

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS, 2025-II

INTELIGENCIA ARTIFICIAL

---

**Proyecto Final**  
**Detección de imágenes enfocada en básquetbol**

---

**Autor:**

Alide Muabi Lara

No. de cuenta: 315321971

Correo: alidemuabil@ciencias.unam.mx

## 1 Descripción del problema

En el mundo moderno actual, la detección de imágenes se ha vuelto parte del día a día en sectores amplios de la sociedad, incluyendo al deporte. En el básquetbol las estrategias han cambiado rápidamente buscando mejorar y explotar las capacidades de nuevos jugadores que revolucionan este deporte y a través de la detección de los jugadores en cancha, el balón y, porque no, los árbitros, abre posibilidades innovadoras para adaptarse y buscar soluciones estratégicas efectivas para mejorar el rendimiento dentro de la cancha.

Para el área del espectáculo, esta herramienta podría dar pie a la automatización de jugadas destacadas, mejorando la experiencia de los aficionados. El análisis deportivo puede verse beneficiado de igual forma.

Este trabajo busca sentar los cimientos para una incorporación de la inteligencia artificial en el ámbito deportivo al crear un modelo capaz de analizar imágenes de partidos a cualquier nivel ya sea profesional, semi-profesional o amateur, siendo capaz de detectar a los jugadores dentro de la cancha, al balón, los aros, los árbitros y en caso de que la imagen incluya marcador, también poder detectar los nombres de los equipos y sus puntos, así como el tiempo restante.

## 2 Modelos de CNN usados

Modelo	Descripción	Arquitectura
<b>YOLOv5nu</b>	Modelo ligero versión nano para detección de objetos con buena velocidad y precisión. Ultralytics mismo hizo la recomendación de utilizar la versión 5nu en lugar de la 5n	Basado en CSPDarknet53, utiliza un cuello PANet (optimiza la fusión de características entre diferentes niveles de la red) y un cabezal de detección anclado.
<b>YOLOv8n</b>	Modelo versión nano presentando iteración avanzada con mejoras en precisión y versatilidad.	Introduce un cuello C2f refinado y un cabezal sin anclajes para mayor flexibilidad
<b>YOLOv11n</b>	Última versión con optimización en precisión, menor uso de memoria y mejor extracción de características.	Mejora la extracción de características con un troncal más eficiente y un cuello avanzado

Cuadro 1: Descripción de Modelos YOLO utilizados

### 3 Métricas Usadas

Métrica	Descripción	Razón
Precisión	Es la proporción de verdaderos positivos entre todos los resultados positivos predichos.	Queremos saber que tanto afecta la existencia de falsos positivos.
Recall	Es la proporción de verdaderos positivos entre todos los resultados positivos reales.	Nos interesa no perder casos de verdaderos positivos en la medida de lo posible.
Latencia (Velocidad de inferencia)	Es el tiempo que tarda el modelo en procesar una imagen.	Se busca tener resultados de manera rápida y efectiva, por lo que saber el tiempo que toma en procesarse una imagen resulta relevante para el proyecto.
mAP@0.5	Es el cálculo del promedio de precisión (mean Average Precision) utilizando un único umbral de intersección sobre unión (IoU) de 0.5 (una predicción se considera correcta si al menos el 50 % de las áreas se superponen).	Es de las métricas más utilizadas en visión por computadora.

Cuadro 2: Descripción de métricas usadas

## 4 Base de datos y preprocesamiento

### 4.1. Base de datos utilizada

Se obtuvo a través de la plataforma Roboflow, teniendo por nombre “Basketball Players Computer Vision Project” (<http://universe.roboflow.com/roboflow-universe-projects/basketball-players-fy4c2>) dando la opción de descarga para un modelo YOLO en específico. Para mantener validez al momento de utilizar los modelos mencionados en la sección 2, se descargó en un solo formato indicado ser para YOLOv8, teniendo las imágenes y anotaciones ya separadas para train, validation y test de la siguiente manera:

- Imágenes para train: 1140 (95 %)
- Imágenes para validation: 32 (3 %)
- Imágenes para test: 24 (2 %)

Para cada imagen había un archivo .txt con el mismo nombre en el que se designaban las “bounding boxes” asociadas, teniendo la posibilidad de representar una de las 9 clases: ‘Ball’ (Balón), ‘Hoop’ (Canasta), ‘Period’ (Periodo o cuarto), ‘Player’ (Jugador), ‘Ref’ (Árbitro), ‘Shot Clock’ (Reloj de tiro), ‘Team Name’ (Nombre del equipo), ‘Team Points’

(Puntos de equipo), 'Time Remaining' (Tiempo restante).

## 4.2. Preprocesamiento

No hubo preprocesamiento como tal, sin embargo al descargar el conjunto de datos se incluía un archivo que detallaba el preprocesamiento realizado por los creadores del conjunto, indicando que para cada imagen se realizó una auto-orientación de información de pixel (con "EXIF-orientation stripping").

### 4.2.1. Aumento de datos

Se crearon 10 versiones de cada foto procesada a través de los siguientes aumentos

- 50 % de probabilidad de volteo horizontal
- Recorte aleatorio entre el 0 % y el 20 % de la imagen
- Rotación aleatoria entre -12° y +12°
- Inclinación aleatoria entre -2° y +2° horizontal y entre -2° y +2° vertical
- Ajuste aleatorio de brillo entre -15 % y +15 %
- Ajuste aleatorio de exposición entre -15 % y +15 %
- Desenfoque gaussiano aleatorio entre 0 y 0.5 píxeles

## 5 Resultados

Modelo	Precisión (%)	Recall (%)	Latencia (ms/img)	mAP@0.5 (%)
YOLOv5nu	84.68	78.45	26.8	82.56
YOLOv8n	85.67	80.43	29.8	83.29
YOLOv11n	83.27	77.27	31.7	79.51

Cuadro 3: Resultados de los mejores modelos obtenidos durante el entrenamiento de 100 épocas para cada caso.

## 5.1. Precisión

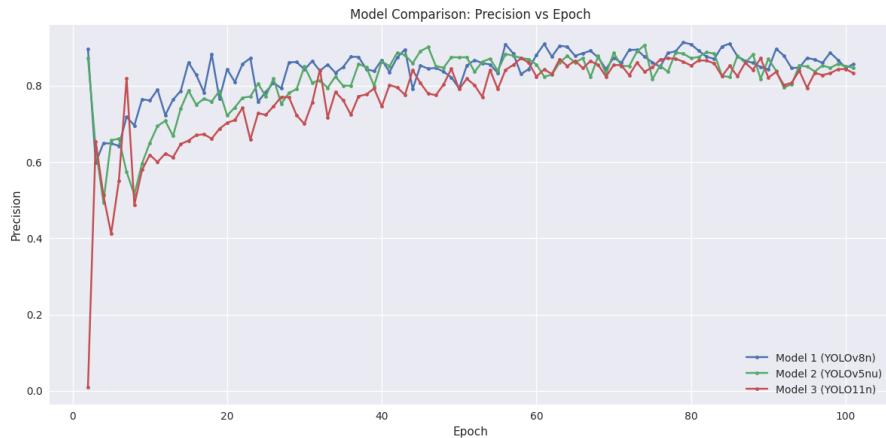


Figura 1: Gráfica de precisión vs. Época

## 5.2. Recall



Figura 2: Gráfica de recall vs. Época

### 5.3. Curvas PR

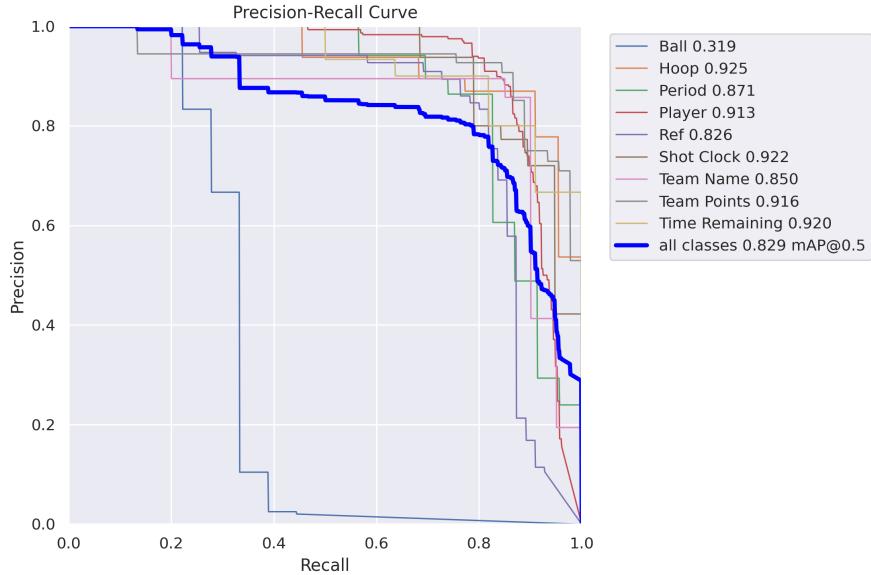


Figura 3: Gráfica de Precisión vs Recall para YOLOv5nu

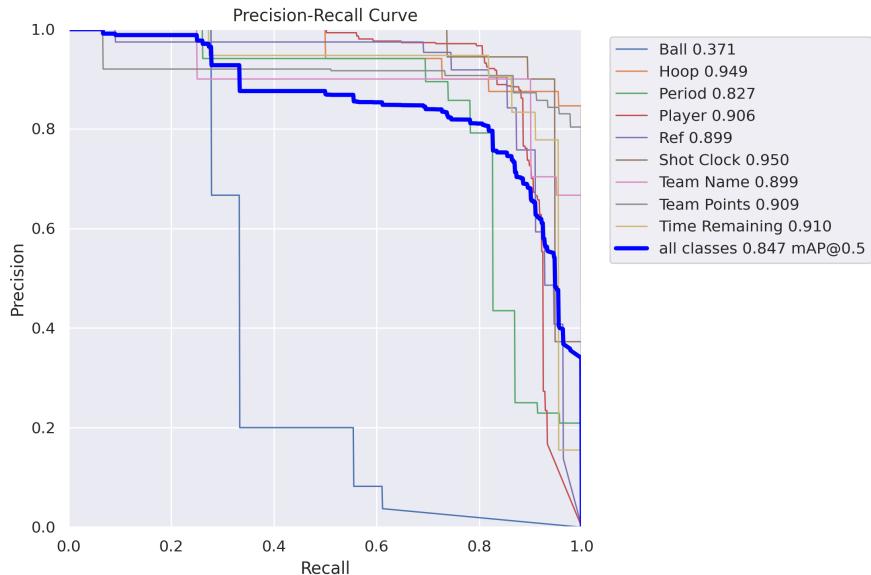


Figura 4: Gráfica de Precisión vs Recall para YOLOv8n

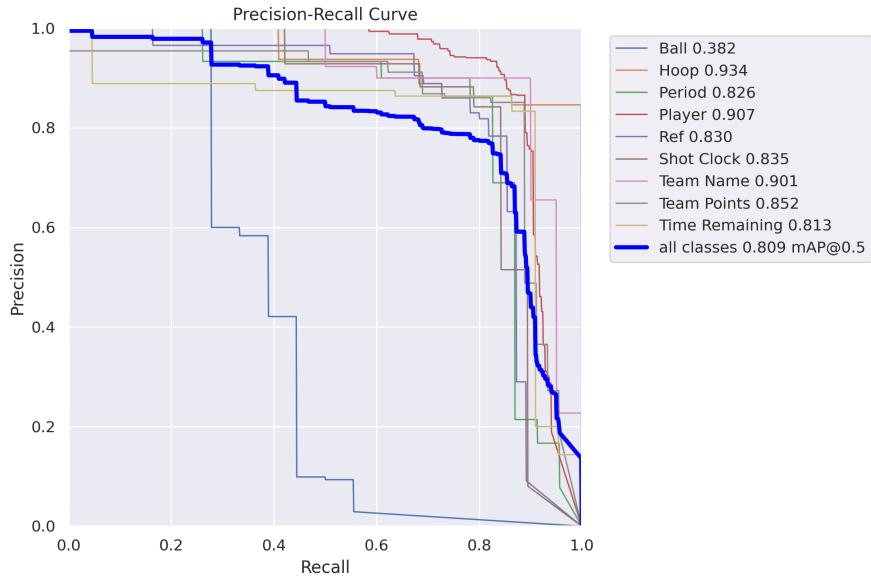


Figura 5: Gráfica de Precisión vs Recall para YOLOv11n

#### 5.4. mAP@0.5

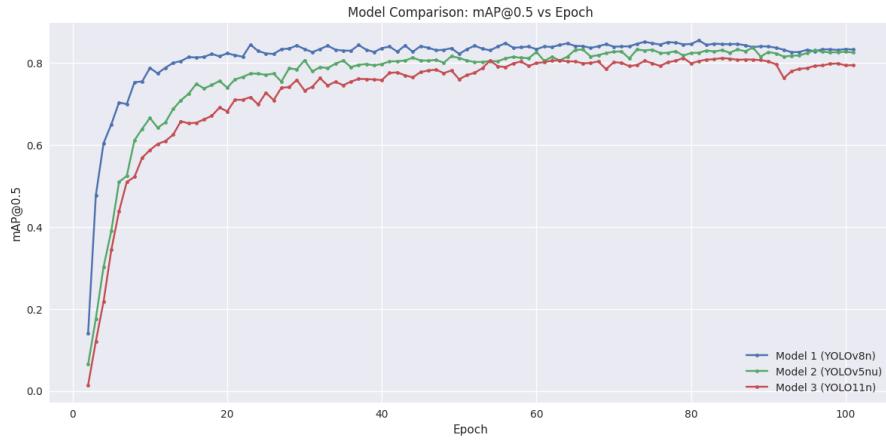


Figura 6: Gráfica de mAP@0.5 vs. Época

## 5.5. Latencia

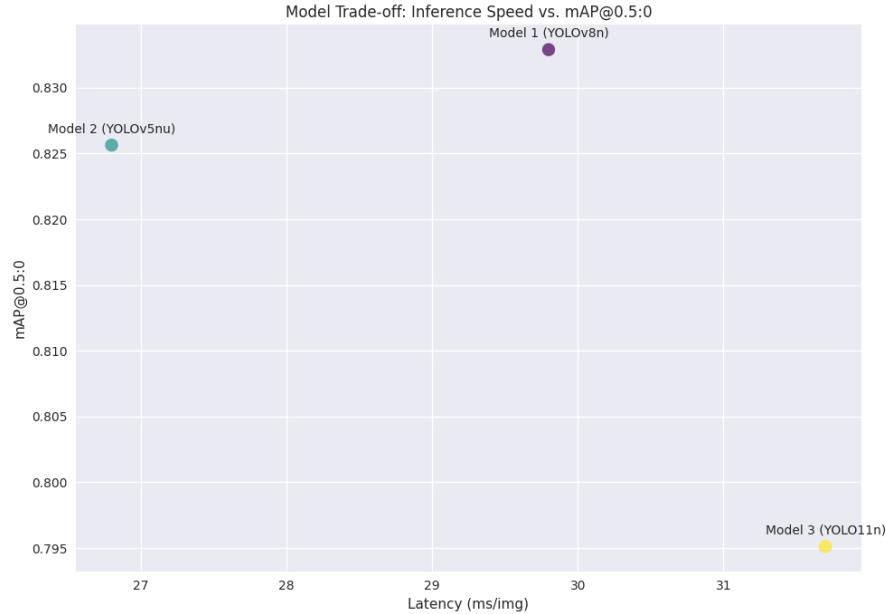


Figura 7: Gráfica de latencia vs.mAP@0.5

# 6 Análisis

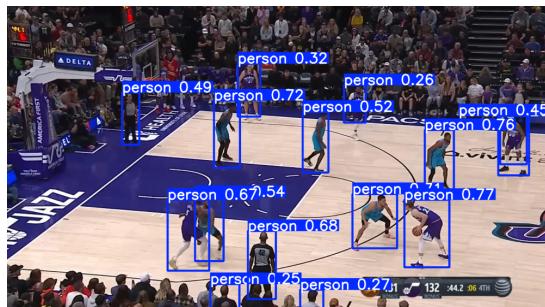
## 6.1. Análisis de métricas

- **YOLOv5nu:** Con una precisión casi del 85 %, YOLOv5nu muestra un rendimiento aceptable (aunque inesperado, como se verá más adelante) hasta que se analiza el recall menor al 80 % dando puerta a la no detección de algunas clases, en particular de la clase 'Ball' (Figura 3), siendo el peor modelo para detectar el balón, pieza sumamente importante en un juego de básquetbol. Es destacable su latencia, al ser la más baja de los 3 modelos, implicando ser el más veloz. Si lo que se busca es mayor velocidad, este modelo no es mala opción.
- **YOLOv8n:** Este modelo es el más preciso superando la marca del 85 % y además es el más sensible con un recall del 80.43 %, dando así el valor más alto de mAP entre los 3 modelos. En el único rubro que no es el mejor viene a ser la latencia, con una diferencia de 3 ms/img con el modelo YOLOv5nu. En general, este modelo casi no presenta limitaciones ni desventajas ante los demás, lo cual era de esperarse pues el dataset utilizado, en teoría, está diseñado para YOLOv8.
- **YOLOv11n:** Este modelo es el más deficiente en todas las métricas sin tener una a destacar. A pesar de ser el modelo más reciente, sus valores no llegan a ser los mejores y además es el modelo más lento y con menor porcentaje de mAP, lo cual lo vuelve el peor modelo de entre los 3.

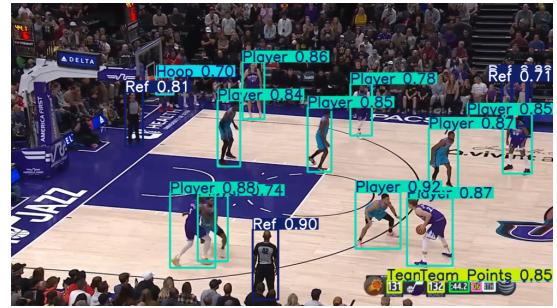
Como mencioné antes, a pesar del rendimiento aceptable de YOLOv5nu y los números deficiencias de YOLOv11n, al hacer un ejemplo con imágenes del grupo test (que no se usaron en ningún momento durante el entrenamiento), nos encontramos con resultados interesantes:

## 6.2. Comparación entre modelos pre-entrenados y entrenados

### YOLOv5nu



a. YOLOv5nu pre-entrenado

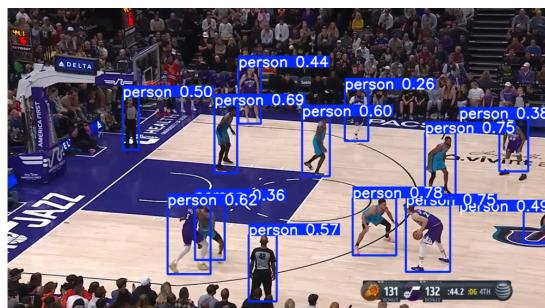


b. YOLOv5nu entrenado para detección de imágenes de básquetbol

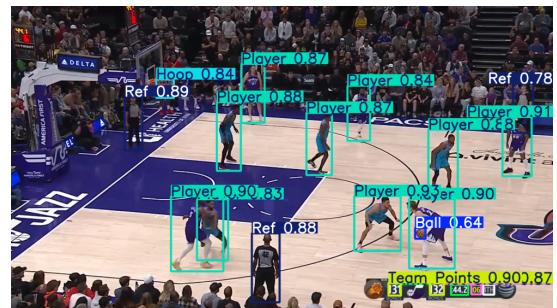
Figura 8: Comparación de Modelo YOLOv5nu pre-entrenado y entrenado

De la Figura 8.b) las detecciones fueron las siguientes: 1 Hoop, 1 Period, 10 Players, 5 Refs, 1 Shot Clock, 2 Team Points, 1 Time Remaining.

### YOLOv8n



a. YOLOv8n pre-entrenado

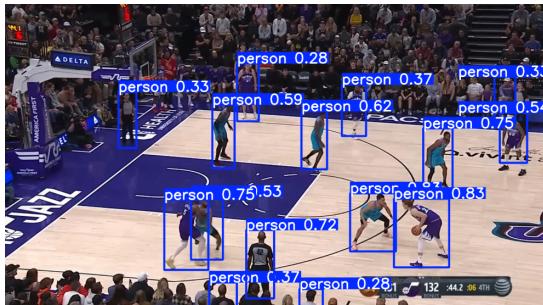


b. YOLOv8n entrenado para detección de imágenes de básquetbol

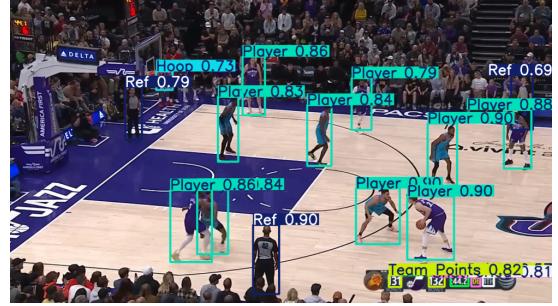
Figura 9: Comparación de Modelo YOLOv8n pre-entrenado y entrenado

De la Figura 9.b) 1 Ball, 1 Hoop, 1 Period, 10 Players, 3 Refs, 1 Shot Clock, 2 Team Points, 1 Time Remaining.

## YOLOv11n



a. YOLOv11n pre-entrenado



b. YOLOv11n entrenado para detección de imágenes de básquetbol

Figura 10: Comparación de Modelo YOLOv11n pre-entrenado y entrenado

De la Figura 10.b) las detecciones fueron las siguientes: 1 Hoop, 1 Period, 10 Players, 3 Refs, 1 Shot Clock, 2 Team Points, 1 Time Remaining.

Comparando estos ejemplos vemos que, de la Figura 8, se detectaron 5 árbitros cuando en el básquetbol solo hay 3 árbitros en cancha, error que no sucede para ningún otro modelo. A pesar de la relativamente alta precisión del modelo entrenado en YOLOv5nu, es el único que cometió un error de sobredetección. Incluso YOLOv11n detectó de manera correcta la situación que se presenta en la imagen.

## 7 Conclusiones

### 7.1. Mejor modelo

Con base en el análisis anterior, el modelo **YOLOv8n entrenado** resulta ser la mejor opción en general para la detección de imágenes enfocada en básquetbol, con una ligera decadencia en la velocidad de inferencia pero con porcentajes sólidos en precisión, recall y mAP@0.5.

### 7.2. Recomendaciones para futuros trabajos

Se sugiere tener hardware capaz de hacer entrenamientos con un tamaño de batch mayor, así como poder entrenar durante más épocas si el tiempo lo permite. Esto podría abrir las puertas a utilizar modelos pre-entrenados más grandes (YOLOv8m, YOLOv11s, ...) que probablemente mejoren los resultados. Este es un primer acercamiento a la detección en vivo que podría traer una infinidad de posibilidades en el ámbito deportivo al momento.