NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

ANDRÉ FARINHA DE CARVALHO

Master/BSc in Electrical and Computer Engineering

AUTOMATIC EVENT DETECTION FOR TENNIS SPORT

THIS THESIS WILL FOCUS ON THE DEVELOPMENT OF A SOLUTION FOR AUTOMATIC EVENTS DETECTION FOR TENNIS MATCHES.

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon
November, 2023

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF ELECTRICAL
AND COMPUTER ENGINEERING

AUTOMATIC EVENT DETECTION FOR TENNIS SPORT

THIS THESIS WILL FOCUS ON THE DEVELOPMENT OF A SOLUTION FOR AUTOMATIC EVENTS DETECTION FOR TENNIS MATCHES.

ANDRÉ FARINHA DE CARVALHO

Master/BSc in Electrical and Computer Engineering

Adviser: Daniel de Matos Silvestre
Assistant Professor, FCT/UNL

Co-adviser: Xavier Marques Frazão
Software Engineer, Six Floor Solutions

Examination Committee

Chair: João Carlos da Palma Goes
Full Professor, FCT/UNL

Rapporteur: João Paulo Salgado Arriscado Costeira
Associate Professor, IST/UL

Adviser: Daniel de Matos Silvestre
Assistant Professor, FCT/UNL

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon
November, 2023

Automatic Event Detection for Tennis Sport

This thesis will focus on the development of a solution for automatic events detection for Tennis matches.

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v6.10.10) [24].

To all my family.

Acknowledgements

First, I want to start by expressing my appreciation for my advisor Professor Daniel Silvestre. His valuable support, knowledge, and our conversations have been crucial in shaping and structuring my thesis throughout this journey.

I would also like to extend my gratitude to the entire Six Floor Solutions team for their generous support since the first day I contacted them. The availability of essential resources and extensive knowledge, especially from software engineer Xavier Frazão, greatly enriched my research and thesis development.

To my friends and colleagues, particularly those from my course, Diogo Oliveira, João Soares, Cezar Petrea, Francisco Silva, Afonso Tavares, João Ferreira and João Bento, I offer my sincere thanks. Your friendship, shared experiences, and useful conversations have been a source of motivation and inspiration.

A special appreciation goes to my girlfriend, Beatriz Salgueiro, for her endless and strong support every single day. Her huge encouragement and understanding have been a constant source of strength and inspiration throughout all this thesis elaboration.

I also need to express a huge thanks to all my family for their love, encouragement, and constant support in every single moment since the beginning. I want to thank you all for the education you gave me and for the person you helped me to become.

This work was partially supported by the Portuguese Fundação para a Ciência e a Tecnologia (FCT) through project FirePuma (https://doi.org/10.54499/PCIF/MPG/0156/2019), through Institute for Systems and Robotics (ISR), under Laboratory for Robotics and Engineering Systems (LARSyS) project UIDB/50009/2020, and through COPELABS, University Lusófona project UIDB/04111/2020.

"Everything is theoretically impossible until it is done." (Robert A. Heinlein)

Abstract

The abundance of multimedia information presents a challenge in filtering out relevant content [22]. This master's dissertation proposes a solution for real-time automatic event detection in tennis TV broadcasts.

After analyzing the current state of the art in image processing and event detection for different sports, we introduce and describe approaches based on computer vision techniques to detect, classify, and label different play moments during tennis matches. We compare the performance of the developed approach to the manual event segmentation by humans and demonstrate the accuracy and quality of our work using the Six Floor Solutions prototype.

Our research aims to provide a reliable and efficient solution for detecting the events in tennis matches and inspire similar approaches in other sports or TV shows.

Keywords: Computer vision, Tennis matches, Event detection

Contents

List of Figures

List of Tables

Acronyms

ACC Accuracy

BPM Beats per minute
BS Background Subtraction

DCR Dominant Color Ratio
DPM Deformable Part Model

FN False Negative
FP False Positive
fps Frames per second

HLS High-level semantics
HMM Hidden Markov Model
HSV Hue, Saturation, Value

LLF Low-level features
LoG Laplacian of Gaussian

MFCC Mel-Frequency Cepstral Coefficient
MLR Mid-level representation
MSE Mean Squared Error

OCR Optical Character Recognition
OT Overlaid-text

PPV Precision
PSNR Peak signal-to-noise ratio

RGB Red, Green, Blue
ROI Region of interest

SIFT Scale-invariant feature transform
SSIM Structural Similarity Index
SVM Support Vector Machine

TN True Negative
TP True Positive

1 Introduction

The tennis sport has captivated audiences for decades. The fast-paced and high-intensity nature of the game offers a unique blend of strategy, skill, and athleticism. "On average, best-of-3 tennis matches last about 90 minutes, while best-of-5 matches last 2 hours and 45 minutes" [17]. During a match, there are a lot of different events, such as aces, faults, and match points. One important element in broadcasting and summarizing the sport is the ability to accurately and quickly detect all those events. In this way, automatic event detection is beneficial by providing real-time, detailed information about a match. This technology is a useful tool for coaches and players to analyze the game tactically at a high level, for fans to gain insight into the performance of their favorite players and for sports TV networks to be able to add more information to the broadcast in real-time. Automatic event detection can also allow a faster and more objective analysis of the match and be a valuable resource for statisticians and betting companies to update information on the fly.

For a better outcome, the detection of tennis events automatically can combine various techniques and technologies, such as machine learning, and computer vision. Together they can be used to extract relevant information from video footage of a match, such as the location of the players, the score and the ball. Machine learning algorithms can be used to analyze this data and make predictions about the match's outcome. Computer vision is a useful tool to track the movement of the players and the ball in real-time.

One of the challenges associated with automatizing event identification, with artificial intelligence, is the need for large amounts of data to train the machine learning algorithms to produce accurate results [44]. Traditional image processing can have difficulties in the detection of court lines due to variable lighting conditions. Furthermore, different camera angles can make it difficult to track the ball and the players accurately. Even with these challenges, automatic event detection in tennis presents a promising proposal.

In this document, we will examine and take advantage of the techniques and technologies used for automatic event detection to develop a robust algorithm that receives a real-time TV stream as input and automatically segments and detects the main events in a tennis match. These events are divided into points, games, and sets. We can identify the exact moment when a tennis point began, its duration, the number of ball touches,

and how the point was won. As well as identifying faults committed by the serving player before a particular point, we can identify whether that point had a repetition or a "Hawk-Eye" associated. In addition, we provide the serving player, the winner of the point, the result at the end and the beginning of the point, and a thumbnail of reference for each point.

1.1 Motivation

Tennis is widely recognized as a sport with significant popularity among individuals worldwide, as documented by various sources [42], [7]. The Tennis Industry Association reports that in 2018, the economy generated by tennis was around $6.19 billion, an increase of $150 million from the previous year [5]. With the growing popularity of live streaming and online platforms, it is predicted that these numbers will continue to increase, as it was verified for different international tournaments [40].

The use of an automatic event detection system in tennis can improve the speed of highlights generation, enrich the commentary and analysis during online broadcasting and provide valuable insights into the match. The development of such a system has multiple benefits, including the ability to leverage technology to enhance performance, the viewer experience, and promote the growth of the sport. Additionally, the algorithm created for this purpose can also inspire similar approaches in other sports and other fields.

1.2 Video Analysis and Image Processing

This section provides an overview of fundamental concepts related to video analysis and the frames within it, image processing tools, and techniques for event detection. Furthermore, we will discuss the challenges and limitations of current techniques and possible approaches to address these issues.

In this dissertation, we will consider a video $\mathcal{V}$ being described as a sequence of frames $F(k)$ numbered using the index k, i.e., $\mathcal{V} = \{F(k) : k \in \mathbb{N}\}$. The effect of a continuous video is achieved through the rapid succession of frames. A video is formed by a certain number of Frames per second (fps). The higher the fps, the smoother the motion will appear in the video[45]. An image, on its turn, is composed of a set of tiny squares or colored dots called pixels, short for "picture element".

In digital imaging, pixels are the elementary building blocks, whether on a computer screen, television or digital camera. The resolution and detail of an image are directly proportional to the number of pixels it contains. Image resolution is typically described as PPI, which refers to the number of pixels per inch of an image. For example, an image with a resolution of 600 ppi will contain 600 pixels in each inch of the image [11]. Therefore, frames $F(k)$ can be represented as matrices where each entry has the numeric value associated with a color, meaning that $F(k) \in \mathbb{R}^{w \times h}$ for a frame of width given by w

and height h. The value stored in $F_{ij}(k)$ is the correspondent pixel intensity in some color scheme. If using the RGB color space, there are going to be stored in each pixel a tuple of 3 decimal values ranging from 0 to 255 (or 0 to FF in hexadecimal representation) for each of the three colors. To illustrate this concept, Figure 1.1, provides a visual representation of how a single pixel is composed in an RGB image with a display resolution of 32×32 pixels.

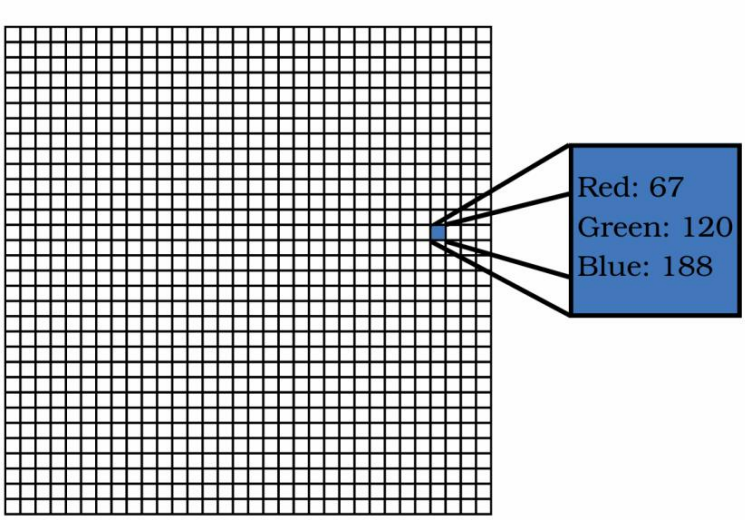


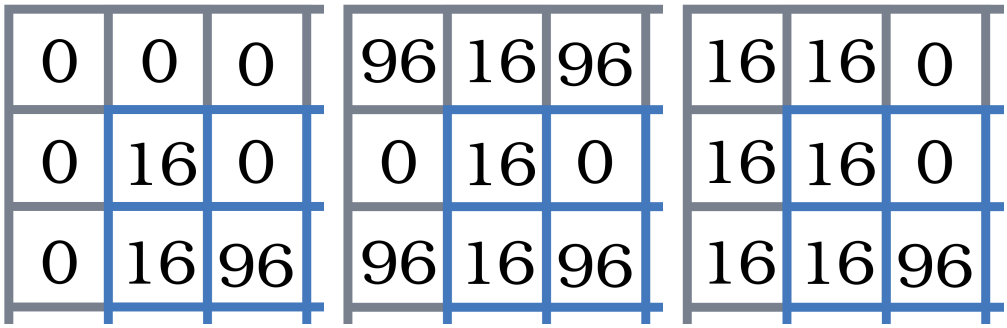
Figure 1.1: Composition of the decimal value for each channel of a Pixel from a Digital Image in RGB with 32×32 resolution

As depicted in the figure above, an RGB image with a single colored pixel is presented. The intensity values of each channel used to obtain the colored dot are displayed in decimal values.

In a traditional image processing problem, the feature extraction problem can be defined as the identification of particular structures in the pixels of each $F(k)$ or across multiple values of k through the use of filters. These correspond to mathematical operations that are utilized to alter, improve, or extract information from an image. These filters operate by processing each pixel by a set of defined rules. The application of a filter to an image is typically achieved by convolution with a kernel, which is a small matrix of numbers that represents the filter in question. To apply a filter to the borders of an image, it is necessary to first "extend" it by adding additional pixels to its margins.

To achieve image border extension, various techniques can be employed. One such technique is the application of a zero padding method [28], which involves adding one or more rows and columns filled with zeroes to the margins of the original image. Another technique is reflection padding[28], in which the margins are mirrored to the adjacent pixels, resulting in outer rows and columns that reflect the pixels within the image. A third technique is replication padding[28], which simply duplicates the pixels present in the margins of the image. A visual representation of the practical application of each technique can be observed in Figure 1.2. The figure illustrates the implementation of zero

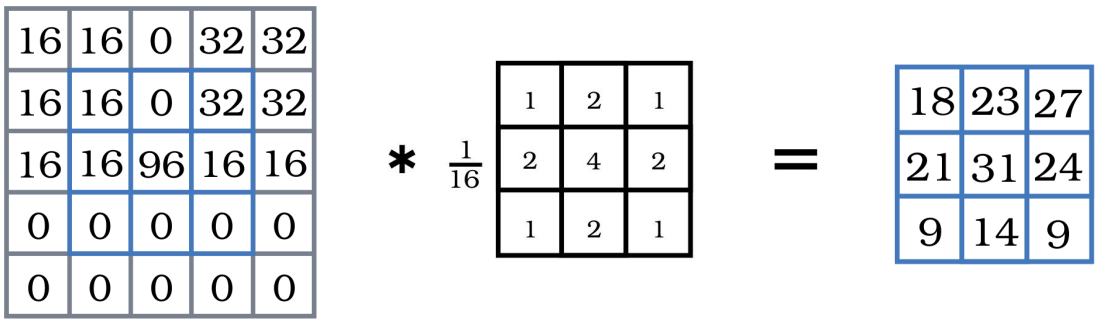
padding, reflection padding, and replication padding through subfigures 1.2(a), 1.2(b), and 1.2(c), respectively.



(a) Zero padding (b) Reflection padding (c) Replication padding

Figure 1.2: Various types of padding around the border of a grayscale image [28].

After extending the margins of images to apply filters to the borders, we shall now proceed to examine a practical example of filter application. For instance, to apply a Gaussian filter to a frame, one must convolve the image with a kernel that corresponds to the Gaussian function. This is a low-pass filter, which means it can be used to reduce noise and blur an image. A visual representation of the application of a 3×3 Gaussian filter kernel to a grayscale image with a resolution of 3×3 can be observed in Figure 1.3.



(a) (b) (c)

Figure 1.3: Illustration of applying a 3×3 Gaussian filter kernel to a 3×3 resolution grayscale image. The convolution process is represented, where each pixel of the image is convoluted with the corresponding kernel values to obtain the final output image. 1.3(a) 3×3 grayscale image using replication padding as the border extension technique. 1.3(b) 3×3 Gaussian kernel. 1.3(c) 3×3 output image with new grayscale values.

As illustrated in Figure 1.3, the process of applying a filter to an image involves using a sliding window that runs the entire image and convolves each pixel and those surrounding it with the filtering kernel. The convolution operation illustrated in the figure above can be represented mathematically as:

$$Y(i,j) = F * G = \sum_{n=-\lfloor \frac{N}{2} \rfloor}^{\lfloor \frac{N}{2} \rfloor} \sum_{m=-\lfloor \frac{M}{2} \rfloor}^{\lfloor \frac{M}{2} \rfloor} F(i-m, j-n) \cdot G(m,n) \tag{1.1}$$

In (1.1), the output image with dimensions $w \times h$ of the convolution at pixel location (i, j) is represented by $Y(I, j)$. The input frame is represented by F and the kernel of the applied filter, with dimensions $M \times N$, is represented by G. To obtain the output value of a particular pixel, the kernel is positioned centred on that pixel. Then, each image pixel is multiplied by its corresponding value within the kernel. The resulting values are then added together to yield the final output value of the central pixel.

Having gained an understanding of the composition of videos, the structure of frames, and techniques for accessing, manipulating and obtaining values from images. The following section presents an overview of several solutions present in the literature on the topic of sports video analysis.

1.3 State-of-the-Art

Over the past years, significant progress has been made in the field of automated event detection and content analysis for sports videos [15],[41],[13]. Advanced software capable of precisely identifying and categorizing various events within a match can provide valuable insights into the sport and assist in evaluating players and teams.

In this chapter, we provide an overview of the current state-of-the-art in tennis event detection and classification. We outline the various events that can occur within a tennis match and the challenges related to their detection and classification. We also review the current approaches utilized for event detection and classification, including both traditional methods and more recent techniques such as the ones pertaining to machine learning. Additionally, we examine some of the advancements in vision systems techniques that can help in the envisioned solution.

1.3.1 Sports video analysis

The analysis of a sports video can be done in several ways. Many of them were implemented to produce an automatic or semi-automatic solution. The goal of sports video analysis is focused on extracting the most valuable information possible at each moment. The elements through which it is feasible to infer this knowledge are given the name *features*. The features that allow extracting direct information from the images or the audio components, such as the color of a certain element or a noise with a certain frequency, are called Low-level features (LLF). However, these LLF by themselves cannot provide direct high-level conclusions from the games.

The labelling of high-level information in sports video follows some steps with the first being identifying low-level audiovisual and textual features present in the frames. A sequence of features can then be grouped to signal a player movement or kick strength,

the identification of a keyword or other textual components. However, these models of how LLF can construct higher contextual information are still insufficient to High-level semantics (HLS) from the analysis of the game. The high-level semantics can be used to support various applications such as tactic analysis, net approaches and even play highlights extraction [10]. In Figure 1.4, a scheme is presented to provide a better understanding of the analysis process and the different levels of information in a sports video.

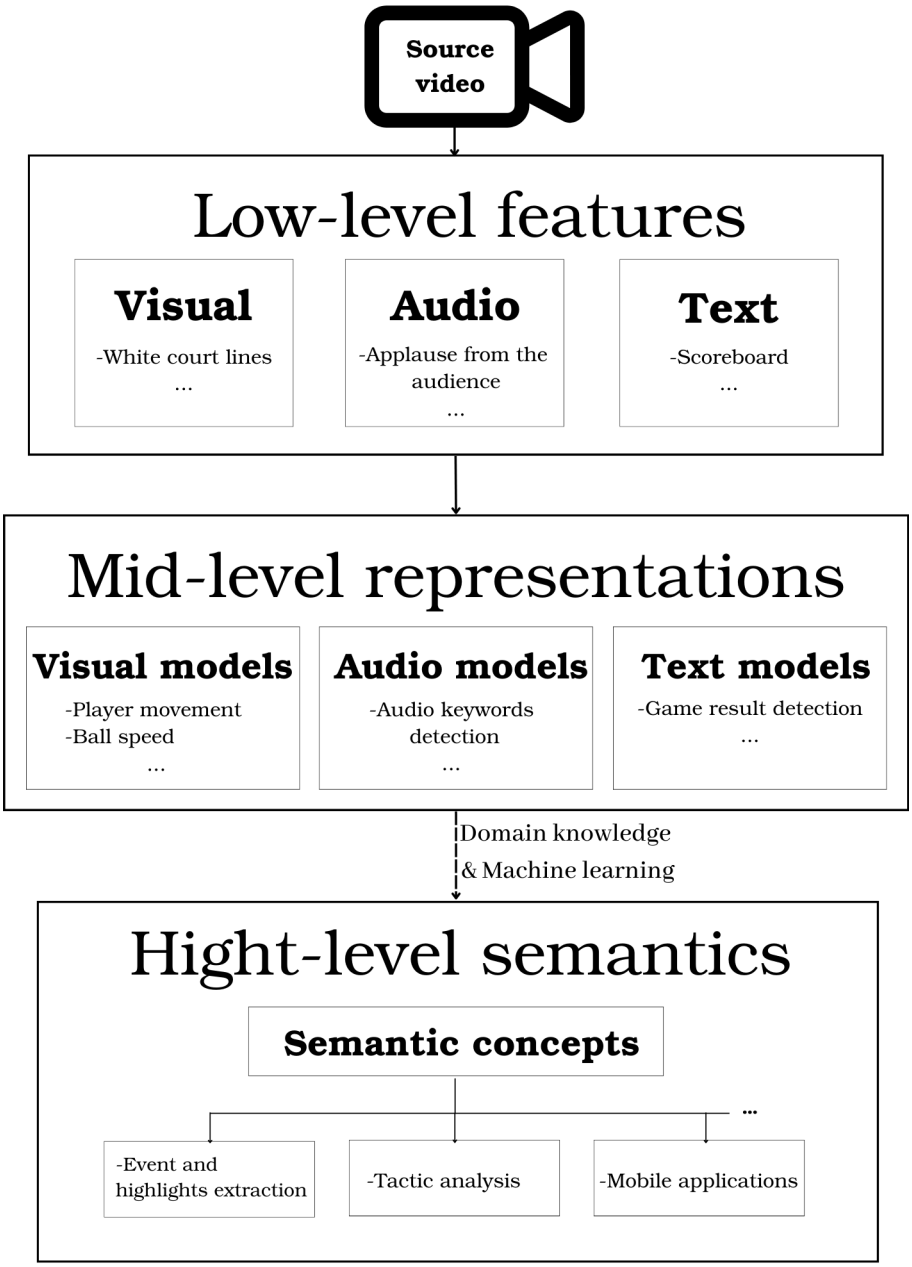


Figure 1.4: Relationship between audiovisual feature levels in sports videos [10]

The features present in sports videos are the basis for constructing event detection models, as shown in Figure 1.4. The scheme presents the three different levels of features

with some examples and the correspondent relationship between them. The following section provides an overview of the predominant features observed in tennis videos, as well as the methods used to extract them.

1.3.2 Characterizing Tennis Match Videos using Feature Analysis

Since features are distinctive patterns within an image, it is important to explore in more detail the most relevant such as edges, corners, and specific color patches. An edge is an area in an image where pixel intensity differs significantly, such as the boundary between an object and its background. There are several algorithms that can be used for edge detection, such as the Sobel operator [20], the Canny edge detector [3], and the Laplacian operator [43].

Corners are points in an image where two edges intersect at a sharp angle. Due to its attribute, this feature could be useful for detecting tennis court endings. There are some known algorithms specialized in corner detection such as the Harris corner detector [9] and the Shi-Tomasi corner detector [38].

The color of specific parts of the image is also a valuable piece of information in image analysis. In the specific case of tennis, the lines on the court are always white. color can be represented in different color spaces such as RGB, Hue, Saturation, Value (HSV) or even in L*a*b*. These types of features can be extracted using techniques such as color histograms and moments.

The various features outlined above can be extracted from an image using a variety of image processing techniques, such as filtering, thresholding, and detection algorithms. Then, they can be utilized for multiple purposes, such as image segmentation, object recognition, and classification. The next section presents some of the resources that have been previously employed to identify and classify the different events within the game.

1.3.3 Visual Feature Analysis in Sports Match Videos

Visual LLFs can be extracted from pixel intensity values, edges, color histograms and some other elements. These features are characterized by being easy to define, and extract and have a weak semantic meaning [1]. However, LLFs alone are not enough to develop a complex project like automatic event detection.

Certain LLFs combined can lead to Mid-level representation (MLR). In some previous research done on event detection for other sports games, it was concluded that MLRs are related to standard camera shooting style transmission practices and also to domain-specific concepts such as the location of the diverse elements on the playing field [15]. The visual elements in a broadcast sports video can be enough to detect and classify events.

1.3.3.1 Play Detection and Tennis Court Segmentation

A first remark regarding tennis streams is that the action can be broken down into distinct parts: when the camera is fixed, parallel to the net, and filming the entire court, these are known as court view shots; when pausing between match moments, these are typically non-court view shots.

In [41], the authors describe a method for splitting tennis videos into individual shots. They use a technique based on histogram differences to separate the video into individual planes. The descriptor called Dominant Color Ratio (DCR) is used to identify court view shots, which contain a large number of the desired pixels by measuring the relative proportion of different colors in the current frame. This descriptor analyses the color composition of a given frame by identifying the most dominant colors and compares them with the histogram of the court view. In these court view shots, the authors use line detection and camera calibration techniques to locate the court position. Afterwards, the players are localized on the tennis court, which enables detecting approaches to the net, different player movements and classifying each play into categories.

Another approach to detecting and segmenting the court can be found in [13]. The algorithm follows the same principles as the approach mentioned above. In the paper, two different approaches to detect the court view shots are presented based on reference histogram patterns for the three different courts of the analysed tournaments (US Open, Roland Garros and Wimbledon). On the first method, which ends to be the chosen one with an accuracy of around 98%, the author starts by calculating, for all the RGB components, the correlation between the histogram of the current frame to the histogram pattern associated with the correspondent tournament for the corresponding RGB channel using:

$$C = CFH * RHP \tag{1.2}$$

where C denotes the correlation (with symbol $*$) between the two histograms CFH associated with the current frame and RHP respecting the reference pattern for the calculated RGB channel. The non-court view frames can be discarded if at least one of the three obtained correlation values for each channel is above a given threshold.

The other presented method results from the calculation of the absolute value of the subtraction between the histogram of the current frame and the pattern associated with the reference histogram of that specific game. The average of the histogram of each component is calculated and compared to a reference limit. In the case at least one of the three mean values is above the threshold, that frame must be discarded as a court view frame. Nonetheless, this method is sensitive to changes in lighting, which may result in incorrect detection during cloudy weather days.

In [13] the author also presents an approach to detect and segment the court lines. The algorithm starts by converting the original RGB frame to grayscale and then a binarization process that converts the grayscale values into black or white pixels. This process uses

the Otsu method [33] to define thresholds that mark the distinction between the binary choice. Otsu's thresholding technique can be calculated using an algorithm for minimizing the intra-class variance by researching for a threshold value that can be mathematically expressed as:

$$
\begin{aligned}
\omega_0(t) &= \sum_{i=1}^{T} p(i) \\
\omega_1(t) &= 1 - \omega_0(t) \\
\mu_0(t) &= \frac{\sum_{i=1}^{T} i \cdot p(i)}{\omega_0(t)} \\
\mu_1(t) &= \frac{\sum_{i=t+1}^{T} i \cdot p(i)}{\omega_1(t)} \\
\sigma^2(t) &= \omega_0(t) \cdot \omega_1(t) \cdot (\mu_0(t) - \mu_1(t))^2 \\
t^* &= \underset{t \in [0,T-1]}{\arg\max}\, \sigma^2(t)
\end{aligned}
$$

Where $p(i)$ is the probability of a pixel having intensity value i, T is the total number of possible intensity values, $\omega_0(t)$ and $\omega_1(t)$ are the class probabilities for the foreground and background respectively, $\mu_0(t)$ and $\mu_1(t)$ are the class means for the foreground and background respectively, and $\sigma^2(t)$ is the intra-class variance. The optimal threshold value, t^*, is the value that maximizes $\sigma^2(t)$.

However, ground color changes depending on the tournament. The US Open is played on a blue acrylic hard court whereas Wimbledon Championship is played on grass and Roland Garros on red clay ground. The author adopts a percentage threshold for the US Open and Roland Garros to deal with the differences. In contrast, due to the degradation of grass on the playing surface due to usage, the court line detection using a single standard threshold value for every part of the image becomes a more significant challenge when attempting to identify every line at Wimbledon. To overcome this issue, the author presents the solution of dividing the frame into four separate zones, utilizing three imaginary horizontal lines. Then, a different percentage level is defined for each zone in order to achieve a final image with the white lines of the court as prominent as possible. The next step is to apply linear filters to keep the lines and remove the white pixels that do not compose the court. The proposal in [13] is to use Roberts for the US Open and Canny for the other two tournaments.

The Roberts filter is an operator used in computer vision for edge detection. The filter consists in a differential operator which aims to approximate the gradient of an image through discrete differentiation. This means replacing the value of the central pixel with the sum of the squares of the differences between diagonally adjacent pixels. In the end, the function returns edges at the points where the gradient of the image is the highest.

The Canny filter is a multi-stage algorithm that is used to detect edges in an image. The underlying principle of this algorithm is similar to that of the Roberts filter, which

also aims to locate the local maxima of the gradient to identify edges. The Canny filter process begins by applying a Gaussian filter to eliminate noise and calculate the gradient magnitude and direction of the image. This is followed by the implementation of gradient magnitude thresholding or lower bound cut-off suppression to discard any spurious responses in edge detection. A double threshold is then applied to determine potential edges. The final step is the implementation of non-maximum suppression and hysteresis thresholding to obtain the final edges as per [6].

After the application of linear filters, the Hough transform [19] is applied to extract the lines from the input candidate points. It is particularly useful for segmenting the tennis court, because it allows the lines detection of any orientation in an image, even if they are partially obscured or hidden by other objects in the analysed frame [2]. The next step is court validation, which verifies that the lines conform to a tennis court. Short lengths, bad placements, or wrong slants are all indicators of lines to be removed. Corners and geometry are also validated to allow for better tracking of the player's position based on these strategic points. To increase the robustness of the algorithm, it is suggested to ensure that the previously selected corners are located at the intersection of two lines. The geometric check is based on the size of the lines and their relationships with each other. The final horizontal lines of the court must be parallel and have the same size proportion, while the vertical lines must have the same size.

1.3.3.2 Players Detection and Tracking

In the literature, a possible way to detect and locate players on the court is using the Deformable Part Model (DPM) [10]. DPM is a type of statistical model that represents objects as a set of parts that can move and deform independently. The model is trained specifically to detect and recognize objects of different sizes and scales. With an accuracy of around 73%, in [25], the DPM works as a tool to identify the players from broadcast sports videos. However, such an approach can lead to false positives given the similarity in shape of spectators and the referee. Moreover, it may fail to detect players when they are partially occluded by other players because the DPM detector applies non-maximum suppression after the object detection. This technique removes duplicate or overlapping detections of lower score or confidence.

Another way to detect and track the players applied to the tennis sport was studied in [41]. In this approach, the core idea is to find a non-dominant-color region surrounded by dominant-color areas. These non-dominant regions are supposed to be the players somewhere inside the dominant color region, representing the court. It is required to detect in the first place the court view shot and only then apply this method.

An alternative way to detect tennis players is to use standard background targeting followed by a simple blob tracker [50]. As during the point there is a fixed camera filming, theoretically, the only moving objects will be the players on the court and the ball. The process consists of binarizing two consecutive frames and then subtracting

them. After that, the resulting image will reveal the differences between the two frames in white and everything else will be black. Using blob detection algorithms such as Scale-invariant feature transform (SIFT) [23], Laplacian of Gaussian (LoG) [21] or even Connected Component Labelling [37], it is possible to detect and isolate players.

1.3.3.3 Tennis Ball Detection and Tracking

Some different tennis ball tracking algorithms can be found in previous projects. A part of them uses multiple cameras, like in [34], while others can do the job with a single transmission camera. We will analyse examples of these last approaches.

One possible way to develop a system for tracking a tennis ball using computer vision techniques can be found in [26]. The developed system uses three different techniques. They differ from each other in how foreground objects are detected and the background model is used.

The first technique, called *background subtraction with verification*, starts by creating a background model by averaging a number of background images. Then comparing the current image to the average background image to identify differences. These differences are considered candidates for the tennis ball, but they are further verified based on various criteria, such as color and size, to confirm their identity. Additionally, candidates located within the area of the player are removed, as these are likely to be false positives.

The second technique, *image differencing* between current and previous images, involves detecting candidates for the tennis ball in the current image by comparing it to the previous image, using image differencing, and identifying differences. There can, however, be a situation known as "double detection" in which the same ball position is detected in both the prior and current images. Background subtraction is performed on the current image to eliminate candidates originating from the previous image. After that a logical AND operation is used between the result of the image differencing and the background subtraction. The region of the ball obtained from the preceding frames is then removed. Rather than using the average image as the background model, the authors use a model that simulates the value of each background pixel as a single Gaussian distribution. This is more robust to noise and variations in lighting conditions. In the case of real-world scenarios, this last step is essential to prevent incorrect detections due to meteorological conditions, such as clouds passing in front of the sun.

The third technique to detect the ball named *adaptative background modelling using a mixture of Gaussians* used in [26], is somehow similar to the second one. The difference between them is that this last one uses a mixture of Gaussians to model the value of the background pixels instead of a single Gaussian. This technique is also more flexible and dynamic because it can adapt the background model to gradual changes in the environment.

There are some other approaches to detect the tennis ball, in [35], for example, the authors suggest that the universal yellow color present in the ball can represent an

important feature for its tracking. Nonetheless, due to its size and fast-moving speed, its color can be intensely affected by the court color it is moving on.

Another possible method for detecting and tracking the tennis ball is presented in [50]. In the article, the authors present a solution for robust detection under low-quality tennis match videos. Their approach starts by extracting foreground-moving objects using a pixel-wise temporal differencing. After that, the detected objects are clustered into blobs together with a small surrounding area. This area is binarised, and an ellipse is fitted to the detected edge pixels on it. Then the normal direction and the gradient for a number of pixels are calculated using a 3x3 Sobel Mask. Finally, an 8-dimensional feature vector is constructed with certain accurate parameters to identify the tennis ball, like the dimensions of the ellipse, the coordinates of the blob centre, and the mean of pixels inside the blob in HSV channels. After having the tennis ball candidates, a particle filter is used to track them. This particle filter assumes probabilities for the location of the ball during its movement. When the ball is near one player there is a change of dynamic model to track it and when the ball starts to move away again, the first dynamic model assumes the place and continues the tracking movement estimation.

The following section provides an in-depth examination of the current state-of-the-art in audio feature analysis within the context of sports match videos.

1.3.4 Audio Feature Analysis in Sports Match Video Context

The aural component of sports video can represent a significant part in semantic event detection [49]. It is possible to isolate the keywords that can support a spectacularity measure of the events. These keywords or some specific sounds can come from the commentators, the referees, the players, or even the audience. As shown in Figure 1.4, audio keywords can represent the MLR, which may lead, through some domain knowledge, to constructing the HLS. These aural MLR can be created from the different audio LLFs supported by vector machines or algorithms. In [49], the authors affirm that based on experimental results, the audio models have the needed characteristics to help event detection in sports videos. These characteristics can include spectral features such as the energy and the power of a signal, and pitch features including the fundamental frequency and harmonic frequencies of a signal. Can also include Mel-Frequency Cepstral Coefficient (MFCC)s, rhythm measured in Beats per minute (BPM), or even temporal features like zero crossing rate, energy entropy, and energy variation.

Some research in detecting audio features was made, and it was found that they can be a valuable tool for detecting events. However, due to the similar audience noise produced in different moments of sports games, it is complicated to distinguish and label events only based on the aural component of those broadcast videos. In our particular case, aural features can be an evaluative tool or a good indicator of the best moments in the tennis game. When combining the visual and the audio features the highlight detector can become more robust and accurate. There is already some research in the area that

can prove the successful harmonious conjugation of visual and audio features in tennis matches [41].

1.3.4.1 Audio Features Extraction

The first step in detecting events is understanding how and when to extract audio features. The chosen method must be as robust as possible, having the ability to distinguish the relevant information present in the audio part of the video. It is also essential to understand when these features appear so that they can be related to the outstanding moments present in tennis matches.

Due to the typically low noise level present during tennis matches, the utilization of aural features may not be as effective in detecting all events. However, it can be useful in identifying specific, notable events and highlights. The audience and commentators in tennis tend to generate minimal noise during most of the matches. Therefore, instances of increased noise levels, such as vigorous applause, can be used as an indicator of an exceptional event.

In [47] the authors introduced the term "audio keyword". It refers to a group of sounds that are specifically related to the actions of players, commentators, audience and referees, within a game. These sounds can be used to convey important information about what is happening in the game.

In later research [48], the authors present an approach to collect the LLF from the audio of the sports video transmissions. The article [48] introduces a new system to identify and classify audio keywords and compares it to a previous one introduced in [47], the hierarchical SVM. The introduced system is an adaptive HMM. It was presented as an improvement of SVM as a keyword classifier system because this one does not consider any contextual information and because to become a robust recognizer it has to be trained with big amounts of data. An adaptive HMM is used to refine the initial HMM, which is obtained from an available training dataset, using a diminutive number of testing data in order to better fit the attributes of a testing sports game video. The developed algorithm to achieve the aural features extraction is based on MFCC and Energy because they show success in experiments in audio keyword recognition and in speech recognition obtained in the author's previous research [47]. The authors, based on different experiences, define the short duration features as 0.2 seconds long (e.g. the ball hitting the racket) and 1 second for the longer ones (e.g. keywords in commentator observations and audience noise). The louder and more excited interventions by the commentators and the audience indicate to be the best highlight indicator when compared with other noises during the game.

In Table 1.1 it is possible to observe both proposed methods and observe how accurate they could be in detecting the audio keywords present in the audio part of a real case scenario. It is notorious that at the frame level, the HMM-based method performance is superior to the SVM keyword classifier approach.

Table 1.1: Comparison between HMM and SVM in detecting some of the audio features present in a basketball video [48]

Audio Keywords	Methods	Recall(%)	Precision(%)
Audience	SVM	83.71	79.52
	HMM	95.74	95.74
Excited Audience	SVM	80.14	81.17
	HMM	85.71	85.71
Commentator	SVM	79.09	78.27
	HMM	98.04	94.34
Excited Commentator	SVM	78.44	82.57
	HMM	86.67	100

Another successful case of the use of HMM-based method to detect the audience applause and noise can be found in [41]. Here, the authors use as audio features the band energy ratio, energy, bandwidth, zero-crossing rate, frequency centroid and MFCC. In tennis, the audience use to be quiet until some outstanding event happened. These remarkable moments could be composed of baseline rallies or even aces. The proposed approach focused on detecting audio effects at the end of each play and labelling it accordingly. If it was detected cheers and applause at the end of it, the play is marked as a possible highlight, otherwise, it is ignored from the aural point of view.

There are already audio feature detectors that are quite efficient, as shown in Table 1.1. Combining them with other feature detectors present in the frames of a tennis broadcast video could improve the robustness of sports event detection and highlight generation.

1.3.5 Textual Features Analysis in Sports Match Videos

The textual information associated with sports broadcasts can be separated into two different categories as described in [15]. The first category involves the extraction and analysis of external sources, including text from websites related to the sport and statistics found on the internet. These data could be very useful for event detection and segmentation due to their detailed information about the players and the description of the precise time when the occurrences happened during the game. The second category is directly related to Overlaid-text (OT). In other words, it is the textual data that appears on the screen during the game. This information may include the result of the game, fouls committed, ball speed and some additional statistics.

1.3.5.1 Textual Features Extraction

In the publication [46], the authors propose an automated event detection method for American football that is based on the integration of features from within the video and

external sources. Specifically, the developed algorithm extracts text from sports websites and fuses this information with other features to detect significant events during the game.

The work in [12] describes a method for extracting and recognizing overlaid text from soccer videos. There are some challenges associated with its recognition in soccer videos, including the presence of motion blur, occlusion, and variations in font and text size. The proposed method to do it involves pre-processing the video frames to remove noise and improve contrast and then applying a text detection algorithm to locate and extract the overlaid text. To achieve this, the Sobel operator is applied to the original frame.

The Sobel operator [22], a convolution-based filter, is used to calculate the gradient magnitude of the image, which is a measure of the change in intensity of the image pixels. The gradient magnitude is then used to identify areas in the image where there are significant changes in intensity, which correspond to edges.

After the application of the Sobel filter to the original image, the authors apply element dilatation [21], binarization, a morphological opening [21] and finally, a connected-component analysis [23] to locate exactly where the surrounding boxes with the textual information are located, as can be seen in Figure 1.5.

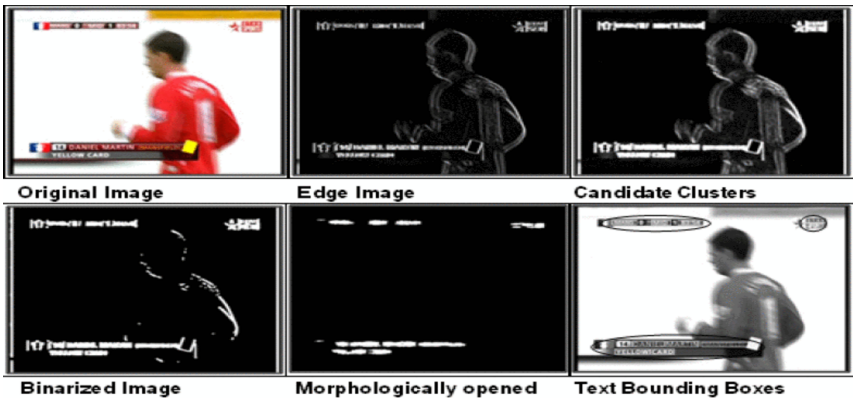


Figure 1.5: Figure from the article [11] to illustrate the detection and extraction process of the Overlaid Text within a TV transmission of a football match.

Upon identification of the regions containing overlaid text, the authors present a technique for recognizing the text through the application of Optical Character Recognition (OCR) technology. The OCR process converts the image of the text into machine-readable characters as described in [31]. To optimize the performance of the OCR software, which is generally more effective with images featuring black text on a white background, an image binarization process is applied. This is achieved by utilizing a threshold value obtained through Otsu's method [24]. The quality of the text recognition is directly proportional to the clarity of the letters within the image.

The proposed method for extracting and recognizing overlaid text from soccer videos results in a set of textual features that can be utilized for event detection. These features, obtained through OCR technology, can be matched against a pre-determined set of soccer-concept keywords. Additionally, the OCR is capable of detecting numerical values present in the score marker, which can serve as a valuable confirmation tool for event detection.

This enables a precise understanding of when a goal or point has occurred during the match.

To effectively classify the various occurrences detected during the match, it is important to organize them into distinct categories. The next section examines the strategies utilized for classifying the events detected during tennis matches.

1.3.6 Event Classification

This section aims to provide an understanding of how low-level data can be used to extract higher-level video features and semantics, as shown in Figure 1.4. We also explore how to distinguish and classify different events in tennis video broadcasting by combining spatial movement, temporal information, and audio effects.

The current section is built upon the research conducted by [13] and [41] on event classification in tennis. Both of these approaches involve connecting various frame-based information with symbolic descriptions of the scene. In the examination of both approaches, four distinct features were taken into consideration. The first feature pertains to the moving distance of a player during a single play, which is then classified as "short", "medium", or "long" in accordance with predetermined criteria. The second one is about the relative position of the player in relation to the court, which is classified as either "near the net" or "away from the net". This last feature considered is determined by the service line. If the analyzed player crosses the service line in the direction of the net at least once, it is classified as "near the net." Contrarily, if the player never crosses the service line, the play is classified as "away from the net." The third one is related to the length of the play, which is also classified into three stages: "short", "medium", and "long". The final feature considered is the presence or absence of sound effects or applause at the end of the play. The combination of these last four elements is sufficient to classify events into a range of different types of plays.

Based on the previously discussed research conducted by [13] and [41], a tennis event can be classified into five distinct categories. This classification is achieved by considering various frame-based information and linking it with symbolic descriptions of the scene.

The first event considered is a fault. It can be characterized as the moment when a serving player fails to place the ball correctly within the designated playing area, when this happens, the point cannot start. In these cases, the camera will often switch to an off-court view. When a fault is committed, the audience is typically silent.

The second type of event is similar in characterization to the first, but it is less common to occur. It is called "Double fault". This type of event will only be detected if, after the first committed fault, there is no change in the scene and the player fails to execute the second serve as well. It is characterized by being longer than a single foul and by low or non-existent player movement. This moment is usually followed by a moment of silence or a sound of astonishment. This sound likely has a spectral pattern distinct from that of applause.

The third event type is called "Ace" or "non-returned service". It is considered an ace when a player successfully serves and the opponent is unable to make any contact with the ball. Contrarily, if the opponent is able to make contact with the ball, throwing it to the net or falling outside of the court, this event is referred to as a non-returner service. In both instances, the audience typically communicates their excitement through applause. In both implemented approaches, [13] and [41], aces and non-returned services are classified as the same.

The fourth, known as a "baseline rally" is created when a player successfully executes a serve and the opponent returns the ball with success. For a playing moment to be considered a baseline rally, both players must remain around the baseline until one of them is unable to return the ball. Consequently, neither player approaches the net during the course of the shot.

The final considered event is the "net approach". This one is considered when, after a player successfully serves, the opponent successfully returns and one or both players approach the net with the intention of putting pressure on their opponent.

The last two events can be detected via image processing alone, however, the audio features present at the end of both plays may provide additional information to classify the level of spectacularity of the point, based on the noise produced by the audience after the play.

In Table 1.2 each one of the described events is classified according to the features they have to group. It explains what specific combinations of mid-level representations can lead to the labelling of the different events during the game.

Table 1.2: Tennis Events-Features Correspondence([13],[41])

	Player Movement	Net relative positioning	Play length	Sound or applauses
Fault	Short	Away	Short	Silent
Double fault	Short	Away	Medium	Silent
Ace or Non-returned service	Short	Away	Short	Noise
Baseline rally	Medium/long	Away	Medium/long	Noise
Net approach	Medium/long	Near	Medium/long	Noise

As it is possible to understand, Table 1.2 describes the events of an entire play, without discriminating the individual moments within it. Nonetheless, in this dissertation, we want to take this topic to another level. To accomplish this goal, we intend to develop an algorithm capable of identifying and distinguishing every single moment during a single play. The developed software will be able to differentiate between a right and a left swing, segment the precise moment when a player approaches the net, and determine the type of shot that the player chooses to execute.

2

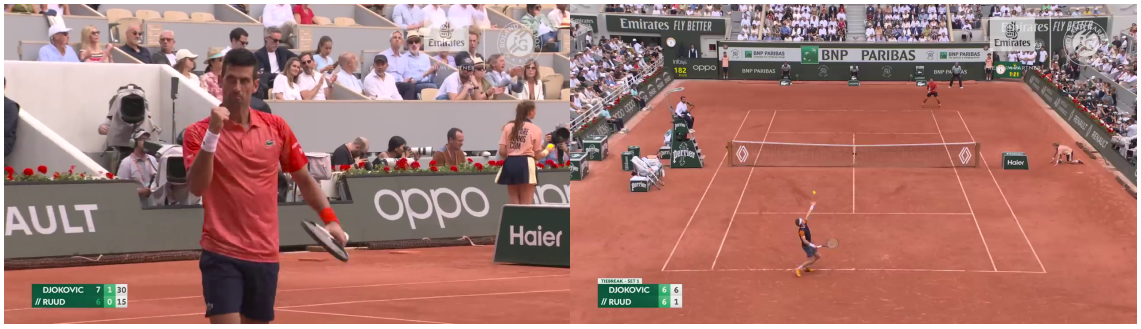
Tennis Event Detection

Having a comprehension of the fundamental concepts related to video analysis and image processing techniques, in this chapter, we present the proposed solution to overcome the engineering problem introduced in the particular case of tennis. We will explain, in detail, our developed work allowing a deep understanding of what kind of high-level characteristics can be extracted from the tennis match and how this information can be accessed in real-time. The videos chosen within the scope of the development and testing of our project have a quality defined as 720p (stands for 1280×720 pixels) and the number of Frames per second (fps) in the tested videos is between 25 and 30. To a better understanding of all the steps taken to achieve the final goal, we separate the major milestones into sections. These sections explain the methodologies, techniques, and approaches to solve every found challenge.

In the first section, we will focus on distinguishing between the transmissions that are capturing tennis court views and those that are not.

2.1 Tennis Court View Detection

In the context of tennis TV broadcasts, we can identify two distinct categories: *court view* and *non-court view*, as shown in Figure 2.1. The focus of this analysis is on detecting *court view* shots. The accurate identification of court view shots is a crucial step in detecting the initiation of a play. During gameplay, the camera is typically positioned behind one of the players and is not subject to significant movement. When adjustments are made, they are typically small and smooth movements to keep up with the players' movements.



(a) *Non-court view shot* (b) *Court view shot*

Figure 2.1: Two examples of the distinct view shots under study.

The developed algorithm to detect the beginning of one play is based on comparing the previous with the current frame of the video. The objective is to identify a change of scenario. The adopted method is based on the Peak signal-to-noise ratio (PSNR) value obtained from its formula. The PSNR is an engineering term used to measure the quality of a signal by comparing the peak signal strength from it to the strength of the signal's noise. It is defined as the ratio between the maximum possible power of a signal and the power of the noise that corrupts the signal affecting the integrity of its representation. PSNR is commonly used to evaluate the quality of image compression and transmission techniques. The higher the PSNR value, the better the quality of the image. This value is typically measured in decibels (dB).

To detect the differences between the current and the last frame, first, we have to calculate the Mean Squared Error (MSE) between them. MSE can be used to measure the difference or dissimilarity between two images. It is defined as the average of the squared differences between the pixel values of the two images. The MSE is a scalar value that ranges from 0 to infinity, with lower values indicating that the two images are more similar.

MSE provides a direct measure of the average pixel difference, while PSNR provides a relative measure of image quality. Together, the equations (2.1) and (2.2) can provide a more robust and accurate assessment of the differences between the images.

$$\text{MSE} = \frac{1}{w \cdot h} \sum_{i=1}^{h} \sum_{j=1}^{w} (\text{F}_{ij}(k) - \text{F}_{ij}(k-1))^2, 1 \leq k \leq \text{K} \tag{2.1}$$

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right) \tag{2.2}$$

As outlined in Equation 2.1, the MSE, calculated in the range of frames in the video, K, is determined by calculating the average of the squared differences between each pixel value of the previous, $F(k-1)$, and the current frame, $F(k)$, both with dimensions $w \times h$. Subsequently, the PSNR in decibels (dB) can be calculated using the formula presented in Equation (2.2), where MAX represents the maximum possible value for a single pixel,

which in this case is 255. The final step is to compare the PSNR value to a predetermined threshold.

A different method to detect scene change was also tested. This other method called Structural Similarity Index Measure (SSIM), takes into account the structural and luminance information of the images. It is more complex and computationally intensive, but it is better at detecting structural differences between two images, such as changes in texture, contrast, and spatial relationships. Considering two different images, x and y, the index equation of this method is presented below in (2.6) resulting from the product of the brightness, contrast and structure components of those images. Assuming all the weights α, β and γ have the same value, defined as 1, the resulting equation to calculate the SSIM can be reduced as shown in (2.7).

$$l(x,y) = \frac{2 \cdot \mu_x \cdot \mu_y + c1}{\mu_x^2 + \mu_y^2 + c1} \tag{2.3}$$

$$c(x,y) = \frac{2 \cdot \sigma_x \cdot \sigma_y + c2}{\sigma_x^2 + \sigma_y^2 + c2} \tag{2.4}$$

$$s(x,y) = \frac{\sigma_x y + c3}{\sigma_x \cdot \sigma_y + c3} \tag{2.5}$$

$$SSIM = l(x,y)^{\alpha} \cdot c(x,y)^{\beta} \cdot s(x,y)^{\gamma} \tag{2.6}$$

$$SSIM = \frac{(2 \cdot \mu_x \cdot \mu_y + c_1)(2 \cdot \sigma_{xy} + c_2)(\sigma_{xy} + c_3)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)(\sigma_x \cdot \sigma_y + c_3)} \tag{2.7}$$

The equation (2.6) can be divided into three distinct components. The luminance component, l, compares the average intensity of the images. It is calculated as presented in (2.3). Where μ_x and μ_y are the average pixel intensities of the images, and $c1$ is a small constant used to prevent division by zero.

The contrast component, c, calculated in the formula (2.4), compares the local contrast of the two images. The σ_x and σ_y represent the standard deviations of their pixel intensities, and $c2$ is a small constant used to avoid division by zero.

Finally, the structure component, s, compares the correlation between the images. It is calculated in (2.5). Where $\sigma_x y$ is the covariance of the pixel intensities of the frames, and the zero division is prevented by using the small constant c3.

One advantage of choosing PSNR over SSIM is that it is faster and easier to calculate. Additionally, PSNR may be preferred when the objective is to compare the quality of images with a high signal-to-noise ratio, such as in image compression, or in the case where the objective is to compare the quality of images that are relatively clear and noise-free.

Upon completion of the PSNR method, if the current frame has met the criteria set by this test, it is assumed that the next play is about to begin. The number of the frame in

which the change of scene is detected is saved for subsequent identification of the duration of play.

The same methodology is used to detect the end of the court view shot. When another change of frame is detected after the play has begun, the number of the frame is subtracted from the number of the one that marks the beginning of the court view shot. The resulting number is multiplied by the number of fps, to determine the duration of the match in seconds.

To enhance the method discussed earlier, we have developed two functions to validate the start and end of a new match scene. These functions primarily concentrate on detecting alterations within the scoreboard. When the scoreboard becomes visible on the screen, it acts as an indicator for the beginning of a new play. Similarly, the moment when it changes or disappears in the broadcast transmission represents the conclusion of a disputed point. All of these observations provide valuable information to construct a more robust detector for identifying key moments within the game.

Once the court view has been identified in a TV tennis broadcast, the next step is to detect the precise placement of the court lines. This next step is crucial for extracting more substantial information on the relative positioning of players throughout the game.

2.2 Tennis Court Lines Detection and Segmentation

The method for detecting and segmenting court lines begins with cropping the image using a Region of Interest (ROI). This initial selection serves to reduce computational complexity and increase the efficiency of subsequent techniques. The ROI is defined as a percentage and is primarily used to remove a significant portion of the surrounding audience from the image.

The next step involves converting the image to grayscale, followed by binarization using the Otsu method. The resulting black-and-white image is then processed using the Canny edge detection function to extract edges.

Before applying the Hough Line detection technique, a dilatation of both vertical and horizontal lines is performed, followed by an erosion with a different kernel applied to the vertical lines only. Both techniques can be mathematically written as shown in (2.8) and in (2.9).

$$(F \oplus G)(i,j) = \bigcup_{m,n \in G} F(i-m, j-n) \tag{2.8}$$

$$(F \ominus G)(i,j) = \bigcap_{m,n \in G} F(i+m, j+n) \tag{2.9}$$

Equation (2.8) is the mathematical representation of the dilatation operation in an image processing context. The F represents the input frame. $\oplus$ denotes the dilation operation signal. G is the structuring element, which defines the shape and size of

the neighbourhood used for the dilation operation. In image processing, the structuring element is usually a small binary matrix or a small neighbourhood of pixels. (x,y) represents the pixel coordinates in the output dilated image, while (i, j) represents the relative positions within the structuring element G. The $F(i-m, j-n)$ denotes the shifted version of the input image F by the relative position (i, j) (moving the structuring element G to the pixel coordinate (x, y)). $\bigcup$ represents the union operator, which takes the union of all shifted versions of the input image F based on the structuring element G at the pixel coordinates (x, y). $(F \oplus G)(x, y)$ represents the result of the dilation operation at the pixel coordinate (x, y) in the output dilated image.

Likewise, in equation (2.9), we present the erosion corresponding formula. The erosion operation ($\ominus$) is conducted between the input frame F and the structuring element G. In the erosion case, the symbol $\bigcap$ denotes the intersection operator, which computes the intersection of all shifted versions of the input image F based on the structuring element G at the pixel coordinate (x, y).

After the application of the previously mentioned operations, the Hough Line detector is then applied to obtain only the candidate lines that compose the court. By analyzing the slope of the output lines, the relative distance between points and the position of all of them, the algorithm composes two vectors. The next step is to remove small lines and multiple ones detected in the same location. Finally, the method correctly displays the vertical and horizontal lines forming the court.

All processes until the final results are present in Figure 2.2 for a Wimbledon match.

The subfigure labelled **1** in the top right corner represents the original frame, captured when the court view is detected. From this initial frame, a series of transformations are applied, including resizing and grayscale transformation, as depicted in subfigure **2**. Subfigure **3** illustrates the application of the Otsu method for image binarization. The Canny edge detector is applied to the image in subfigure **4**, followed by vertical and horizontal dilation, as shown in subfigure **5**. The Hough line detector algorithm is then applied, and the resulting lines are displayed in blue for horizontal lines and green for vertical lines as in subfigure **6**. Further, a process of comparison and elimination is carried out to eliminate small lines and multiple lines in the same space, resulting in the final subfigure **7**, where the vertical lines are displayed in blue, and the horizontal lines are displayed in green.

The successful detection of all court lines leads us to the identification of the four constituent corners of the court. This final task is based on comparing each detected line and subsequently labelling the corners of the court based on their respective distances from the corners of the frame. By comparing the endpoints of the detected lines with each other, it becomes possible to identify them.

Precisely identifying court lines is crucial in establishing the positional relationship between players and the court. This initial detection is a foundation for achieving more accurate player detection and tracking. Subsequently, the following section explains the comprehensive methodology and the developed algorithms employed to achieve robust

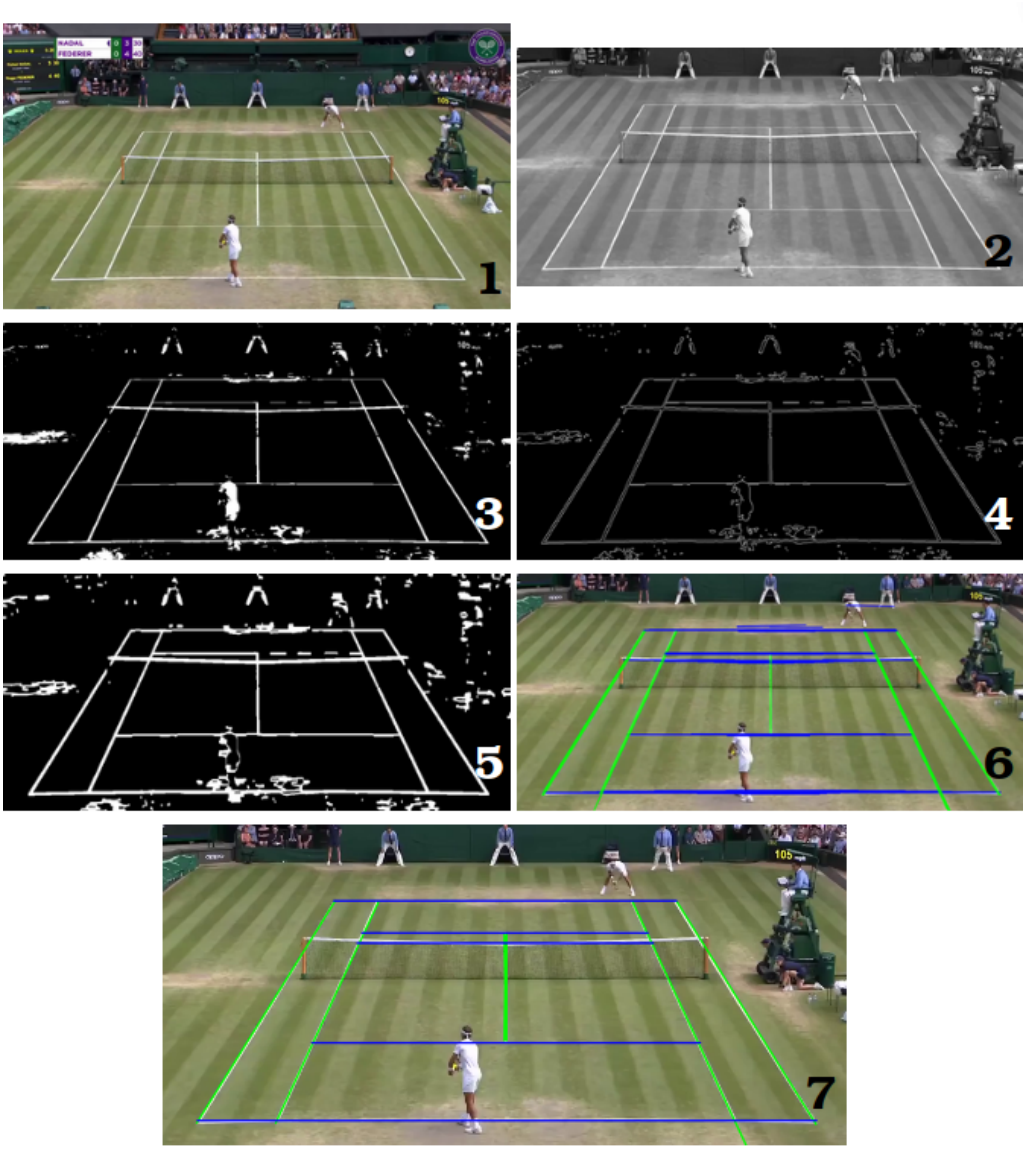


Figure 2.2: The sequentially numbered process images resulting from applying the Tennis Court Line Detection algorithm to a Wimbledon match: **1**- The original RGB frame, in which the court scene is identified. **2**- A resized image that has been converted to grayscale. **3**- Binarized image, obtained using the Otsu method. **4**- The application of a Canny edge detector. **5**- Vertical and horizontal dilatation. **6**- The application of the Hough line detector, in which the vertical lines are represented in green and the horizontal lines in blue. **7**- The final line segmentation, in which all detected lines are filtered.

player detection and tracking.

2.3 Players Detection and Tracking

During the points disputes, the camera is fixed, making only small movements to follow the main actions. Around the court area are only referees, the ball boys and girls and the players. The first ones should remain stationary in the same position throughout the play,

so theoretically, the players will be the only moving objects along the court.

The adopted technique to identify the players on the tennis court is called Background Subtraction (BS). BS is a prevalent and extensively employed technique used to generate a foreground mask. This mask, which comprises binary image data, effectively identifies the pixels that correspond to the moving objects within the observed scene. By utilizing stationary cameras, BS calculates the foreground mask through a subtraction operation conducted between the current frame and a background model. This background model contains the stationary elements of the scene or, in a wider context, any components that possess the characteristics of a background under the observed scene [7].

In our case, we employ a binarization process on both images, followed by subtracting the current from the previous frame. Given that the only entities in motion between consecutive frames are the players, the resultant image of this operation should contain the players as white foreground elements, while everything else appears black. However, achieving this outcome is not as straightforward as it may initially seem, owing to minor camera movements and the motions of the referees. Consequently, the subtracted image exhibits additional white blobs beyond just the players. Further image manipulation techniques are necessary to accurately distinguish the players from these extraneous elements.

The initial image processing operation applied is known as Opening, which involves the sequential execution of two distinct Morphological Transformations: Erosion (2.9) followed by Dilatation (2.8). This combined approach effectively eliminates noise from the original image, as explained in [8].

After the removal of smaller blobs, a contour-finding function is employed to identify the contours within the image. Subsequently, the focus shifts towards locating two blobs within the game area, corresponding to human-sized dimensions, which represent the players. Upon successful player detection, their positions are recorded, allowing for faster and more precise tracking. The tracking algorithm narrows its search scope to the surrounding areas of the players' most recent positions. A maximum distance threshold is defined to limit the potential locations of the players, with constant updates to the current position.

Having successfully identified the players, the subsequent task revolves around analyzing their movements and ball-striking actions at different moments during the gameplay. To accomplish this objective, a specialized ball-tracking algorithm was developed. Recognizing that the tennis ball always maintains a yellow color, for every tournament, a function to ignore other colors is applied to the analyzed frame. First, the original Red, Green, Blue (RGB) color space is converted to the Hue, Saturation, Value (HSV) space. In this way, it is possible to ignore the variances in the brightness and luminosity present in the frames and consider only the desired color. Subsequently, the algorithm utilizes this yellow filter on the Hue channel of the HSV, resulting in a black-and-white representation where all the white elements fall within the defined threshold for the yellow color, as determined by the used function [9].

The subsequent stage is similar to the players' identification process, but this time, the filters are defined to identify the tennis ball. The algorithm searches for elements within the defined court area, adhering to an additional surrounding threshold, and possessing dimensions corresponding to a tennis ball. This enables the accurate detection and tracking of the tennis ball throughout the course of the game. After the ball detection, its position is stored in a variable.

The result of the application of both algorithms combined with the corner identification algorithm is shown in the figure above.

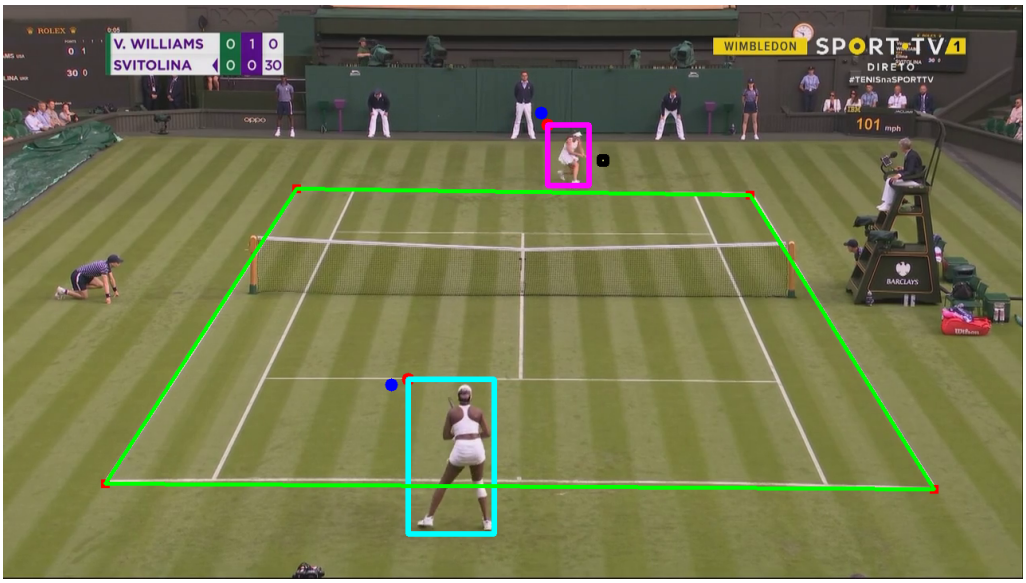


Figure 2.3: The generated image is a culmination of the combined implementation of the players' tracking algorithm and the ball tracking algorithm. Within the image, the presence of a magenta rectangle represents the bounding box containing the player located farthest from the camera, while a cyan rectangle delineates the current position of the player closest to the camera. At the top corners of both rectangles, red circles are used to depict the top left corner of the presently detected position, while a blue circle represents the previous top left corner of the tracked player's position. Additionally, the current identified position of the ball is enclosed within a black circle.

As can be seen in Figure 2.3, both players are surrounded by rectangles of distinct colors, and near the top left corner two different colors can also be observed. The red circle represents the identification of the player's detected position at the top left corner, while the blue circle signifies the identification of the same point in the previous player-detected location. The distance between them is used to calculate the displacement of the player frame from one frame to another, providing valuable information on their movement. Furthermore, Figure 2.3 displays a black circle, which corresponds to the output position generated by the ball tracking algorithm, effectively tracking the tennis ball's location during the game. To facilitate a comprehensive understanding of the outer court boundaries and the corner locations, resulting from the application of the algorithms explained in section 2.2, we have chosen to display the elements in green and

red, respectively.

With successful detection of the players and the ball accomplished, our subsequent goal is to extract higher-level information from the game. In this pursuit, valuable data can be obtained, including the identification of the winning point shot and the determination of the total number of shots that have taken place during each play. These insights provide meaningful and relevant details that contribute to a comprehensive analysis of the tennis match dynamics and outcomes.

2.4 Shot Segmentation and Counting

To accomplish the detection and identification of the type of applied shot, we utilize an algorithm for articulated human pose estimation, which enables the detection of the player's body movements and ball-striking actions. For this purpose, we use a pre-trained Caffe model [2]. The chosen model relies on the MPII Human Pose dataset [1]. This model based on MPII produces 16 output points. These points are different articulations and body parts. They are the head, the neck, right and left shoulders, the elbows, both wrists, the hips, the knees, the ankles, the chest and the background. In the developed algorithm for pose estimation, we apply it to a single player at each time. The output predictions of the network will be a 4 dimensions matrix. We ignore the first dimension because it is the image ID and we only pass one image to the network at each time. The last two dimensions are the height and the width of the output map, respectively. Finally, we have the second dimension indicating the index of the keypoints. For MPII it generates 44 points, we only use the first 16 that correspond to the keypoints of the body parts predictions. After that, we get the location of each point within the image by finding the maxima of the confidence map of that keypoint. We also use a threshold to eliminate wrong detections.

Through a comparative analysis of the points derived from the players' pose estimation, it becomes possible to verify the type of shot performed by the player. Combining this information with the specific location of the players on the tennis court, we can obtain high-level information on the play and enrich the metadata extracted from the tennis game. In Figure 2.4 it is possible to understand how the pose estimation points can be useful in identifying the type of tennis shot applied.

(a) Forehand (b) Backhand

Figure 2.4: This figure showcases two distinct types of tennis shots and their corresponding identification through skeleton analysis. Sub-figure (a) illustrates a *forehand* shot, while sub-figure (b) depicts a *backhand* shot.

The different types of shots identified can be distinguished from the other by analysing the body positioning. In 2.4(a), the pose estimation points distributed along the player's body allow the developed algorithm to identify a *forehand*. Similarly, subfigure 2.4(b), presents the articulated body when the player is executing a *backhand* shot. If both the right and left wrists are detected on the right side of the neck during the shot movement, it will be classified as a *forehand*. Conversely, if the same occurs on the left side, it will be categorized as a *backhand*. Another criterion to determine the shot type is the positioning of the knees in relation to the ankles of the players. Therefore, if both knees are positioned on the right side of the ankles, it is considered a *forehand*. On the contrary, if the right knee is on the left of the right ankle and the left knee is detected on the left of the left ankle, the shot is classified as a *backhand*. We apply these criteria to the player closest to the camera. For the player on the other side of the net, we reverse the relationship between the body parts. Also, these shot type detections are calculated for right-handed players, so inverting the final classification is sufficient to determine the type of shot for left-handed players. It is possible to know in advance whether a certain player is left-handed or to calculate it at the beginning of the match after detecting how the player is performing the service. The other two types of shots are classified as *Ace* and *Overhead*. The last is detected when the right, or the left, wrist is detected above the player's head and he is close to the net of the court. The *Ace* is defined when there were no detected shots during the point.

Upon successfully detecting how the point was won and documenting the number of shots involved, the subsequent crucial aspect is to determine the specific timing of the point's victory and the corresponding match status both before and after that key moment. This contextual information is essential in understanding the game's overall progression and can provide valuable insights into the dynamics of player performance and strategies.

2.5 Score Detection and Extraction

To improve event detection and classification, score information is collected at the end of each play. This allows precise determination of the serving player and point winner.

The proposed algorithms are designed to adapt to the variations in scoreboard design across different tournaments, such as Wimbledon and Grand Slam or Roland Garros. The algorithm begins by identifying the scoreboard location, then by resizing the image to zoom in on the score and performing image binarization to obtain a black-and-white image of the score, with a white background and black letters and numbers. However, in some tournaments, as present in Figure 2.5, the scoreboard may have a region where the numbers are white on a colored background. To address this, a mask is applied to invert the black-and-white elements in this region, ensuring that the complete score information is captured in black. Finally, the algorithm proceeds to identify the serving player and obtain the points information of the ended play.

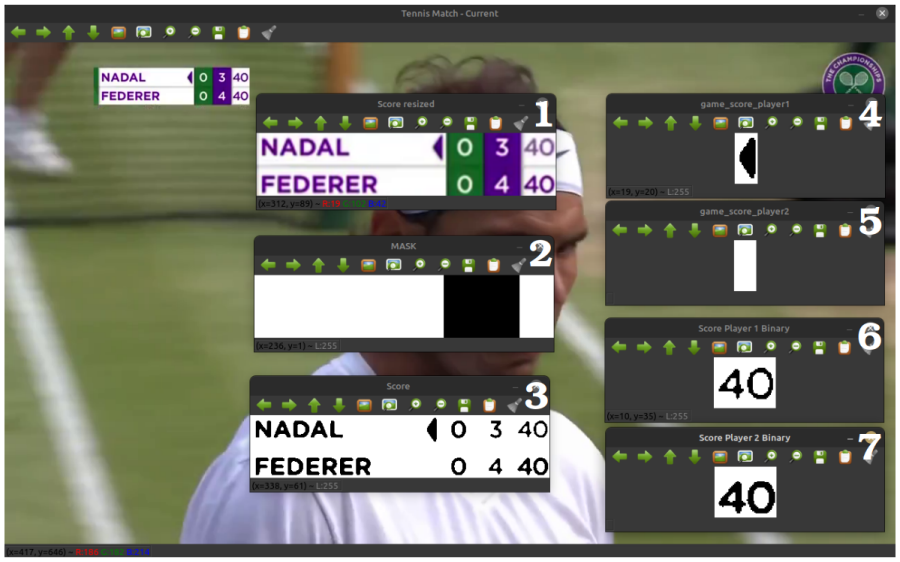


Figure 2.5: The Tennis Score Information Extraction Process Applied to a Wimbledon Match: 1-Extraction and resizing of the detected scoreboard. 2- Mask used to define the region where black and white pixels are inverted. 3- Obtainment of a scoreboard with black letters and numbers on a white background. 4,5- Identification of the region of the scoreboard indicating the serving player (Player 1 or Player 2, respectively) 6,7- Identification of Player 1 and Player 2 scores, respectively, for the current set.

Figure 2.5 shows the process of tennis score information extraction applied to a match from the Wimbledon tournament. The original frame from the match can be seen in the background. The first subfigure, **1**, presents the scoreboard with a double zoom-in. The second subfigure, **2**, shows the mask used to invert the black and white pixels in the specified region. This region is where the foreground is white and the background is black, and a color inversion is applied only to the pixels where the mask is white. Subfigure **3** presents the scoreboard with all the letters and numbers in black and the background

in white, providing better results for the OCR method. Subfigures **4** and **5** enable the determination of the serving player by comparison. Finally, subfigures **6** and **7** display the results of the current set for player 1 and player 2, respectively.

Next, the prepared images are converted into textual information by applying the Optical Character Recognition (OCR) technology using the Tesseract engine [3]. This information is then stored to enhance the identification of events.

In televised tennis broadcasts, remarkable events are occasionally highlighted through replays. When players contest the referees' decision regarding the ball's point of impact on the ground, they may request confirmation through challenges. These particular moments present in TV transmissions hold the potential for generating metadata about the game and serve to differentiate the highly disputed plays from others. In the next section, we introduce the developed algorithms to detect and identify these significant moments within the broadcast.

2.6 Special Events Detection

Upon analysing the TV transmissions of tennis matches, we have identified two distinctive moments that possess the potential to generate valuable metadata for objectively classifying the attractiveness of a point. The first determined moment is known as replay, which is commonly employed to emphasize outstanding points.

In the context of Wimbledon matches, before presenting a replay, the Wimbledon logotype is showcased as a watermark on the screen with a flipping animation. Due to the logo's dynamic movement, variable size, and presentation as a watermark, its detection and identification present considerable challenges.

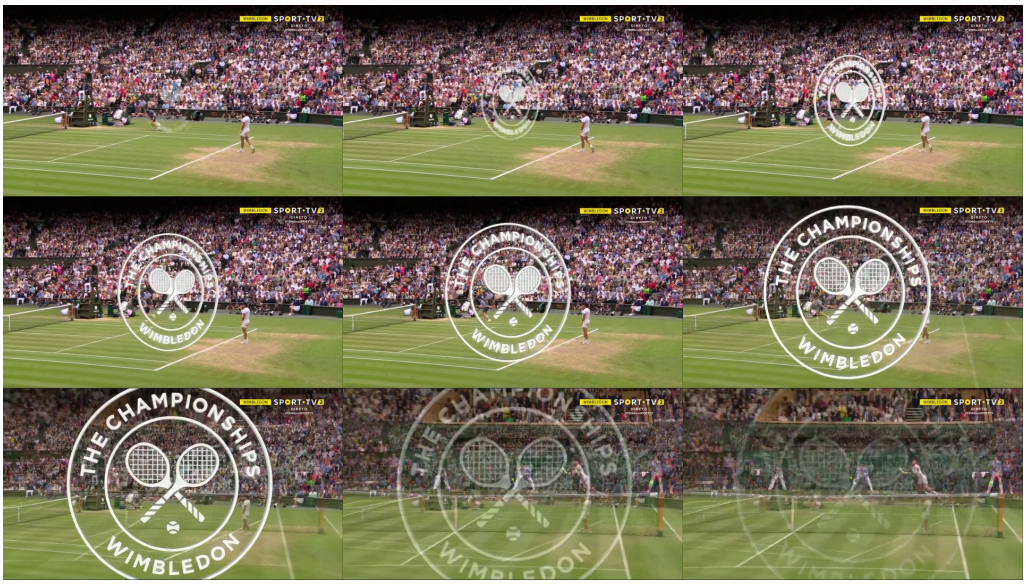


Figure 2.6: Introduction to the replay of the Wimbledon tournament consisting of a sequence of frames with the Wimbledon logo as a watermark.

As presented in Figure 2.6, the Wimbledon logotype watermark signalizes the beginning of a replay. Having it in mind, our approach starts by saving a preprocessed black-and-white reference image, containing the isolated logotype. After detecting the ending of a play, a fixed threshold is applied to eliminate darker noise from the frame. Subsequently, the image is converted to grayscale and then to black-and-white. Following these preprocessing steps, we subtract the obtained frame and the reference image and then count the number of resulting white pixels. If the total of white pixels within the subtracted image falls below a predetermined threshold, the algorithm signals the presence of replay, and the corresponding flag is set to true. Otherwise, the replay search will continue until the beginning of a new play.

The other special event observed in tennis matches is commonly referred to as *Challenge*, or *Hawk-eye*. Challenges in tennis present players with the opportunity to contest a line call made by the umpires. As part of the established rules, each player is entitled to a maximum of three unsuccessful challenges per set. At the conclusion of a point, if a player disagrees with the referee's decision regarding the ball's landing position on the ground, they can opt to call for a challenge. A successful challenge allows the player to retain the same number of them for the duration of that set.

Before displaying the video generated with the *Hawk-Eye*, on the TV transmission, a small symbol is exhibited always in the same location on the screen. Our algorithm for identifying these moments involves the direct application of the PSNR between a Region of interest (ROI), where the *Hawk-Eye* analysis indicator appears on the screen, and an image containing the corresponding symbol stored locally. Once the endpoint of a play is detected, the algorithm initiates the search for a frame where the replay indicator associated with the challenge is expected to appear.

With the development of these algorithms to detect and identify every event, we are now prepared to subject them to testing using real TV broadcast transmissions. This crucial phase will verify the effectiveness and robustness of the algorithms, contributing to the successful realization of our objectives in analyzing tennis match broadcasts.

2.7 Final Tests and Conclusion

This last section is dedicated to presenting the obtained results, analyzing them and bringing some conclusions about the proposed solutions for the engineering problem presented in the tennis context.

To validate the quality of the developed algorithm we decided to test it for all the Grand Slam tournaments. These four tournaments are considered the most prestigious individual competitions in tennis around the world [4]. They are the Australian Open, Roland Garros, US Open and Wimbledon. All the tests were conducted using a computer equipped with an Intel Core i5-7300HQ CPU of 2,50GHz. The videos have an HD (720p) resolution, meaning they consist of 1280 x 720 pixels. They are composed of 25, 29 or 30 frames per

second (fps). In Figure 2.7 we can observe the main differences in the characteristics of the courts and the different scoreboard styles for the tournaments mentioned above.

(a) Roland Garros

(b) US Open

(c) Australian Open

(d) Wimbledon

Figure 2.7: The 4 screenshots of the tennis match transmissions used to test the implemented algorithm. These tennis matches are from the tournaments including the Grand Slam. Roland Garros (a), US Open (b), Australian Open (c) and Wimbledon (d) are considered "the most prestigious individual competitions in tennis" [4].

A total of four recorded TV broadcast tennis match transmissions were used to evaluate the quality of the developed algorithms. The matches include the Wimbledon 2023 men's final, the US Open female final 2023, the Australian Open male first round of 2023, and the Roland Garros 2020 men's final. Table 2.1 summarizes the results of those tests.

Table 2.1: The tested tennis matches (detected | real)

	Australian Open	Roland Garros	US Open	Wimbledon
Points [detected | real]	[233 | 234]	[182 | 183]	[157 | 159]	[333 | 334]
Games [detected | real]	[37 | 37]	[26 | 26]	[25 | 25]	[46 | 46]
Sets [detected | real]	[4 | 4]	[3 | 3]	[3 | 3]	[5 | 5]
Processing time (without shot counting algorithms)	42m12s	48m33s	37m51s	76m20s
Processing time (with all the algorithms)	156m32s	299m50s	152m20s	512m37s
Match duration	165m08s	151m50s	103m50s	299m44s

By analysing Table 2.1, we can observe optimistic results on automatic event detection. The total accuracy on the 4 videos, of the point detection algorithm, was approximately 99.37%, while the accuracy for the set and game identification was 100%. We can also confirm the viability of the project application in live TV tennis match streams because, for every scenario, the processing time of detecting the points, games, sets, and all the associated metadata is always less than half of the tennis match duration.

For the tested Wimbledon match, we decided to test the algorithm for counting ball touches and the winning shot for each point. When these algorithms are working, the total process execution time triples. Although the algorithm was successful in detecting the winning shot at several points, the overall results obtained are not yet very promising, and it will need to be improved in terms of execution time and accuracy in the future. The algorithm to identify a player's challenge call was also tested for the same game. In order to assess the algorithm's performance, we use the terms True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN) [5]. The results obtained were [14 | 0 | 4 | 315], which means we got 14 correct detections of the *Challenge* call made by the players, 315 correct detections of points without the use of the *Hawk-Eye*, and 4 plays where it was used but not detected. The Accuracy (ACC) and the Precision (PPV) of the method can be calculated based on the formulas 2.10 and 2.11, respectively. For this Wimbledon match, the ACC for the method under test is 0.9880 and the PPV is 1.

$$\text{ACC} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.10}$$

$$\text{PPV} = \frac{TP}{TP + FP} \tag{2.11}$$

A replay detection algorithm for the Roland Garros and the US Open matches was also tested and its precision and accuracy were evaluated using the same method as referred to before (TP | FN | FP | TN). The results were [44 | 33 | 30 | 161] and [25 | 0 | 6 | 126] respectively. According to the outcomes for the Roland Garros, 49 replays were correctly detected, 21 replays were not detected, 23 replays were incorrectly detected, and 126 points were correctly marked without replay presence. The obtained results gave us an ACC of 0.6538 and 0.9688, and a PPV of 0.5946 and 1, respectively, according to 2.10 and 2.11.

There have been numerous challenges encountered during the development of the final solution. To simplify the organization and presentation of the fundamental steps involved in achieving the desired results, this chapter was divided into logical sections.

The first section introduces the identification of the court view shots caught on camera. This primary challenge has become harder specifically to identify the points that start being filmed from the above with the application of a zoom-in until the desired resolution, or the ones that are filmed from a camera positioned at the level of the players. In the following section, we encountered the difficulty of identifying the court lines due to the meteorological conditions or the number of matches on the same court in a short period. To overcome this we had to build a robust line filtration algorithm. Next, to count the

number of shots during every point, to identify how the point was won and where on the court, we developed an algorithm to identify the players and track them. In this section and the following one the major difficulties were distinguishing the players from the ball boys and girls and making a robust algorithm able to ignore the camera movements and keep tracking the players and the ball on the court. Every tournament has its design so the way the score is displayed is also unique for each one. Understanding how, when and where the score appears and disappears is crucial to extracting useful data from it. We decided to dedicate another section to demonstrate the difficulties related to the topic and how we adapted the developed algorithm for the different tournaments. To conclude the automatic event detection we added the identification of replays and the challenges analysis. Both events can work as an objective way to measure the spectacularity of one point that could be useful in highlighting generations.

In the developed solution, we can find some promising advancements and the creation of tools that could have opened some exploration paths in the automatic identification of events in tennis and can also inspire future solutions in other sports.

3

Prototype Overview

The intrinsic value of this thesis is the automatic detection of events with associated metadata within a tennis match. This data can be used for highlights generation, real-time updates in betting websites and statics generation. In this section, we will present the acquired metadata associated with the detected events using a prototype. Since this thesis was developed with the support of Six Floor Solutions, the prototype to validate visually the outcomes of the proposed solution is the one they already use. In this chapter, we explain the available functionalities of the prototype and how to take advantage of it.

The prototype offers a range of functionalities, enabling us to validate the efficiency of the developed algorithm in identifying events of different tennis matches. After collecting all the data associated with a point, when the implemented algorithm detects the end of the subsequent point, it generates a JSON message. The POST method is used to send that message as an HTTP request to the server so that the information can be stored. Subsequently, the data about that specific point becomes accessible for consultation on the prototype, displaying the stored data on the server.

Using the prototype, events can be filtered by points, games, or sets and even personalized filters can be set up to view only the events of interest. Figures 3.1, 3.2 and 3.3 illustrate one example of such filters.

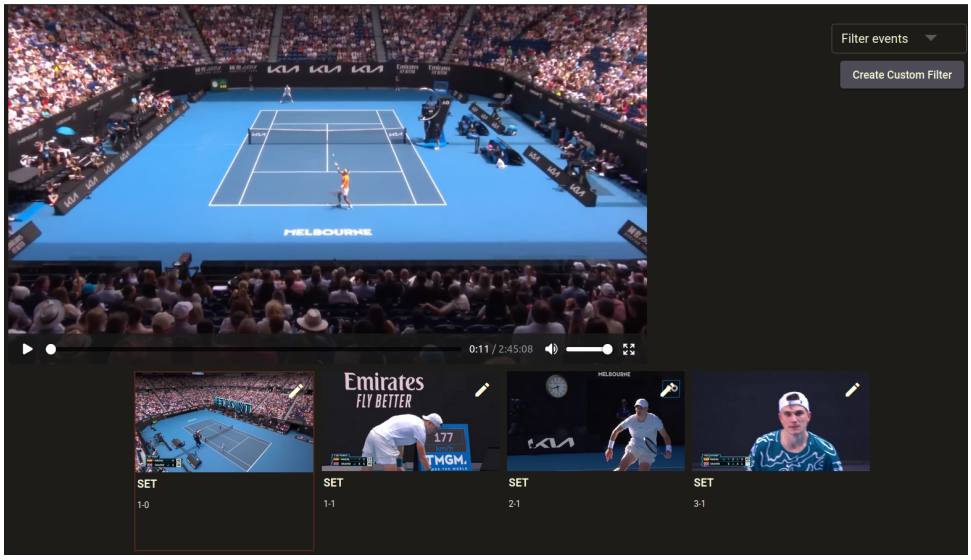


Figure 3.1: Screenshot of the prototype view of all the events "SET" detected for the Australian Open male first round of 2023.

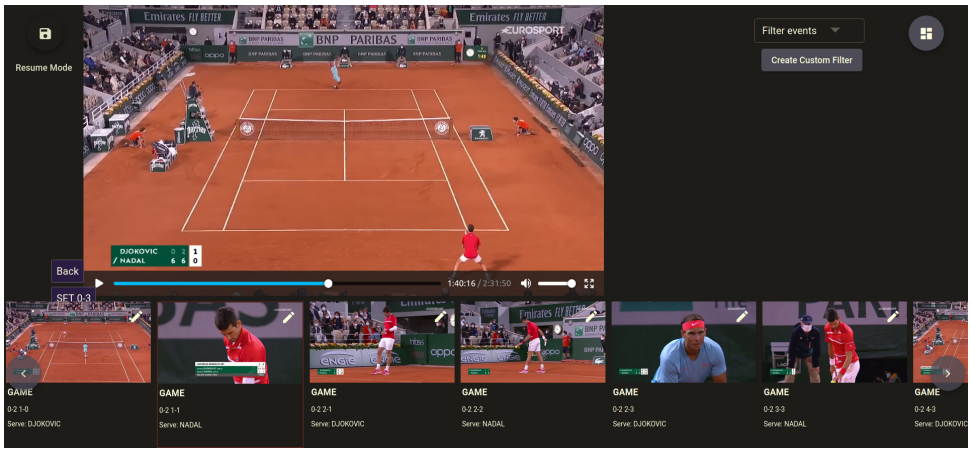


Figure 3.2: Screenshot of the prototype view for the detected "GAME" events in the 3rd set of the Roland Garros 2020 men's final.

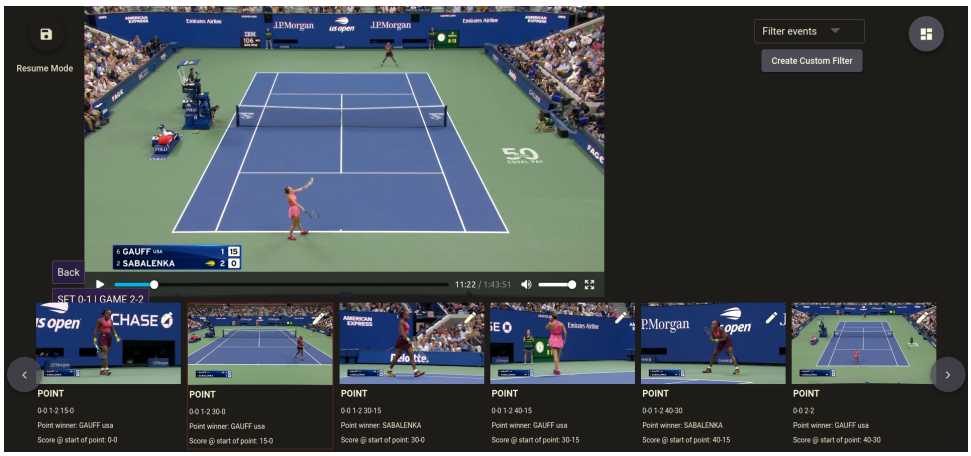


Figure 3.3: Screenshot of the prototype view for the detected "POINT" events in the 4th game of the 1st set of the US Open female final 2023.

As can be seen in the tree figures above, the prototype allows us to validate the detected events. It shows the detected sets, games and points for different tournaments. Next, in Figure 3.4 we display the detected metadata associated with the detected event. In this case, the event was a point.

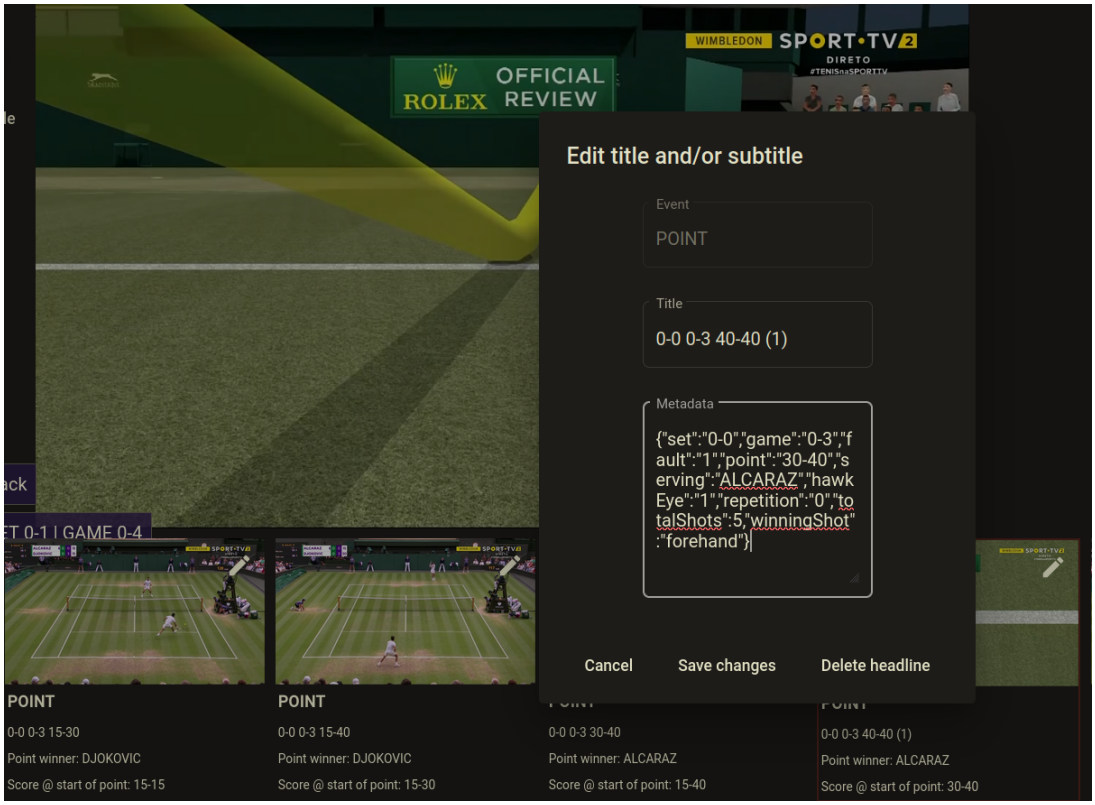


Figure 3.4: Screenshot of the prototype view for the extracted metadata of the 6th point of the 4th match in the 1st set of the Wimbledon 2023 men's final. The metadata is a field inside the JSON message. Its content is: "{"set": "0-0", "game": "0-3","fault": "1","point": "30-40","serving": "ALCARAZ","hawkEye": "1","repetition": "0","totalShots": 5,"winningShot": "forehand"}".

In Figure 3.4, the metadata about the detected point can be visualized. This metadata comprises information about the outcome of the current game, set, and the preceding point. It also indicates the serving player for that point, whether there was a challenge (referred to as "Hawk Eye") during that point if the game had a replay associated with it, and if the serving player committed a fault while serving. As evident in the figure, the metadata associated with the challenge call is marked as "true" (equivalent to "1"), confirming the accuracy of the detected data. This metadata is a field inside the JSON message that is sent to the server.

Figure 3.5 presents the application of a customized filter.

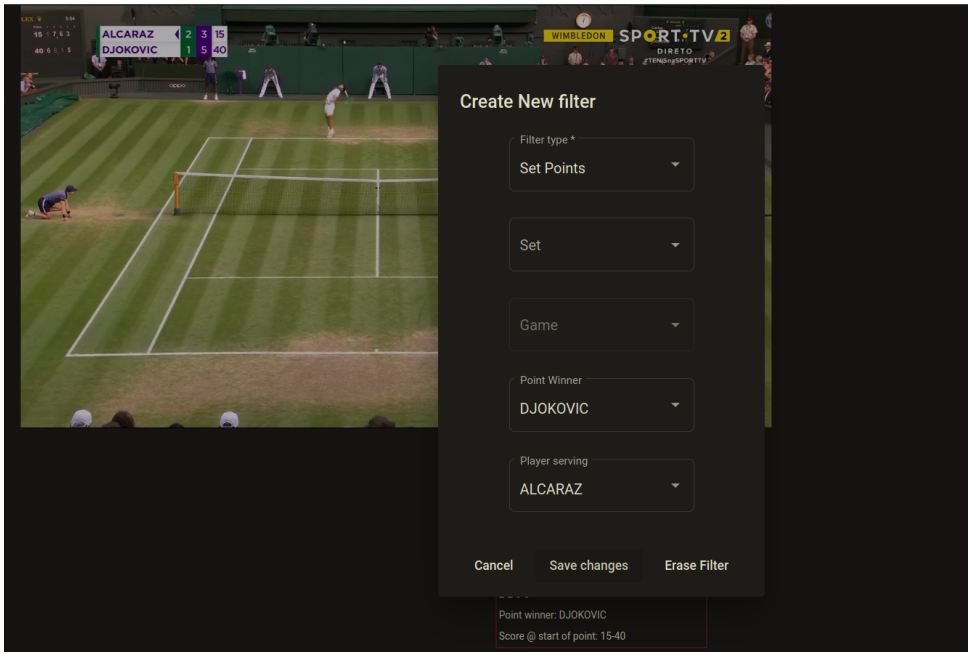


Figure 3.5: Screenshot of the prototype view of a customized filter for the Wimbledon 2023 men's final. In the figure, the filter has been configured to display only "Set Points" served by "Alcaraz" and won by "Djokovic".

Figure 3.5 presents one example of a possible custom filter. In this case, the chosen filter was "Set Points" served by the player Alcaraz and won by Djokovic. In the figure, it can be seen that there is only one detected event correspondent.

It was demonstrated in this chapter that the prototype can be useful to visually validate the automatic event detection solution that was developed. This step is crucial in supporting the transition to live television broadcasting of tennis matches after validating multiple detections for different events and associated metadata.

4

Conclusions

Significant advancements have been made in the field of automatic event detection in recent years. The automation of sports event detection not only enhances the efficiency of this task in comparison to manual human performance but also presents prospects for real-time performance and statistical analysis. Furthermore, it enhances the user experience by allowing them to selectively review the desired events during matches. The ongoing progress in developing algorithms for event detection could contribute to the expansion and progress of the sports industry.

In our investigation, we decided to solve the problem of automatically detecting and categorizing the events that occur during live tennis match broadcasts. To accomplish this objective, we developed specific algorithms based on conventional image processing and vision systems. These algorithms were designed to intake televised transmissions as input and generate, as output, a comprehensive compilation of all the events and relevant statistics. Those algorithms have the ability to identify the duration of each event, the outcome, the number of shots taken, the server, the precise moment at which the game starts, and the presence of any associated repetition. Our algorithms were prepared to only rely on the visual features presented in the video. To achieve a complete automatic generation of the scoreboard, potential future works may involve integrating additional visual elements displayed on the screen, such as the recognition of double faults, as well as accurately determining the winner of each point. Furthermore, this engineering challenge can be improved by incorporating the audio features present in the television broadcast. It is worth mentioning that the detection and categorization of events in televised streams can be extended beyond the domain of sports and can find practical applications in other fields.

In the world of multimedia, automatically detecting and identifying what events are occurring on the fly can save time and work. We hope our approach can inspire others in different sports or even other areas.

Bibliography

[1] "(PDF) INTERACTION BETWEEN MODULES IN LEARNING SYSTEMS FOR VISION APPLICATIONS". URL: https://www.researchgate.net/publication/249862831_INTERACTION_BETWEEN_MODULES_IN_LEARNING_SYSTEMS_FOR_VISION_APPLICATIONS (cit. on p. 7).

[2] I. J. of Advanced Research in Computer Engineering & Technology (IJARCET) ijarcet. "A Survey on Line Detection Techniques using Different Types of Digital Images". In: *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* (2019-01), pp. 43.1–43.8. DOI: 10.5244/C.1.43. URL: https://www.academia.edu/42907468/A_Survey_on_Line_Detection_Techniques_using_Different_Types_of_Digital_Images (cit. on p. 10).

[3] M. Ali and D. Clausi. "Using the Canny edge detector for feature extraction and enhancement of remote sensing images". In: *International Geoscience and Remote Sensing Symposium (IGARSS)* 5 (2001), pp. 2298–2300. DOI: 10.1109/IGARSS.2001.977981 (cit. on p. 7).

[4] M. Andriluka et al. "2D Human Pose Estimation: New Benchmark and State of the Art Analysis". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014 (cit. on p. 26).

[5] T. I. Association. "TENNIS INDUSTRY AT A GLANCE". In: (2019) (cit. on p. 2).

[6] *Canny Edge Detector — OpenCV 2.4.13.7 documentation*. URL: https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html (cit. on p. 10).

[7] T. Connected. *Tennis Popularity Continued to Surge in 2022 - Tennis Connected*. 2023-01. URL: https://tennisconnected.com/home/2023/01/12/tennis-popularity-continued-to-surge-in-2022/ (cit. on p. 2).

[8] *Deep Learning based Human Pose Estimation using OpenCV | LearnOpenCV*. URL: https://learnopencv.com/deep-learning-based-human-pose-estimation-using-opencv-cpp-python/ (cit. on p. 26).

[9] K. G. Derpanis. "The Harris Corner Detector". In: (2004) (cit. on p. 7).

[10] P. Felzenszwalb, D. Mcallester, and D. Ramanan. "A Discriminatively Trained, Multiscale, Deformable Part Model". In: (2008) (cit. on p. 10).

[11] *Frame Rate: A Beginner's Guide | The TechSmith Blog*. URL: https://www.techsmith.com/blog/frame-rate-beginners-guide/ (cit. on p. 2).

[12] *GitHub - tesseract-ocr/tesseract: Tesseract Open Source OCR Engine (main repository)*. URL: https://github.com/tesseract-ocr/tesseract/ (cit. on p. 28).

[13] J. E. González. "Automatic event detection for tennis broadcasting". 2011-07. URL: https://citeseerx.ist.psu.edu/document?repid=rep1%5C&type=pdf%5C&doi=608c918b752220406b53ccecc079bc6ccda2c365 (cit. on pp. 5, 6, 8, 9, 15–17).

[14] *Grand Slam tennis tournaments | ITF*. URL: https://www.itftennis.com/en/itf-tours/grand-slam-tournaments/ (cit. on pp. 30, 31).

[15] A. A. Halin. "SOCCER VIDEO EVENT DETECTION VIA COLLABORATIVE TEXTUAL, AURAL AND VISUAL ANALYSIS". 2011. URL: https://www.academia.edu/15757026/SOCCER_VIDEO_EVENT_DETECTION_VIA_COLLABORATIVE_TEXTUAL_AURAL_AND_VISUAL_ANALYSIS (cit. on pp. 5, 7, 14).

[16] A. A. Halin, M. Rajeswari, and D. Ramachandram. "Overlaid text recognition for matching soccer-concept keywords". In: *Proceedings - Computer Graphics, Imaging and Visualisation, Modern Techniques and Applications, CGIV* (2008), pp. 235–241. DOI: 10.1109/CGIV.2008.34 (cit. on p. 14).

[17] *How Long do Tennis Matches Last? - My Tennis HQ*. URL: https://mytennishq.com/how-long-do-tennis-matches-last/ (cit. on p. 1).

[18] *How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification - MachineLearningMastery.com*. URL: https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/ (cit. on p. 32).

[19] *Image Transforms - Hough Transform*. URL: https://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm (cit. on p. 10).

[20] S. Israni and S. Jain. "Edge detection of license plate using Sobel operator". In: *International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016* (2016-11), pp. 3561–3563. DOI: 10.1109/ICEEOT.2016.7755367 (cit. on p. 7).

[21] H. Kong, H. C. Akakin, and S. E. Sarma. "A generalized laplacian of gaussian filter for blob detection and its applications". In: *IEEE Transactions on Cybernetics* 43 (6 2013-12), pp. 1719–1733. ISSN: 21682267. DOI: 10.1109/TSMCB.2012.2228639 (cit. on p. 10).

[22] J. Li et al. "New challenges in multimedia research for the increasingly connected and fast growing digital society". In: ACM Press, 2007, p. 3. ISBN: 9781595937780. DOI: 10.1145/1290082.1290086 (cit. on p. vi).

[23] T. Lindeberg. "Scale invariant feature transform". In: *Scholarpedia* 7 (5 2012), p. 10491. DOI: 10.4249/SCHOLARPEDIA.10491. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-62443 (cit. on p. 10).

[24] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/master/template.pdf (cit. on p. ii).

[25] W. L. Lu et al. "Learning to track and identify players from broadcast sports videos". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (7 2013), pp. 1704–1716. ISSN: 01628828. DOI: 10.1109/TPAMI.2012.242 (cit. on p. 10).

[26] J. Mao. "Tracking a tennis ball using image processing techniques". In: (2006). URL: https://harvest.usask.ca/handle/10388/etd-08302006-125935 (cit. on p. 11).

[27] *OpenCV: How to Use Background Subtraction Methods*. URL: https://docs.opencv.org/4.x/d1/dc5/tutorial_background_subtraction.html (cit. on p. 23).

[28] *OpenCV: Morphological Transformations*. URL: https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html (cit. on pp. 15, 24).

[29] *OpenCV: Sobel Derivatives*. URL: https://docs.opencv.org/3.4/d2/d2c/tutorial_sobel_derivatives.html (cit. on p. 15).

[30] *OpenCV: Structural Analysis and Shape Descriptors*. URL: https://docs.opencv.org/3.4/d3/dc0/group__imgproc__shape.html (cit. on p. 15).

[31] *OpenCV: Thresholding Operations using inRange*. URL: https://docs.opencv.org/3.4/da/d97/tutorial_threshold_inRange.html (cit. on p. 24).

[32] N. Otsu. "THRESHOLD SELECTION METHOD FROM GRAY-LEVEL HISTOGRAMS." In: *IEEE Trans Syst Man Cybern* SMC-9 (1 1979), pp. 62–66. ISSN: 00189472. DOI: 10.1109/TSMC.1979.4310076 (cit. on p. 15).

[33] *Otsu's Thresholding Technique | LearnOpenCV*. URL: https://learnopencv.com/otsu-thresholding-with-opencv/ (cit. on p. 8).

[34] G. Pingali, A. Opalach, and Y. Jean. "Ball tracking and virtual replays for innovative tennis broadcasts". In: *Proceedings-International Conference on Pattern Recognition* 15 (4 2000), pp. 152–156. ISSN: 10514651. DOI: 10.1109/ICPR.2000.902885 (cit. on p. 11).

[35] G. S. Pingali, Y. Jean, and I. Carlbom. "Real time tracking for enhanced tennis broadcasts". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1998), pp. 260–265. ISSN: 10636919. DOI: 10.1109/CVPR.1998.698618 (cit. on p. 11).

[36] *RGB Image Prioritization Using Convolutional Neural Network on a Microprocessor for Nanosatellites*. URL: https://www.researchgate.net/figure/Variations-of-the-paddings-around-the-border-used-in-the-convolutional-layer-of-the-CNN_fig3_347339965%20[accessed%2028%20Sep,%202023 (cit. on pp. 3, 4).

[37] A. Rosenfeld. "Connectivity in Digital Pictures". In: (1970) (cit. on p. 10).

[38] J. Shi and Tomasi. "Good features to track". In: IEEE Comput. Soc. Press, 1994, pp. 593–600. ISBN: 0-8186-5825-8. DOI: 10.1109/CVPR.1994.323794 (cit. on p. 7).

[39] R. Smith. "An overview of the tesseract OCR engine". In: *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR* 2 (2007), pp. 629–633. ISSN: 15205363. DOI: 10.1109/ICDAR.2007.4376991 (cit. on p. 15).

[40] *Tennis Viewership Statistics 2023 | (numbers charts)*. URL: https://tennisracketball.com/guide/tennis-viewership-statistics/ (cit. on p. 2).

[41] M.-C. Tien et al. "Event detection in tennis matches based on video data mining". In: *2008 IEEE International Conference on Multimedia and Expo*. 2008, pp. 1477–1480. DOI: 10.1109/ICME.2008.4607725 (cit. on pp. 5, 7, 10, 12, 13, 15–17).

[42] E. Veroutsos. "The Most Popular Sports In The World". In: (2022-10) (cit. on p. 2).

[43] X. Wang. "Laplacian operator-based edge detectors". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (5 2007-05), pp. 886–890. ISSN: 01628828. DOI: 10.1109/TPAMI.2007.1027 (cit. on p. 7).

[44] *What is Machine Learning and How Does It Work? In-Depth Guide*. URL: https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML (cit. on p. 1).

[45] *What is Resolution? - All About Images - Research Guides at University of Michigan Library*. URL: https://guides.lib.umich.edu/c.php?g=282942&p=1885350 (cit. on p. 2).

[46] H. Xu and T. S. Chua. "Fusion of AV features and external information sources for event detection in team sports video". In: *ACM Transactions on Multimedia Computing, Communications and Applications* 2 (1 2006), pp. 44–67. ISSN: 15516865. DOI: 10.1145/1126004.1126007. URL: https://www.researchgate.net/publication/220214385_Fusion_of_AV_features_and_external_information_sources_for_event_detection_in_team_sports_video (cit. on p. 14).

[47] M. Xu et al. "Creating audio keywords for event detection in soccer video". In: *Proceedings - IEEE International Conference on Multimedia and Expo* 2 (2003), pp. 281–283. ISSN: 1945788X. DOI: 10.1109/ICME.2003.1221608 (cit. on p. 13).

[48] M. Xu et al. "Audio keyword generation for sports video analysis". In: *ACM Multimedia 2004 - proceedings of the 12th ACM International Conference on Multimedia* (2004), pp. 758–759. DOI: 10.1145/1027527.1027702. URL: https://www.researchgate.net/publication/221571405_Audio_keyword_generation_for_sports_video_analysis (cit. on pp. 13, 14).

[49] M. Xu et al. "Audio keywords generation for sports video analysis". In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 4 (2 2008-05). ISSN: 15516857. DOI: 10.1145/1352012.1352015. URL: https://dl.acm.org/doi/10.1145/1352012.1352015 (cit. on p. 12).

[50] F. Yan, W. Christmas, and J. Kittler. "A Tennis Ball Tracking Algorithm for Automatic Annotation of Tennis Match". In: *British Machine Vision Conference* 2 (2005), pp. 619–628. URL: https://openresearch.surrey.ac.uk/esploro/outputs/conferencePresentation/A-Tennis-Ball-Tracking-Algorithm-for/99511961802346 (cit. on pp. 10, 11).