Análisis de videos de baloncesto usando Deep Learning y Machine Learning

Ureta, Estuardo ure17010@uvg.edu.gt1, Graf, Oliver 171291.2, [1]Departamento de Facultad de Ingeniería, Departamento de Bioinformática, Universidad del Valle de Guatemala. 01 de mayo, 2021

***Abstract*— La detección y el seguimiento automatizados de jugadores en los juegos deportivos de equipo son de creciente importancia en los últimos años. A medida que las ganancias del entretenimiento de los deportes aumentan sustancialmente, los equipos invierten más en la recopilación de estadísticas sobre sus atletas. Ciertas estadísticas, como la distancia recorrida durante un partido, pueden proporcionar información sobre la salud del jugador, por ejemplo. Además, la detección de jugadores en tiempo real puede ser valiosa para identificar la formación y la estrategia del oponente, y puede dar una idea de la probabilidad de que una determinada jugada tenga éxito. Esto puede conducir a mejores estrategias y personalización de las mismas dependiendo del oponente. La automatización potencial del reconocimiento de los movimientos humanos, comúnmente conocida como reconocimiento de la actividad humana (HAR), se puede lograr a través de enfoques de modelos de aprendizaje profundo o de machine learning. Las entradas de datos comunes se obtienen a partir de unidades de medida inercial (IMU) o visión. En este estudio, la primera parte busca dar uso de herramientas open source para la detección de objetos. Así mismo la detección y seguimiento en tiempo real de la pelota y los jugadores tratando de generar datos útiles para el análisis en tiempo real de videos específicos de baloncesto. Posteriormente, en la segunda etapa, trabaja en un mapeo 2D a manera de estadístico ya que esto presenta mayor valor a un entrenador de baloncesto que, por ejemplo la velocidad de un jugador.**

Keywords—video analysis, detection, localization, deep learning

I. INTRODUCTION

El análisis del rendimiento en la ciencia del deporte ha experimentado cambios recientes considerables, debido en gran parte al acceso a la tecnología mejorada y al aumento de las aplicaciones de la informática (Wei-Lwun, 2001). El análisis de notación manual o la codificación en los deportes, incluso cuando lo realizan analistas capacitados, tiene limitaciones. Estos métodos suelen requerir mucho tiempo, son de naturaleza subjetiva y son propensos a errores y prejuicios humanos. Automatizar el reconocimiento del movimiento deportivo y su aplicación a la codificación tiene

el potencial de mejorar tanto la eficiencia como la precisión del análisis del rendimiento deportivo.

Nuestro proyecto es relevante porque es capaz de recopilar información sobre la posición de los jugadores en la cancha, así como datos relacionados con el estilo de juego de cada equipo, información que podría ser crucial para la victoria de los partidos, si bien analizado por el entrenador del equipo. De manera más general, nuestro proyecto podría adaptarse al seguimiento de los jugadores de cualquier partido deportivo.

II. LITERATURE REVIEW

Los sistemas inteligentes de análisis de video deportivo tienen muchas aplicaciones comerciales y han generado mucha investigación en la última década. Anteriormente, las personas se centraban en análisis de alto nivel, como el resumen de video (Cheng y Chiou-Ting, 2006) y el reconocimiento de eventos de disparo (Huang, Shih, y Chao, 2006). Recientemente, con la aparición de algoritmos precisos de detección y seguimiento de objetos, las tendencias se han desplazado hacia un análisis más detallado de videos deportivos, como el seguimiento e identificación de jugadores.

A. RCNN

a. Fast

No es más que un mecanismo de training que alterna el ajuste fino para las tareas de la propuesta regional y el ajuste fino para la detección de objetos.

El modelo Faster R-CNN se compone de dos módulos: una red convolucional profunda responsable de proponer las regiones y un detector Fast R-CNN que utiliza las regiones. La red de propuestas regionales toma una imagen como entrada y genera una salida de propuestas de objetos rectangulares. Cada uno de los rectángulos tiene una puntuación de objetividad. Clasificados y localizados mediante un cuadro delimitador y una segmentación semántica que clasifica cada píxel en un conjunto de categorías.

b. Masked

Mask R-CNN es conceptualmente simple: R-CNN más rápido tiene dos salidas para cada objeto candidato, una etiqueta de clase y un desplazamiento de cuadro abundante; a esto le agregamos una tercera rama que sale de la máscara del objeto. Mask R-CNN es, por tanto, una idea natural e intuitiva. Pero la salida de máscara adicional es distinta de las salidas de clase y caja, lo que requiere la extracción de un diseño mucho más fino de un objeto. A continuación, presentamos los elementos clave de Mask R-CNN, incluida la alineación píxel a píxel, que es la principal pieza que falta en Fast / Faster R-CNN.

B. Mapeo de la cancha vía homografía

Una homografía es una transformación en perspectiva de un avión (en nuestro caso, una cancha de baloncesto) de una vista de cámara a otra diferente. Básicamente, con una transformación de perspectiva, puede mapear puntos 3D en una imagen 2D utilizando una matriz de transformación.

Se identifica a los jugadores con Detectron2 y con los modelos Mask R-CNN puede identificar fácilmente los objetos en una imagen. Con tecnología PyTorch, este es un proyecto de código abierto de Facebook, implementa algoritmos de detección de objetos de última generación. Es sorprendente lo que puede detectar.

Necesitaremos filtrar a las personas y, de hecho, trabajar solo con los jugadores que están en la cancha. La imagen usada tiene todos los jugadores agrupados porque es el comienzo de un juego, como resultado, solo se encontraron 8 de cada 10 jugadores.

El modelo de segmentación panóptica de COCO detecta el techo, las paredes y el suelo

y los colorea en consecuencia. Esta será una entrada muy interesante para la detección de la cancha. Detectron2 también es compatible con la estimación de la postura humana (detección de puntos clave) que usaremos en el futuro para clasificar las acciones de baloncesto de los jugadores.

D. Xception y Keras Aplications; Xception es una red neuronal convolucional que tiene 71 capas de profundidad. Puede cargar una versión pre-entrenada de la red entrenada en más de un millón de imágenes de la base de datos ImageNet (Vera-Rodriguez et al., 2019).

La red previamente entrenada puede clasificar imágenes en 1000 categorías de objetos, como teclado, mouse, lápiz y muchos animales y en este caso rayos-x. Como resultado, la red ha aprendido representaciones de características ricas para una amplia gama de imágenes.

La red tiene un tamaño de entrada de imagen de 299x299. Para redes más capacitadas en MATLAB, por ejemplo. Las aplicaciones de Keras son modelos de aprendizaje profundo que están disponibles junto con pesos previamente entrenados. Estos modelos se pueden utilizar para la predicción, la extracción de características y el ajuste detallado. Los pesos se descargan automáticamente al crear una instancia de un modelo (SINGH, 2019).

E. Tensor Flow: Tensor Flow es una plataforma de código abierto de extremo a extremo para el aprendizaje automático. Tiene un ecosistema integral y flexible de herramientas, bibliotecas y recursos comunitarios que permite a los investigadores impulsar el estado de la técnica en ML y a los desarrolladores crear e implementar fácilmente aplicaciones impulsadas por ML.

Tensor Flow fue desarrollado originalmente por investigadores e ingenieros que trabajaban en el equipo de Google Brain dentro de la organización de investigación de inteligencia de máquinas de Google para realizar investigaciones de aprendizaje automático y redes neuronales profundas. El sistema es lo suficientemente general como para ser aplicable en una amplia variedad de otros dominios (SINGH, 2019).

III. METHODOLOGY

Con el objetivo de utilizar video de deporte de una corta duración y utilizando APIs de reconocimiento de imágenes para así extraer características o features de un video genérico. Se realizó el proyecto en python ya que posee numerosas herramientas para trabajar con modelos de la rama de Machine Learning y Deep Learning. La API implementadas fueron OpenPose la cual esta disponible en línea y representa el primer sistema de multi-personas en tiempo real para detectar conjuntamente puntos clave del cuerpo humano, la mano, el rostro y el pie (en total 135 puntos clave) en imágenes individuales o frames en este caso. También se utilizó Tensorflow de Keras para generar un mask RCNN. Puesto que en el dominio del aprendizaje automático, se ha establecido que las CNN nos brindan una representación de características más precisa en comparación con otros métodos. Esta investigación se puede dividir en módulos:

1. Detección de la cancha: encuentra las líneas de la cancha
2. Detección individual: detecta personas
3. Clasificación de color: separe a estas personas en dos equipos

4. Seguimiento de jugadores: mantiene la información de las posiciones cuadro por cuadro.

5. Mapeo: traducir a una cancha 2D como estadísticos

Los datos del proyecto consistirán en múltiples videos de YouTube recortados para hacer nuestro análisis. Así como algunos video de partidos del equipo de baloncesto UVG y otros lugares. Principalmente, seleccionaremos videos en los que podamos ver todas las líneas principales de la cancha para realizar con precisión la homografía.

A. Detección de cancha:

En esta sección se utilizaron algunas imágenes de Youtube de partidos en la NBA como los utilizados normalmente. Estas imágenes se convirtieron inicialmente de modelo de color BGR a HSV (tono, saturación y valor). Luego nos enfocamos en el plano H con el fin de crear un modelo binario del sistema. Luego, se vio la erosión y dilatación de la imagen con el fin de deshacernos de los artefactos que no estaban relacionados con la cancha. Posteriormente, se exploro el uso del detector de bordes Canny para detectar las líneas en nuestro sistema. Finalmente, realizaremos la transformada de Hough para detectar las líneas rectas en el sistema. Las cuales son circuladas en rojo y luego con funciones de Python se pueden unir los puntos y generar un polígono que representa la cancha (Figura. 2). Todo esto para elegir alguna opción que nos permita dilucidar la cancha.

B. Detección de jugadores

Nuestro enfoque es utilizar redes neuronales convolucionales como TensorFlow. Se pueden encontrar más detalles sobre las diferentes arquitecturas de modelos en A 2019 Guide to Object Detection.

Mask R-CCN nos permite segmentar el objeto en primer plano del fondo como se muestra en este ejemplo de Mask R-CNN y la imagen a continuación.

Imagen 2. Imagen del video de output tras realizar la red neuronal convolucional

C. Detección y clasificación de equipos basada en colores

La detección de objetos localiza la presencia de un objeto en una imagen y dibuja un cuadro delimitador alrededor de ese objeto, en nuestro caso sería una persona. Con estas imágenes de cada persona localizada luego se puede evaluar utilizando Kmeans para separar en cluster a los colores de cada imagen en un archivo csv que muestra la cantidad de rojo, verde, azul en el especto de luz y así le da un porcentaje de probabilidad de estar en un equipo o el otro (Figura 3)

D. Mapping

Una homografía es una transformación en perspectiva de un avión (en nuestro caso, una cancha de baloncesto) de una vista de cámara a otra diferente. Básicamente, con una transformación de perspectiva, puede mapear puntos 3D en una imagen 2D utilizando una matriz de transformación. Existen numerosos ejemplos de Python sobre cómo usar el algoritmo de homografía OpenCV.

Básicamente se proporcionó un mini-mapa de los jugadores. Osea, una vista de arriba hacia abajo de la cancha con los diferentes jugadores representados como círculos de colores apartir de un video. Al tener las dimensiones de la cancha, podemos encontrar una matriz de homografía de 3x3 que se

calcula usando una transformada afín. Luego, la posición de cada jugador se multiplica por la matriz de homografía que los proyecta en la cancha modelo.

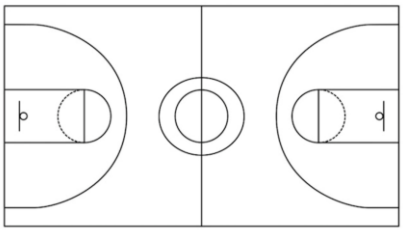

Imagen 2. Imagen donde se representarán los jugadores con puntos en vivo

La recuperación de la posición de cada jugador se logra utilizando el siguiente código de Python:

```
from shapely.geometry import Point, Polygon


color = [255, 0, 0]   # BLUE
thickness = 2
radius = 2


i  = 0
for box in pred_boxes:

  # Include only class Person
  if pred_classes[i] == 0:

    x1 = int(box[0])
    y1 = int(box[1])

    x2 = int(box[2])
    y2 = int(box[3])


    xc = x1 + int((x2 - x1)/2)
    player_pos = (xc, y2)


    court = Polygon(src_pts)
```

Imagen 3. Posiciones de jugadores extraído deLearning to track and identify players from broadcast sports videos Lu, Wei-Lwun

El método DefaultPredictor.predictor devuelve una lista de coordenadas rectangulares (pred_boxes) de cada objeto identificado. Las clases de objeto se almacenan en pred_classes, donde los objetos de persona se marcan como 0. Debido a que la detección automática de la cancha aún no está lista, tuve que proporcionar las coordenadas del polígono de la cancha manualmente.

Dibujaremos un círculo azul para cada jugador iterando sobre las coordenadas predicadas de los objetos encontrados (cajas). Solo deberíamos incluir objetos Persona que estén posicionados dentro de las coordenadas del polígono de la cancha.

IV. RESULTADOS AND DISCUSIÓN

A. Detección de cancha:

Tras generar múltiples alteraciones a la imagen ya mencionadas en la metodología notamos que la mejor alteración visualmente útil para identificar la cancha es la hace trabajo de resaltar las esquinas o edges como es llamada normalmente. Esta conciere a la primera imagen. Los puntos generados por la transformada de Hough no fueron precisos e incluso se salieron del rango deseado. Por lo tanto, este no fue un buen método para la obtención de las líneas de una cancha de baloncesto. Si los puntos de la Transformada hubieras sido precisos una simple función nos permitiría generar un polígono que representa la cancha en su posición respectiva.

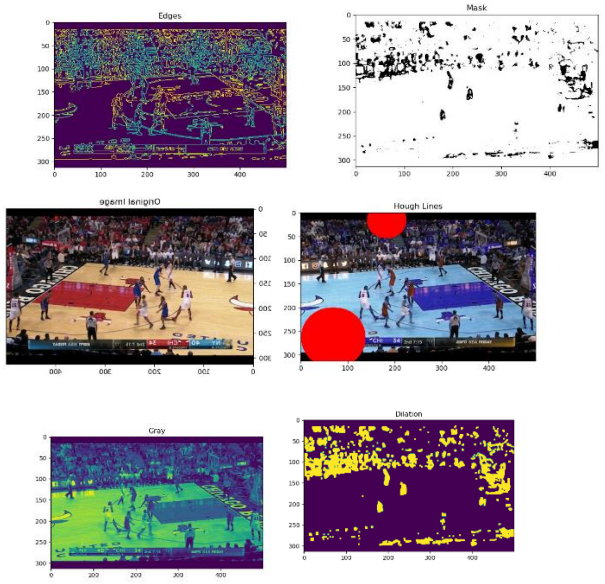

Imagen 4. Cada una de las alteraciones de imagen para la detección de la cancha.

B. *Detección de jugadores*

El modelo presentado en este documento es una extensión de la arquitectura Faster R-CNN descrita anteriormente. También permite la estimación de poses humanas.

En este modelo, los objetos se clasifican y localizan mediante un cuadro delimitador y una segmentación semántica que clasifica cada píxel en un conjunto de categorías. Este modelo amplía Faster R-CNN al agregar la predicción de máscaras de segmentación en cada región de interés. El Mask R-CNN produce dos salidas; una etiqueta de clase y un cuadro delimitador.

Este enfoque utiliza una red neuronal convolucional de retroalimentación que produce una colección de cuadros delimitadores y puntuaciones para la presencia de ciertos objetos. Se agregan capas de características convolucionales para permitir la detección de características en múltiples escalas. En este modelo, cada celda del mapa de características está vinculada a un conjunto de cuadros delimitadores predeterminados.

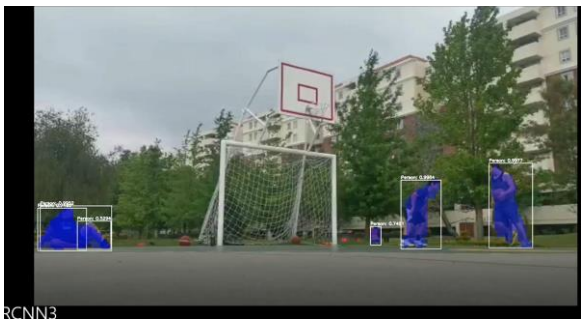

Imagen 6. Output3 tras realizar la red neuronal convolucional

Imagen 7. Imagen del video de output tras realizar la red neuronal convolucional

Imagen 5. Output2 tras realizar la red neuronal convolucional

C. Detección y clasificación de equipos basada en colores

Para reducir el número de detecciones de falsos positivos, utilizamos el hecho de que los jugadores del mismo equipo usan camisetas cuyos colores son diferentes a los de los espectadores, árbitros y el otro equipo. Específicamente, entrenamos un clasificador de regresión logística que asigna parches de imagen a etiquetas de equipo (Equipo A, Equipo B), donde los parches de imagen están representados por histogramas de color RGB. Tenga en cuenta que es posible agregar características de color al detector DPM y entrenar un detector de jugadores para un equipo específico. Sin embargo, requiere datos de entrenamiento etiquetados más grandes, mientras que el método propuesto solo necesita unos pocos ejemplos.

team	red	green	blue	percentage
team_a_4.jpg	148	144	76	41
team_a_4.jpg	4	4	3	35
team_a_4.jpg	87	85	69	23
team_a_3.jpg	132	125	89	34
team_a_3.jpg	6	5	3	47
team_a_3.jpg	210	206	149	18
team_a_2.jpg	161	162	146	20
team_a_2.jpg	0	0	0	60
team_a_2.jpg	146	138	63	18
team_a_1.jpg	2	1	1	52
team_a_1.jpg	183	176	64	25
team_a_1.jpg	156	148	129	22
team_b_4.jpg	2	1	1	73
team_b_4.jpg	135	128	120	12
team_b_4.jpg	71	66	59	13
team_b_5.jpg	114	107	100	22
team_b_5.jpg	6	6	7	75
team_b_5.jpg	254	254	251	1
team_b_2.jpg	90	84	75	18

Tabla 1. CSV de los colores de cada jugador encontrado a mano y su porcentaje de pertenecer a un equipo

D. Mapping

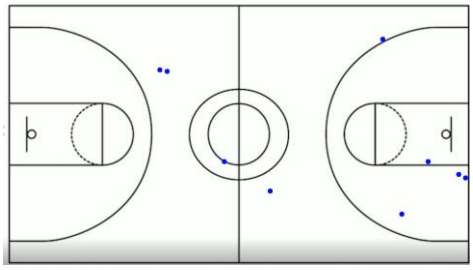

Imagen 8. Ejemplo de mini mapa en un segundo especifico.

El mapping fue un éxito pero para futuras practicas seria agradable realizar una aplicación la cual una ambos videos. Tanto el original como el representado en dimensión 2D. De esta manera se apreciaría más el cambio e tiempo real. No obstante si se logra dilucidar y conectar los movimientos del video original con el de las homografía.

V. CONCLUSION

1. Nuestro marco puede extenderse fácilmente a la estimación de poses humanas. Modelamos la ubicación de un punto clave como una máscara única y adoptamos Mask R-CNN para predecir Kmasks, una para cada tipo de Kkeypoint (por ejemplo, hombro izquierdo, codo derecho). Esta tarea ayuda a demostrar la flexibilidad de Mask R-CNN.

2. Recomendamos en futuras practicas realizar una app web que perita unir todas las herramientas. No obstante, este grupo de trabajo no esta familiarizado con dichas herramientas y tecnologías web por lo que no se implementó.

3. El mappeo 2D fue una muy buena manera de ver el flujo del partido y represnetta un aporte bastante algo para un entrenador de baloncesto actualmente.

REFERENCES

[1] Cust, E. E., Sweeting, A. J., Ball, K., & Robertson, S. (2018). Machine and deep learning for sport-specific movement recognition: a systematic review of model development and performance. Journal of Sports Sciences, 1–33. doi:10.1080/02640414.2018.1521769

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh,
S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei. “ImageNet Large Scale Visual Recognition Challenge”, *Inyernational Journal of Computer Vision*, vol. 115, issue 3, pp 211-252, 2015.

[3] S. Karen, and A. Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

[4] Wei-Lwun Lu, Jo-Anne Ting, James J. Little, Kevin P. Murphy,“Learning to Track and Identify Players from Broadcast Sports Videos,” IEEE transactions on pattern analysis and machine intelligence, 2011.

[5] Dollar, Piotr, et al. “Pedestrian detection: An evaluation of the state of the art.” Pattern Analysis and Machine Intelligence, IEEE Transactions on 34.4 (2012): 743-761.

[6] Scott Parsons and Jason Rogers, “Basketball Player Tracking and Automated

Analysis,” EE368 final project, Spring 2013/2014.

[7] Matthew Wilson and Jerry Giese, “Basketball Localization and Location Prediction,” EE368 final project, Winter 2013/2014..

[8] K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition" *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778

[9] Chih-Chieh Cheng and Chiou-Ting Hsu,Fusing of Audio anad MotionInformation on HMM-Based Highlight Extraction for Baseball Games, IEEETransactions on Multimedia8(2006), no. 3, 585–599.

[10] Chung-Lin Huang, Huang-Chia Shih, and Chung-Yuan Chao,SemanticAnalysis of Soccer Video Using Dynamic Bayesian Network, IEEETransactions on Multimedia8(2006), no. 4, 749–760.