Tennis Ball Tracking IACV

Felipe Azank and Felipe Bagni



Table of contents

01

02

03

Problem

Literature Review Methodology

04

05

06

Implement ation

Analysis & Results

Conclusions





01&02 Problem & Literature Review

Problem Description

Objective

Given an input video of a tennis rally in a standard view, obtain:

- The time (frame) in which the ball bounces in the court or is hit by a racket;
- The position of the ball in those frames.

This allows, for instance:

- Valuable match insights;
- Assess player performance;
- Strategize gameplay;
- Enhance training techniques.





Literature Review



Ball tracking

Previous works have demonstrated the effectiveness of CNN techniques in accurately identifying and following objects within video sequences, particularly in dynamic environments like sports. [2]

Court detection

Court detection was done using the Hough Transform. In this project, the OpenCV function cv2.HoughLinesP() was applied to detect the court lines. These detected lines are categorized into horizontal and vertical groups, after which lines corresponding to the same one are merged. [3][5]

Image rectification

The literature behind this approach was based on what was seen during the Lessons and Practices of the discipline itself. [1][4]



Methodology

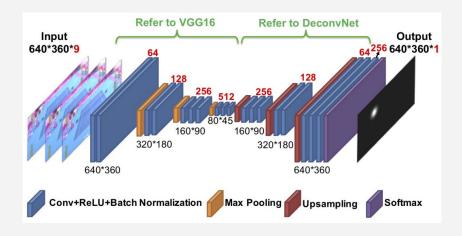
Tracking of the ball

TrackNet

In order to obtain the coordinates of the ball during the rally, we used TrackNet, a Convolutional Neural Network specialized on high speed ball detection in sports.

The output of this phase should be (for each frame):

 The coordinate X and Y of the tennis ball identified in that frame (w.r.t. the video, not the tennis court)





Bounce moment identification

Derivative Approach

A bounce is detected when the ball's vertical velocity (first derivative of position) reaches zero and its acceleration (second derivative) becomes positive, marking a local minimum as the ball hits the ground and bounces. Racket hits are identified by sudden changes in both the magnitude and direction of the velocity, signaling the ball has been struck.

The output of this phase should be:

• The frames in which a bounce or racket hit occurs

Neighbourhood Values Approach

The second approach applied a simpler solution, basically looking at which values of the X and Y coordinates (separately) the data shows a local minima. After that, a filtering of the peaks was applied to avoid the identification of the same bounce more than once



Court Identification

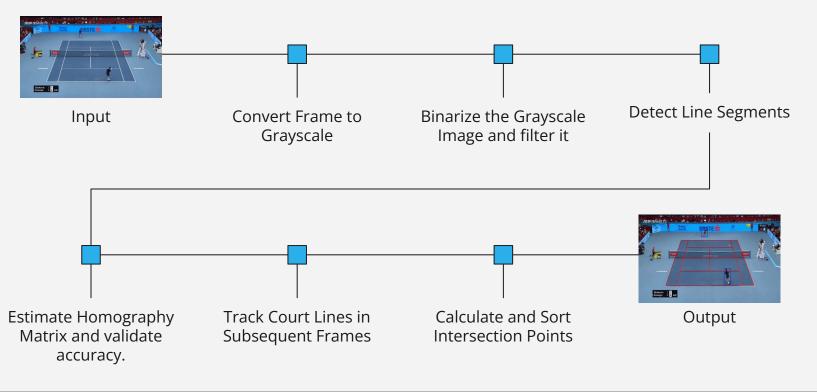


Image Rectification

Step 1 - Transformation Matrix

The transformation from the original coordinates to the top view were obtained using a transformation matrix based on the camera's intrinsic and extrinsic parameters, which helps to remove the effects of perspective distortion.

This matrix is one output of the Court Identification

Step 2 - Coordinate Transform

With the matrix M we can then obtain the rectified version by:

$$rectified_coord = M \times OG_coords$$

And then, obtaining the final coordinates by:

$$\mathbf{rectified_coord} = \begin{bmatrix} x' \\ y' \\ w \end{bmatrix} \qquad \begin{aligned} x_{\text{rectified}} &= \frac{x'}{w} \\ y_{\text{rectified}} &= \frac{y'}{w} \end{aligned}$$

The output of this phase should be:

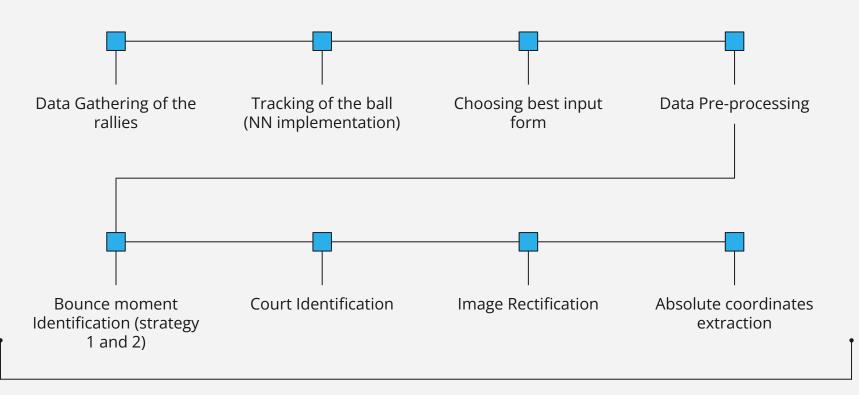
 The location of the ball in the rectified image on every frame in which a bounce was identified





O4 Implementation

Implementation Summary



Data Gathering of the rallies

Main Data Souce

The main data source for the project was the match between **Jannik Sinner and Daniil Medvedev** for the Australian Open 2024 Final. The full match video was obtained in Youtube and many "cuts" were made manually in order to select the "rallies" in which a considerable amount of bounces could be obtained.







Ball Tracking and coordinates

TrackNet Implementation

For the tracking of the ball, we used an existing implementation of TrackNet available on GitHub. Several modifications to the original code to better suit our project requirements. Specifically, we adjusted the code to extract the full coordinates of the ball from each detected frame.



Total params: 10,719,104 (40.89 MB)
Trainable params: 10,707,744 (40.85 MB)
Non-trainable params: 11,360 (44.38 KB)

Layer (Type)	Delpel Maye	Parse
input_layer (toputinger)	(box, x, 300, 500)	
	(More, 64, 760, 640)	1,5
activation (Activation)	(tore, 60, 360, 680)	
matificarealization	(bore, 66, 565, 840)	2,1
(MatcMares) station) condt_1 (torvitt)	(6000, 60, 580, 600)	36,0
activation 1 (Activation)	(block, 60, 580, 600) (block, 60, 260, 600)	
Batch corealization 3 (Matchesonalization)	(Nice, 64, 365, 640)	2,1
(Mancheseallindian) Manchese (Mancheses)	(None, 64, 386, 586) (None, 64, 386, 320) (None, 128, 386, 320) (None, 128, 386, 320)	
max_panting2g (*su*coling20)	(Mars, 44, 180, 330)	79.5
conduct a tronsal	(box, 128, 200, 210)	79,5
compd 2 (Convil) actualism 2 (Actualism) sant nerellisties 2 (Matchines)	(No. 120, 100, 200)	100
(Math/Mar/46) 1594 (40)		
constd_8 (forvill)	(Now, 128, 288, 229)	107,1
activation_8 (Artivation)	(Marry, 128, 188, 128)	
Marcilloreal Lization 3 (Marcilloreal Lization)	(Marr, 128, 188, 228)	1.89
man_pankingFE_1 (ManPankingFE) compde_4 (ccev22)	(tore, 120, 40, 160)	
comide_6 (conv22)	(boor, 25s, 40, 140)	299,1
activation 4 (Activation)	(box, 250, 50, 100)	
Batch cores linties 4 (Butcheres) linties	(Nov., 250, 90, 189)	
Interconnected that love		369,3
condt 5 (Condt) activistics 5 (Activistics)	(bine, 256, 60, 580) (bine, 256, 60, 580)	568,5
	(bone, 200, 90, 140).	
Satch neresization & (Matchesization)		
convint in (convint)	(Nove, 200, 90, 160) (Nove, 256, 90, 160)	384,8
activation & (Activation)	(Nice, 256, 90, 187)	
Batch cores) Iration 5 (National restion)	(tone, 256, 90, 180)	
man partings (Peneturings)	(Nove, 256, 81, 89)	
condit ((covit)	(book, 552, 45, 88)	1,189.1
conds f (ConvS) activation f (Activation)	(1000, 512, III, III)	
manch cormitration ?	(Mor, 112, 15, 81)	
convict & (Convict)	(1004, 112, 16, 10)	2.353.3
activation & (Artivation)	(mor, 512, 45, 89)	-
Setif rereally attem # (Mutchester)	(1004, 512, 15, 8H	
(RETCHOOM) LIM LOV)		
comdd # (convit)	(1604, A12, 45, 89)	-2,169,0
activation # (Activation)	(Nove, 512, 45, 89) (Nove, 512, 45, 89)	
Batch_coreslitation 9 (Butcheres) tations	(Nov. 512, 10, 88)	
up_compling2d (6)55000[limp20)	(Now, 552, 90, 580)	
com24_38 (Covv30)	(1604, 256, Wt. 180)	1,179,1
activation,18 (activation) batch normalization 38 (Matchingraphication)	(Micro, 200, 90, 160)	
Batch normalization DR (Matchester) 2245 2011	(Nov., 256, 40, 160)	
condit 11 (Covato)	(900e, 256, 90, 360)	349,1
activation 11 (artivation)	(tore, 25s, 90, 180)	
Satis corealization is conscionated tractions	(book, 250, 90, 160)	
(Bancaloreallist cos)		
condit 12 (Condit)	(Nov., 158, W., 188)	588,8
activation 12 (Activation)	(boos, 25s, 80, 150) (boos, 25s, 90, 160)	
batch normalization E2 (MEDCHARGES) LIACION)		
up_complicate_1 (synonylingth)	(Mare, 270, 500, 620)	
condd_12 (CovdD)	(Mare, 228, 200, 220)	295,1
Activation 12 (Activation) Earth rerealization 52 (Matcharran) 1785[an]	(boos, 130, 300, 120)	
Match nerealization 58 (Matchescop);sation)	(800, 120, 100, 100)	
comdit at (Covidt)	(Nove, 128, 286, 326)	147,1
condit of (Cowdo)	(None, 120, 200, 520) (None, 120, 200, 520)	
Extraction_14 (Activation) Eath_normalization_14 (BatcHornalization)	(bece, 120, 300, 120)	10
op compliance 2 (molecularity)	(boos, 129, 368, add)	
	(Nove, 66, 365, 642)	79.7
activation 15 (Activation)	(bine, 64, 780, 680)	14,0
batch careal institute th	(bice, 60, 500, 600)	2,0
Batch nerselization 25 (Watchernelization)		
condd_36 (Cess30)	(took, bit, bit, 640)	.00,1
activation,16 (Activation)	(hors, se, san, saly	
Nation cornelization 16 (National Continue)	(bose, bd; 880, b80)	2,1
condd_37 (Celv30)	(toos, 25s, 36s, 46s)	107,7
activation II (artivation)	(toos, 250, 300, 440)	
Satist pares) (ration_SP (Manchestation)	(Mor. 258, 368, 869)	2,5
(Butchernatization)		
restage (fectage)	(Nove, 276, 235881) (Nove, 23888, 264) (Nove, 23888, 264)	
permite (Permite) activation (# (Errovation)		

Trainable parass: 10,701,701 (40.85 Mb)

Choosing Best Input Form

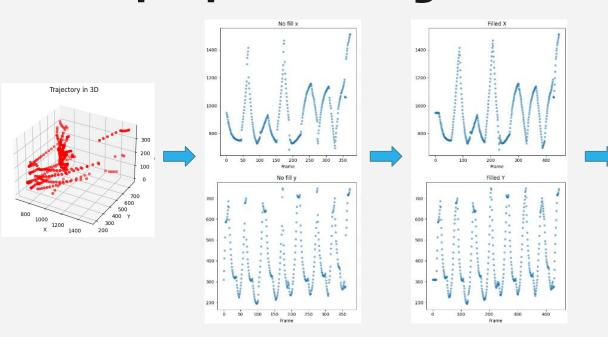
Best input for the tracknet

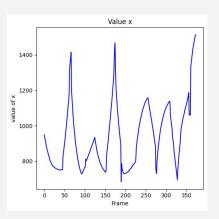
During the implementation, 3 methods were tested as the input. The metric to choose the best one was done considering the amount of identified frames.

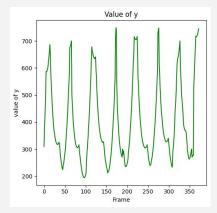


After some tests, the input that yielded the best results was the first one (given that it's the most similar to the train test of the Tracknet).

Data preprocessing



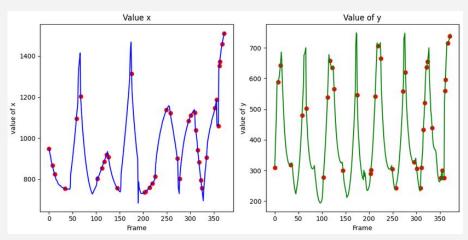




Bounce moment identification

Strategy 1

Using the gradient() function from NumPy to find the derivatives and find local minima using the change of sign from the second derivative of the positions where the first derivative equals zero.

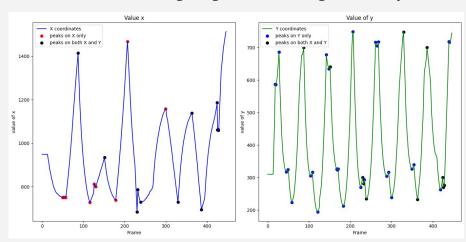


The results of this strategy were not as good as expected, due to inflection points introduced by fitting a continuous trajectory on the discrete points distribution.

Bounce moment identification

Strategy 2

Using the find_peaks() function from SciPy to find the peaks (trend change) in the coordinates and doing it again on the negative array to find the valleys of the data.



After obtaining the peaks, we applied a padding, ignoring peaks identified in a range of 4 frames.

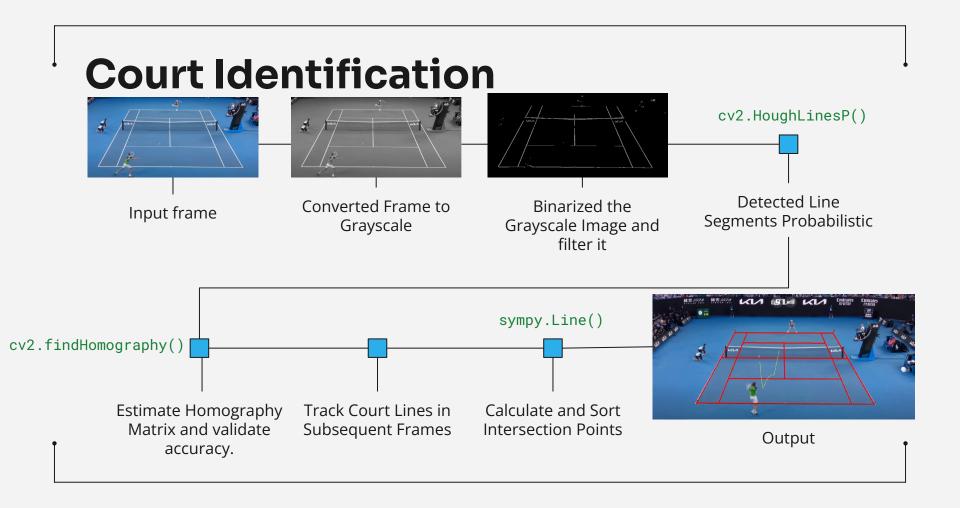
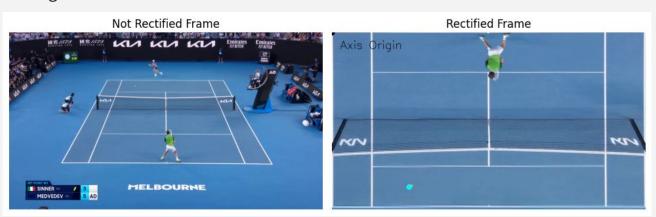


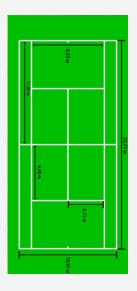
Image Rectification

From the output of the court detection, we could obtain the matrix

$$M = \begin{bmatrix} 5.05 & 2.79 & -3847.30 \\ 0.02 & -3.95 & 3122.66 \\ -0.000005 & 0.0030 & 1.00 \end{bmatrix}$$

That generated the rectification:



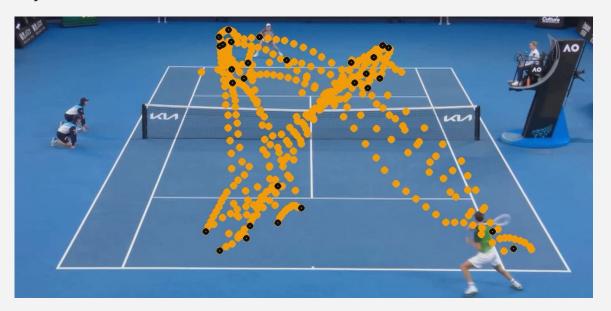




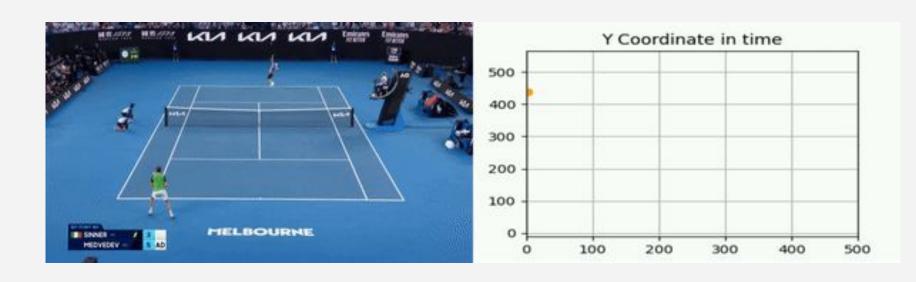
O5Analysis & Results

Tracking and bouncing results

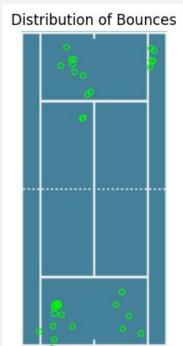
The tracking of the ball showed a quite good accuracy to the intuitive reality of the track. The bounces were correctly identified in the overall as well.

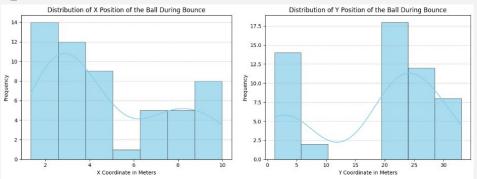


Tracking and bouncing results



Absolute position of the bounces





Padding	Amount of Bounces Detected
no padding	54
1	37
4	28
5	21
real	28



Conclusions

Conclusion



Key Takeaways

- The tracking of the ball was mostly satisfiable, as well as the data imputation;
- The bounce identification has a high precision (everytime it identifies a bounce, it's a bounce);
- Traditional IACV techniques are really handy for current problems.

Improvements

- Increase robustness of the system to make a good result in any tennis match recording;
- Make the system less dependable of padding and frame rate.

Future Features

- Further refactor the code used during the development in order to make it easier to other to use it;
- Use a bigger frame rate and better "find peak" strategy to improve the coordinate accuracy.

Bibliography

- [1] V. Caglioti. 2d reconstruction (image rectification). Lecture Slides, 2023. Course: Image Analysis and Computer Vision.
- [2] Y.-C. Huang, I.-N. Liao, C.-H. Chen, T.-U. İk, and W.-C. Peng. Tracknet: A deep learning network for tracking high-speed and tiny objects in sports applications, 2019. URL https://arxiv.org/abs/1907.03698.
- [3] A. Liberman. Highlights generation for tennis matches using computer vision, natural language processing and audio analysis, 2022. URL https://openworks.wooster. edu/independentstudy/9837. Paper 9837.
- [4] L. Magri. Image analysis and computer vision luca magri's website, 2023. URL https://magrilu.github.io/iacv/. Accessed: September 4, 2024.
- [5] SciPy Community. Scipy: Scientific computing tools for python, 2024. URL https://docs.scipy.org/doc/scipy/index.html. Accessed: 2024-08-29.



Thanks!

Do you have any questions?

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**