

# Bloque 13: Glosario

## Glosario Cronológico del Proyecto

### Fase 1 y 2: El Nacimiento del Dataset y Primeros Pasos con YOLO

- **Dataset:** Tu colección de imágenes de bolas de billar y sus etiquetas.
- **Etiquetado (Labeling) y Bounding Box:** El proceso de dibujar cajas rectangulares ( `xmin` , `ymin` , `xmax` , `ymax` ) sobre los objetos y asignarles una **Clase** (ej. `white` ).
- **YOLO (You Only Look Once):** La arquitectura de red neuronal que elegimos por su equilibrio entre velocidad y precisión.
- **Framework de Deep Learning:** La biblioteca de software usada, que fue **PyTorch** a través de la librería **Ultralytics**.
- **Formato de Anotaciones YOLO (.txt):** El formato final de tus etiquetas, con coordenadas normalizadas ( `center_x` , `center_y` , `width` , `height` ) entre 0 y 1.
- **División del Dataset (Train/Val/Test):** La práctica fundamental de separar los datos para **entrenar**, **validar** durante el proceso y **probar** al final.
- **Aprendizaje por Transferencia (Transfer Learning):** La técnica de usar un modelo **pre-entrenado** (como `yolov8n.pt` ) como punto de partida para acelerar y mejorar el aprendizaje.
- **Época (Epoch):** Un ciclo completo de entrenamiento sobre todo tu dataset.
- **ROCm (Radeon Open Compute):** La plataforma de software de AMD que te permitió usar la potencia de tu **GPU** para acelerar el entrenamiento, un paso crucial que redujo los tiempos de días a horas.
- **Checkpoint ( `best.pt` ):** El archivo más importante generado durante el entrenamiento, que guarda los pesos del modelo en la época en que obtuvo el mejor rendimiento sobre el conjunto de validación.

---

### Fase 3, 4 y 5: Optimización y Diagnóstico Avanzado

- **Sobreajuste (Overfitting):** El primer gran problema que diagnosticamos, donde el modelo se "aprendía de memoria" el set de entrenamiento y

perdía su capacidad de generalizar. Lo identificaste al ver que la pérdida en validación se estancaba mientras la de entrenamiento seguía bajando.

- **Parada Temprana (Early Stopping):** La primera contramedida que aplicamos, usando el parámetro `patience` para detener el entrenamiento cuando el modelo dejaba de mejorar, evitando así el sobreajuste excesivo.
- **mAP (Mean Average Precision):** La métrica clave que usamos para medir objetivamente el rendimiento del modelo. `mAP50` es la estándar y `mAP50-95` la más estricta.

### Tu Propuesta: Análisis "Ciego vs. Inseguro"

Aquí tuviste una idea fundamental. Propusiste realizar una inferencia con un umbral de confianza muy bajo ( $\text{conf}=0.25$ ) para diferenciar entre bolas que el modelo no veía en absoluto (estaba "Ciego") y bolas que sí veía pero con poca seguridad (estaba "Inseguro"). Este diagnóstico nos confirmó que el problema principal no era de ceguera, sino de falta de confianza debido al

**"Domain Gap"** (la diferencia entre las fotos del dataset y tus fotos reales). 1111

- **Brecha de Dominio (Domain Gap):** El concepto que explica por qué un modelo que funciona bien en el laboratorio (con los datos del dataset) puede fallar en el mundo real (con tus fotos de prueba). Las diferencias en iluminación, cámara y ángulos crean esta brecha.

## Fase 6 y 7: El Desafío "Black Edition"

- **Pre-etiquetado (Pre-labeling):** Tu iniciativa de usar el modelo ya entrenado como un "asistente" para generar las primeras etiquetas en el nuevo dataset "Black Edition", acelerando masivamente el proceso de etiquetado manual en **Label Studio**.
- **Aumentación de Datos con `Albumentations`:** La técnica con la que creaste un dataset sintético y variado para las bolas "Black Edition", generando cientos de imágenes nuevas a partir de unas pocas, para combatir el sobreajuste.

## Analogía: El Síndrome del "Ave Exótica"

Para entender por qué el modelo se obsesionaba con las nuevas bolas "Black Edition", usamos la analogía de un observador de aves que, tras ver 1000 gorriones comunes, ve una única foto de un "Diamante Azul" rarísimo. Su cerebro memoriza tanto las características únicas del ave exótica que luego confunde un simple brillo en la cola de un gorrión con el ave rara. Esto representó cómo tu modelo le daba un peso desmedido a las características únicas de la clase minoritaria ("Black Edition"). 2222

- **Olvido Catastrófico:** El problema más avanzado que enfrentaste. Al re-entrenar el modelo experto en bolas clásicas con los nuevos datos, el nuevo conocimiento "pisaba" y destruía lo que ya sabía.

## Analogía: El Cirujano Experto

Para solucionar el olvido catastrófico, imaginamos el modelo como un "cirujano experto". Re-entrenarlo con una tasa de aprendizaje alta era como darle instrucciones "a gritos", lo que lo abrumaba y hacía que olvidara sus habilidades. La solución fue "susurrarle" las nuevas instrucciones, es decir, aplicar un

**ajuste fino (fine-tuning)** con una tasa de aprendizaje ( Ir0 ) muy baja. 3333

- **Entrenamiento por Fases:** La solución definitiva que implementaste, que consistía en:
  1. **Congelar el Backbone:** Entrenar solo la "cabeza" de la red para que aprendiera las nuevas clases sin dañar el conocimiento base.
  2. **Fine-Tuning Completo:** Descongelar todo el modelo y re-entrenarlo con una tasa de aprendizaje mínima para un pulido final y de alta precisión.

## Fase 8 y 9: La Arquitectura Híbrida Definitiva

Tu Propuesta Clave: La Arquitectura Híbrida ("Ojo" y "Cerebro")

Ante la persistencia del sesgo, tomaste la decisión estratégica más importante del proyecto: abandonar la idea de un único "supermodelo" y diseñar una arquitectura de dos etapas, una solución de ingeniería mucho más robusta y profesional. 4

- **El "Ojo" (Percepción):** Un modelo YOLO ( `Modelo_Hibrido_v1` ) entrenado con un set de clases simplificado, cuyo único trabajo es detectar las bolas de la forma más precisa posible.
- **El "Cerebro" (Razonamiento):** Un módulo de post-procesamiento que interpreta la salida del "Ojo" para tomar la decisión final.

Tu Propuesta: Del Voto al "Meta-Modelo"

No satisfecho con una lógica de votación manual para el "Cerebro", propusiste reemplazarla por un modelo de Machine Learning. 5 Esto elevó el proyecto a un nivel superior, aplicando una técnica conocida como

**Model Stacking o Meta-aprendizaje. 6**

- **Ingeniería de Características:** El proceso en el que definiste qué información extraer de la salida de YOLO (recuentos, confianzas, etc.) para usarla como "features" y alimentar a tu nuevo modelo.
- **Random Forest:** El algoritmo de Machine Learning que elegiste para actuar como "Cerebro", entrenándolo para clasificar el contexto de la mesa ("classic" o "black\_edition") a partir de las características generadas por el "Ojo", logrando una precisión casi perfecta.

## Fase 10: Presentación del Proyecto

- **Flask:** El framework de Python con el que construiste una aplicación web interactiva para demostrar y presentar tu trabajo.
  - **Refactorización:** La buena práctica de ingeniería de software que aplicaste al separar la lógica de la IA en su propio módulo ( `motor_inferencia.py` ), haciendo que el código de la web ( `app.py` ) fuera más limpio, seguro y fácil de mantener.
- 

Este glosario ahora no solo define los conceptos técnicos, sino que también narra la historia de tu aprendizaje y de cómo tus decisiones y análisis críticos guiaron el proyecto hacia una solución final de gran calidad. ¡Un trabajo realmente completo!