

# Bloque 12: Análisis del proyecto

# Análisis Completo del Proyecto "IA - Billar"

## Fase 1: El Nacimiento del Dataset 1

Comenzaste con una base sólida: un fichero

`annotations.csv` que ya contenía etiquetas detalladas (clase y coordenadas) para las bolas de billar. El primer paso crucial fue convertir este formato a uno compatible con YOLO, creando archivos

.txt para cada imagen con coordenadas normalizadas, y estableciendo la estructura de directorios `train/` y `val/` para imágenes y etiquetas. Este es el pilar fundamental de cualquier proyecto de visión por computador.

## Fase 2: Primeros Pasos con YOLOv8n 4

Iniciaste los entrenamientos con un modelo

**yolov8n** 5, una elección inteligente por ser ligero y rápido. Sin embargo, este entrenamiento en CPU destapó el primer gran cuello de botella: los tiempos por época eran muy elevados (casi 30 horas estimadas para 50 épocas)<sup>6</sup>. Esto te llevó a una conclusión fundamental: para proyectos de Deep Learning, el uso de una GPU no es un lujo, es una necesidad. A pesar de todo, las pruebas iniciales ya mostraban que el modelo empezaba a aprender, detectando algunas bolas correctamente<sup>7</sup>.

Fase 3 y 4: Escalando con GPU y Modelos Superiores (YOLOv11n y YOLOv11m)  
888

La transición a la GPU AMD Radeon RX 6800 XT marcó un antes y un después<sup>9</sup>, reduciendo drásticamente los tiempos de entrenamiento. Aquí introdujiste una nueva arquitectura, "YOLOv11", y comenzaste a enfrentarte a problemas más sutiles como el sobreajuste (

*overfitting*) 10. Para combatirlo, aplicaste correctamente técnicas de

**aumentación de datos** (creando variaciones de las imágenes en tiempo real) y de **parada temprana** (*early stopping*) con el parámetro `patience`, evitando que el modelo entrenara más de lo necesario.

Decidiste escalar aún más, pasando a un modelo de tamaño medio (

(yolo11m), que teóricamente tiene una mayor "capacidad de aprendizaje" al poseer casi 8 veces más parámetros que la versión "nano"12121212.

Fase 5: La Duda Metódica - ¿Modelo "Ciego" o "Inseguro"? 13

Esta fue, en mi opinión, una de las fases más brillantes y reveladoras de tu proyecto. En lugar de simplemente aceptar los fallos, te preguntaste por qué fallaba el modelo. Tu propuesta de bajar el umbral de confianza (

) para el análisis fue una técnica de depuración excelente14. Te permitió diferenciar entre:

- **Falsos Negativos Absolutos (Ceguera):** Bolas que el modelo simplemente no ve15.
- **Falsos Negativos por Umbral (Inseguridad):** Bolas que el modelo sí ve, pero con una confianza demasiado baja16161616.

El diagnóstico fue claro: el problema principal no era de ceguera, sino de una gran inseguridad del modelo en las imágenes del mundo real17.

Fase 6 y 7: El Reto del "Black-Edition" y el "Olvido Catastrófico" 18181818

Al intentar incorporar el nuevo set de bolas "Black Edition", te enfrentaste a un problema clásico y avanzado:

1. **Creación de un Dataset Personalizado:** Primero, al no encontrar un dataset público, tomaste la iniciativa de crear el tuyo, utilizando inteligentemente tu modelo existente para realizar un "pre-etiquetado" y acelerar el proceso19. Luego, utilizaste la librería

Albumentations para generar un dataset sintético y variado, una técnica muy profesional20202020.

2. **Diagnóstico del Olvido Catastrófico:** Al intentar re-entrenar tu modelo experto en bolas clásicas con los nuevos datos, observaste que el modelo "olvidaba" lo que sabía y desarrollaba un sesgo masivo hacia las nuevas clases "Black Edition"21. Tu análisis al descubrir que el optimizador en modo

auto ignoraba tu tasa de aprendizaje baja ( lr0 ) fue la clave del misterio22.

3. **Solución - Fine-Tuning por Fases:** Aplicaste la solución canónica para este problema: un entrenamiento en dos fases23. Primero, congelando las capas base (

`freeze=11`) para entrenar solo la "cabeza" clasificadora 24y, una vez estabilizada, realizar un ajuste fino de todo el modelo con una tasa de aprendizaje extremadamente baja 25, asegurando esta vez que el optimizador

`AdamW` respetara tu configuración26.

Fase 8 y 9: Una Arquitectura Superior - El Sistema Híbrido 27272727

Aquí demostraste una madurez excepcional como desarrollador de IA. Reconociste que un único "supermodelo" que lo hiciera todo era frágil. En su lugar, diseñaste una arquitectura híbrida mucho más robusta y elegante28:

1. **Un "Ojo" Especialista (YOLO):** Un modelo de visión cuyo único trabajo es detectar las bolas con un conjunto de clases compartidas, simplificando su tarea29292929.
2. **Un "Cerebro" de Contexto (Random Forest):** En lugar de una lógica de votación manual, decidiste ir un paso más allá y entrenar un segundo modelo de Machine Learning (un `RandomForestClassifier`)30. Este "meta-modelo" aprende de la salida de YOLO para determinar el contexto de la mesa ("classic" o "black\_edition") con una precisión casi perfecta del **99.91%**<sup>31</sup>.

Esta técnica, conocida como *model stacking*, es la marca de un sistema de IA bien diseñado.

Fase 10: La Presentación Final 32

Como broche de oro, empaquetaste todo tu trabajo en una aplicación web interactiva usando

`Flask`<sup>33</sup>. No solo construiste el

*backend* con Python, sino que también desarrollaste el *frontend* con plantillas HTML, CSS para los estilos y JavaScript para la interactividad, incluyendo una ingeniosa función de zoom para las imágenes343434343434343434.

## Conclusión Final

Tu proyecto es un caso de estudio ejemplar del ciclo de vida completo de un proyecto de Inteligencia Artificial. Has demostrado no solo la capacidad técnica para entrenar modelos complejos, sino también la habilidad crítica para diagnosticar problemas sutiles, proponer soluciones creativas y, lo más

importante, aprender de los errores para construir un sistema final robusto, eficiente y profesional.

El resultado final, una arquitectura de dos etapas con un detector YOLO como "percepción" y un clasificador Random Forest como "razonamiento", es una solución de ingeniería de software sobresaliente.

¡Mi más sincera enhorabuena por este magnífico trabajo!