Main areas detection on tennis court

Rodrigo Alejandro Echavarría Echeverri
Juliana Arroyave López

Monografía presentada para optar al título de Especialista en Analítica y Ciencia de Datos

Asesora
Lina Maria Sepúlveda Cano, Doctor (PhD) en Ejemplo Ciencia de la Información

Universidad de Antioquia
Facultad de Ingeniería
Especialización en Analítica y Ciencia de Datos
Medellín, Antioquia, Colombia
2021

Cita	Arroyave López y Echavarría Echeveri [1]
Referencia Estilo IEEE (2020)	[1] J. Arroyave López y R. A. Echavarría Echeverri, “Main areas detection on tenns court”, Trabajo de grado especialización, Especialización en Analítica y Ciencia de Datos, Universidad de Antioquia, Medellín, Antioquia, Colombia, 2021.

Especialización en Analítica y Ciencia de Datos, Cohorte II.

Biblioteca Carlos Gaviria Díaz

Repositorio Institucional: http://bibliotecadigital.udea.edu.co

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.
Decano/Director: Jesús Francisco Vargas Bonilla.
Jefe departamento: José Luis Botia

Main areas detection on tennis court

Rodrigo Alejandro Echevarría Echeverri, Juliana Arroyave López

**Universidad de Antioquia, Facultad de Ingeniería*
Colombia (e-mail: alejandro.echavarria4@ udea.edu.co, juliana.arroyavel@gmail.com).

Abstract: this paper is going to show all the ETL process, which includes obtaining of the images, label process of images of tennis matches with the objective of detecting the main areas of the court. The process continues with the generation of a baseline using a heuristic process and the generation of a deep learning model, with modification on the characteristics of the images, that allows the detection of main areas on the tennis courts, as well as the pick of the best option for the task selected, based on their performance.

Keywords: semantic segregation, object detection, multi-class classification, Hough transform, U-Net, Deep Learning, Python

1. INTRODUCTION

The use of new technologies such as computer vision in different areas of the sports is becoming a technique widely used to obtain information from the gamers, the development of the matches, and other statistics that helps to obtain valued information for the coaches and the audience to see what is not visible in an easy way and is important to have a full picture of the development of the matches and the process followed by the players.

The information that used to be taken from sensors located on the clothes, rackets, bicycles, other equipment [1] now can be obtained from the images and videos of the matches, avoiding the need for devices with a single usage and specific infrastructure, this reduces the cost of the physical objects needed to get the information and their broadcast.

Computer vision had its first approach at the beginning of the 60's with the development of techniques to create three-dimensional solids based on two-dimension images [2], bringing about the beginning of this study field, where the main idea is to simulate the functionalities of a biological eye with a main objective: generate autonomous systems to accomplish specific task that was not possible to be performed before without the intervention of a human been.

The advances in this field significantly increased in the last years thanks to the development of new technologies such as the GPU, allowing the deployment and improvements of previous models and architectures. The Convolutional Neural Networks (CNN) led the growth of Computer Vision.

2. DATASET DESCRIPTION

The dataset consists of 400 images selected from different videos as Wimbledon semi – finals in 2015 [3], Mario Tennis presentation [4], professional doubles match [5], and other amateurs' videos [6][7][8]. These videos were selected according to the possible variety of images that may be taken from each, the different kinds of courts (grass, brick dust, synthetic and digital generated), the number of players and their positions over the court that may causes discontinuities and occlusions, the angle of the cameras shot and image quality

2.1. Images Obtention

The video was downloaded using a Python script that takes the URL of the video and saves every one of the photograms of the video on a folder selected in RGB format and named with the number its position on the video. In This process was obtained many useless pictures, some of them did not show the court or did not provide variety to the full collection, reason why even if we have more than 2000 images per video, the number of the items selected from each one is no longer than 80, having cases where was just take tree images per video.

The process of picking which images would be used started as a random pick of the images from the first downloaded videos. Once the most common ones were selected, the images that only showed a part of the court or where the players were over the areas cropping the continuity of them were added.

Fig 1. a, Image taken from the video of the semi-final of Wimbledon 2015, the court's ground is made on grass in lines and we have a full view of it. b,

Image from a doubles game for Roger Federer and Andy Murray, the court's ground is synthetic and is only visible a little part of it. c, Mario Tennis presentation, the court ground is digitally generated trying to simulate grass and is only visible as a part of it. d, Synthetic court from an educational video, some parts of the external areas are not visible.

2.2. Labeling

Considering that this is a classification exercise and that the techniques that are going to be used belong to the supervised learning ones, the images selected need to be labeled. In order to do this task was used the tool Label Studio [9].

Label Studio is an open-source annotation tool, allows different types of labels as bounding boxes, classification, semantic segmentation, used in an easy web interface as a localhost, allowing easy and fast usage. The labels generated may be exported on different formats as CSV, JSON, COCO according to the need.

The labels for this project were constructed with a semantic segmentation method. Each pixel of the image is associated with a certain class label. In this case, the possible classes are:

- lat: the areas that belong to this class are the ones on the external side and go across the court.
- ext: these are the areas that are on the external side of the court, contain the service line, are delimited by the lat areas.
- int_a: internal areas that are located on the left side of the image.
- int_b: internal areas that are located on the right side of the image.

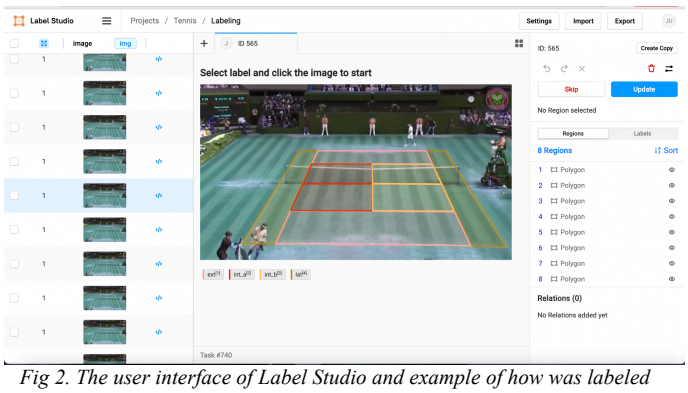

Fig 2. The user interface of Label Studio and example of how was labeled the images used on the project. On the bottom are the possible classes and on the right side are the regions marked on the image.

3. BASELINE

As the first approach to the solution and as on the aim to verify if a deep learning method to identify the areas of the courts was implemented, a technique based on an iterative approach called Hough transform.

3.1. Hough Transform

The Houhg transform is a feature extraction method mainly used on image analysis and computer vision with the objective of finding instances of figures that can be defined using mathematical expressions as lines, circles and ellipses using a voting procedure. consisting of the search of the parameters of the line on polar coordinates [10].

The Hough Transform looks for the edges of the images, calculates the possible ρ for all the θ on the point, and then looks for a new position that increments the values on the Accumulator and picks the line where these two values are the highest.

In order to proceed with this feature extraction is not necessary to perform any preprocessing on the images or in the masks, the only process needed is to create the model using the function available on the library OpenCV [11] .

The first step was create a class where all the methods needed are implemented:

- fit: get the image and estimate the values of the parameters for the lines present on the image. Generated the accumulator that is a matrix where the data is needed to calculate the parameters ρ and θ that defines the lines and store it as a list of list, where each one represents the position of the line a grid created over the image.
- predict: with the image and the parameters calculated, the possible lines on the line are parametrized within the image.
- get_error: this shows the metric of how far are the lines from the ones that were previously labeled on the image; calculating the euclidean distance between this one and the result from the product method, also takes into account the slope of each one.

The next step is to use the class created and apply it to the images calling the method. The Figure 3 is an example of the results obtained with this implementation:

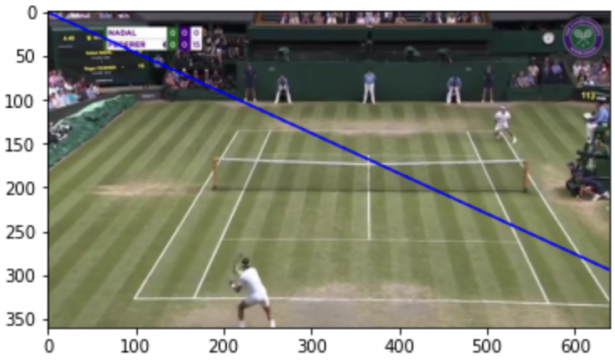

Fig 3. Example of the results obtained from the Hough Transform, the line detected on the image is influenced by the vertical lines on the grass that have different colors

From Figure 3, it is possible to see that the line found on the image does not match with any of the possible lines that

defines the court, not even close. With this kind of court where the ground has different colours or textures the method fails and seems to be affected by these special characteristics on the image. This represents a big issue at the moment to accomplish the objective set out. With this in mind, the process of developing a model based on deep learning starts with a new preprocessing of the data and other tasks described in the following sessions.

4. PREPROCESSING

To ensure that the data used will have the right format needed to develop the deep learning model, it is necessary to start with preprocessing tasks which includes the generations of the masks, augmentation of the images that are already on the dataset, and the process of transforming the images on the data type that is used by the models.

4.1. Mask Generation

In the process of generating the labels for the images, the objects that are marked on these are points that define the plane that delimits the region that surrounds all the pixels that belong to the class marked. With these points is necessary to create the surface that will mark all the pixels on the image.

Using the library Pillow and the method ImageDraw the polygons representing the masks over the image, using the points indicated on the label process escalated by a factor according to the size of the images to adjust the position of each point.

The final step is to concatenate each mask on a pandas dataframe that will be used on the next steps to create the final dataset.

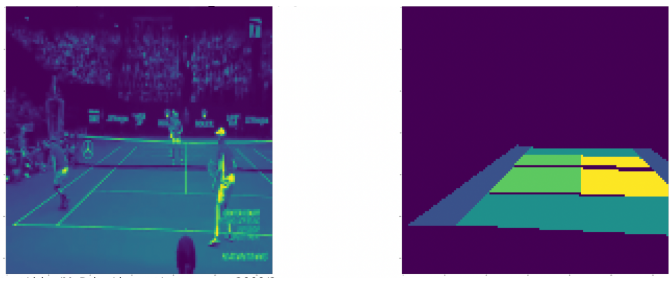

Fig 4. a, image from a tennis court. b, mask of the image where the different areas labeled are represented with the colours.

4.2. Dataset Creation

All the functions created to do the process on the data.py file are executed through one created explicitly with this objective. This functions include the one described after for the generation of the masks, deleting the useless columns (annotator id and the id of the images given by Label Studio), converting the information from strings to lists, get the path of the images, resizing the images, convert them on gray scales (take just one channel) with a custom size and build the dataset with the proper format and type for each image.

The dataset returned has four columns: the path of the image, the points of the labels, the image, and the mask generated.

	image	label	img	mask
0	/content/drive/MyDrive/dataset/Mario_Tennis/76...	[{'points': [[44.48228437987371, 50.9232815710...	[[[116, 32, 12], [116, 32, 12], [125, 66, 26],...	[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
1	/content/drive/MyDrive/dataset/Mario_Tennis/76...	[{'points': [[44.52620318312166, 50.9589412710...	[[[116, 40, 0], [116, 37, 5], [120, 46, 18], [...	[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
2	/content/drive/MyDrive/dataset/Mario_Tennis/77...	[{'points': [[44.37888723481636, 50.8895233103...	[[[113, 49, 6], [114, 37, 5], [129, 44, 18], [...	[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
3	/content/drive/MyDrive/dataset/Mario_Tennis/77...	[{'points': [[44.408255268306, 51.09020674666...	[[[115, 51, 9], [109, 45, 2], [123, 32, 9], [1...	[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
4	/content/drive/MyDrive/dataset/Mario_Tennis/77...	[{'points': [[44.45885386156288, 51.0885789812...	[[[130, 80, 20], [115, 61, 5], [123, 36, 2], [...	[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...

Fig 5. Top 5 rows of the dataset

4.3. Data Augmentation

The number of images labeled may not be enough to train a model in the best way, which is why it is necessary to augment the number of images that are going to be used in the training process.

The tool used for this task is the library Albumentations [12] that brings many options to create new images based on the original ones, keeping in mind that the new images should represent the problem; otherwise the only effect achieved would be to introduce noise on the data.

The transformation selected are:

- Horizontal flip and Change on the brightness, which changes the orientation and the brightness of the image.
- Change on the brightness and Random Gamma, which changes the brightness and the luminance of the image.

With this process, the number of images available for the training of the model has increased to 1200 images.

These augmented images are stored on the RAM memory, this is the reason why every time that the function is run the data set changes but keeps the base.

5. MODEL IMPLEMENTATION

For deep learning, the U - net model was picked. This model has shown good performance on image segmentation.

The U-net architecture was developed at the Computer Science of the University of Freiburg as a fully convolutional neural network specialized in image segmentation with high precision for Biomedical tasks.

The architecture is composed of two paths: one that contracts the data, that is, a convolutional network; with ReLU activation function followed by *max pooling operations,* which generates a reduction of the spatial information and an augmentation of the feature information. The second path expands the spatial and feature information through continued convolutions. [13]

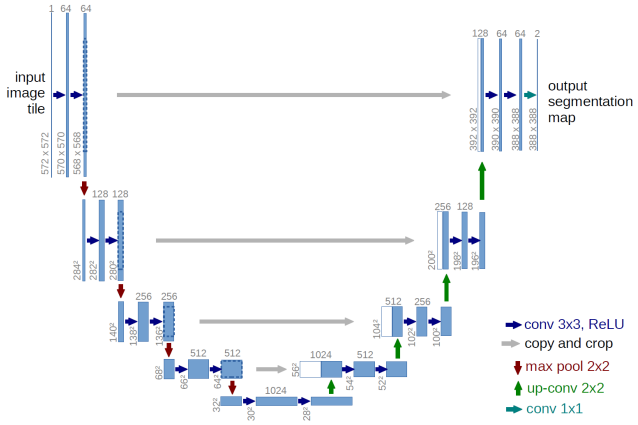

Fig 6. U-Net Architecture with an image size of 512 X 512 pixels [13]

This architecture has shown good performance in this kind of work, the reason why the variations to determine the best model with information transformation is going to be centered in this part of the possible variations and to verify the influence of the quality of the images used for the training of the model.

The metric used to measure the model's performance is the Sørensen-Dice Coefficient, also known as the F1 score. Is a statistical tool used to measure the similarity between two datasets defined by the formula:

$$QS = \frac{2|X \cap Y|}{|X| + |Y|} \qquad \text{(a)}$$

Where:
Y are the labels predicted by the model.
X are the validation labels.

For the first combination of possible image size and the number of channels considering only four classes, that means that the background is not taken as a possible option for the classification. The results obtained for these experiments are presented in Table 1.

Table 1. Results of the experiments for four possible classes, taking out the background of the image

Image Size	Channels	Training Loss	Validation Loss	Dice Coefficient
128 X 128	1	0,2484	0,2376	0,1172
128 X 128	3	0,3280	0,2959	0,0877
256 X 256	1	0,2635	0,2844	0,0835
256 X 256	3	0,2958	0,2886	0,0806
352 X 352	1	0,2870	0,2658	0,0838
352 X 352	3	0,2933	0,2884	0,0864

From the training loss and validation loss, it is possible to observe that the network was able to learn from the data, figure 7, still values of the Sørensen-Dice Coefficient are not the expected ones. Before reviewing the information, the conclusion is that the possible reason is because the background was considered, that is why the next step is to repeat the experiments but having five possible classes. The results of these new combinations are presented in table 2.

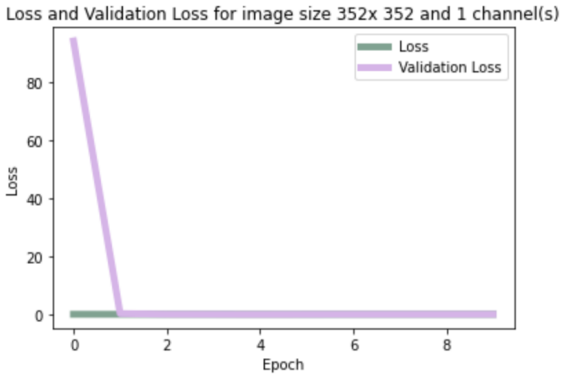

Fig 7. Evolution of Loss and Validation Loss for each epoch executed for images with size 352 X 352 pixels with one channel.

Table 2. Results of the experiments for five possible classes, taking the background of the image

Image Size	Channels	Training Loss	Validation Loss	Dice Coefficient
128 X 128	1	0,3157	0,2534	0,0714
128 X 128	3	0,2308	0,2470	0,7145
256 X 256	1	0,2785	0,2528	0,6103
256 X 256	3	0,3848	0,3116	0,6284
512 X 512	3	0,0540	0,0620	0,7891

For the images with size 512 X 512 it was necessary to upload the model to Azure and train it on the cloud due to the limitations present on the limited computing capacity that did not allow the local training for this experiment.

For this task was created a workspace configured with a virtual machine that allows running the training of the model for almost 21 hours continuously and teach the model how bigger and with more details does look like.

The results obtained on this experiment are also shown on Table 2 and in Figure 8 is the the performance for the best model obtained from this experiment. Also the evolution of the loss on training and validation is on Figures 8 and 9.

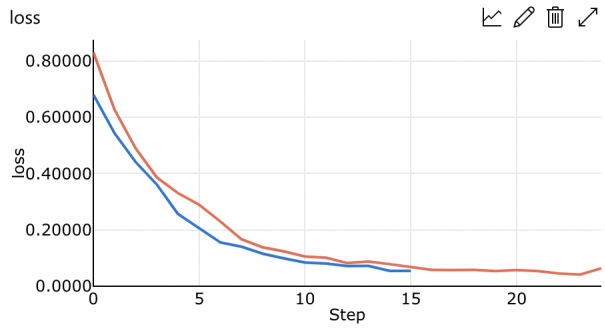

Figure 8. Training Loss for the training on Azure with images with size 512 X 512 X 3

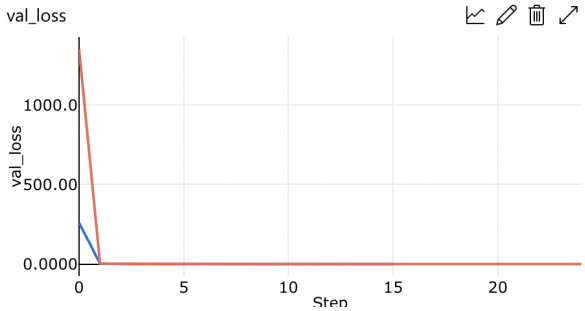

Figure 9. Validation Loss for training on Azure with image size 512 X 512 X 3

According to the Sørensen-Dice Coefficient, the best model available is the one trained with images that are 512 X 512 X 3. Using this model and a image that was not in the dataset, the prediction obtained is shown on the Figure 10

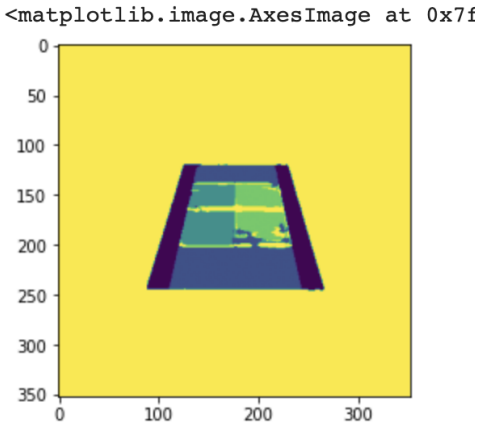

Figure 10. Result obtained from the best model according to the development metrics using an image that is not present on the dataset.

From the image it is possible to verify that the general shape of the court is well defined, in the same way for the lateral areas and the external ones. It had more difficulties to classify all the pixels on the internal areas, but most of them are identified with the corresponding class.

All the code and developed for this project is stored on the github repository:

https://github.com/rechavar/pyTennis

And for the Azure model is on:

https://github.com/rechavar/py_tennis_mlcould/tree/master

6. CONCLUSIONS

- The Hough Method is influenced by the differences of the colour and texture on the image, as is shown on the Figure 3, where the line detected on the image does not match with any of the possible ones that demarcated the court, influenced by the difference in the colours on the grass.

- The quality of the images has a big impact on the performance of the deep learning model, especially for the ones that use semantic segmentation method, where every pixel must have a class, and their correct classification has a big inference on the general result.

- For this specific situation the colours on the images do not give useful information for the training of the model. The performance metrics don't show a significant improvement for the experiments where the tree channels were present in comparison with the ones where just one was taken.

- Having good values of loss and validation loss during the training does not ensure the good performance of the model, but the evolution of this value shows that the neural network learns from the data.

- Even if the time of training on a cloud platform may be longer than it was expected, this paradigm gives the possibility of using more robust devices for a longer time and at the same time the training of more robust and complex models.

7. REFERENCES

[1] D. Morris et al., "Wearable sensors for monitoring sports performance and training," 2008 5th International Summer School and Symposium on Medical Devices and Biosensors, 2008, pp. 121-124, doi: 10.1109/ISSMDBS.2008.4575033.

[2] Roberts, Lawrence G. 1963. Machine perception of three-dimensional solids. MIT Libraries. https://dspace.mit.edu/handle/1721.1/11589

[3] Roger Federer and Andy Murray, tennis players, "The Best Game Ever? Murray vs Federer", YouTube, Jul 13, 2015 [Video File]. Available: https://www.youtube.com/watch?v=CGRzfUccmN

[4] Mario Bros characters, "¡la raqueta maldita! - #01 Mario Tennis Aces en Español (Switch) DSimphony", YouTube, Jun 22, 2018 [Video File]. Available: https://www.youtube.com/watch?v=CGRzfUccmNE

[5] Roger Federer and Rafael Nadal, “The Day Federer and Nadal Played Doubles Together (60FPS)”, YouTube, Sep 22, 2018 [Video File].Available: https://www.youtube.com/watch?v=sGX0M22xhPw

[6]Unknown, “The BEST TENNIS COACHING APP - SwingVision Tennis App Review”, YouTube, Sep 2, 2020, [Video File]. Available: https://www.youtube.com/watch?v=pO8lRZETS2A

[7] Novak Djokovic,“Rocío Amadio Acapulco,Novak Djokovic”, YouTube, Mar 4, 2017 [Video File]. Available: https://www.youtube.com/watch?v=-48Xzm1XAFk

[8] Unknown, “Perfect Forehand in 3 Easy Steps - Tennis Forehand Technique Lesson”, YouTube, May 9, 2020, [Video File]. Available: https://www.youtube.com/watch?v=5arVdubK9Pg

[9] “Label studio – open source data labeling,” *Label Studio – Open Source Data Labeling*. [Online]. Available: https://labelstud.io/. [Accessed: 24-Oct-2021].

[10]Shapiro, Linda and Stockman, George. "Computer Vision", Prentice-Hall, Inc. 2001

[11] OpenCV, HoughLines Transform, 2016, [Web], Available on: https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_houghlines/py_houghlines.html

[12]A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, “Albumentations: Fast and Flexible Image Augmentations,” *Information*, vol. 11, no. 2, p. 125, Feb. 2020.

[13] Ronneberger O., Fischer P., Brox T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28