

Escuela de Postgrados



Maestría en Ciencia de Datos

Aprendizaje Profundo y Operaciones de Aprendizaje Profundo

Trabajo Final:

Entrenamiento de Modelos de Aprendizaje Profundo para la
Detección de la Pelota en Partidos de Handball

Integrantes: Alvaro Ciganda, Ana Romero y Silvia Motta

Profesor: Dr. Enrique Márquez

Montevideo, junio 2024

1- Introducción	3
Contexto y Motivación	3
Objetivo	3
Alcance y limitaciones	3
2. Revisión de Literatura	4
Conclusiones de la Literatura	4
3. Análisis de Datos	4
4. Modelos y Métodos	5
4.1 Modelo baseline YOLOv8s con entrenamiento en COCO	5
4.2 Segmentación avanzada con SAM	6
4.3 Generación de Datos Sintéticos para entrenar a YOLO	6
4.3.1 Elementos a insertar	6
4.3.2 Imágenes de fondo	7
4.3.3 Inserción de imágenes	7
4.3.4 Entrenamiento	8
4.3.5 Resultados	8
4.3.6 Análisis de errores	9
4.4 Entrenamiento de YOLO con datasets reales	10
4.4.1 Pruebas Realizadas	10
4.4.2 Evaluación en video de una nueva cancha	11
4.4.3 Evaluación del modelo elegido	12
Gráficas de Métricas de Entrenamiento y Validación	12
Matriz de Confusión	13
4.4.4 Resultados y discusión	13
4.4.5 Limitaciones y Desafíos	15
4.4.6 Análisis de Errores	15
5. Conclusiones	16
6. Referencias	17

1- Introducción

Contexto y Motivación

El análisis de contenido de imágenes y videos deportivos ha ganado relevancia tanto en la investigación como en la industria debido a su potencial para optimizar el entrenamiento de los deportistas y proporcionar indicadores valiosos para los entrenadores y analistas deportivos. En particular, el handball es un deporte rápido y dinámico, donde el seguimiento y la detección de la pelota juegan un papel crucial en la comprensión del juego y en la elaboración de estrategias.

En el marco de nuestro trabajo final de maestría estamos explorando la aplicación de técnicas avanzadas de aprendizaje profundo en la detección para el análisis de videos de handball en Uruguay, enfocándonos en la detección y el seguimiento de jugadores y la identificación de sus actividades. Uno de los desafíos más complejos es la detección de la pelota. La pelota de handball se mueve a gran velocidad y a menudo está parcialmente oculta o fuera de la vista debido a la interacción con los jugadores.

Objetivo

El objetivo de este proyecto es desarrollar y entrenar un modelo de aprendizaje profundo capaz de detectar la pelota en videos de partidos de handball. En particular partidos del ámbito uruguayo. Para lograr esto, se establecerán y medirán las siguientes métricas de rendimiento:

- mAP50: La precisión promedio media (mean Average Precision) a un umbral de 0.50 de Intersection over Union (IoU), que evalúa la precisión de las detecciones del modelo a un nivel de solapamiento básico entre la predicción y la verdad.
- mAP50-90: La precisión promedio media a múltiples umbrales de IoU entre 0.50 y 0.90, que proporciona una evaluación más robusta de la precisión del modelo.

Alcance y limitaciones

El proyecto se estructura en las siguientes fases:

- Colección y preprocesamiento de datos: Se recopilaron videos de partidos de handball uruguayos, los cuales serán procesados y anotados utilizando Roboflow.
- Obtención de datos baseline: Se evaluará la detección de la clase ‘sports ball’ del modelo You Only Look Once (YOLO) entrenado con el dataset COCO y la segmentación obtenida usando Segment Anything Model (SAM) contra las anotaciones realizadas.
- Entrenamiento de modelos: Se entrenará YOLO con datos reales y con datos sintéticos.
- Evaluación y comparación de modelos: Los modelos entrenados se evaluarán utilizando las métricas mAP50 y mAP50-90, comparando su eficacia en un conjunto de validación que no ha sido visto por los modelos durante el entrenamiento.
- Análisis de resultados: Se analizarán los resultados obtenidos para identificar el modelo más eficiente.
- Análisis de errores: se analizará si existe uno o varios patrones en los errores detectados para entender cómo mejorar la precisión de un modelo. En el contexto de detección de objetos, esto implica revisar las predicciones fallidas y detectar tendencias comunes en los errores.
- Mejoras: se explorarán posibles mejoras del modelo tomando como input los análisis de resultados y el análisis de errores.

Las limitaciones del proyecto incluyen el tiempo disponible para el desarrollo y el entrenamiento de los modelos, la calidad y cantidad de datos de entrenamiento y los recursos computacionales de GPU y memoria necesarios para el procesamiento y entrenamiento de los mismos.

2. Revisión de Literatura

El análisis de contenido en deportes, especialmente en escenarios de handball, presenta desafíos significativos debido a la naturaleza dinámica y rápida de estos juegos. En el artículo “Analysis of Movement and Activities of Handball Players Using Deep Neural Networks” de Host, Probar y Ivasic-Kos (Host et al., 2023) [1], se utilizaron redes neuronales profundas para la detección y seguimiento de jugadores, así como para el reconocimiento de sus acciones en videos de handball. Este estudio se centró en aplicar métodos como YOLO y MASk R-CNN para la detección de jugadores y pelotas, comparando diferentes configuraciones para seleccionar el mejor detector, y utilizando algoritmos de seguimiento como DeepSORT y BoT SORT para mantener la identificación de las jugadoras a través de los frames. Este trabajo es el más reciente que hemos encontrado sobre el tema y proporciona una base sólida y actualizada para abordar los desafíos planteados en el objetivo del proyecto. En sus resultados muestran que para la detección de la pelota obtuvieron valores AP de 23.07 utilizando YOLOv7 y de 35.44 utilizando YOLOv3 y expresan que sería poco probable que usar un nuevo modelo o reentrenar con más datos resolvería el problema de detección de la pelota.

Por otro lado, Alwi et al., (2020) [2] se enfocaron en la detección de la pelota de handball en un ambiente experimental. Evaluaron modelos de aprendizaje profundo como YOLOv2, YOLOv3 y Faster R-CNN para identificar la pelota en diversas condiciones: flotando, contra la red y diferentes tamaños.

Además, Wang et al., (2022) [3] proporcionan una revisión exhaustiva de cómo se han aplicado técnicas de visión por computadora y aprendizaje profundo en deportes como el fútbol y el baloncesto. Por ejemplo, el uso de YOLOv3 para detectar jugadores y la combinación con algoritmos de seguimiento como SORT y redes LSTM para el reconocimiento de acciones en baloncesto, muestran la eficacia de integrar la detección y el seguimiento en el análisis deportivo. Estas metodologías se han adaptado con éxito a diferentes deportes, demostrando la versatilidad de las arquitecturas de detección de objetos en datasets deportivos personalizados.

Conclusiones de la Literatura

La revisión de la literatura indica que la detección y seguimiento de objetos en deportes de equipo, como el handball, son áreas de investigación activas con aplicaciones significativas. Los modelos basados en redes neuronales profundas, como YOLO y Mask R-CNN, han demostrado ser efectivos para la detección de jugadores y la clasificación de acciones. Sin embargo, la detección de objetos como la pelota en handball sigue siendo un desafío debido a su tamaño, movimiento rápido y frecuente ocultación.

Los estudios revisados destacan la importancia de la personalización de modelos y la creación de datasets específicos para mejorar la precisión en tareas de detección en deportes. Además, la combinación de algoritmos de seguimiento y detección, así como el uso de modelos que consideran tanto la información espacial como temporal, son estrategias prometedoras para abordar estos desafíos. En resumen, la literatura revisada proporciona una base sólida para el desarrollo de modelos de detección de la pelota en videos de handball, subrayando la necesidad de adaptar y mejorar continuamente las técnicas de visión por computadora para obtener resultados precisos y robustos.

3. Análisis de Datos

En esta sección, se presenta un análisis de los datos utilizados en el proyecto. incluyendo su origen y características principales.

Para el proyecto, se recopilaron y procesaron tres datasets principales, cada uno proveniente de diferentes complejos de la Federación Uruguaya de Handball ([Complejos](#)) para proveer a los modelos de una variedad de condiciones de juego y entornos.

1- Test01 - Espacio Polideportivo Municipio G (Lezica), este dataset contiene 291 imágenes de resolución estándar 640 x 640.

2- Test02 - Espacio Polideportivo Municipio G (Lezica), este dataset contiene 945 imágenes de resolución estándar 1920 x 1080.

3- Test03 - Sporthalle Colegio Alemán: este dataset incluye 905 imágenes con alta resolución 1920 x 1080.

4- Test04- Scuola Italiana: comprende 1000 imágenes con resolución estándar 640 x 640.

5- COCO 2017 - Dataset de uso abierto en Computer Vision. Se utilizaron 10.000 imágenes como fondo en entrenamiento de datos sintéticos.

6- SINTÉTICA - A partir de imágenes de Test02 se recortaron imágenes con 30 visualizaciones distintas de la pelota.

Distribución de clases: dado que la pelota es el objeto principal de interés, hemos revisado la distribución de las imágenes para asegurarnos de que la pelota esté presente de manera significativa y en diferentes contextos (ej., en movimiento, parcialmente oculta, etc.).

Diversidad de entornos: los datasets se han seleccionado entre distintas categorías y equipos para representar diferentes condiciones de juego, lo que ayuda a minimizar el sesgo que podría resultar de entrenar el modelo en un entorno único.

Diferencias en origen: cada dataset proviene de una cancha diferente, lo que introduce variabilidad en términos de iluminación, fondo y condiciones de juego.

Esta diversidad es crucial para entrenar un modelo robusto que pueda funcionar bien en condiciones no vistas durante la fase de entrenamiento.

4. Modelos y Métodos

En esta sección, describimos los enfoques y técnicas utilizados para entrenar y evaluar el modelo YOLO para la detección de la pelota en videos de handball. La elección de YOLO como el modelo principal para este proyecto se basó en la revisión de la literatura. Varios estudios como se describen en la sección 2- “Revisión de Literatura” han demostrado que YOLO es altamente efectivo para tareas de detección de objetos en tiempo real debido a su capacidad de realizar detección rápida y precisa con un solo paso de inferencia.

Abordamos el problema desde tres líneas principales: la generación de datos sintéticos, el uso de datasets reales de manera independiente y combinada y la evaluación de datos segmentados con SAM. Cada enfoque se diseñó para explorar y mejorar diferentes aspectos del rendimiento del modelo en condiciones variadas. A continuación, se detallan los modelos y métodos aplicados en cada línea de trabajo.

4.1 Modelo baseline YOLOv8s con entrenamiento en COCO

Como modelo base se evaluaron las métricas de YOLOv8s entrenado con COCO con el dataset anotado de Test02. Con el editor de texto Notepad++ se reemplazó en todos los archivos el índice de clase 0 original por el

32 que corresponde a ‘sports ball’ en YOLO. Se obtuvo un valor de mAP50 de 0.522 y un valor de mAP50-95 de 0.204.

4.2 Segmentación avanzada con SAM

En esta etapa nos planteamos detectar la clase pelota utilizando conjuntamente dos herramientas muy ponderadas actualmente: GROUNDING DINO y SAM.

Lo primero que realizamos fue la configuración del ambiente, esto significó la instalación de la versión sam_vit_h para SAM (versión huge) y para la versión de Grounding Dino se utilizó SwinT_OGC la cual posee un Swin Transformer backbone optimizado para la detección de objetos.

A modo de validación, hacemos un proceso end to end de una sola imagen y a continuación realizamos todo el procesamiento de las imágenes de train del dataset Test02 con Grounding Dino para la detección del objeto pelota. A continuación predecimos usando el predictor de SAM pasándole como prompting la información generada por DINO en el paso anterior.

Como resultado obtenemos las máscaras generadas con la predicción de SAM y las guardamos en un directorio en particular (MASK) lo que nos facilitará su exportación, descarga y reutilización. Realizamos ploteo de una cantidad acotada de imágenes para validar los resultados obtenidos.

Con el objetivo de guardar información para una posterior utilización, realizamos las transformaciones para guardar las anotaciones en Pascal VOC XML.

También realizamos la comparación de los bounding boxes detectados en las anotaciones realizadas al conjunto de datos (700 imágenes) y las predichas con el modelo SAM. Haciendo un resumen entre los True Positive (TP), los False Negative (FN) y False Positive (FP), vemos que el modelo detecta 344 FN, 71 TP y 29 FP siendo estos valores los que nos planteamos a mejorar en pruebas a continuación de este proyecto. Hay que tener en cuenta que si los bounding boxes detectados no tienen intersección entre ellos, se contará como 1 FN y 1 FP. Para aquellas imágenes en las que no fue ni anotada la pelota ni tampoco detectada por SAM (293 en total), no cuentan dentro de los números manejados anteriormente.

Result	
FN	344
TP	71
FP	29
Name: count, dtype: int64	

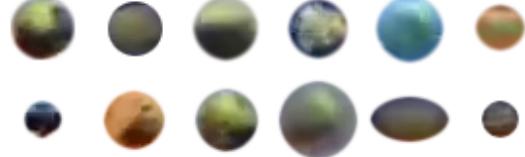
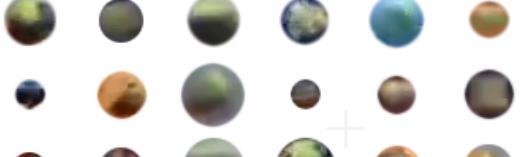
Como próximo paso y para profundizar en el análisis, se puede obtener la confianza del score cuando se realiza la predicción de SAM (ya que la misma es calculada y se encuentra entre los datos devueltos por SAM), y de esta manera poder comprar además de la información de los bounding boxes, también la información de confianza.

4.3 Generación de Datos Sintéticos para entrenar a YOLO

4.3.1 Elementos a insertar

Como elemento a insertar se utilizaron pelotas obtenidas del dataset Test02. La vista de la pelota varía notablemente debido a las condiciones de iluminación, distancia y occlusiones parciales. Se probaron dos selecciones de imágenes, por un lado se eligieron 12 vistas de la pelota de forma arbitraria y por otro se utilizaron las etiquetas asignadas a la clase pelota en el dataset Test02 para extraer a archivos nuevos recortes de esas regiones y aplicarles KNN. Se conformaron 5 clusters de los cuales finalmente se eligieron 30 vistas.

Utilizando el software de edición de imágenes GIMP, se recortó cada pelota dejando fondo transparente y un borde difuso de 2 píxeles. Cada imagen de la pelota fue guardada como archivo png con canal de transparencia.

Conjunto de 12 pelotas	Conjunto de 30 pelotas
	

4.3.2 Imágenes de fondo

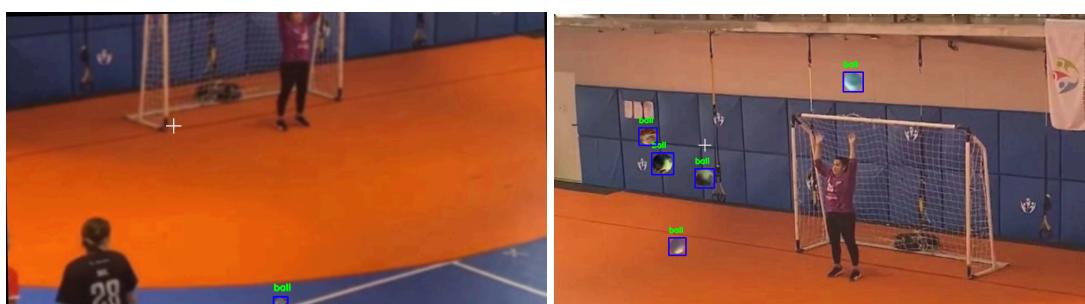
Se realizaron pruebas utilizando las primeras 1.000 imágenes del dataset COCO 2017 por una lado, e imágenes del dataset Test02 donde no se encontraba la pelota por otro lado. En este segundo caso la cantidad de imágenes sin pelota es de 88 y se utilizó augmentación (crop, mirror, rotación, blur, brillo, contraste y saturación) para llegar a 968 imágenes.

4.3.3 Inserción de imágenes

Se creó una función que sobre las imágenes de fondo inserta un número dado de pelotas incluyendo parámetros de modificación de tamaño, rotación, espejado y adaptación al brillo y color del sector de la imagen de fondo sobre el que se inserta la pelota.

Se realizaron dos tipos de inserciones con dos cantidades de pelotas distintas llegando a cuatro pruebas para la imágenes de fondo de COCO y cuatro para las imágenes de fondo de Test02 sin pelota. Las cantidades fueron 1 o 5 y las formas de inserción para el conjunto de 12 pelotas se le asignó a cada una una probabilidad (según observación visual del dataset) y en el de 30 pelotas cada una tenía la misma probabilidad.

Ejemplos de imágenes de COCO (arriba) y del dataset de Test02 (abajo) con 1 pelota insertada (izquierda) y con 5 (derecha):



4.3.4 Entrenamiento

Se realizaron 8 entrenamientos de 30 épocas partiendo de YOLO preentrenado con COCO y usando los pesos de yolov8s. Se utilizó el ancho de imagen de 640 pixeles. Se verificó que pasar al ancho de 800 pixeles en el entrenamiento no mejoraba los valores de mAP y sí el tiempo de procesamiento de 30 segundos a casi 600 segundos por época.

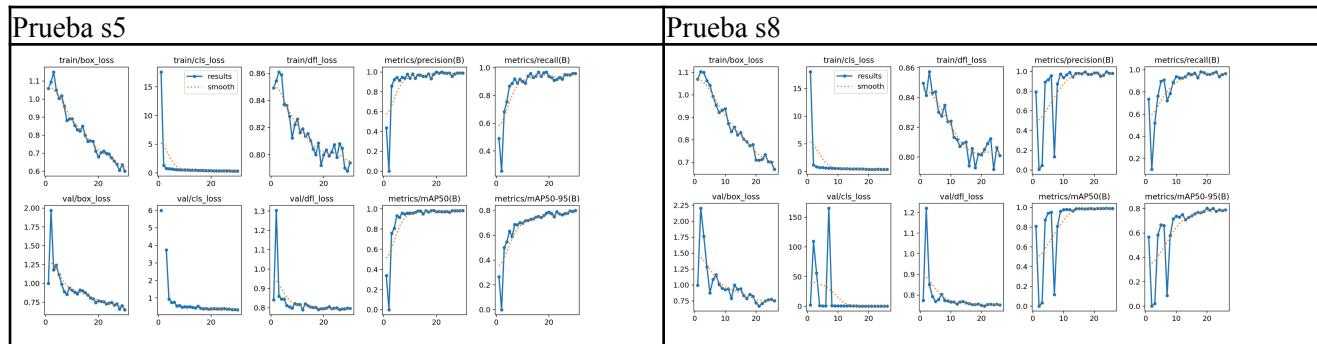
4.3.5 Resultados

Prueba	Modelo preentrenado	Dataset	Entrada	train/val	En validación del dataset sintético		En validación de data real Test02	
					mAP50	mAP50-95	mAP50	mAP50-95
Prueba s1	yolov8s	COCO + 1 pelota elegida de 12	640x640	700/100	0.995	0.907	0.329	0.158
Prueba s2	yolov8s	COCO + 5 pelotas elegidas de 12	640x640	700/100	0.995	0.931	0.102	0.059
Prueba s3	yolov8s	COCO + 1 pelota elegida de 30	640x640	700/100	0.995	0.902	0.254	0.152
Prueba s4	yolov8s	COCO + 5 pelotas elegidas de 30	640x640	700/100	0.995	0.911	0.266	0.124
Prueba s5	yolov8s	Test02-2 sin CLASE BALL + 1 pelota elegida de 12	800x800	677/98	0.984	0.798	0.508	0.224
Prueba s6	yolov8s	Test02-2 sin CLASE BALL + 5 pelotas elegidas de 12	800x800	677/98	0.994	0.873	0.442	0.180
Prueba s7	yolov8s	Test02-2 sin CLASE BALL + 1 pelota elegida de 30	800x800	677/98	0.983	0.804	0.351	0.135
Prueba s8	yolov8s	Test02-2 sin CLASE BALL + 5 pelotas elegidas de 30	800x800	677/98	0.987	0.833	0.394	0.195

En todas las pruebas los valores de mAP50 y de mAP50-95 resultaron muy buenos. Con imágenes de fondo del dataset COCO, el valor de mAP50-95 fue mejor que con imágenes de fondo del dataset Test02 por la aparición de falsos positivos en éste último. Al evaluar el modelo entrenado con los datos anotados de data real de Test02, ambos valores bajaron notablemente y varían según las imágenes de fondo, la cantidad de pelotas y forma de selección.

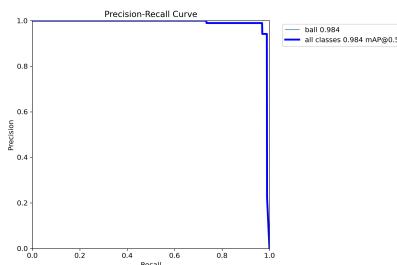
Las imágenes de fondo tomadas del mismo dataset donde se evaluaron resultados resultó mejor que el uso de imágenes genéricas del dataset COCO.

Con respecto al conjunto de pelotas disponibles para insertar y la cantidad de pelota insertadas los resultados varían en distintas direcciones. Usando el conjunto de 12 pelotas y variando entre elegir 1 o 5 pelotas, la precisión disminuye (0.73 y 0.64) y el recall aumenta (0.28 y 0.32). Usando el conjunto de 30 pelotas y variando entre elegir 1 o 5 pelotas, la precisión aumenta notablemente (0.38 y 0.70) y el recall apenas aumenta (0.27 y 0.28).



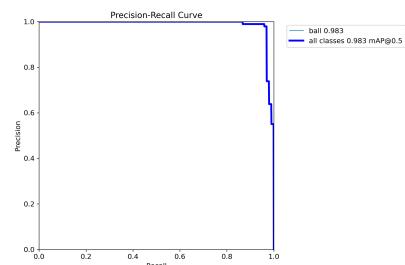
Training val

Pred/True	ball	bg
ball	93	1
bg	5	0



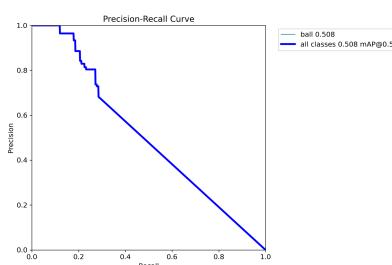
Training val

Pred/True	ball	bg
ball	462	8
bg	25	0



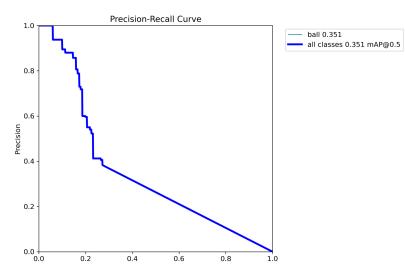
Datos reales val

Pred/True	ball	bg
ball	48	3
bg	103	0



Datos reales val

Pred/True	ball	bg
ball	53	30
bg	98	0



El mejor resultado del entrenamiento con imágenes sintéticas se obtuvo al entrenar con imágenes de fondo del dataset Test02 y agregando solamente una pelota por imagen a partir de las 12 seleccionadas y con asignación de probabilidad por pelota. El valor de mAP50 de ese modelo (0.508) no superó al baseline (0.522) y su valor de mAP50-95 (0.224) apenas superó al del baseline (0.204).

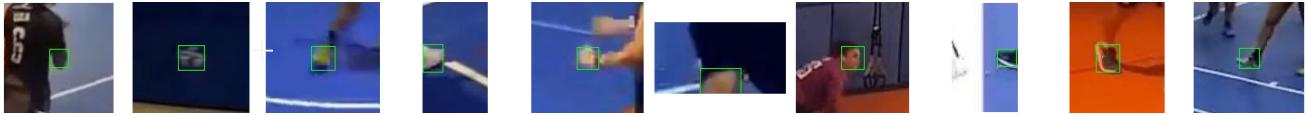
4.3.6 Análisis de errores

Para el análisis de errores se desarrolló un script que compara las predicciones con las anotaciones emparejándolas por mayor IoU. Si no hay intersección o está bajo determinado umbral, se considera falso positivo (FP) a la predicción obtenida y como falso negativo (FN) a la anotación existente.

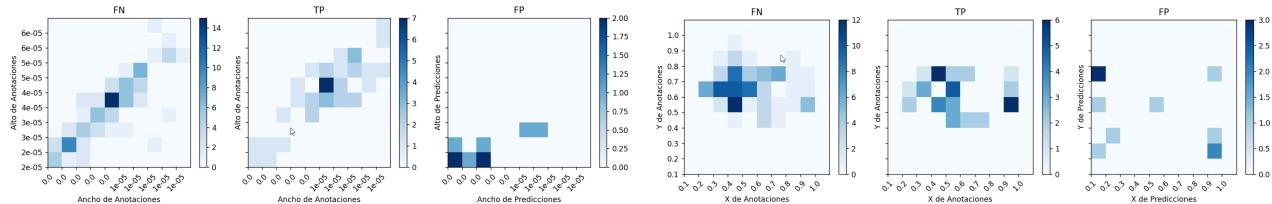
Este análisis permitió detectar que los falsos negativos corresponden a algunos errores de anotación, pelota difusa en movimiento, vistas de imágenes no representadas en la selección y sobre todo escenas donde la pelota está parcialmente oculta. Para intentar solucionarlo se podrían agregar más vistas de la pelota y explorar cómo agregar representaciones de la pelota parcialmente oculta. Ejemplos de falsos negativos:



Los falsos positivos son pocos y la mayoría corresponden a calzado o manos cerradas sin pelota. Para solucionarlo estos casos particulares se podrían agregar imágenes sintéticas de calzados como ruido sin etiquetar para que el modelo refuerce la diferencia entre los calzados y las pelotas. Ejemplo de falsos positivos:



Distribución de tamaños y posiciones de FN, TP y FP:



Los falsos negativos se distribuyen de forma similar a los verdaderos positivos (TP) mientras que los falsos positivos parecen ser más frecuentes en pequeños tamaños y sobre el borde izquierdo.

4.4 Entrenamiento de YOLO con datasets reales

Para encontrar el mejor modelo, probamos varias configuraciones de entrenamiento utilizando datasets reales capturados en diferentes canchas. En cada caso, comenzamos con un modelo YOLO v8 preentrenado con el dataset COCO, aprovechando la capacidad del modelo para reconocer una amplia variedad de objetos en diferentes contextos antes de ajustarlo específicamente a la detección de la pelota en el handball. Los datasets fueron anotados en Roboflow, el conjunto Test02 fue anotado manualmente por el equipo mientras que el dataset Test03 fue anotado automáticamente (Roboflow) y validado manualmente. A continuación, se describe un resumen de las pruebas realizadas, seguidas de una explicación detallada del modelo seleccionado para el entrenamiento final.

4.4.1 Pruebas Realizadas

Prueba 1: Custom dataset Test01 (versión 3).

En esta prueba inicial con datos reales, utilizamos un dataset con la cancha de Test01 anotando manualmente 5 clases diferentes: ball, player, goalkeeper, goal y referi. Durante el procesamiento, se aplicaron el resize uniformemente de 640 x 640 y stretch. Se utilizaron técnicas de augmentacion en Roboflow, como ser flip, crop, rotation y brightness, lo que nos generó un datatest de 699 imágenes. El modelo se entrenó con imágenes de entrada de tamaño 640 x 640, con 612 imágenes para entrenamiento y 58 para validación, utilizando 25 épocas de entrenamiento. Se obtuvo una precisión de 0.732, un recall de 0.456 y un mAP50 de 0.512.

Prueba 2: Custom Dataset Test02 (versión 1).

Este dataset contiene las mismas cinco clases que Test01 pero más imágenes a mejor resolución y en esta ocasión no se aplicaron técnicas de preprocesamiento ni de augmentation, lo que permitió evaluar el modelo en condiciones más simples. Utilizamos un total de 661 imágenes para el entrenamiento y 189 para validación, y al igual que en la prueba 1 se utilizaron 25 épocas para entrenar. Una diferencia fundamental en esta prueba con respecto a la primera fue la precisión y el cuidado con el que se realizaron las anotaciones. Este conjunto de datos fue anotado manualmente con un alto grado de detalle, intentando que cada objeto estuviera correctamente etiquetado, lo que se reflejó en los resultados.

Se logró una mejora considerable con respecto a la prueba 1. La precisión (P) alcanzó un valor de 0.902, el recall (R) 0.662 y además el mAP 50 se elevó a 0.773. **Estos resultados destacan que el enfoque meticuloso en la anotación de los datos tiene un impacto sustancial en el rendimiento del modelo.**

Pruebas 3, 4 y 5: Custom Dataset Test02 (versión 2).

En las pruebas 3, 4 y 5, el enfoque estuvo en la detección exclusiva de la pelota utilizando el dataset Test02 (versión 2) que contiene sólo anotaciones de pelotas. Cada prueba se diseñó para evaluar cómo diferentes niveles de augmentación al momento de entrenar afectan el rendimiento de YOLO. En estas pruebas se utilizaron 189 imágenes de validación y 661 imágenes de entrenamiento.

En la Prueba 3 sin augmentación, los resultados muestran una precisión de 0.85, un recall de 0.742 y una mAP50 de 0.781 además un mAP50-95 de 0.386. Estos resultados indican que el modelo pudo aprender efectivamente a detectar la pelota sin necesidad de augmentacion, beneficiándose de la calidad de las anotaciones.

En la Prueba 4, aplicamos augmentación básica de YOLO, incluyendo técnicas como el mosaico, mixup y el volteo horizontal. Los resultados mostraron una precisión de 0.853, un recall de 0.682 y una mAP50 de 0.724, junto con un mAP50-95 de 0.369. Aunque la precisión se mantuvo alta, el recall y la mAP50 disminuyeron ligeramente, lo que sugiere que la augmentación básica introdujo variabilidad que el modelo no aprovechó completamente para mejorar la detección de todas las instancias de la pelota.

En la Prueba 5, se utilizó una augmentación completa de YOLO con una amplia gama de transformaciones como ajustes de saturación, valor, brillo, rotaciones, cizallamiento, escalado y contraste. Estas técnicas aumentan la complejidad y diversidad de las imágenes de entrenamiento. Los resultados mostraron una precisión de 0.835, un recall de 0.773 y una mAP50 de 0.829, además de un mAP50-95 de 0.379. La augmentación completa mejoró significativamente el recall y la mAP50, indicando que el modelo pudo capturar más instancias de la pelota y manejar mejor la variabilidad en las condiciones del juego. Aunque la precisión disminuyó ligeramente, la mejora en el recall y la mAP50 sugiere que la augmentación completa ayudó al modelo a generalizar mejor.

Estas pruebas muestran cómo diferentes niveles de augmentación de datos influyen en el rendimiento del modelo YOLO. Esto resalta la importancia de seleccionar el nivel adecuado de augmentación para equilibrar la precisión y la capacidad de generalización en la detección de objetos.

Prueba 6: Custom Dataset Test02 (Versión 3)

En la Prueba 6, se utilizó la plataforma Roboflow para aplicar una augmentación avanzada, incluyendo saturación, brillo, contraste y transformaciones geométricas como rotaciones y escalado, lo que generó un dataset de 1983 imágenes de entrenamiento y 189 de validación.

Esta prueba demostró que la augmentación avanzada con Roboflow mejoró significativamente la precisión del modelo YOLO en la detección de la pelota. Sin embargo, la Prueba 5 logró un mejor equilibrio general especialmente en términos de recall y mAP50. Ambas pruebas subrayan la importancia de elegir adecuadamente las técnicas de augmentación y utilizar procesamiento acelerado para optimizar el rendimiento del modelo en la detección de objetos en entornos dinámicos como el handball.

4.4.2 Evaluación en video de una nueva cancha

Después de obtener resultados prometedores en las pruebas con el dataset Test02, decidimos evaluar la capacidad de generalización del modelo YOLO usando un video de una cancha de handball diferente, que no se había utilizado durante el entrenamiento. Al correr el modelo para detectar la pelota en este nuevo entorno, sorprendentemente, se detectaron muy pocas pelotas, a pesar de las buenas métricas obtenidas en las pruebas anteriores. **Esta limitación en la generalización** del modelo nos indica que, aunque el modelo puede funcionar bien en condiciones controladas y conocidas, su capacidad de generalizar a nuevos entornos sigue siendo limitada. Para mejorar esta capacidad, es fundamental entrenar el modelo con datos más variados y realizar pruebas exhaustivas en diferentes condiciones reales.

Después de observar que el modelo YOLO no logró detectar la pelota en un video de una cancha diferente no vista durante el entrenamiento inicial, decidimos evaluar cómo afectaría la combinación de datos de múltiples canchas al rendimiento del modelo. Para ello, realizamos las Pruebas 7 y 8, donde utilizamos datasets

combinados de dos canchas distintas, Test02 y Test03. El objetivo fue mejorar la capacidad de generalización del modelo, exponiéndolo a una mayor diversidad de condiciones de juego.

Prueba 7: Dataset Combinado de Test02 y Test03

El uso del modelo YOLOv8 con datos combinados resultó en una alta precisión (0.867) y una buena capacidad de recall (0.715), mAP50 de 0.801 y un mAP50-95 de 0.402. Combinar datos de distintas fuentes mejoró notablemente los resultados obtenidos. Aumentar la variabilidad al entrenar resulta muy importante para la generalización.

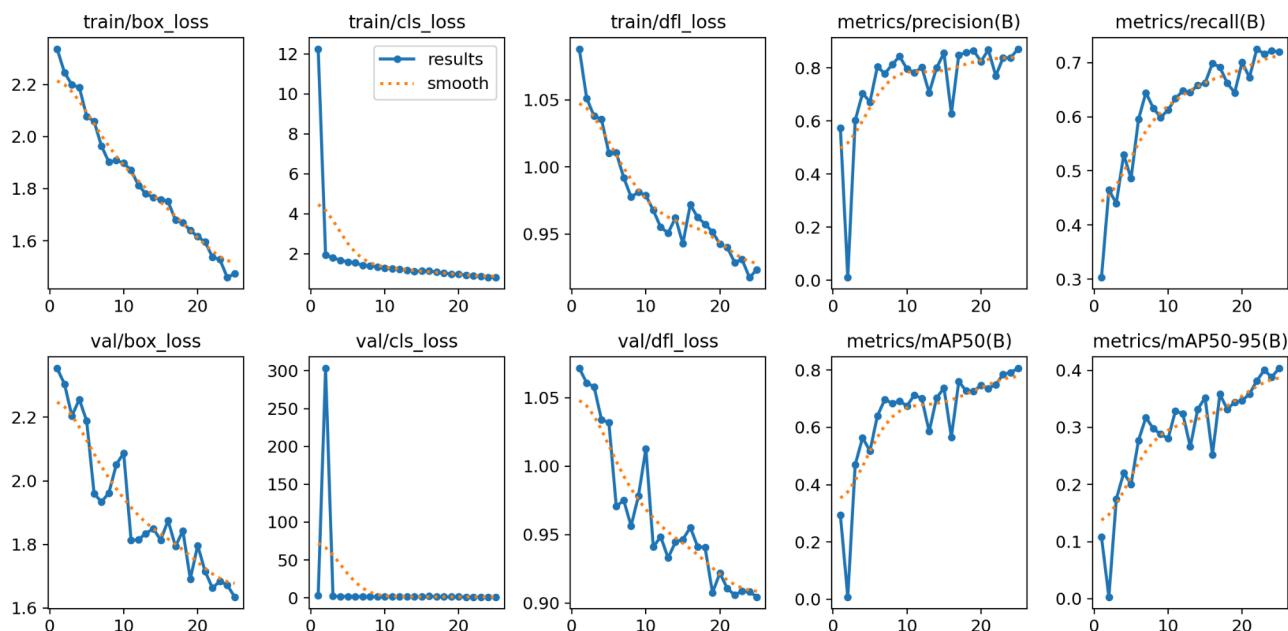
Prueba 8: dataset combinado de Test02 y Test03 con Modelo YOLOv8 con backbone congelado: En la Prueba 8, la inclusión de Resnet50 como backbone resultó en una mejora en el recall (0.741), lo que sugiere que esta configuración ayudó al modelo a detectar más instancias de la pelota. Prueba 8 tiene un mAP50 de 0.779 y un mAP50-95 de 0.396.

En términos de mAP50 y mAP50-95, la Prueba 7, utilizando YOLOv8 estándar con el dataset combinado de dos canchas, es ligeramente mejor que la Prueba 8 con YOLOv8 y el Resnet50 congelado.

4.4.3 Evaluación del modelo elegido

El modelo entrenado en la Prueba 7 fue seleccionado como el más eficaz, a continuación se presentan y analizan las gráficas de métricas de entrenamiento y validación junto a la matriz de confusión para entender mejor el rendimiento del modelo.

Gráficas de Métricas de Entrenamiento y Validación

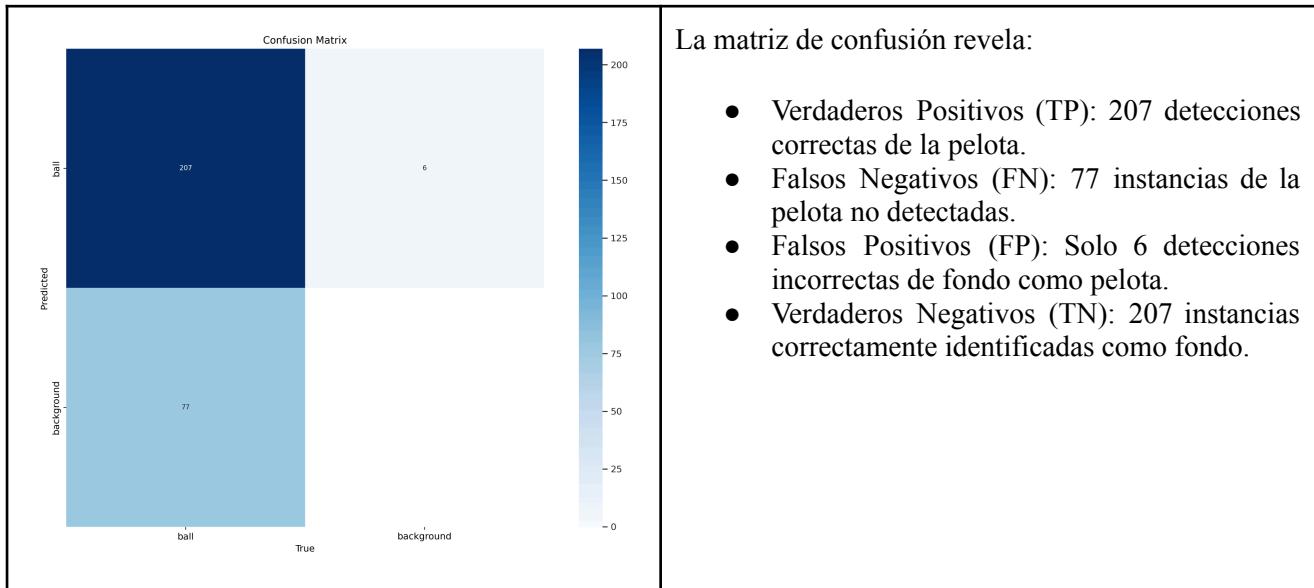


Las gráficas de entrenamiento y validación muestran mejoras consistentes a lo largo de 25 épocas:

- **Train/Box Loss:** Disminuye de 2.2 a 1.6, indicando una mejora continua en la precisión de las predicciones de las ubicaciones de las cajas.
- **Train/Cls Loss:** Baja rápidamente de 12 a 2, estabilizándose, lo que refleja una mejora en la clasificación de la pelota.

- **Train/DFL Loss:** Desciende de 1.05 a 0.9, mejorando la precisión de la predicción de las ubicaciones.
- **Val/Box Loss:** Disminuye de 2.4 a 1.8, mostrando una buena generalización del modelo.
- **Val/Cls Loss:** Reduce de 2.25 a 1.25, reflejando una mejora en la clasificación en los datos de validación.
- **Val/DFL Loss:** Baja de 1.05 a 0.9, consistente con las mejoras en el entrenamiento.
- **Precisión y Recall:** Mantienen una alta precisión (~0.8) y un buen recall (~0.75), lo que sugiere una buena capacidad del modelo para realizar detecciones correctas y capturar la mayoría de las instancias de la pelota.
- **mAP50 y mAP50-95:** mAP50 se estabiliza cerca de 0.8 y mAP50-95 alrededor de 0.4, lo que indica una alta precisión en detecciones con diferentes umbrales de solapamiento.

Matriz de Confusión



4.4.4 Resultados y discusión

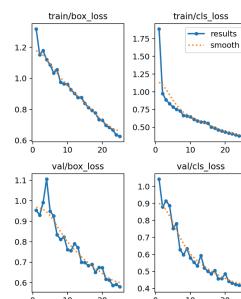
A continuación, se presenta una tabla que resume los resultados obtenidos en las diferentes pruebas descritas en la sección anterior. La tabla incluye detalles sobre el modelo utilizado, el dataset, el tamaño de entrada de las imágenes, y las métricas claves de rendimiento como la precisión (P), el recall (R), el mAP50 y el mAP50-95.

Prueba	Modelo preentrenado	Dataset	Entrada	train/val	(P)	(R)	mAP50	mAP50-95
Prueba1	yolov8s	TEst01-3	800x800	612/58	0.732	0.456	0.512	0.153
Prueba2	yolov8s	Test02-2	800x800	661/189	0.902	0.662	0.773	0.345
Prueba3	yolov8s	Test02-2 solo CLASE BALL	800x800	661/189	0.851	0.742	0.781	0.386
Prueba4	yolov8s	Test02-2 solo CLASE BALL y AUGMENTATION	800x800	661/189	0.853	0.682	0.724	0.369
Prueba5	yolov8s	Test02-2 solo CLASE BALL y Aumentación Completa	800x800	661/189	0.835	0.773	0.829	0.379

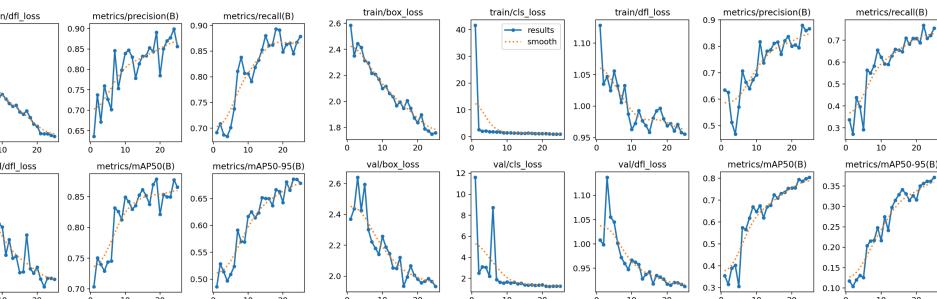
Prueba6	yolov8s	Test02-2 solo clase BALL y AUMENTACION en ROBOFLOW	800x800	1983 / 189	0.811	0.748	0.799	0.361
Prueba7	yolov8s	Test03 y Test02	800x800	700/100	0.867	0.715	0.801	0.402
Prueba8	yolov8s/backbone freeze	Test03 y Test02	800x800	700/100	0.827	0.741	0.779	0.396

Presentamos un análisis de las gráficas de métricas de entrenamiento y validación para tres pruebas claves realizadas: Prueba 1, Prueba 5 y Prueba 7. A través de este análisis, compararemos cómo el modelo responde a la inclusión de múltiples clases (Prueba 1), el uso exclusivo de la clase pelota (Prueba 5) y la combinación de datasets de diferentes canchas (Prueba 7), proporcionando una visión integral sobre la eficacia y la capacidad de generalización del modelo en diversos escenarios de juego.

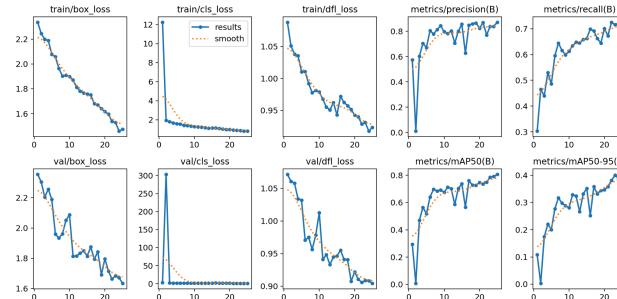
Prueba 1



Prueba 5



Prueba 7



Al observar las tres series de gráficas de métricas de entrenamiento y validación para las pruebas 1, 5 y 7 podemos observar que:

- 1) En las tres pruebas, hay un descenso consistente tanto en las pérdidas de entrenamiento como en las de validación. Esto indica que los modelos están aprendiendo de manera efectiva durante el proceso de entrenamiento.
- 2) En la Prueba 1, la pérdida de clasificación es mayor y más variable, lo que podría reflejar la complejidad añadida de manejar múltiples clases simultáneamente (ball, player, goalkeeper, arco y referí).
- 3) La Prueba 1 al incluir las cinco clases, no sólo pelota y la detección de player y referí, es muy buena por lo que mAP50 y mAP50-95 son muy buenos en general.
- 4) La Prueba 5, con enfoque en una sola clase y augmentación completa, presenta una reducción más efectiva y estable en la pérdida de la clasificación, lo que indica la especialización.
- 5) Las gráficas de precisión en las tres pruebas muestran un aumento general a lo largo de las épocas de entrenamiento. Tanto en la Prueba 5 como la Prueba 7 tienen una precisión alta y consistente. Este es un punto interesante sobre el cual reflexionar. La alta precisión alcanzada en la Prueba 5 durante el entrenamiento no se tradujo en una buena generalización a nuevas canchas.

- 6) Hacia el final de las épocas de entrenamiento, las pérdidas tienden a estabilizar lo cual indica que los modelos están alcanzando un punto de convergencia.
- 7) Las métricas mAP50 y mAP50-95 mejoran a lo largo del tiempo en todas las pruebas. Esto refleja una alta precisión en las detecciones con diferentes niveles de solapamiento (IoU), lo cual es crucial para la detección precisa de objetos en un entorno dinámico como el handball.

Las diferencias observadas en las métricas entre las pruebas indican que el uso de preprocesamiento y técnicas de augmentación, como en las pruebas 5, tiene un impacto notable en el rendimiento del modelo.

La Prueba 7, que combina datos de dos canchas, muestra mejoras significativas en las métricas, sugiriendo que la combinación de datasets de diferentes entornos puede ayudar a mejorar la capacidad de generalización del modelo.

Todas las pruebas demuestran que los modelos están aprendiendo de manera efectiva, como lo indica la disminución consistente de las pérdidas y la mejora en las métricas de evaluación.

La combinación de datasets y la aplicación de augmentación de datos son estrategias clave para mejorar la robustez y la capacidad de generalización del modelo, especialmente en entornos deportivos dinámicos.

Las configuraciones que incluyen augmentación avanzada y la combinación de múltiples datasets (como la Prueba 7) parecen ser las más efectivas para mejorar la precisión y la capacidad de generalización del modelo en la tarea específica de detección de la pelota.

La Prueba 5 (augmentación completa) y la Prueba 7 (datasets combinados de dos canchas) mostraron el mejor rendimiento global.

La Prueba 5 obtuvo un alto recall y mAP50, indicando que el modelo es eficaz en capturar la mayoría de las instancias de la pelota, incluso en condiciones variadas.

La Prueba 7 logró la mejor mAP50-95, sugiriendo una excelente capacidad para manejar variabilidad y condiciones de juego diferentes.

4.4.5 Limitaciones y Desafíos

Una limitación observada fue la dificultad del modelo para generalizar a nuevas canchas y condiciones de iluminación que no estaban presentes en los datos de entrenamiento. Esto se evidenció cuando el modelo no detectó la pelota en videos de una cancha nueva no utilizada en el entrenamiento inicial.

La calidad de las anotaciones manuales puede influir significativamente en el rendimiento del modelo. Se observó que una anotación más cuidadosa y detallada, como en la Prueba 2, contribuyó a un mejor rendimiento del modelo.

La limitación de recursos computacionales como GPU y memoria, son fundamentales en el entrenamiento de estos modelos.

La variabilidad de datos de entrenamiento para mejorar la robustez del modelo y su capacidad de generalización.

4.4.6 Análisis de Errores

Los errores más frecuentes en la detección de la pelota son debido a su pequeño tamaño, la velocidad rápida de movimiento, la oclusión frecuente por los jugadores y algunas más, a la incorrecta anotación. La pelota a menudo se confunde con el fondo o no se detecta en absoluto cuando está en el aire o cerca de otros objetos.

Los falsos positivos son pocos y la mayoría corresponden a problemas de anotación y detección de calzado o partes del cuerpo como pelotas. Ejemplos de falsos positivos:

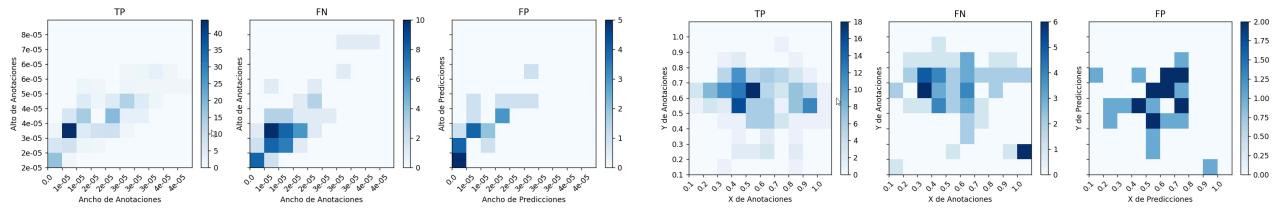


Los falsos negativos corresponden a algunos errores de anotación, pelota difusa en movimiento, pelotas muy pequeñas y sobre todo escenas donde la pelota está parcialmente oculta. Ejemplos de falsos negativos:



La revisión de las anotaciones a partir de este análisis permitiría reentrenar y obtener mejores resultados.

Distribución de tamaños y posiciones de TP, FN y FP:



Tanto los falsos negativos como los falsos positivos son más frecuentes en tamaños menores. Los falsos positivos parecen concentrarse en el medio de la imagen pero los verdaderos positivos también, en parte puede explicarse porque en general la cámara sigue a la pelota. Se destacan tanto falsos negativos como falsos positivos en el lado inferior derecho que corresponden a anotaciones automáticas y predicciones incorrectas en la zona de anuncios publicitarios.

5. Conclusiones

El objetivo principal de este proyecto es desarrollar un modelo eficiente para la detección de la pelota en videos de handball. Hemos cumplido con nuestro objetivo inicial y además hemos logrado resultados que han superado nuestras expectativas en varios aspectos que serán valiosos insight para nuestro trabajo final de maestría.

El uso del modelo YOLO v8 pre-entrenado con COCO demostró ser un muy buen modelo para la detección de la pelota en videos de handball, especialmente cuando se entrena con datos diversos y se aplican técnicas de augmentación. Comparar diferentes configuraciones y combinar datos de diversas canchas mostró que la combinación de datasets mejora la capacidad del modelo para generalizar. La Prueba 7, con YOLOv8 pre-entrenado en COCO y entrenado con un dataset combinado de dos canchas, fue la más efectiva en términos de equilibrio entre precisión y generalización logrando unas métricas de mAP50 de 0.801 y mAP50-95 de 0.402 superiores a las métricas bases de mAP50 0.522 y de mAP50-95 de 0.204.

Las anotaciones detalladas y precisas de los datasets tuvieron un impacto directo en el rendimiento del modelo. Las pruebas con anotaciones manuales cuidadosas resultaron en mejores métricas, destacando la necesidad de mantener altos estándares de calidad en la preparación de los datos.

La capacidad del modelo para generalizar a nuevos entornos puede ser mejorada significativamente mediante la inclusión de datos de entrenamiento que representen una amplia variedad de condiciones de juego y entornos. Esto es crucial para aplicaciones prácticas en análisis deportivo.

El análisis detallado de los errores nos permitió identificar patrones y áreas de mejora para futuros trabajos.

La técnica de imágenes sintéticas nos parecía prometedora pero no resultó como lo esperado. Los modelos convergieron muy bien en el mundo sintético pero no tuvieron buen desempeño con datos reales.

En cuanto a la utilización de SAM, demostró ser una buena herramienta para acelerar tiempos de etiquetado y detección de objetos.

Recomendaciones y líneas futuras del proyecto.

Recolectar y anotar más datos de una variedad de canchas y condiciones de juego para mejorar la robustez y la capacidad de generalización del modelo.

Verificar las anotaciones antes de iniciar el proceso de entrenamiento y utilizar analítica para encontrar problemas. Por ejemplo; relaciones alto/ancho de las anotaciones, cantidad de instancias de anotaciones por imagen.

Mejorar la forma en que se integran las imágenes sintéticas a las de fondo para que se puedan generalizar los resultados. También agregar imágenes de fondo de otras canchas y evaluar el entrenamiento con más imágenes. Adicionalmente se podría usar un dataset híbrido con entre un 10% y 15% de datos reales anotados que según algunos foros consultados mejoraría la performance de los modelos sintéticos.

Evaluar agregar pasos de preprocesamiento antes de entrenar y predecir como ser, convertir las imágenes a escalas de grises para uniformizar los fondos.

Para el caso de la utilización de SAM, recomendamos tener mayor variabilidad en el conjunto de datos, ya sea en entornos como en condiciones de la imagen.

6. Referencias

1. Host, K., Pobar, M., & Ivisic-Kos, M. (2023). Analysis of movement and activities of handball players using deep neural networks. *Journal of Imaging*, 9(4), 80. <https://doi.org/10.3390/jimaging9040080>
2. Alwi, A. A., Ibrahim, A. N., Shapiee, M. N. A., Ibrahim, M. A. R., Razman, M. A. M., & Khairuddin, I. M. (2020). Ball classification through object detection using deep learning for handball. *Mekatronika*, 2(2), 49-54. <https://doi.org/10.15282/mekatronika.v2i2.6751>
3. Wang, H., Li, Z., Liu, Z., & Shi, Z. (2022). A comprehensive review of computer vision in sports. *Computer Vision and Image Understanding*, 210, 103252. <https://doi.org/10.1016/j.cviu.2021.103252>
4. Listado de complejos de la Federación Uruguaya de Handball: <https://handball.com.uy/complejos>.