

# Procesamiento de Video de un Partido de Fútbol

## Informe Trabajo Final

Facultad De Ingeniería, Universidad De Cuenca  
GRÁFICOS POR COMPUTADORA

Freddy L. Abad L., Edisson S. Reinozo T., David E. Santos L.

{[freddy.abadl](mailto:freddy.abadl@ucuenca.edu.ec), [edisson.reinozo](mailto:edisson.reinozo@ucuenca.edu.ec), [david.santos](mailto:david.santos@ucuenca.edu.ec)}@ucuenca.edu.ec

**Resumen-** En el contexto del mundo globalizado actual, donde una persona de un lugar recóndito puede tener acceso a un amplio repertorio de campeonatos de fútbol, surge la necesidad de procesar estos juegos para generar conocimiento en base a la información existente. El procesamiento de video conforma una disciplina amplia basada en el procesamiento de imágenes. Procesar un partido de fútbol significa identificar a los jugadores en campo, su número de camiseta, la ocurrencia de alguna falta de juego o la obtención de un gol. Proponer este trabajo significa mantener un ambiente controlado permitiendo así ya sea a periodistas, aficionados o cuerpo directivo mejorar en la toma de decisiones que concluyan en una mejor estrategia de juego. Este procesamiento de video se implementa en distintas fases: segmentación de la imagen, identificación de contornos, aproximación de objetos, rastreo a través de los frames de video, el dibujo de las secciones en movimiento y el análisis de faltas, goles, velocidad de jugador-balón. Cada fase, prevé diversos problemas los cuales se abordan en este informe. Estos problemas no son nuevos, por lo que en la actualidad existen metodologías detalladas, modelos pre entrenados, por lo cual este informe detalla su uso. Sin duda el campo de la visión por computadora, presenta muchas ventajas antes diversas situaciones en las cuales explorar y explotar. Todos estos se pueden abordar usando la API OpenCV para Python en 2D.

**Palabras Clave-** OpenCV, Contorno, Segmentación, Futbol, Tracking.

### I. INTRODUCCIÓN

En la actualidad el reconocimiento de elementos en un video es fundamental en el campo de la visión por computadora y sus derivados usos en la Inteligencia Artificial. Este proceso se puede realizar mediante, varios algoritmos y modelos ya desarrollados, la eliminación de “ruido”, el filtrado de elementos, el trazo e identificación de elementos en búsqueda [1]. Este informe propone reconocer jugadores de fútbol, identificarlos, así como a un balón y su trayectoria. El proceso de filtrado permite delimitar el área a analizar, así se tiene menor cantidad de datos a procesar, segregando las imágenes [1] que no aportan a los objetivos de esta práctica. Este proceso pertenece a un concepto mayor, el cual es la Segmentación de Imágenes. Elsevier (2016), define a la segmentación de imágenes como “el proceso de dividir una imagen digital en varias partes o grupos de píxeles u objetos”. Esto con el fin de simplificar la representación de una imagen o el procesamiento como “imágenes independientes”. Este concepto se usa para localizar objetos y para encontrar los límites de estos dentro de una imagen. [1]. Posterior a la segmentación de imágenes, se necesita identificar a los elementos del frame, mediante métodos desarrollados de alta precisión para el reconocimiento de por ejemplo una persona en un área específica, o caso contrario ignorarla, así mismo

la trayectoria que toma cada elemento, y analizar por ejemplo su velocidad.

### II. OBJETIVOS

#### A. Objetivo General

1. Procesar jugadores y balón (Object Track) en un video de fútbol.

#### B. Objetivo Específico

1. Utilizar metodologías para la segmentación de imágenes y reconocimiento de contornos apropiados para un entorno supervisado por capturas de video ya sea con una calidad alta o baja.
2. Identificar jugadores en movimiento y en acercamientos mediante funciones provistas por la API de OpenCV 2D para Python.
3. Identificar el balón y su velocidad mediante funciones provistas por la API de OpenCV 2D para Python.
4. Identificar un gol segmentando el campo de fútbol mediante funciones provistas por la API de OpenCV 2D para Python.
5. Optimizar el código fuente, utilizando prácticas de programación que faciliten el mantenimiento en el tiempo de este.
6. Medir la eficacia del algoritmo de rastreo ante uno o varios videos de prueba.

### III. MARCO TEORICO

Los conceptos teóricos, necesarios para entender la metodología y finalidad de este proyecto, se explican

a través de fases. Estas fases obedecen a las etapas abordadas por el objetivo principal, las cuales son explicadas en la sección IV.

A continuación, se definen los conceptos de cada algoritmo y metodología abordado.

#### A. Detección de jugadores en el campo de fútbol.

La fase de detección de jugadores en el campus de fútbol se realizó en dos fases: segmentación e identificación.

##### a. Segmentación de imágenes

En el campo de la visión artificial, la segmentación es el proceso de dividir una imagen digital en varias partes u objetos. Con la finalidad de eliminar las áreas innecesarias a procesar para determinado objetivo, además de “simplificar y/o cambiar la representación de una imagen en otra más significativa y más fácil de analizar”. [12]

Para realizar la segmentación de imágenes, existen múltiples métodos, entre ellos el detector de bordes mediante la delimitación por color [13]. Para tal fin, se usa el concepto de “espacios de color”, los cuales son una forma de representar los canales de color presentes en la imagen, que le dan a la imagen un matiz particular. Los espacios de color más populares son: RGB (rojo, verde, azul), CMYK (cian, magenta, amarillo, negro), HSV (tono, saturación, valor), entre otros. [2]

El espacio de color que se usa para la delimitación por color, es el HSV, el cual tiene la ventaja de que el canal de matiz (H) modela el tipo de color, lo cual conviene en tareas de procesamiento de imágenes con segmentación de objetos en función de su color. La variación de la saturación (S) puede entenderse como qué tan fuerte es el color que vemos; pasa de no saturada a representar sombras de gris y completamente saturada (sin componente blanco). El canal de valor (V) describe el brillo o la intensidad del color. [2] Para entender el esquema de color usado ver la figura 1.

El proceso de segmentación por color es realmente sencillo, se realiza definiendo rangos de color en HSV aplicando una máscara, así se puede recortar una porción de la imagen. El proceso de aplicar una máscara a una imagen para segmentar objetos se muestra en la figura 2.

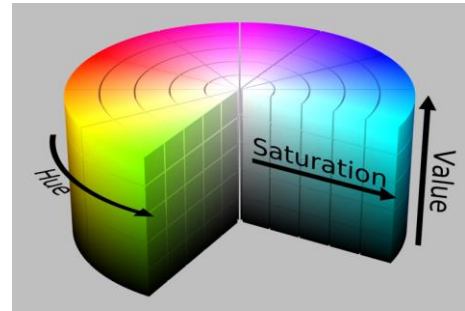


Figura 1: Espacio de Color HSV para la segmentación por color. Fuente: [2]

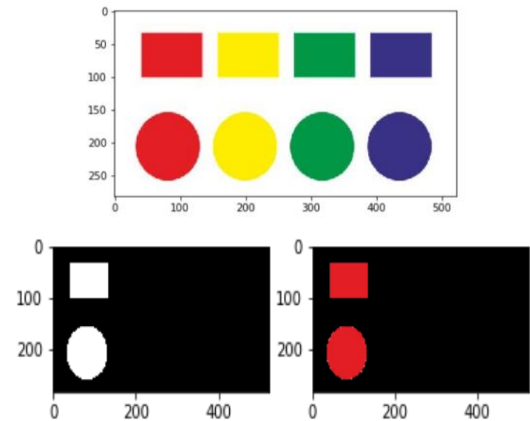


Figura 2: Segmentación de imagen mediante una máscara de color rojo HSV. Fuente: [2]

##### b. Identificación de personas

El proceso de identificar personas, parte del resultado de segmentar la imagen de interés, en el caso de estudio se segmentó el campo de juego por filtración de color. Para la identificación de personas, se usa como base el concepto de identificación de objetos. [3] En Machine Learning existen los siguientes métodos para lograr este objetivo:

1. R-CNN y sus variantes, Fast R-CNN y Faster R-CNN. La cual es una técnica de detección de objetos y jerarquías de funciones ricas en papel para la detección precisa de objetos y la segmentación semántica. Usa la búsqueda selectiva para extraer solo 2000 regiones de la imagen llamadas propuestas de región. Así, en lugar de tratar de clasificar una gran cantidad de regiones, puede trabajar con 2000 regiones. Así se optimiza el costo temporal de cálculo. Ver la figura 3

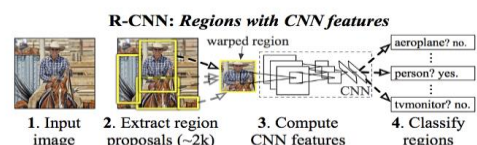


Figura 3: R-CNN para la identificación de objetos. Fuente: [4]

2. YOLO (You Only Look Once), a comparación de otros algoritmos de

detección de objetos que usan regiones para localizar el objeto dentro de la imagen. YOLO identifica la imagen en su totalidad, donde una única red convolución predice los cuadros delimitadores y las probabilidades de clase para estos cuadros. YOLO es más rápido (45 fps) que otros algoritmos de detección de objetos. La limitación de este es la identificación de objetos pequeños dentro de la imagen. El proceso de identificación se puede observar en la figura 4.

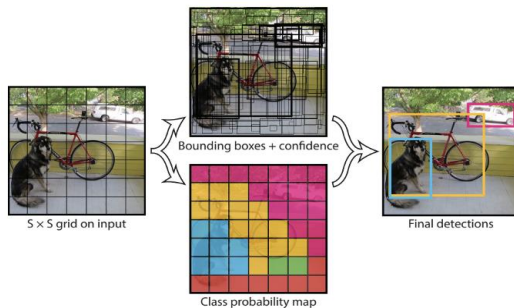


Figura 4: YOLO para la identificación de objetos. Fuente: [4]

#### c. Identificación de equipo

La identificación de equipo parte del segmento de imagen identificado como persona en el proceso anterior. Obtenido este segmento, se debe identificar el equipo mediante el color de camiseta, dado esto como proceso se asemeja a la delimitación del campo de juego. El proceso de segmentación por color, se hace con los mismos principios de esquemas de color HSV que se identificó en la sección III.A.a.

#### B. Identificar los rostros de jugadores en acercamientos.

El reconocimiento facial es uno de los campos más estudiados de la visión por computadora y la inteligencia artificial. Dado esto, tiene múltiples métodos los cuales aplicar para obtener los resultados requeridos.

Algunos de los métodos que se pueden utilizar son: Redes Neuronales Convolucionales (Clasificación de Imágenes), PCA - Principal Component Analysis, PINGO, Análisis de Textura de Piel, entre otras.

El caso de estudio implementa Deep Learning, mediante una API basada en DLIB (Herramienta desarrollada en C++ con algoritmos de aprendizaje automático [5]) llamada Face Recognition. Esta API permite reconocer y manipular rostros de manera sencilla. El modelo tiene una precisión del 99,38%. Su uso parte de una imagen referenciada que identifique a la persona a reconocer, esta herramienta permite optimizar su reconocimiento mediante modelos

pre entrenados con rostros similares a los de entrada. [6] Ver imagen 5.

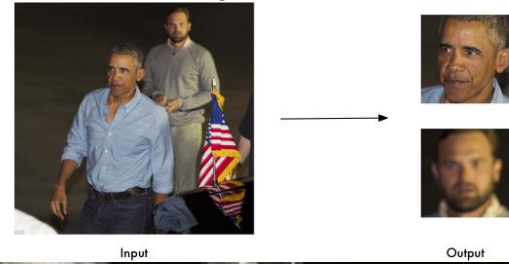


Figura 5: Identificación de rostros y asignación de etiquetas mediante la API face\_recognition.

Fuente: [6]

#### C. Identificar el número de la camiseta de los jugadores.

La identificación de número de persona parte de la fase de identificación de equipo, dado que al pertenecer a un equipo el segmento de imagen puede tener un número que identifique al jugador. Esta técnica permite eliminar falsos positivos causados por terceras personas en el campo, o jueces de línea - árbitros. La identificación de números puede darse por múltiples métodos partiendo por redes convolucionales que clasifiquen con una data de entrenamiento, o con métodos como OCR.

- OCR- Optical Character Recognition, Reconocimiento óptico de caracteres: Este es un método para la digitalización de textos a partir de una imagen y obteniendo símbolos o caracteres que pertenecen a un determinado alfabeto. [7] Ver imagen 6. Su funcionamiento se basa en dos subprocesos: reconocimiento de patrones (reconocimiento de los caracteres en su totalidad) o detección de características (detectando las líneas individuales y los caracteres de los trazos). [8] En el caso de estudio, al ser números impresos de mejor identificación se aplica el reconocimiento de patrones, mediante modelos pre entrenados de las herramientas utilizadas. Ver figura 6.

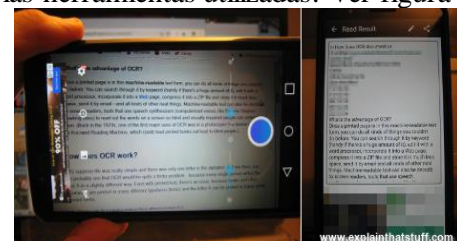






Figura 6: Proceso de reconocimiento de texto mediante OCR Fuente: [8]

#### D. Análisis de gol

La detección de un gol consta de una técnica que va en conjunto de otras adicionales como el rastreo del balón y la identificación de la guardería. Se considera como una anotación cuando el balón cruza completamente las líneas de demarcación de la portería.

Actualmente, existen sistemas dedicados a esta tarea dentro de algunas competencias futbolísticas, tal es el caso del sistema de Video Assistant Referee, que consiste en un conjunto de cámaras de video con software integrado que realiza el tracking de los elementos del juego. Cuando ocurre una controversia el árbitro principal puede acudir a este sistema, obtener retroalimentación de lo ocurrido y tomar una decisión con más contexto en lo acontecido. Este sistema es capaz de detectar si el balón ha ingresado a la portería ya que mantiene un modelado digital de todos los elementos del juego. Ver figura 7.



Figura 7: Reconocimiento de anotación del sistema VAR. Fuente: [8]

#### E. Demarcación de campo

La demarcación de las líneas de campo de juego consiste en identificar las líneas que limitan el campo y trazar sobre ellas líneas digitales que dentro del software se encuentran posicionadas en un espacio de 2 dimensiones, junto con sus respectivas coordenadas. Esta fase ayuda a otros tipos de reconocimientos como es el caso del reconocimiento de una anotación.

A través del modelo de la portería, y de la línea de su base, se puede identificar si las coordenadas del balón se encuentran dentro de la zona de anotación.

En otros sistemas como es el caso de VAR, esta demarcación se utiliza en combinación con técnicas de trazado para identificar posiciones de los jugadores en fuera de juego.

Existen varios métodos para realizar este reconocimiento. El principal es a través de cámaras estáticas y suspendidas en la posición superior del campo de juego. De esta forma delimita esta zona y entrega un modelo al sistema que además se encuentra realizando otro tipo de procesamientos.

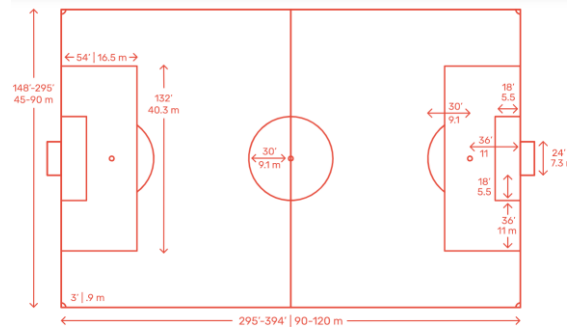


Figura 8: Reconocimiento de las líneas de demarcación del juego. Fuente: [8]

#### F. Identificación del Balón.

La identificación del balón en el campo de juego es igual al proceso de identificación de personas (jugadores) mediante el método YOLO detallado en la sección III.A.a. Este proceso parte de la segmentación de la imagen entre las secciones que pertenecen al campo de fútbol y las que no.

#### G. Registro de velocidad de velocidad.

El registro de velocidad parte de la identificación del balón, y para su cálculo hace uso del concepto de distancia euclidiana. Existen otros cálculos algebraicos que se realizaron y que se detallan en la sección IV-Metodología.

La distancia euclidiana corresponde a la distancia comúnmente calculada entre dos puntos de un espacio euclídeo, deducida a partir del teorema de Pitágoras [10] [11]. Esta ecuación se define por la fórmula representada en la figura 9 y 10

$$d_E(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

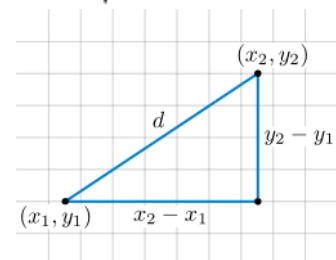


Figura 9: Distancia euclidiana y su representación gráfica. Fuente: [10]

$$\bar{v} = \frac{\Delta \mathbf{r}}{\Delta t}$$

*Figura 10: Fórmula de velocidad usada para el rastreo del balón. Fuente: [11]*

#### IV. METODOLOGÍA

El proyecto se implementó en distintas fases subyacentes, las cuales se definieron en la sección III (Marco Teórico), a continuación, se detalla su implementación.

##### A. Detección de jugadores en el campo de fútbol.

La detección de campo como se definió en la sección de Marco Teórico se realiza mediante la segmentación de imagen mediante la delimitación por color, primero del campo de juego y posteriormente del equipo al cual pertenece una persona. La identificación de personas se realizó mediante el método YOLO. Para la segmentación por color se identifica un rango de valores HSV que “discriminen” la imagen y filtre los objetos no necesarios. Los rangos en este caso se definen en el siguiente formato HSV Máximo - HSV Mínimo. La figura 11 muestra esta segmentación por delimitación de color. Las herramientas utilizadas para la delimitación de color son funciones proporcionadas por la API OpenCV 2D y numpy, aplicada en Python. Las herramientas utilizadas para el reconocimiento de jugadores son las proporcionadas por Redmon, Joseph and Farhadi, Ali [9] incluido los pesos y etiquetas requeridas e implementadas mediante Python. Ver figura 12, 13 y 14.

##### B. Identificar los rostros de jugadores en acercamientos.

El proceso de identificación de rostros de jugadores en acercamientos se identifica en las figuras 15. Este paso parte del proceso de segmentación e identificación de jugadores y equipo. Este proceso se realizó usando la API face recognition [6]. Previo a la aplicación de esta API se preprocesa el video, extrayendo los rostros a identificar y dado esto se tomó imágenes referenciales de la Web para el modelo de reconocimiento facial.

##### C. Identificar el número de la camiseta de los jugadores.

El resultado del proceso de reconocimiento de número de camiseta de los jugadores se muestra en las figuras 16. Para este proceso se empleó la herramienta Tesseract y PyOCR, estas dos herramientas permiten el reconocimiento de caracteres.

##### D. Análisis de gol

Para nuestro estudio se realizó un reconocimiento de las líneas de demarcación

junto con los postes verticales de la portería y se determinó a través de la formulación de funciones que definen estas líneas si el balón se encuentra o no dentro de esa sección. Se fijó un ángulo que determine si las líneas serán verticales de este modo junto con el preprocesamiento de la imagen y de la reconstrucción de houghlines se pudieron identificar los postes. En cada frame se identifican los postes y se evalúa si el balón está entre ellos. Ver figura 18.

##### E. Demarcación de campo

Para nuestro estudio nos valimos de filtros HSV para poder quedarnos únicamente con el campo de juego dentro de la imagen y posteriormente realizar el reconocimiento con la función de las líneas de Hough. Este módulo brinda soporte al módulo de verificación de la anotación ya que permite identificar las líneas horizontales de la base de la portería. A la imagen resultado de presentar únicamente la cancha se aplica el reconocimiento de líneas con el método houghlines, el cual identifica los contornos donde estaban las líneas de juego originales. Ver figura 17.

##### F. Identificación del Balón.

La identificación del balón usa el método YOLO de la sección IV.A. Su identificación parte de la segmentación de imágenes por delimitación de colores bajo el esquema de colores HSV. Esto permite identificar solo el balón que se encuentra en el campo de juego. A partir de esta segmentación, se busca los objetos similares a los “ball” que se encuentran en el modelo preentrenado de YOLO v3. El resultado de identificación de balón se detalla en la figura 19.

##### G. Registro de velocidad de velocidad.

Esta fase del parte de la identificación de balones, la velocidad es posible calcular partiendo de la fórmula detallada en la figura 20. Para tal, se debe tener una secuencia de frames que contenga el balón, donde se identificará las coordenadas del centro del balón, y la distancia recorrida será a partir de los píxeles recorridos y haciendo un proceso análogo a la distancia calculada en metros lineales. Esta distancia será tomada en cuenta además del tamaño promedio de un balón de fútbol (diámetro entre 21,65 y 22,29 cm). Ver figura 20. Dado esto, se puede obtener una distancia aproximada (mas no certera) del balón, esto sirve para además de calcular la velocidad, dejar una traza del recorrido del balón. Obtenido la distancia recorrida se procede con el cálculo del tiempo el cual se

cuenta desde el frame inicial donde se identifica el balón hasta el momento de calcular (proceso repetitivo). Ver figura 20.

## V. RESULTADOS

### Video Demostrativo:

<https://streamable.com/ins44d>

El resultado de la segmentación de campo de juego y de jugador se identifica en la figura

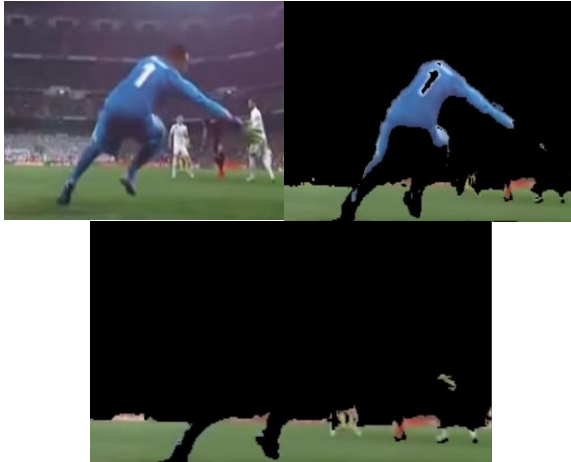


Figura 11: Imagen referencial e imagen filtrada con valores HSV.

Fuente: Autoría Propia

El resultado de la etapa “Detección de jugadores en el campo de fútbol.” se identifica en las figuras

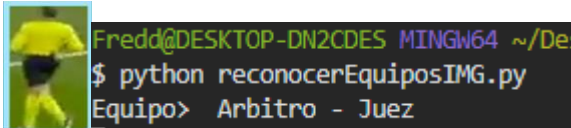


Figura 12: Implementación de YOLO para la identificación de personas además de segmentación por color con esquemas HSV para el reconocimiento de un árbitro o juez de línea.

Fuente: Autoría Propia

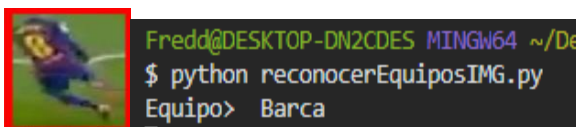


Figura 13: Implementación de YOLO para la identificación de personas además de segmentación por color con esquemas HSV para el reconocimiento de un jugador del BARCA. Fuente: Autoría Propia

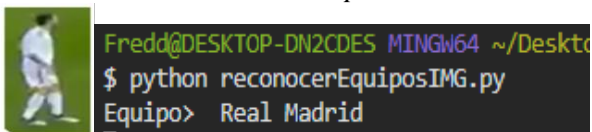


Figura 14: Implementación de YOLO para la identificación de personas además de segmentación por color con esquemas HSV para el reconocimiento de un jugador del Real Madrid. Fuente: Autoría Propia

El resultado de la etapa “Identificar los rostros de jugadores en acercamientos.” se identifica en las figuras.



Figura 15: Reconocimiento Facial. Fuente: Autoría Propia

El resultado de la etapa “Identificar el número de la camiseta de los jugadores.” se identifica en las figuras

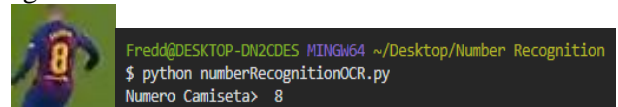


Figura 16: Uso de OCR para la identificación del número de camiseta. Fuente: Autoría Propia

El resultado de la etapa “Demarcación de campo” se identifica en las figuras.



Figura 17: Reconocimiento de líneas de demarcación. Fuente: Autoría Propia

El resultado de la etapa “Análisis de gol” se identifica en las figuras a través de la constatación frecuente del estado del balón frente a las líneas que limitan la zona de anotación



Figura 18: Reconocimiento de anotación. Fuente: Autoría Propia

El resultado de la etapa “Identificación del Balón” se identifica en las figuras



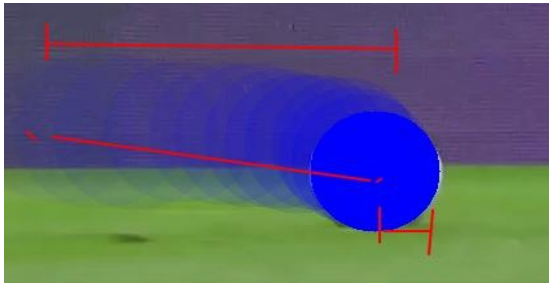


Figura 19: Reconocimiento de balón y traza de su trayecto. Fuente: Autoría Propia

El resultado de la etapa “Registro de velocidad de velocidad” se identifica en las figuras



Figura 20: Cálculo de la velocidad de un balón. Fuente: Autoría Propia

## VI. CONCLUSIONES

Este informe permitió una exploración teórica y práctica basta para tener un conocimiento de las herramientas existentes de visión por computadora para el procesamiento de video. Además, se puede concluir que:

- La metodología de segmentación por limitación de color permite delimitar áreas de procesamiento, disminuyendo la carga computacional, sin embargo, requiere tiempo adicional para encontrar los rangos HSV máximo y mínimos adecuados.
- La API face\_recognition permite un reconocimiento facial sencillo y rápido en base a su modelo pre entrenado, para tal se debe optar por imágenes claras y nítidas que permitan reconocer los rostros con mayor precisión.
- El uso de YOLO v3 permite la identificación de personas: jugadores, árbitros o jueces. El uso adecuado de este método de segmentación depende también de la reprocesamiento de cada frame, ya que en el caso de estudio se debió delimitar el campo de juego para eliminar los aficionados que se tomaban como falsos positivos.
- El uso de OCR para el reconocimiento del número de camiseta depende grandemente del tipo de imagen que se tiene, en el caso de estudio se obtuvo un buen rendimiento con imágenes con un contraste de colores opacos

pero un mal rendimiento con imágenes con contraste en colores claros.

- El resultado del reconocimiento de la anotación presenta gran inestabilidad principalmente por la calidad de imagen. El reconocimiento planteado depende de si se halló o no las líneas de demarcación de esta sección de anotación. Cambios en el brillo pueden afectar el filtro HSV de modo que el reconocimiento no sea posible.
- El reconocimiento de las líneas de demarcación es efectivo más no eficiente, tiene un alto consumo computacional.

## VII. RECOMENDACIONES

Este proyecto logró los objetivos principales y específicos, sin embargo, se identificó limitaciones, las cuales se detalla a continuación y se recomienda que hacer en estos casos:

- El uso de YOLO v3 es óptimo con el uso de una tarjeta gráfica de buena capacidad, puesto que el procesamiento que hace imagen por imagen requiere un gran costo computacional.
- El uso de OCR -Tesseract se recomienda con imágenes de equipos con uniformes opacos, ya que OCR tiende a no reconocer números o letras de imágenes claras.
- El uso de la API face\_recognition es recomendable para tareas sencillas, ya que permite una rápida y sencilla implementación, sin embargo, para tareas de reconocimiento facial más precisas se recomienda el uso de Redes Neuronales convolucionales para la correcta clasificación de rostros y la obtención de mayor precisión.
- No se recomienda el reconocimiento de una anotación a través de las cámaras con tomas con la finalidad de entretenimiento (TV) de fútbol. Debe existir un sistema dedicado a esta labor con cámara estáticas, esto debido a la importancia de tener continuidad en los frames contiguos.

## VIII. BIBLIOGRAFÍA

- [1] Elsevier. (2016, marzo 31). GPU-based Parallel Implementation of Swarm Intelligence Algorithms - 1st Edition. Recuperado 13 de mayo de 2020, de <https://www.elsevier.com/books/T/A/9780128093627>
- [2] Detección de objetos por colores en imágenes con Python y OpenCV. Aug 29, 2019. Gastón Di Giuseppe Fecha de consulta: 08:36, julio 12, 2020 desde <https://medium.com/@gastonace1/detección-de-objetos-por-colores-en-imágenes-con-python-y-opencv-c8d9b6768ff>

- [3] DAY 75-100 DAYS MLCODE: YOLO Object Detection. January 24, 2019. Pavan Tiwari. Fecha de consulta: 08:36, julio 12, 2020 desde <https://tekworld.org/2019/01/24/day-75-100-days-mlcode-yolo-object-detection/#page-content>
- [4] R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms. Rohith Gandhi. Jul 9, 2018 Fecha de consulta: 08:36, julio 12, 2020 desde <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>
- [5] DNN for Facial Recognition. Documentación Dlib. Fecha de consulta: 08:36, julio 12, 2020 desde [http://dlib.net/dnn\\_face\\_recognition\\_ex.cpp.html](http://dlib.net/dnn_face_recognition_ex.cpp.html)
- [6] FACIAL RECOGNITION. Adam Geitgey. 2017. Fecha de consulta: 08:36, julio 12, 2020 desde [https://github.com/ageitgey/face\\_recognition](https://github.com/ageitgey/face_recognition)
- [7] Reconocimiento óptico de caracteres. (2020, 26 de junio). Wikipedia, La enciclopedia libre. Fecha de consulta: 08:46, julio 12, 2020 desde [https://es.wikipedia.org/w/index.php?title=Reconocimiento\\_%C3%B3ptico\\_de\\_caracteres&oldid=127245235](https://es.wikipedia.org/w/index.php?title=Reconocimiento_%C3%B3ptico_de_caracteres&oldid=127245235).
- [8] Optical character recognition (OCR). Chris Woodford. March 28, 2020. Fecha de consulta: 08:46, julio 12, 2020 desde <https://www.explainthatstuff.com/how-ocr-works.html>
- [9] YOLOv3: An Incremental Improvement. Redmon, Joseph and Farhadi, Ali. 2018 Fecha de consulta: 08:46, julio 12, 2020 desde <https://pjreddie.com/darknet/yolo/>
- [10] Distancia euclidiana. (2020, 23 de mayo). Wikipedia, La enciclopedia libre. Fecha de consulta: 03:53, julio 13, 2020 desde [https://es.wikipedia.org/w/index.php?title=Distancia\\_euclidiana&oldid=126300410](https://es.wikipedia.org/w/index.php?title=Distancia_euclidiana&oldid=126300410).
- [11] Velocidad. (2020, 8 de julio). Wikipedia, La enciclopedia libre. Fecha de consulta: 03:59, julio 13, 2020 desde <https://es.wikipedia.org/w/index.php?title=Velocidad&oldid=127577716>.
- [12] Segmentación (procesamiento de imágenes). (2020, 17 de abril). Wikipedia, La enciclopedia libre. Fecha de consulta: 08:36, julio 12, 2020 desde [https://es.wikipedia.org/w/index.php?title=Segmentaci%C3%B3n\\_\(procesamiento\\_de\\_im%C3%A1genes\)&oldid=125275296](https://es.wikipedia.org/w/index.php?title=Segmentaci%C3%B3n_(procesamiento_de_im%C3%A1genes)&oldid=125275296).
- [13] Detector de bordes. (2019, 3 de diciembre). Wikipedia, La enciclopedia libre. Fecha de consulta: 08:37, julio 12, 2020 desde [https://es.wikipedia.org/w/index.php?title=Detector\\_de\\_bordes&oldid=121759705](https://es.wikipedia.org/w/index.php?title=Detector_de_bordes&oldid=121759705).