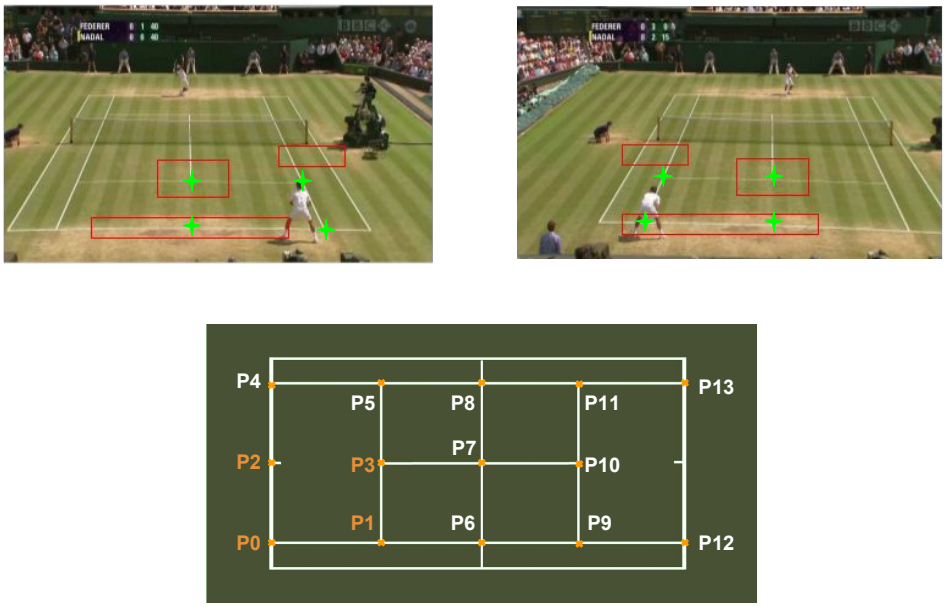

Figure 5.2: Homography Estimation: We detect intersecting court lines and thus point correspondences, and estimate the homography to the standard court coordinates.

However, Player level analyses requires us to build correspondence of players across the *points* in a match. This is due to the fact that, in a match, the players change court sides during designated intervals and in-match events. Thus it becomes important to re-identify them. We exploit the fact that the Players' outfit have distinct patterns and remains largely consistent through out the match. We segment the player foreground regions by subtracting the background obtained using moving average background subtraction method [31]. We extract color histograms of the player foreground regions from 10 random frames in each *point* to create 2 player features per *point*. We cluster these features using Gaussian Mixture Model into 2 clusters, one for each player in a match. The method is ineffective in Wimbledon matches due to the tournament rules which enforces a white dress code for the players. In those match videos, we resort to manual annotations.

Evaluation: The precision and recall of player detection model is respectively 98.3% and 99.7%. The failure cases involve ball boys getting detected as players. We discard such detections by measuring the distance from all the previous frame detections and setting an empirical threshold over that distance. The accuracy after evaluating player re-identification model on our dataset excluding the Wimbledon matches is 97.2%.

5.4.3 Stroke Recognition

We adapt the Temporal Convolutional Network (TCN) variants described by [47] to classify player strokes (see Figure 5.1). Specifically, we employ the Encoder Decoder TCN (ED-TCN) as the authors

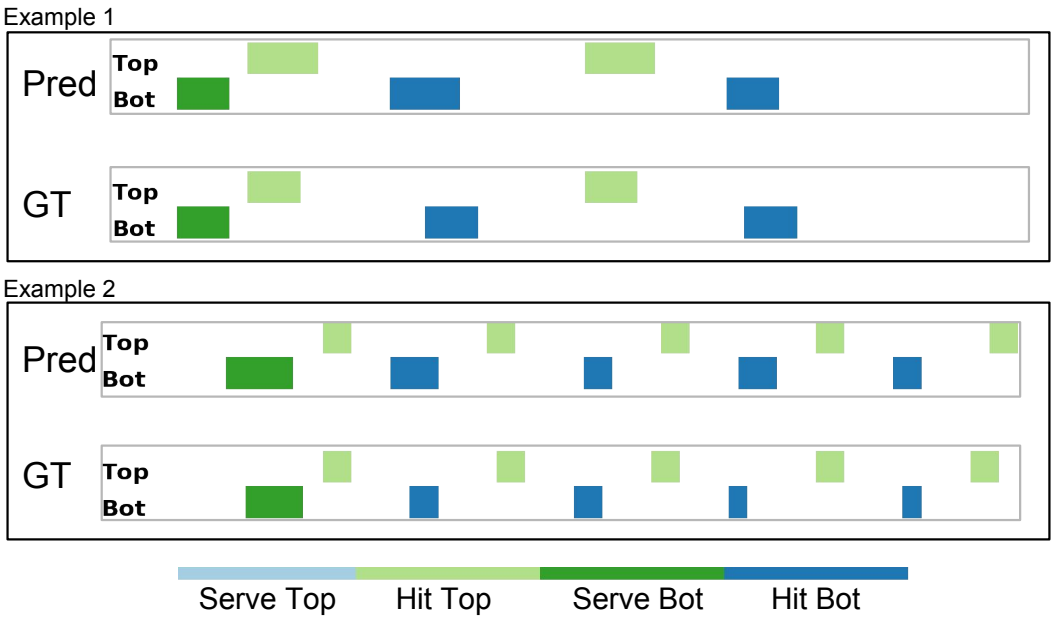

Figure 5.3: Qualitative Results (Stroke Detection): It can be observed that the model is able to localize temporally and accurately classify player strokes. *(Best viewed in Color)*

note superior performance over the Dilated TCN variant and is arguably a simpler model [47]. The encoder network hierarchically convolves on the temporal input feature (extracted for each frame) and max-pools the outputs from the ReLU activation function. The decoder network hierarchically performs deconvolutions over the upsampled outputs from the encoder network. The decoder output is then classified into individual classes using a softmax activation. The network is trained using the categorical cross entropy loss with weighted classes. For every frame, we extract HOG features from a fixed sized box centered at both the player bounding boxes and concatenate them. The video frame is rescaled to 640×480. HOG features are extracted from a 112×112 box centered at centroid of the 'PlayerTop' bounding box and similarly a 224×224 box centered at 'PlayerBottom' bounding box to account for perceptive size differences. The cell sizes used are 64 and 32 respectively.

As we aim to obtain a robust model for further analysis, we restricted the *stroke* classes to *serve* and hit by either players, instead of finer action categories. The proposed model works well for temporal localization of *strokes*. (see Figure 5.3).

Evaluation: The filter size $d = 5$ and sample rate $s = 4$ were found to be the optimal parameters for ED-TCN. The per-frame classification accuracy of model is 84.19% and the segmental edit score is 88.98% . It should be noted that the temporal extents of player actions are subjective in nature.

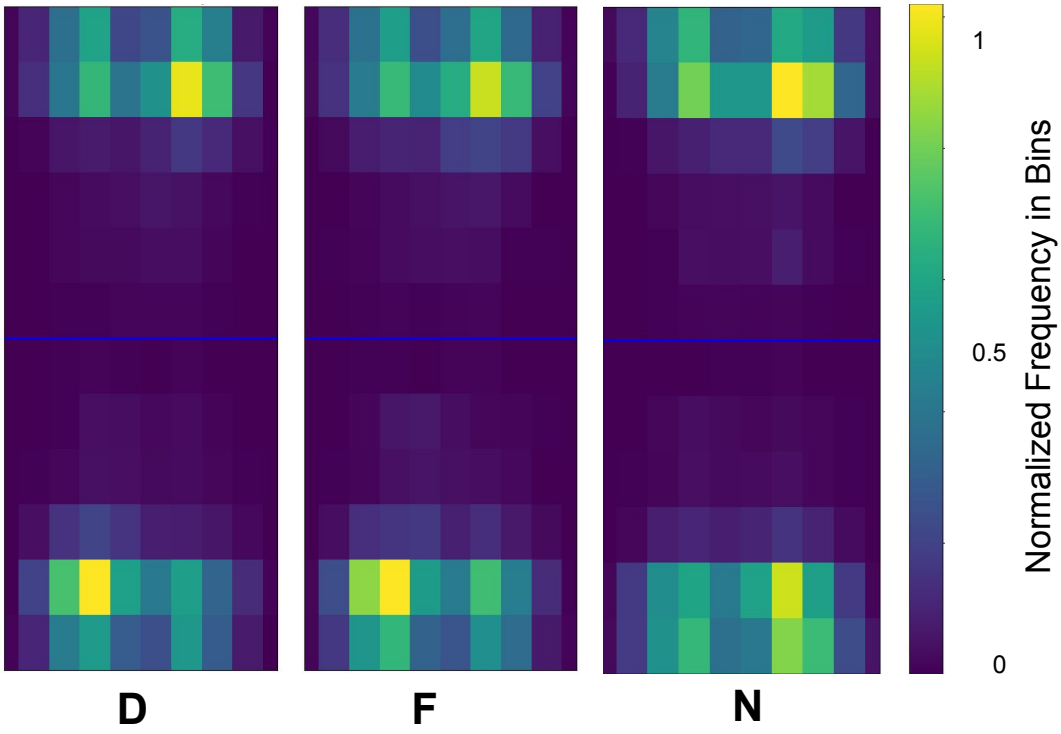

Figure 5.4: Aggregated and discretized player locations: D, F and N stand for Djokovic, Federer and Nadal respectively and the top positions depict the opponent. It can be observed that the three players have a distinctive style in terms of positioning and placement.

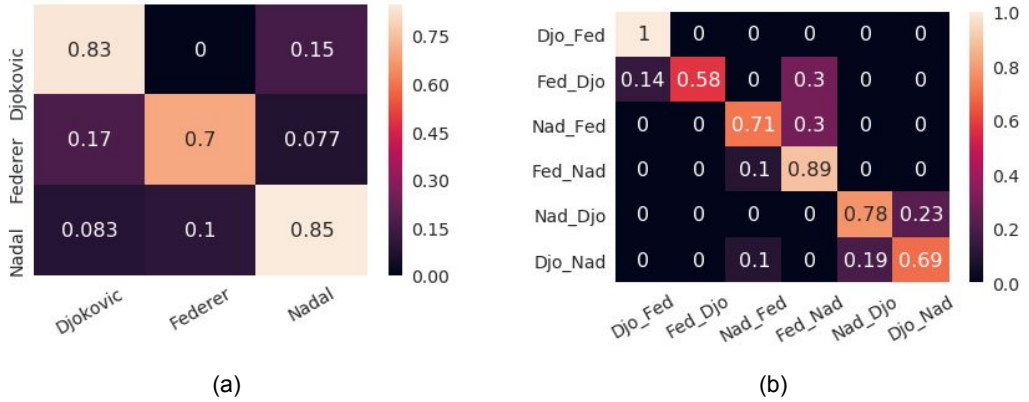

Figure 5.5: Confusion matrices: (a) identifying the winner of each *set* and (b) identifying the winner and the opponent in each *set* (first player in the label is the set winner)

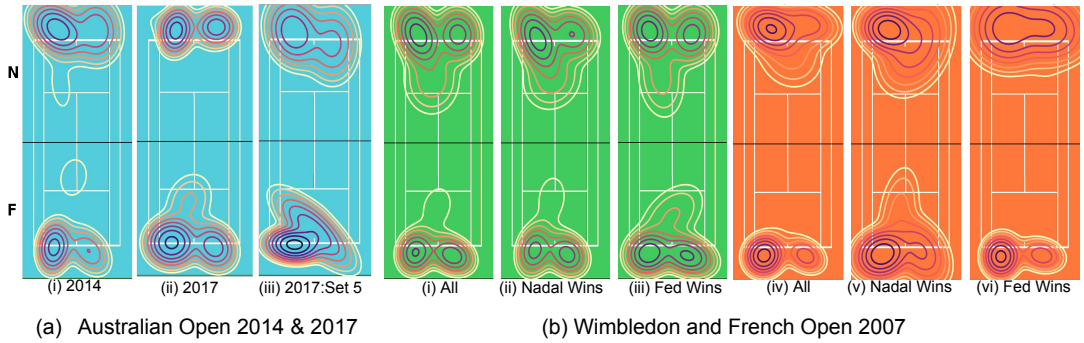

Figure 5.6: **Federer vs Nadal** Heatmaps for (a-i) All sets, 2014 Australian Open Semifinals (a-ii) All sets, 2017 Australian Open Finals (a-iii) 5^{th} set, 2017 Australian Open Finals (b-i) All rallies, 2007 Wimbledon Finals (b-ii) All rallies won by Nadal (b-iii) All rallies won by Federer (b-iv) All rallies, 2007 French Open Finals (b-v) All rallies won by Nadal (b-vi) All rallies won by Federer *(Best Viewed in Color)*

5.5 Individuality of Players

Before we attempt to analyze the mined data, the first question that arises is, if the data itself can be used to discriminate between players strategies. In the absence of strategy labels, it is very hard to evaluate the efficacy of the data. Inspired by Lucey et al [52] work on evaluating team behavior without such labels, we attempt at identifying the identity of the winner and the opponent given the spatiotemporal data of the players.

We first extract both the players' locations when they hit a *shot* (midpoint of the hit segment) and exclude the *serve* locations. We homogenize the data for the winner of a *set* by marking all their locations on the bottom half of the court. We now divide the court into a grid of size $k \times k$ and aggregate the player locations (see Figure 5.4). The histograms are aggregated and normalized for each *set* for the identification problem. We perform MaxAbs scaling of each feature and apply PCA to reduce the dimensionality to 35 dimensions.

To evaluate our hypothesis, we classified the identity of the winning player of the *set* given this feature vector. We employed a $3:1$ train-test split. We used a RBF-SVM and our model is 80.0% accurate (random predictor is accurate to 32.6% on the test data). Using the same method, jointly identifying both the winner and the opponent in the set resulted in model accuracy of 80.0% (random predictor is accurate to 16% on the test data), implying that the players have distinctive strengths and weaknesses which partially manifest in their footwork. We plot the confusion matrix for both the experiments in Figure 5.5. The optimal k is 12, the SVM parameters C and γ are 10^5 and 10^{-3} respectively. We perform two more experiments. Excluding the opponent locations resulted in a lower accuracy of 74.29% and only considering the locations (of both players) at the last *shot* meant the accuracy went down to 63.86%. Thus, we can infer that the opponent's positioning holds discriminative information and the hits preceding to the winning hit are strategically important [99].

5.6 Spatiotemporal Analysis of Rivalries

5.6.1 Player Position Heatmaps

We observed that player positions are a discriminative aspect of player style, and wished to visualize these patterns and analyze them in a match. Inspired by Wei et al [100]'s visualizations, we gather the player position data at their respective last hits in the *rally* based on the *rally winner* and *set winner*. We then learn probability distribution functions (pdf's) using these data-points and generate position heatmaps.

5.6.1.1 Federer vs Nadal, Australian Open, 2014 and 2017

Nadal won the 2014 Semifinals, while Federer won the 2017 Finals and our observations from the heatmaps include (see Figure 5.6-(a)),

- Federer is more aggressive in approaching the net in 2017 and was not pushed to his *backhand position* as often (see Figure 5.6-(a) (i) and (ii)) .
- Surprisingly when Federer is pushed to his *backhand position* in the final *set* (see Figure 5.6-(a) (iii)), he still prevailed. This can be attributed to his improved *backhand shot* (Discussed in Section 5.8.2)

5.6.1.2 Federer vs Nadal, French Open and Wimbledon, 2007

Federer won the 2007 Wimbledon Finals against Nadal, while he lost the 2007 French Open Finals.

- Federer's playing style was more aggressive in Wimbledon where he approached the net more often (see Figure 5.6-(b) (i) and (iii)) when compared to his positions in the French Open match (see Figure 5.6-(b) (iv) and (vi))
- Federer was pushed towards his *backhand position* in French Open when he lost (see Figure 5.6-(b) (v)) when compared to Wimbledon (see Figure 5.6-(b) (ii)). The trends reverse for Nadal in French Open and Wimbledon respectively (see Figure 5.6-(b) (iii) and (vi)).

5.6.2 Relationship Between Court Coverage and Player Speed

We wish to understand the relationship between player speed and court coverage, do fast players also cover a lot of ground? Further, we attempt to observe patterns in the deciding *sets* and the correlation between the average speeds and court coverage.

For calculating court coverage, we follow the method similar to [62]. We utilize our court grid aggregations and calculate the fraction of grids that were covered by the player in their half of the court.

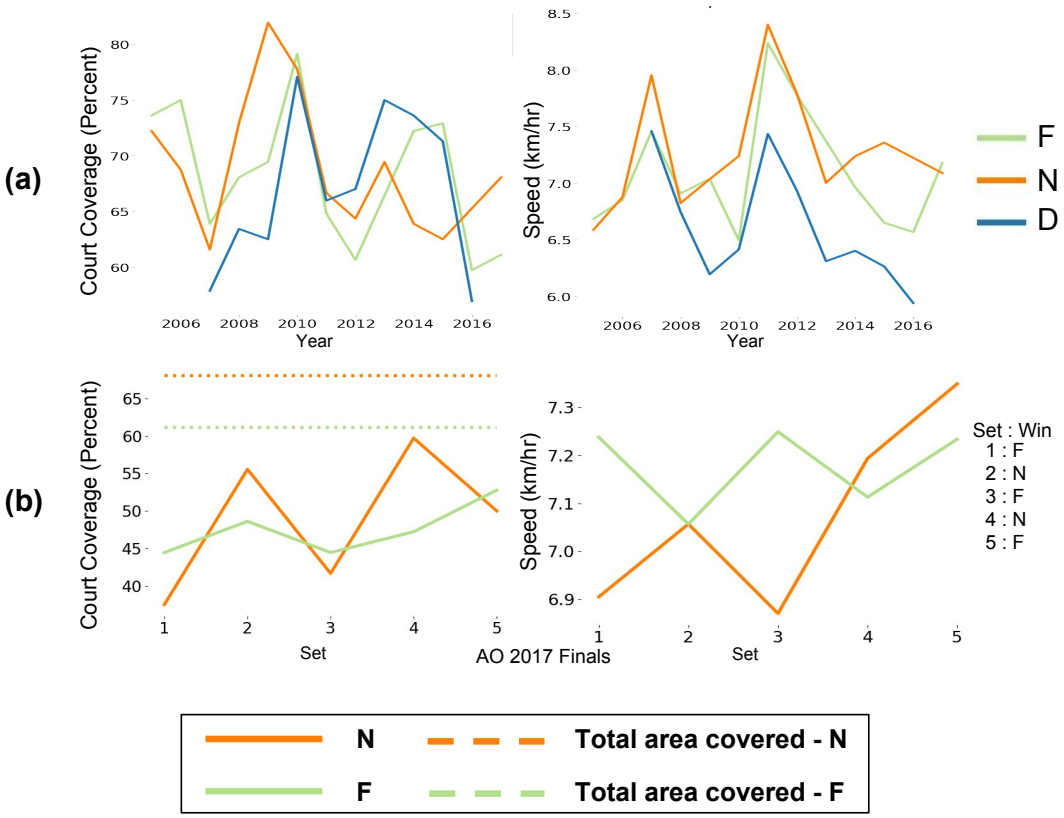

Figure 5.7: Court Coverage and Player Speed: (a) Averaged over matches in a year (b) For each set of 2017 Australian Open Finals *(Best Viewed in Color)*

Next, we calculate the speed of the players of a *rally* using the coordinates extracted from a *rally* by employing an extended Kalman Filter. Our major observations include,

1. We observe that Djokovic has the slowest average speed while Nadal has the highest among the three (see Figure 5.7 (a)). However, despite his slow speed, it does not considerably affect the average court coverage.

2. Taking 2017 Australian Open Finals as an example, we can make few observations (Figure 5.7 (b)). Excluding the last *set*, we observe that the highest area covered and average speed in a *set* correspond to the *set* winner. This is possibly because players strategize to conserve energy for further play. Also, we can observe an increase in game intensity in the final *set*, both in terms of area covered and average speeds.

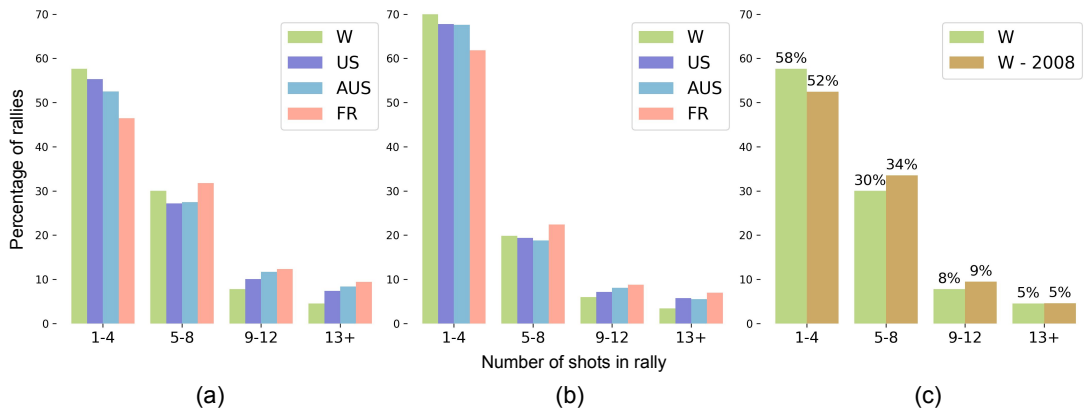

Figure 5.8: Rally Length Analysis - Grand Slams: Plotting the win percentage rally length for each Slam (a) Our Mined Data (b) Tennis Abstract Data (c) Disparity in rally lengths in 2008 Wimbledon from other Wimbledon matches *(Best Viewed in Color)*

5.7 Rally Length Analysis

Rally lengths are an important factor in understanding player styles. Players with longer *rally* lengths are expected to have strong *returns* while players with shorter *rallies* are expected to have strong *serves* [43]. We measure the number of *shots* in a *rally* by counting the player stroke predictions in a *rally*.

5.7.1 Contrast among Grand Slams

We computed the *rally* length distribution for all matches in each Grand Slam (see Figure 5.8 (a)). Based on our analysis, we observe that the *rally* lengths relate inversely to the pace of the sport in each Grand Slam i.e., Wimbledon (Grass Court), US Open and Australian Open (Hard Courts), French Open (Clay Court), in the decreasing order [43]. We also plotted the *rally* length graph from the Tennis Abstract Data [71] (see Figure 5.8 (b)), and can observe a correspondence in the trends.

5.7.2 Player Inferences

We wish to understand the differences in playing style of the Big Three and how it effects *rally* lengths. We should note that the *serving player* always has an advantage, and thus we measure the win percentage in three cases, for each player, (a) irrespective of who *serves*, (b) *serving player* wins and (c) *non-serving* player wins. From our analysis, following observations can be made,

1. We can clearly observe the advantage that a *server* holds from Figure 5.9. For a *non-server*, creating an advantageous situation means reducing the *server's* advantage and thus engaging in longer *rallies*. Conversely, a player with stronger *serves* would prefer short *rallies*.

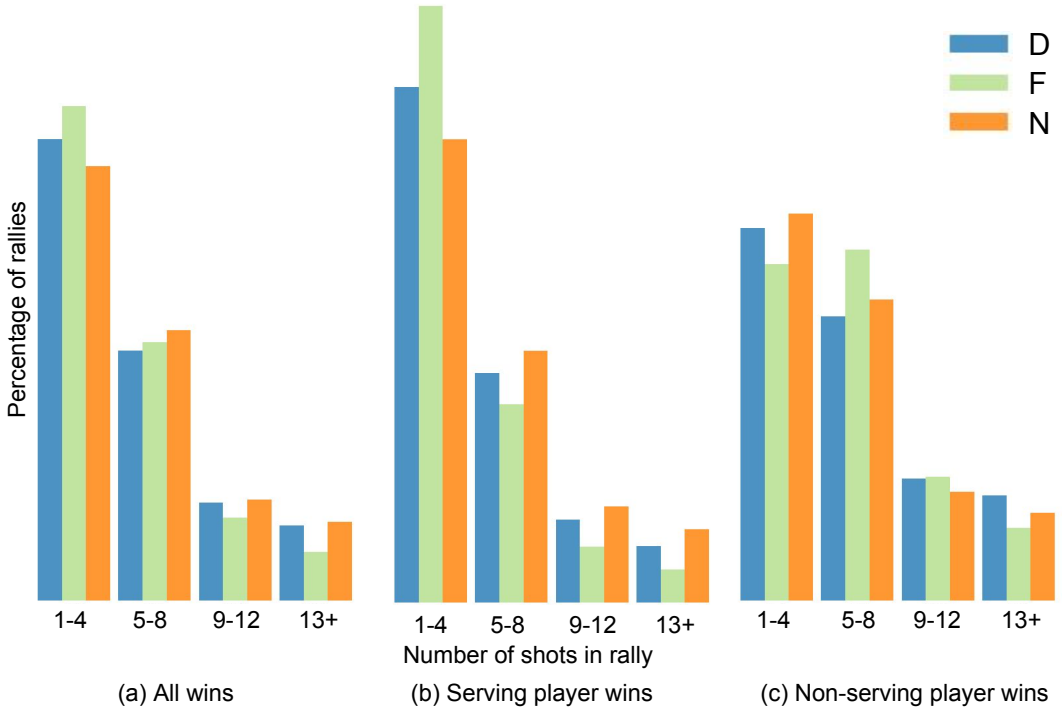

Figure 5.9: Rally length Analysis - The Big Three: We can observe the clear distinction in playing styles with Federer and Nadal with Federer preferring shorter rallies while Nadal prefers drawn out rallies. *(Best Viewed in Color)*

Tournament	Federer	Nadal	Djokovic
Wimbledon	14	7	10
US Open	14	4	14
Aus Open	10	13	9
Roland Garros	7	27	11
Total	**45**	**51**	**44**

Table 5.1: **The Big Three: Head-to-Head.** Sets won by each player in all Grand Slam matches in dataset

2. We note that Federer has the highest percentage of short *rally* wins, which follows from the fact that he is considered a strong *server*. Nadal and Djokovic, however, show a preference for longer *rallies*, typical for players considered strong *returners*.

3. Federer's strong *serves* biases him to prefer shorter *rallies*, thus Wimbledon and French Open (Section 5.7.1) are the best and least suited tournaments respectively (See Table 5.1 for winning statistics). Similar inference can be extended to Nadal, for whom French Open and Wimbledon are the best and least suited tournaments respectively.

4. **The Curious Case of 2008 Wimbledon Final:** Federer is the favorite at Wimbledon matches, however he lost to Nadal in the 2008 final. We can observe a 6% decrease from the Wimbledon average (52% vis-à-vis 58% in our dataset) in the win percentage of very short *rallies* ((Figure 5.8 (c))) and a corresponding 5% increase in longer *rallies*. Thus, it can be inferred that Nadal played a game to his strengths (strong *returns*) and capitalized to win the match.

5.8 Case Studies

5.8.1 Return Pressure

In this section, we analyze the return pressure exerted by players on a *server*, which is defined as the average amount of time the *returner* gives the *server* to react to their first *shot* [43], lowering his *serve* advantage. We measure the time difference between the *returnee's* first *shot* and *server's* second *shot* in a *rally*. Lower the time difference, higher is the return pressure.

As we can observe in Figure 5.10, Djokovic fares considerably better compared to Federer and Nadal respectively (in concurrence with [43]). Nadal applies the least return pressure among the three which may be attributed to his preference to stay far behind the *court baseline* on most *serves*.

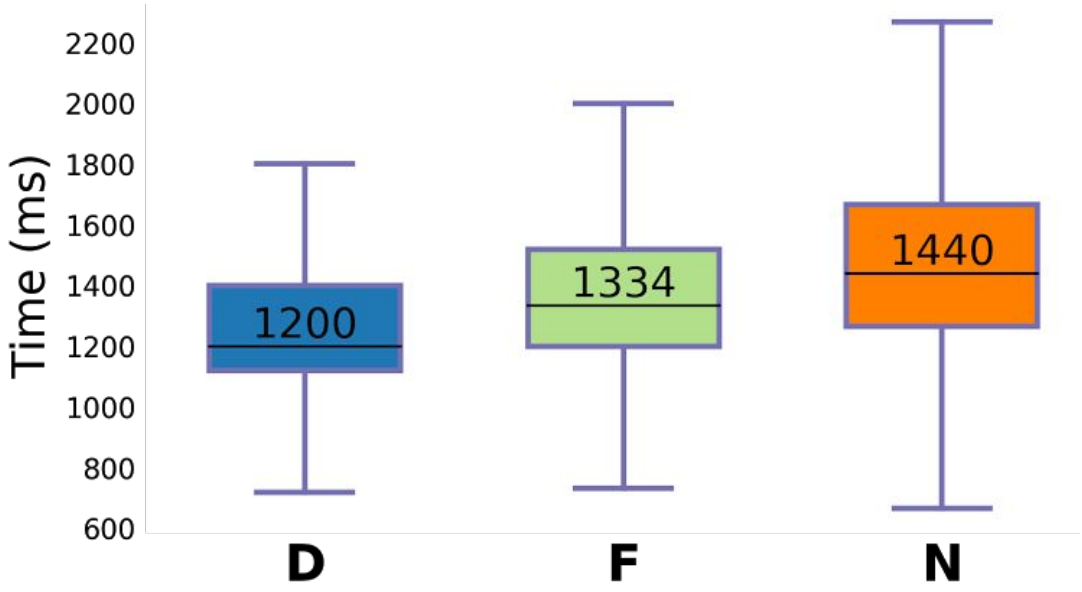

Figure 5.10: Return Pressure: We plot the return pressure (in ms) using box plots and mark the median over all matches for each player. Djokovic proves to be a better returner with higher return pressure.

5.8.2 Federer's backhand 2014-2017

One of the major factors discussed in Federer's win over Nadal in 2017 Australian Open Finals was his improved *backhand*. Federer credited his switch to a bigger racket frame that he started using in the year 2014 [92]. In the absence of fine-grained information about the *shots* at *forehand* or *backhand* classes, we utilize the notion of dominant positions. We try to analyze improvement in his *backhand* by measuring the number of times an opponent *serves* to Federer's *backhand position* against his *forehand position*.

We measure the difference between Federer's horizontal coordinates at the opponent's *serve* and at Federer's *serve-return*. If the difference is *positive*, we classify that the *return* is played from a *forehand position*. If the difference is negative, then it is played from a *backhand position*. The sign of the direction is dependent on which side of the player faces the camera, i.e. upper/lower half of court.

We can observe that the percentage of *serves* made to Federer's *backhand position* was higher earlier and has dropped significantly, indicating that the opponents have caught on that his *backhand* has improved. Further, we obtained the actual *shot* data for the *serve-returns* from Tennis Abstract [71] for the matches and plotted that trend along with our data in Figure 5.11 and observe that the trend is pretty similar in both cases.

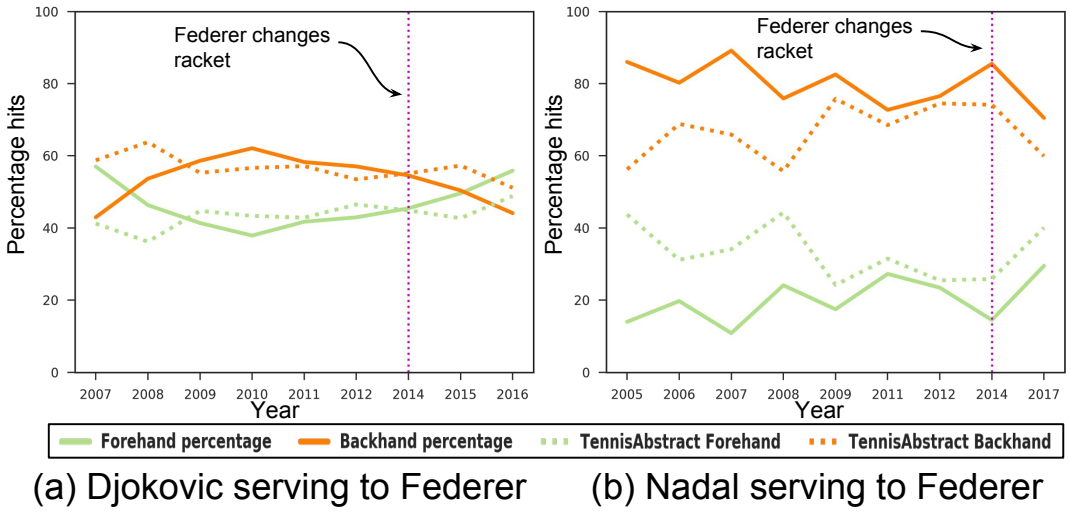

Figure 5.11: Percentage of serves to Federer's backhand position: (a) By Djokovic (b) By Nadal. Concurrently, we have plotted the percentage of actual backhand and forehand serve-returns played by Federer (from Tennis Abstract Data)

5.9 Discussions

We have demonstrated the effectiveness of computer vision methods to perform data mining of broadcast videos, we can characterize minute differences between playing styles and thus analyze both matches and players. In summary, we observe that players have distinctive styles of play. Players with strong *serves* (like Federer) try to play shorter *rallies* and thus prefer faster courts which promote such play. Similarly, strong *returners* (like Nadal) prefer courts which promote longer *rallies*. However, strong *returners* may have different characteristics (such as return pressure) depending on their court preference, as observed in disparity between Djokovic and Nadal.

Our approach is a an important step towards research in the direction of sports reasoning, planning and generating recommendations for sports. Such methods can provide corrective course of action and defensive strategy against the opponents. Lastly, computer vision based analytics methods are cost-effective. These can be extended to other sports and deployed to provide real-time insights for amateur players with minor modifications to the approach.

Chapter 6

Conclusions

The frameworks that we have presented in this thesis address the problem of sports video analytics and retrieval. Our methods operate on broadcast videos and do not require any specialized equipments or sensors. We have restricted our focus to tennis and badminton sports, however, the computer vision approaches can be applied to different sports with modifications. In Section 2.1 we discussed some of the other work in the space of computer vision and sports while in Section 2.2 we discussed some of the machine learning techniques that we put to use while building our frameworks.

Chapter 3 presents an automatic way to index broadcast videos of tennis matches. The method relies on extracting play segments by exploiting the property that the camera is behind the baseline while the sport is being played. Then the scorecard is extracted automatically by exploiting the fact that the scorecard is static in the video segment. The score are extracted using popular scene text spotting and recognition pipelines. Our major contribution is a scores refinement algorithm that takes in raw outputs of the scene text recognition algorithm, and then utilizes the scoring system modeled as an automaton to fix errors in the scores. Section 3.4 provides an evaluation of different parts of our method and demonstrate a simple application of automated event tagging.

In Chapter 4, we present an end-to-end framework for automatic analysis of broadcast badminton videos. We build our pipeline on off-the-shelf object detection [68], action recognition and segmentation [47] modules. Analytics for different sports rely on these modules making our pipeline generic for various sports, especially racket sports (tennis, badminton, table tennis, etc.). Although these modules are trained, fine-tuned, and used independently, we could compute various useful as well as easily understandable metrics, from each of these modules, for higher-level analytics. The metrics could be computed or used differently for different sports but the underlying modules rarely change. This is because broadcast videos of different sports share the similar challenges.

In Chapter 5, we have demonstrated the effectiveness our framework to perform data mining of broadcast videos. Through the mined data, we are able to characterize minute differences between playing styles of different players and thus analyze both matches and players. Our framework is cost-effective and can be extended to other sports and deployed to provide real-time insights for amateur players with minor modifications to the approach.

Related Publications

1. SmartTennisTV: Automatic indexing of tennis videos, Anurag Ghosh, C.V. Jawahar, National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2017 (**Best Paper Award**)

2. Towards structured analysis of broadcast badminton videos, Anurag Ghosh, Suriya Singh, C.V. Jawahar, Winter Conference on Applications of Computer Vision (WACV), 2018

3. Mining and Analyzing Tennis and Badminton Strategies from Broadcast Videos, Anurag Ghosh*, Rakesh Jasti*, Suriya Singh, C.V. Jawahar, 2018 (Under Preparation) (*Joint work)

Bibliography

[1] Dartfish: Sports performance analysis. http://dartfish.com/. 7, 25

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *TPAMI*, 2017. 32

[3] S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018. 13

[4] G. Bertasius, H. S. Park, X. Y. Stella, and J. Shi. Am i a baller? basketball performance assessment from first-person videos. In *Proc. ICCV*, 2017. 7, 9, 28, 42

[5] V. Bettadapura, C. Pantofaru, and I. Essa. Leveraging contextual cues for generating basketball highlights. In *Proc. ACMMM*, 2016. 9, 42

[6] B. L. Bhatnagar, S. Singh, C. Arora, and C. V. Jawahar. Unsupervised learning of deep feature representation for clustering egocentric actions. In *Proc. IJCAI*, 2017. 28

[7] C. Bialik and N. Silver. Tennis has a big three-and-a-half, 2014. https://fivethirtyeight.com/features/andy-murray-tennis-big-four/. 43

[8] A. Bialkowski, P. Lucey, P. Carr, Y. Yue, S. Sridharan, and I. Matthews. Large-scale analysis of soccer matches using spatiotemporal tracking data. In *Proc. ICDM*, 2014. 42

[9] J. Biswas, J. P. Mendoza, D. Zhu, B. Choi, S. Klee, and M. Veloso. Opponent-driven planning and execution for pass, attack, and defense in a multi-robot soccer team. In *Proc. AAMAS*, 2014. 40

[10] C. Calvo, A. Micarelli, and E. Sangineto. Automatic annotation of tennis video sequences. In *Joint Pattern Recognition Symposium*, 2002. 44

[11] S. Careelmont. Badminton shot classification in compressed video with baseline angled camera. 2013. 10

[12] D. Cervone, A. DAmour, L. Bornn, and K. Goldsberry. Pointwise: Predicting points and valuing decisions in real time with nba optical tracking data. In *Proc. MITSSAC*, 2014. 28

[13] J. Chen, H. M. Le, P. Carr, Y. Yue, and J. J. Little. Learning online smooth predictors for realtime camera planning using recurrent decision trees. In *Proc. CVPR*, 2016. 7

[14] J. Chen, L. Meng, and J. J. Little. Camera selection for broadcasting soccer games. In *Proc. WACV*, 2018. 7, 8

[15] S. Chen, Z. Feng, Q. Lu, B. Mahasseni, T. Fiez, A. Fern, and S. Todorovic. Play type recognition in real-world football video. In *Proc. WACV*, 2014. 8, 27

[16] W.-T. Chu and S. Situmeang. Badminton Video Analysis based on Spatiotemporal and Stroke Features. In *Proc. ICMR*, 2017. 7, 10, 25, 27, 33

[17] D. Connaghan, P. Kelly, and N. E. O'Connor. Game, shot and match: Event-based indexing of tennis. In *Proc. CBMI*, 2011. 9, 18

[18] F. Destelle, A. Ahmadi, K. Moran, N. E. O'Connor, N. Zioulis, A. Chatzitofis, D. Zarpalas, P. Daras, L. Unzueta, J. Goenetxea, et al. A multi-modal 3d capturing platform for learning and preservation of traditional sports and games. In *Proc. ACM Multimedia*, 2015. 40, 42
[19] T. Dierickx. Badminton game analysis from video sequences. 2014. 10
[20] T. D'Orazio, M. Leo, N. Mosca, P. Spagnolo, and P. L. Mazzeo. A semi-automatic system for ground truth generation of soccer video sequences. In *Proc. AVSS*. IEEE. 42
[21] L.-Y. Duan, M. Xu, T.-S. Chua, Q. Tian, and C.-S. Xu. A mid-level representation framework for semantic sports video analysis. In *Proc. ACM Multimedia*, 2003. 8, 40, 41
[22] A. Fathi and J. M. Rehg. Modeling actions through state changes. In *Proc. CVPR*, 2013. 28
[23] A. Fathi, X. Ren, and J. M. Rehg. Learning to recognize objects in egocentric activities. In *Proc. CVPR*, 2011. 29, 30
[24] P. Felsen, P. Agrawal, and J. Malik. What will happen next? forecasting player moves in sports videos. In *Proc. ICCV*, 2017. 7
[25] B. Ghanem, M. Kreidieh, M. Farra, and T. Zhang. Context-aware learning for automatic sports highlight recognition. In *Proc. ICPR*, 2012. 7, 9, 17, 25, 27
[26] S. Giancola, M. Amine, T. Dghaily, and B. Ghanem. Soccernet: A scalable dataset for action spotting in soccer videos. In *Proc. CVPR Workshops*, 2018. 8
[27] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *Proc. CVPR*, 2016. 12, 18, 20, 22, 23
[28] A. Hanjalic. Generic approach to highlights extraction from a sport video. In *Proc. ICIP*, 2003. 9, 17, 27
[29] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. 44
[30] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 25, 42
[31] J. Heikkilä and O. Silvén. A real-time system for monitoring of cyclists and pedestrians. *Proc. IVC*, 2004. 31, 45
[32] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In *Proc. ECCV*, 2016. 28
[33] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 1997. 28
[34] N. Homayounfar, S. Fidler, and R. Urtasun. Sports field localization via deep structured models. In *Proc. CVPR*, 2017. 8
[35] Y.-P. Huang, C.-L. Chiou, and F. E. Sandnes. An intelligent strategy for the automatic detection of highlights in tennis video recordings. *Expert Systems with Applications*, 36(6):9907–9918, 2009. 9, 17
[36] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori. A hierarchical deep temporal model for group activity recognition. In *CVPR*, 2016. 7, 8, 28, 30, 42
[37] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014. 12, 18, 20
[38] H. Jiang, Y. Lu, and J. Xue. Automatic soccer video event detection based on a deep neural network combined cnn and rnn. In *Proc. ICTAI*, 2016. 8
[39] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proc. BMVC*, 2010. 8

[40] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proc. CVPR*, 2014. 8, 30, 42

[41] M. H. Kolekar, K. Palaniappan, and S. Sengupta. Semantic event detection and classification in cricket video sequence. In *Proc. ICVGIP*, 2008. 8

[42] M. H. Kolekar and S. Sengupta. Bayesian network-based customized highlight generation for broadcast soccer videos. *IEEE Transactions on Broadcasting*, 2015. 18

[43] S. Kovalchik. Stats on the T, 2017. Retrieved from http://on-the-t.com/. 51, 53

[44] S. Kovalchik, M. K. Bane, and M. Reid. Getting to the top: an analysis of 25 years of career rankings trajectories for professional womens tennis. *Journal of sports sciences*, 2017. 41

[45] H. M. Le, P. Carr, Y. Yue, and P. Lucey. Data-driven ghosting using deep imitation learning. 8, 42

[46] C. Lea, A. Reiter, R. Vidal, and G. D. Hager. Segmental spatiotemporal cnns for fine-grained action segmentation. In *Proc. ECCV Workshops*, 2016. 28

[47] C. Lea, R. Vidal, A. Reiter, and G. D. Hager. Temporal convolutional networks: A unified approach to action segmentation. In *Proc. ECCV Workshops*, 2016. 13, 25, 28, 32, 33, 34, 42, 45, 46, 56

[48] S. Liao, Y. Wang, and X. Y. Research on scoreboard detection and localization in basketball video. 2015. 18, 20

[49] C. Liu, Q. Huang, S. Jiang, L. Xing, Q. Ye, and W. Gao. A framework for flexible summarization of racquet sports video using multiple modalities. *CVIU*, 2009. 9, 18

[50] W.-L. Lu, J.-A. Ting, J. J. Little, and K. P. Murphy. Learning to track and identify players from broadcast sports videos. *TPAMI*, 2013. 8

[51] P. Lucey, A. Bialkowski, P. Carr, Y. Yue, and I. Matthews. How to get an open shot: Analyzing team movement in basketball using tracking data. 9, 42

[52] P. Lucey, D. Oliver, P. Carr, J. Roth, and I. Matthews. Assessing team strategy using spatiotemporal data. In *Proc. KDD*, 2013. 42, 48

[53] A. Maksai, X. Wang, and P. Fua. What players do with the ball: A physically constrained interaction modeling. 2016. 8

[54] R. Martín and J. M. Martínez. Automatic players detection and tracking in multi-camera tennis videos. In *Human Behavior Understanding in Networked Sensing*, pages 191–209. Springer, 2014. 8, 42

[55] J. P. Mendoza, J. Biswas, P. Cooksey, R. Wang, S. Klee, D. Zhu, and M. Veloso. Selectively reactive coordination for a team of robot soccer champions. In *Proc. AAAI*, 2016. 40

[56] M. Mentzelopoulos, A. Psarrou, A. Angelopoulou, and J. García-Rodríguez. Active foreground region extraction and tracking for sports video annotation. *Neural processing letters*, 2013. 9, 11, 17, 28

[57] M. Merler, D. Joshi, Q.-B. Nguyen, S. Hammer, J. Kent, J. R. Smith, and R. S. Feris. Automatic curation of golf highlights using multimodal excitement features. In *Proc. CVPR Workshops*, 2017. 42

[58] G. Miao, G. Zhu, S. Jiang, Q. Huang, C. Xu, and W. Gao. A real-time score detection and recognition approach for broadcast basketball video. In *Proc. ICME*, 2007. 18, 19

[59] H. Miyamori and S.-I. Iisaku. Video annotation for content-based retrieval using human behavior analysis and domain knowledge. In *Proc. FG*, 2000. 9, 17

[60] M. Mlakar and M. Luštrek. Analyzing tennis game through sensor data with machine learning and multi-objective optimization. In *Proc. UbiComp*, 2017. 7, 9, 25, 28, 40

[61] D. Monaghan, F. Honohan, A. Ahmadi, T. McDaniel, R. Tadayon, A. Karpur, K. Morran, N. E. O'Connor, and S. Panchanathan. A multimodal gamified platform for real-time user feedback in sports performance. In *Proc. ACM Multimedia*, 2016. 40, 42

[62] S. V. Mora and W. J. Knottenbelt. Spatio-temporal analysis of tennis matches. In *KDD Workshop on Large-Scale Sports Analytics*, 2016. 42, 49

[63] D. T. Nguyen, W. Li, and P. O. Ogunbona. Human detection from images and videos: A survey. *Pattern Recognition*, 2016. 11, 28

[64] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016. 33

[65] M. Phomsoupha and G. Laffaye. The science of badminton: Game characteristics, anthropometry, physiology, visual fitness and biomechanics. In *Sports Medicine*, 2015. 36

[66] V. Ramanathan, J. Huang, S. Abu-El-Haija, A. Gorban, K. Murphy, and L. Fei-Fei. Detecting events and key actors in multi-person videos. In *Proc. CVPR*, 2016. 7, 8, 9, 28, 30, 42

[67] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proc. CVPR*, 2016. 12

[68] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proc. NIPS*, 2015. 10, 11, 25, 31, 42, 44, 56

[69] V. Renò, N. Mosca, M. Nitti, T. DOrazio, C. Guaragnella, D. Campagnoli, A. Prati, and E. Stella. A technology platform for automatic high-level tennis game analysis. *Proc. CVIU*, 2017. 42

[70] V. Ren, N. Mosca, M. Nitti, T. DOrazio, C. Guaragnella, D. Campagnoli, A. Prati, and E. Stella. A technology platform for automatic high-level tennis game analysis. *CVIU*, 2017. 7, 9, 25, 27

[71] J. Sackmann. Tennis abstract, 2017. Retrieved from http://www.tennisabstract.com/. 41, 51, 54

[72] K. P. Sankar, S. Pandey, and C. Jawahar. Text driven temporal segmentation of cricket videos. In *Proc. ICVGIP*. 2006. 8

[73] L. Sha, P. Lucey, S. Sridharan, S. Morgan, and D. Pease. Understanding and analyzing a large collection of archived swimming videos. In *Proc. WACV*, 2014. 27, 40, 42

[74] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proc. CVPR Workshops*, 2014. 10

[75] R. A. Sharma, B. Bhat, V. Gandhi, and C. Jawahar. Automated top view registration of broadcast football videos. 2018. 7, 8

[76] R. A. Sharma, K. P. Sankar, and C. Jawahar. Fine-grain annotation of cricket videos. In *Proc. ACPR*, 2015. 7, 8

[77] B. Shi, X. Bai, and C. Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *TPAMI*, 2016. 12, 18, 20

[78] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *Proc. ICCV*, 2011. 11, 28

[79] G. A. Sigurdsson, O. Russakovsky, and A. Gupta. What actions are needed for understanding human actions in videos? In *Proc. ICCV*, 2017. 5

[80] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 2016. 40

[81] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Proc. NIPS*, 2014. 10, 13, 28, 34

[82] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2014. 11, 25

[83] S. Singh, C. Arora, and C. Jawahar. First person action recognition using deep learned descriptors. In *Proc. CVPR*, 2016. 25, 28, 29

[84] S. Singh, C. Arora, and C. V. Jawahar. Trajectory aligned features for first person action recognition. *Pattern Recognition*, 2017. 28, 30

[85] R. Smith. An overview of the tesseract ocr engine. In *Proc. ICDAR*, 2007. 18, 20

[86] K. Soomro, A. Roshan Zamir, and M. Shah. UCF101: a dataset of 101 human actions classes from videos in the wild. 2012. 10, 30

[87] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 42

[88] S. Stein and S. J. McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proc. UbiComp*, 2013. 29, 30

[89] G. Sudhir, J. C. M. Lee, and A. K. Jain. Automatic classification of tennis video for high-level content-based retrieval. In *Proc. CAIVD*, 1998. 44

[90] M. Sukhwani and C. Jawahar. Tennisvid2text: Fine-grained descriptions for domain specific videos. In *Proc. BMVC*, 2015. 7, 9, 17, 27, 42

[91] M. Sukhwani and C. Jawahar. Frame level annotations for tennis videos. In *Proc. ICPR*, 2016. 7, 9, 18, 28

[92] K. Tandon. Roger Federer credits switch to bigger racquet for improved backhand, 2017. http://www.tennis.com/pro-game/2017/03/roger-federer-racquet-change-backhand-rafael-nadal-indian-wells/64840/. 54

[93] K. Teachabarikiti, T. H. Chalidabhongse, and A. Thammano. Players tracking and ball detection for an automatic tennis video annotation. In *Proc. ICARCV*, 2010. 42

[94] M. R. Tora, J. Chen, and J. J. Little. Classification of puck possession events in ice hockey. In *Proc. CVPR Workshops*, 2017. 7, 8

[95] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proc. ICCV*, 2015. 28

[96] K.-C. Wang and R. Zemel. Classifying nba offensive plays using neural networks. In *Proc. MITSSAC*, 2016. 28, 40

[97] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proc. CVPR*, 2015. 25, 28

[98] X. Wei, P. Lucey, S. Morgan, P. Carr, M. Reid, and S. Sridharan. Predicting serves in tennis using style priors. In *Proc. KDD*, 2015. 9, 42

[99] X. Wei, P. Lucey, S. Morgan, M. Reid, and S. Sridharan. the thin edge of the wedge: Accurately predicting shot outcomes in tennis using style and context priors. In *Proc. MITSSAC*, 2016. 9, 48

[100] X. Wei, P. Lucey, S. Morgan, and S. Sridharan. 'sweet-spot': Using spatiotemporal data to discover and predict shots in tennis. In *Proc. MITSSAC*, 2013. 9, 40, 42, 49

[101] C. J. Wu. On the convergence properties of the em algorithm. *The Annals of statistics*, 1983. 15

[102] C. Xu, J. Wang, H. Lu, and Y. Zhang. A novel framework for semantic annotation and personalized retrieval of sports video. *Transactions on Multimedia*, 2008. 9, 17

[103] F. Yan, W. Christmas, and J. Kittler. All pairs shortest path formulation for multiple object tracking with application to tennis video analysis. In *Proc. BMVC*, 2007. 8, 9, 17

[104] F. Yan, J. Kittler, D. Windridge, W. Christmas, K. Mikolajczyk, S. Cox, and Q. Huang. Automatic annotation of tennis games: An integration of audio, vision, and learning. *IVC*, 2014. 7, 9, 11, 17, 25, 28, 42

[105] F. Yoshikawa, T. Kobayashi, K. Watanabe, and N. Otsu. Automated service scene detection for badminton game analysis using CHLAC and MRA. 10, 27, 35

[106] Y. Zhang, X. Zhang, C. Xu, and H. Lu. Personalized retrieval of sports video. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*, 2007. 17

[107] X. Zhou, L. Xie, Q. Huang, S. J. Cox, and Y. Zhang. Tennis ball tracking using a two-layered data association approach. *IEEE Transactions on Multimedia*, 2015. 9, 17

[108] G. Zhu, C. Xu, Q. Huang, W. Gao, and L. Xing. Player action recognition in broadcast tennis video with applications to semantic analysis of sports game. In *Proc. ACM Multimedia*, 2006. 42