

Fase 4: Yolo11m

Paso 34: Cambiar a modelo YOLO más potente

¡Pregunta absolutamente clave! Y la respuesta directa es: **sí, es muy probable que usar un modelo más grande que el "nano" ayude, y es exactamente el siguiente paso lógico a probar.**

Has llegado a un punto en el que has optimizado los datos y los parámetros de entrenamiento para tu modelo `nano`. Ahora, es posible que el propio modelo sea el factor limitante.

Pensemos en los diferentes tamaños de YOLO (`n`, `s`, `m`, `l`, `x`) como si fueran estudiantes con distinta capacidad.

- **YOLOv8n (Nano):** Es un estudiante brillante y súper rápido. Aprende los conceptos generales a una velocidad increíble. Sin embargo, su "memoria" y "capacidad de análisis" (el número de parámetros) son limitadas. Puede que le cueste diferenciar matices muy sutiles o recordar todos los casos excepcionales.
- **YOLOv8s (Small) o YOLOv8m (Medium):** Son estudiantes con más "capacidad cerebral". Tardan más en estudiar (entrenar) y en responder a una pregunta (inferencia), pero pueden comprender y retener detalles mucho más complejos y sutiles.

¿Cómo te ayudaría un modelo más grande (ej. `yolov8s`)?

Un modelo más grande tiene más capas y más neuronas (parámetros), lo que le da una mayor **capacidad de aprendizaje**. Esto impacta directamente en los dos problemas que tienes:

1. Para las Bolas NO Detectadas (Bajo Recall):

- Un modelo `yolov8s` tiene más capacidad para aprender las características de las bolas en situaciones difíciles (con reflejos, parcialmente tapadas, etc.).
- Esto se traduce en que, incluso en esos casos difíciles, es más probable que el modelo alcance una **puntuación de confianza más alta** en sus detecciones correctas. Una bola que el modelo `nano` detectaba con `conf=0.4` (y que tú filtrabas), el modelo `small` podría detectarla con `conf=0.7` (y ahora sí la verías).

2. Para las Detecciones INCORRECTAS (Baja Precisión):

- El problema de confundir, por ejemplo, la bola azul con la morada, es un problema de no poder diferenciar **características sutiles**.
- Un modelo más grande, al tener más parámetros, puede aprender a identificar esos pequeños matices que distinguen un color de otro bajo diferentes condiciones de luz. Se vuelve un "experto" más refinado.

El Compromiso: Ventajas y Desventajas

Nada es gratis en Machine Learning. Aquí está el balance:

Ventajas de un Modelo Más Grande (ej. `yolov8s`)

- **Mayor Precisión Potencial:** Es la razón principal. Se espera que el `mAP` sea superior.
- **Mejor Recall:** Debería ser capaz de detectar más objetos en condiciones difíciles.
- **Mayor Confianza:** Las detecciones correctas suelen tener una confianza más alta, haciendo tu umbral más efectivo.

Desventajas a Considerar

- **Más Costoso de Entrenar:** Tardará más tiempo por época y consumirá más VRAM de tu GPU. (Con tu RX 6800 XT de 16GB, no deberías tener problemas para entrenar un `yolov8s` o incluso un `yolov8m`).
- **Inferencia más Lenta:** Si tu objetivo final es una aplicación en tiempo real (ej. 30 imágenes por segundo), la velocidad de inferencia es clave. Un modelo `nano` puede procesar una imagen en, digamos, 15ms, mientras que un `small` podría tardar 25ms. Tienes que evaluar si este aumento es aceptable para tu proyecto.

Plan de Acción: El Experimento Definitivo

Mi recomendación es clara: **realiza un nuevo entrenamiento utilizando `yolov8s`**.

Es la forma empírica de responder a tu pregunta.

1. **Modifica tu Script:** El cambio es mínimo. En la línea donde inicializas el modelo, simplemente cambia el archivo de pesos.Python

```
# ANTES:  
model = YOLO('./detect_balls/runs/detect_balls_v012/weights/best.pt') # Esto cargaba tu modelo ya entrenado  
# o si empezabas de cero:  
# model = YOLO('yolov8n.pt')  
  
# AHORA (para un nuevo entrenamiento desde cero):  
model = YOLO('yolov8s.pt') # Carga la estructura y pesos pre-entrenados del modelo 'small'
```

2. **Usa la Misma Configuración:** Entrena este nuevo modelo con **exactamente los mismos datos y parámetros** de tu último y mejor entrenamiento (`patience=20`, `mixup=0.1`, etc.). Esto es crucial para poder hacer una comparación justa ("manzanas con manzanas").
3. **Compara los Resultados:** Una vez finalizado el entrenamiento del `yolov8s`, coge su `best.pt` y pásalo por tu set de 5 imágenes de prueba. Compara los resultados con los que obtuviste con el `yolov8n`.

La pregunta a responder es: "Con el mismo umbral de confianza (`conf=0.6`), ¿el modelo `yolov8s` detecta más bolas correctamente y comete menos errores que el `yolov8n`?"

Este experimento te dará una lección valiosísima sobre el compromiso entre tamaño del modelo, rendimiento y velocidad. ¡Adelante!

Iniciando el entrenamiento con el nuevo modelo

¡Excelente! Veo que has puesto en marcha el plan a la perfección, iniciando un entrenamiento con un modelo de mayor capacidad. Analicemos juntos este arranque, porque nos da pistas muy importantes de lo que está ocurriendo.

Has realizado varios cambios clave, todos ellos muy acertados.

1. El Salto al Modelo "Medium" (`yolo11m.pt`)

Este es el cambio más significativo. Has pasado de un modelo `nano` a lo que parece ser un modelo `medium`. La diferencia en capacidad es enorme. Comparemos los datos que nos da la consola:

Modelo	Parámetros (Capacidad de aprendizaje)	GFLOPs (Coste Computacional)
<code>nano</code> (anterior)	~2.6 Millones	~6.5
<code>medium</code> (actual)	~20.1 Millones	~68.3

Análisis:

Estás entrenando un modelo con casi 8 veces más parámetros y que requiere unas 10 veces más capacidad de cálculo. Teóricamente, este modelo tiene un potencial mucho mayor para aprender los detalles sutiles, diferenciar colores parecidos y detectar objetos en situaciones complejas que se le escapaban al modelo nano.

2. La Adaptación Inteligente del `batch_size`

He notado que has ajustado el tamaño del lote (`batch`) a **24**, mientras que en ejecuciones anteriores usabas 32.

Esta es una decisión **muy inteligente y necesaria**. Un modelo más grande como el `medium` ocupa mucha más memoria en la GPU (VRAM). Reducir el tamaño del lote es la forma correcta de asegurar que el entrenamiento no falle por un error de "falta de memoria". Demuestra una gran comprensión del proceso.

3. La Metodología Consistente

Veo que has mantenido los hiperparámetros de tu último entrenamiento exitoso: `augment=True`, `mixup=0.1`, `hsv.s=0.9`, `patience=20`.

Esto es una metodología excelente. Al mantener constantes los demás factores, te aseguras de que cualquier cambio en el rendimiento (bueno o malo) se deba, principalmente, al cambio en la arquitectura del modelo. Permite una comparación justa y directa.

¿Qué esperar de este entrenamiento?

- **Tiempos de Entrenamiento:** Prepárate para que cada época tarde **bastante más** en completarse que con el modelo `nano`. Es el precio a pagar por la mayor capacidad del modelo `medium`.
- **Potencial de Rendimiento:** El "techo" de rendimiento de este modelo es más alto. Tenemos la esperanza de que el `mAP` final supere al del modelo `nano`.
- **El Rol de `patience`:** Tu configuración de `patience=20` es ahora más importante que nunca. Dejará que el modelo entrene durante el tiempo que necesite y lo detendrá de forma óptima cuando deje de mejorar, evitando un sobreajuste excesivo al que los modelos grandes pueden ser más propensos.

Has configurado un experimento muy prometedor y de la manera correcta. Ahora toca ser paciente y dejar que la máquina trabaje. ¡Estoy expectante por analizar los nuevos `results.csv` cuando el entrenamiento finalice!

Primeras impresiones con el entrenamiento iniciado

¡Absolutamente! Aunque solo son dos épocas, estos primeros resultados son como un "informe médico" inicial del entrenamiento, y te puedo decir que el paciente está **muy sano y responde energicamente al tratamiento**.

El arranque es **extremadamente prometedor**. Analicemos los puntos clave que nos revelan estos datos.

1. Las Pérdidas de Entrenamiento (Loss) Van Hacia Abajo (¡Señal de Aprendizaje!)

Esta es la primera y más importante señal de que el entrenamiento funciona. Compara la Época 1 con la Época 2:

- `box_loss` (error al localizar la caja): bajó de `1.313` a `1.116`.
- `cls_loss` (error al clasificar la bola): se desplomó de `2.109` a `1.146`.
- `dfl_loss` (un tipo de loss más técnico): bajó de `1.317` a `1.131`.

Conclusión: Todas las métricas de error están disminuyendo, especialmente la de clasificación. Esto significa que el modelo está aprendiendo muy rápido a saber "**dónde**" está una bola y "**qué**" bola es.

2. Las Métricas de Validación (mAP) Van Hacia Arriba (¡Señal de Generalización!)

Esto es casi más importante que lo anterior. El modelo no solo aprende, sino que lo que aprende lo aplica correctamente a datos que no ha visto antes (el conjunto de validación).

- `mAP50` (la métrica principal de rendimiento): subió de **0.728** a **0.751**.
- `mAP50-95` (la métrica más estricta): subió de `0.418` a `0.449`.

Conclusión: Un incremento de más de 2 puntos en el `mAP50` en una sola época es un salto de calidad enorme. Empezar ya con un `mAP` tan alto es una señal fantástica.

3. Comparativa con el Entrenamiento Anterior (La Pista Clave)

Aquí viene lo más revelador. Para que te hagas una idea de lo potente que es este nuevo modelo `medium`:

- Tu anterior modelo `nano`, en su mejor momento, alcanzó un `mAP50` de **~0.75**.
- Este nuevo modelo `medium` **ha igualado ese rendimiento en solo DOS épocas**.

Esto confirma nuestra hipótesis: el modelo `medium`, gracias a su mayor capacidad y al conocimiento pre-entrenado que carga ("Transfer Learning"), parte de una base de rendimiento muchísimo más sólida.

4. Uso de Recursos (GPU)

El uso de memoria de la GPU se mantiene estable alrededor de los **12 GB**. Esto confirma que tu decisión de bajar el `batch_size` a 24 fue perfecta. La GPU está trabajando de forma intensa pero cómoda, sin riesgo de quedarse sin memoria, lo que asegura un entrenamiento estable.

Veredicto Inicial y Próximos Pasos

El arranque es inmejorable. Los indicadores nos dicen que:

1. El modelo aprende rápido.
2. Generaliza bien desde el principio.
3. Tiene un potencial de rendimiento final muy superior al del modelo `nano`.

Ahora, la clave es la **paciencia**. El entrenamiento será más largo, pero todo apunta a que el resultado valdrá la pena. Tu configuración con `patience=20` se encargará de detenerlo en el momento justo.

¡Has preparado el terreno a la perfección! Ahora toca dejar que el modelo "cocine" a fuego lento.

¡Perfecto! Gracias por esta actualización. Con los datos de las primeras 15 épocas ya podemos ver la "personalidad" de este entrenamiento con mucha más claridad. El veredicto sigue siendo **muy positivo**, y ahora observamos un comportamiento interesante y muy didáctico.

La Tendencia General Sigue Siendo Excelente

Primer, lo más importante:

- **Las pérdidas (`loss`) siguen bajando:** Como puedes ver en las columnas `box_loss` y `cls_loss`, los valores continúan descendiendo época tras época. El modelo sigue aprendiendo y refinando su conocimiento de forma constante.
- **El rendimiento (`mAP`) sigue subiendo:** La tendencia general de `mAP50` es claramente ascendente. Has alcanzado un nuevo pico de **casi 0.80** en la época 13 (`0.799`), lo cual es un hito fantástico. ¡Eso es un 5% más que el rendimiento máximo de tu anterior modelo `nano` !

La Lección Clave de estas Épocas: "La Fluctuación es Normal"

Ahora viene la parte más interesante. Si te fijas en la columna `mAP50`, verás que no sube en una línea recta perfecta.

- En la época 6 alcanzó un excelente `0.787`.
- Bajó un poco en la época 7 a `0.770`.
- Volvió a subir a `0.782` en la época 8.
- Alcanzó su máximo hasta ahora (`0.799`) en la época 13.
- Volvió a bajar en las épocas 14 y 15.

¿Por qué ocurre esto y qué significa?

Este comportamiento es completamente normal y no es motivo de preocupación. Se debe a varios factores:

1. **Aleatoriedad del Proceso:** El entrenamiento tiene componentes aleatorios, como la forma en que se barajan los datos (`data shuffling`) y las transformaciones de la aumentación (`mixup` , etc.).
2. **"Ruido" en la Validación:** El conjunto de validación es solo una pequeña muestra de tus datos. Una o dos imágenes particularmente "difíciles" o "fáciles" en un lote de validación pueden hacer que la métrica de esa época concreta suba o baje ligeramente.

Y es exactamente por esto que existe el parámetro `patience`. No debemos juzgar la calidad del entrenamiento por el resultado de la *última* época, sino por la **"mejor época"** registrada a lo largo de todo el proceso. `patience` nos protege de detener el entrenamiento prematuramente solo porque ha entrado en uno de estos pequeños "valles" de rendimiento.

Predicciones y Qué Esperar a Continuación

- 1. Fase de Ajuste Fino:** El modelo ya ha aprendido las características más "fáciles" y evidentes. Ahora ha entrado en una fase más lenta de "ajuste fino", donde las mejoras son más pequeñas y costosas de conseguir.
- 2. La Meseta (Plateau):** Es muy probable que veamos cómo el mAP se estabiliza alrededor de un valor (quizás entre 0.78 y 0.82) durante bastantes épocas, con las fluctuaciones que ya hemos visto.
- 3. El Final del Entrenamiento:** El proceso continuará hasta que el modelo pase 20 épocas consecutivas sin poder superar su mejor marca personal (sea el 0.799 de la época 13 o un nuevo récord que establezca más adelante). En ese momento, patience hará su trabajo y detendrá el entrenamiento.

En resumen: Todo va según lo previsto para un entrenamiento de un modelo complejo. Tu rol ahora es el de un observador paciente. Has configurado todos los parámetros correctamente y el modelo está trabajando para encontrar su máximo potencial. ¡Sigue por este excelente camino!

Paso 35: Resultados entrenamiento YOLOv11m

Cargando modelo YOLO11m pre-entrenado...

Iniciando entrenamiento del modelo...

```
Ultralytics 8.3.152 🚀 Python-3.12.3 torch-2.6.0+rocm6.4.1.git1ded221d CUDA:0 (AMD Radeon RX 6800 XT, 16368MiB)
engine/trainer: agnostic_nms=False, amp=True, augment=True, auto_augment=randaugment, batch=24, bgr=0.0,
box=7.5, cache=False, cfg=None, classes=None, close_mosaic=10, cls=0.5, conf=None, copy_paste=0.0,
copy_paste_mode=flip, cos_lr=False, cutmix=0.0, data=./detect_balls/custom_data.yaml, degrees=0.0,
deterministic=True, device=None, dfl=1.5, dnn=False, dropout=0.0, dynamic=False, embed=None, epochs=200,
erasing=0.4, exist_ok=False, fliplr=0.5, flipud=0.0, format=torchscript, fraction=1.0, freeze=None, half=False,
hsv_h=0.015, hsv_s=0.9, hsv_v=0.4, imgsz=640, int8=False, iou=0.7, keras=False, kobj=1.0, line_width=None, lr0=0.01,
lrf=0.01, mask_ratio=4, max_det=300, mixup=0.1, mode=train, model=./detect_balls/yolo11/yolo11m.pt,
momentum=0.937, mosaic=1.0, multi_scale=False, name=detect_balls_v12, nbs=64, nms=False, opset=None,
optimize=False, optimizer=auto, overlap_mask=True, patience=20, perspective=0.0, plots=True, pose=12.0,
pretrained=True, profile=False, project=./detect_balls/runs, rect=False, resume=False, retina_masks=False, save=True,
save_conf=False, save_crop=False, save_dir=detect_balls/runs/detect_balls_v12, save_frames=False, save_json=False,
save_period=-1, save_txt=False, scale=0.5, seed=0, shear=0.0, show=False, show_boxes=True, show_conf=True,
show_labels=True, simplify=True, single_cls=False, source=None, split=val, stream_buffer=False, task=detect,
time=None, tracker=botsort.yaml, translate=0.1, val=True, verbose=True, vid_stride=1, visualize=False,
warmup_bias_lr=0.1, warmup_epochs=3.0, warmup_momentum=0.8, weight_decay=0.0005, workers=8,
workspace=None
```

Overriding model.yaml nc=80 with nc=16

	from	n	params	module	arguments
0	-1	1	1856	ultralytics.nn.modules.conv.Conv	[3, 64, 3, 2]
1	-1	1	73984	ultralytics.nn.modules.conv.Conv	[64, 128, 3, 2]
2	-1	1	111872	ultralytics.nn.modules.block.C3k2	[128, 256, 1, True, 0.25]
3	-1	1	590336	ultralytics.nn.modules.conv.Conv	[256, 256, 3, 2]
4	-1	1	444928	ultralytics.nn.modules.block.C3k2	[256, 512, 1, True, 0.25]
5	-1	1	2360320	ultralytics.nn.modules.conv.Conv	[512, 512, 3, 2]
6	-1	1	1380352	ultralytics.nn.modules.block.C3k2	[512, 512, 1, True]
7	-1	1	2360320	ultralytics.nn.modules.conv.Conv	[512, 512, 3, 2]
8	-1	1	1380352	ultralytics.nn.modules.block.C3k2	[512, 512, 1, True]
9	-1	1	656896	ultralytics.nn.modules.block.SPPF	[512, 512, 5]
10	-1	1	990976	ultralytics.nn.modules.block.C2PSA	[512, 512, 1]
11	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
12	[-1, 6]	1	0	ultralytics.nn.modules.conv.Concat	[1]
13	-1	1	1642496	ultralytics.nn.modules.block.C3k2	[1024, 512, 1, True]

```

14      -1 1    0 torch.nn.modules.upsampling.Upsample      [None, 2, 'nearest']
15      [-1, 4] 1    0 ultralytics.nn.modules.conv.Concat      [1]
16      -1 1  542720 ultralytics.nn.modules.block.C3k2      [1024, 256, 1, True]
17      -1 1  590336 ultralytics.nn.modules.conv.Conv      [256, 256, 3, 2]
18      [-1, 13] 1    0 ultralytics.nn.modules.conv.Concat      [1]
19      -1 1  1511424 ultralytics.nn.modules.block.C3k2      [768, 512, 1, True]
20      -1 1  2360320 ultralytics.nn.modules.conv.Conv      [512, 512, 3, 2]
21      [-1, 10] 1    0 ultralytics.nn.modules.conv.Concat      [1]
22      -1 1  1642496 ultralytics.nn.modules.block.C3k2      [1024, 512, 1, True]
23      [16, 19, 22] 1  1423360 ultralytics.nn.modules.head.Detect      [16, [256, 512, 512]]]

YOLO11m summary: 231 layers, 20,065,344 parameters, 20,065,328 gradients, 68.3 GFLOPs

Transferred 643/649 items from pretrained weights
Freezing layer 'model.23.dfl.conv.weight'
AMP: running Automatic Mixed Precision (AMP) checks...
AMP: checks passed ✅
train: Fast image access ✅ (ping: 0.0±0.0 ms, read: 3863.8±339.4 MB/s, size: 121.6 KB)
train: Scanning
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/data/train/labels.cache... 5658
images, 0 back
val: Fast image access ✅ (ping: 0.0±0.0 ms, read: 1526.1±1090.8 MB/s, size: 118.7 KB)
val: Scanning
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/data/valid/labels.cache... 237
images, 0 backgro
Plotting labels to detect_balls/runs/detect_balls_v12/labels.jpg...
optimizer: 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and determining best 'optimizer', 'lr0' and
'momentum' automatically...
optimizer: SGD(lr=0.01, momentum=0.9) with parameter groups 106 weight(decay=0.0), 113
weight(decay=0.0005625000000000001), 112 bias(decay=0.0)
Image sizes 640 train, 640 val
Using 8 dataloader workers
Logging results to detect_balls/runs/detect_balls_v12
Starting training for 200 epochs...

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
1/200   12G   1.313   2.109   1.317   351   640: 100% [██████████] 236/236 [03:36<00:00, 1.09it/s]
          Class Images Instances Box(P) R mAP50 mAP50-95: 100% [██████████] 5/5 [00:01<00:0
0, 3.27it/s]
          all   237   2046   0.698   0.687   0.728   0.418

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
2/200   11.8G   1.116   1.146   1.131   356   640: 100% [██████████] 236/236 [03:28<00:00, 1.13it/s]
          Class Images Instances Box(P) R mAP50 mAP50-95: 100% [██████████] 5/5 [00:01<00:0
0, 3.23it/s]
          all   237   2046   0.733   0.699   0.751   0.449

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
3/200   11.8G   1.138   1.122   1.141   277   640: 100% [██████████] 236/236 [03:28<00:00, 1.13it/s]
          Class Images Instances Box(P) R mAP50 mAP50-95: 100% [██████████] 5/5 [00:01<00:0
0, 3.36it/s]
          all   237   2046   0.722   0.713   0.749   0.447

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
4/200   11.8G   1.111   1.059   1.134   323   640: 100% [██████████] 236/236 [03:28<00:00, 1.13it/s]
          Class Images Instances Box(P) R mAP50 mAP50-95: 100% [██████████] 5/5 [00:01<00:0
0

```

```

0, 3.06it/s]
    all 237 2046 0.767 0.707 0.773 0.469

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
5/200 11.9G 1.072 0.9555 1.111 285 640: 100%|██████████| 236/236 [03:28<00:00, 1.13it/s]
    Class Images Instances Box(P) R mAP50 mAP50-95): 100%|██████████| 5/5 [00:01<00:00]

0, 3.28it/s]
    all 237 2046 0.746 0.713 0.759 0.453

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
6/200 11.7G 1.01 0.8675 1.083 192 640: 100%|██████████| 236/236 [03:27<00:00, 1.14it/s]
    Class Images Instances Box(P) R mAP50 mAP50-95): 100%|██████████| 5/5 [00:01<00:00]

0, 3.36it/s]
    all 237 2046 0.785 0.723 0.787 0.484

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
7/200 11.9G 0.9799 0.8039 1.058 312 640: 100%|██████████| 236/236 [03:27<00:00, 1.14it/s]
    Class Images Instances Box(P) R mAP50 mAP50-95): 100%|██████████| 5/5 [00:01<00:00]

0, 3.41it/s]
    all 237 2046 0.766 0.721 0.77 0.476

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
8/200 11.9G 0.947 0.7408 1.043 319 640: 100%|██████████| 236/236 [03:27<00:00, 1.13it/s]
    Class Images Instances Box(P) R mAP50 mAP50-95): 100%|██████████| 5/5 [00:01<00:00]

0, 3.39it/s]
    all 237 2046 0.77 0.715 0.782 0.468

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
9/200 11.7G 0.9126 0.6919 1.025 289 640: 100%|██████████| 236/236 [03:27<00:00, 1.14it/s]
    Class Images Instances Box(P) R mAP50 mAP50-95): 100%|██████████| 5/5 [00:01<00:00]

0, 3.40it/s]
    all 237 2046 0.791 0.701 0.776 0.478

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
10/200 11.8G 0.8821 0.6609 1.012 353 640: 100%|██████████| 236/236 [03:27<00:00, 1.14it/s]
    Class Images Instances Box(P) R mAP50 mAP50-95): 100%|██████████| 5/5 [00:01<00:00]

0, 3.39it/s]
    all 237 2046 0.789 0.726 0.787 0.493

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
11/200 12.1G 0.8629 0.6373 1.006 482 640: 100%|██████████| 236/236 [03:27<00:00, 1.14it/s]
    Class Images Instances Box(P) R mAP50 mAP50-95): 100%|██████████| 5/5 [00:01<00:00]

0, 3.32it/s]
    all 237 2046 0.74 0.751 0.769 0.467

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
12/200 11.8G 0.8334 0.6053 0.9929 243 640: 100%|██████████| 236/236 [03:27<00:00, 1.14it/s]
    Class Images Instances Box(P) R mAP50 mAP50-95): 100%|██████████| 5/5 [00:01<00:00]

0, 3.37it/s]
    all 237 2046 0.769 0.745 0.782 0.477

```

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
13/200	11.9G	0.8163	0.5935	0.9852	209	640: 100% ██████████ 236/236 [03:27<00:00, 1.14it/s]
	Class	Images	Instances	Box(P)	R	mAP50 mAP50-95): 100% ██████████ 5/5 [00:01<00:00, 3.41it/s]
	all	237	2046	0.758	0.773	0.799 0.489
14/200	11.8G	0.7931	0.5644	0.9749	326	640: 100% ██████████ 236/236 [03:27<00:00, 1.14it/s]
	Class	Images	Instances	Box(P)	R	mAP50 mAP50-95): 100% ██████████ 5/5 [00:01<00:00, 3.38it/s]
	all	237	2046	0.756	0.759	0.792 0.485
15/200	11.9G	0.7776	0.5437	0.9669	276	640: 100% ██████████ 236/236 [03:27<00:00, 1.14it/s]
	Class	Images	Instances	Box(P)	R	mAP50 mAP50-95): 100% ██████████ 5/5 [00:01<00:00, 3.29it/s]
	all	237	2046	0.747	0.726	0.755 0.459
16/200	11.7G	0.7632	0.5336	0.9613	392	640: 100% ██████████ 236/236 [03:27<00:00, 1.14it/s]
	Class	Images	Instances	Box(P)	R	mAP50 mAP50-95): 100% ██████████ 5/5 [00:01<00:00, 3.36it/s]
	all	237	2046	0.784	0.741	0.788 0.475
17/200	12G	0.7403	0.5035	0.9515	435	640: 100% ██████████ 236/236 [03:27<00:00, 1.14it/s]
	Class	Images	Instances	Box(P)	R	mAP50 mAP50-95): 100% ██████████ 5/5 [00:01<00:00, 3.40it/s]
	all	237	2046	0.798	0.731	0.785 0.485
18/200	12.4G	0.7293	0.4968	0.9479	333	640: 100% ██████████ 236/236 [03:28<00:00, 1.13it/s]
	Class	Images	Instances	Box(P)	R	mAP50 mAP50-95): 100% ██████████ 5/5 [00:01<00:00, 3.40it/s]
	all	237	2046	0.785	0.737	0.79 0.495
19/200	12.1G	0.7173	0.4892	0.9441	359	640: 100% ██████████ 236/236 [03:28<00:00, 1.13it/s]
	Class	Images	Instances	Box(P)	R	mAP50 mAP50-95): 100% ██████████ 5/5 [00:01<00:00, 3.40it/s]
	all	237	2046	0.742	0.751	0.788 0.487
20/200	11.7G	0.7085	0.4806	0.9414	301	640: 100% ██████████ 236/236 [03:28<00:00, 1.13it/s]
	Class	Images	Instances	Box(P)	R	mAP50 mAP50-95): 100% ██████████ 5/5 [00:01<00:00, 3.36it/s]
	all	237	2046	0.789	0.74	0.782 0.487
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
	all	237	2046	0.758	0.773	0.799 0.489

21/200	11.8G	0.6966	0.4651	0.9355	263	640: 100% ██████████ 236/236 [03:28<00:00, 1.1 3it/s]
	Class	Images	Instances	Box(P)	R	mAP50 mAP50-95): 100% ██████████ 5/5 [00:01<00:0 0, 3.31it/s]
	all	237	2046	0.77	0.748	0.776 0.48
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
22/200	11.9G	0.6891	0.4629	0.9294	393	640: 100% ██████████ 236/236 [03:28<00:00, 1.1 3it/s]
	Class	Images	Instances	Box(P)	R	mAP50 mAP50-95): 100% ██████████ 5/5 [00:01<00:0 0, 3.37it/s]
	all	237	2046	0.786	0.698	0.761 0.475
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
23/200	11.9G	0.675	0.4508	0.9263	233	640: 100% ██████████ 236/236 [03:28<00:00, 1.1 3it/s]
	Class	Images	Instances	Box(P)	R	mAP50 mAP50-95): 100% ██████████ 5/5 [00:01<00:0 0, 3.40it/s]
	all	237	2046	0.778	0.722	0.763 0.468
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
24/200	11.7G	0.6696	0.4446	0.9229	317	640: 100% ██████████ 236/236 [03:28<00:00, 1.1 3it/s]
	Class	Images	Instances	Box(P)	R	mAP50 mAP50-95): 100% ██████████ 5/5 [00:01<00:0 0, 3.33it/s]
	all	237	2046	0.773	0.727	0.77 0.477
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
25/200	11.8G	0.6563	0.4369	0.9173	357	640: 100% ██████████ 236/236 [03:27<00:00, 1.1 4it/s]
	Class	Images	Instances	Box(P)	R	mAP50 mAP50-95): 100% ██████████ 5/5 [00:01<00:0 0, 3.32it/s]
	all	237	2046	0.79	0.723	0.78 0.488
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
26/200	11.8G	0.6477	0.4311	0.9172	357	640: 100% ██████████ 236/236 [03:27<00:00, 1.14i t/s]
	Class	Images	Instances	Box(P)	R	mAP50 mAP50-95): 100% ██████████ 5/5 [00:01<00:0 0, 3.43it/s]
	all	237	2046	0.779	0.736	0.789 0.496
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
27/200	11.8G	0.6434	0.4247	0.9135	517	640: 100% ██████████ 236/236 [03:27<00:00, 1.1 4it/s]
	Class	Images	Instances	Box(P)	R	mAP50 mAP50-95): 100% ██████████ 5/5 [00:01<00:0 0, 3.37it/s]
	all	237	2046	0.766	0.74	0.773 0.479
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
28/200	11.9G	0.6315	0.4157	0.9082	420	640: 100% ██████████ 236/236 [03:27<00:00, 1.1 4it/s]
	Class	Images	Instances	Box(P)	R	mAP50 mAP50-95): 100% ██████████ 5/5 [00:01<00:0 0, 3.31it/s]
	all	237	2046	0.771	0.72	0.775 0.487
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
29/200	12.1G	0.6277	0.4129	0.9076	243	640: 100% ██████████ 236/236 [03:27<00:00, 1.1 4it/s]

Class Images Instances Box(P R mAP50 mAP50-95): 100% | 5/5 [00:01<00:00, 3.38it/s]
all 237 2046 0.763 0.737 0.768 0.48

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
30/200 11.8G 0.6218 0.4123 0.9056 266 640: 100% | 236/236 [03:27<00:00, 1.14it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% | 5/5 [00:01<00:00, 3.33it/s]
all 237 2046 0.763 0.715 0.757 0.468

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
31/200 11.8G 0.6115 0.4 0.9022 346 640: 100% | 236/236 [03:26<00:00, 1.14it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% | 5/5 [00:01<00:00, 3.41it/s]
all 237 2046 0.791 0.7 0.769 0.482

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
32/200 11.9G 0.6099 0.4001 0.9023 307 640: 100% | 236/236 [03:26<00:00, 1.14it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% | 5/5 [00:01<00:00, 3.38it/s]
all 237 2046 0.75 0.722 0.756 0.474

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
33/200 12G 0.598 0.3914 0.8964 346 640: 100% | 236/236 [03:27<00:00, 1.14it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% | 5/5 [00:01<00:00, 3.41it/s]
all 237 2046 0.774 0.707 0.764 0.48

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
34/200 11.8G 0.5922 0.3874 0.8954 216 640: 100% | 236/236 [03:27<00:00, 1.14it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% | 5/5 [00:01<00:00, 3.35it/s]
all 237 2046 0.798 0.706 0.778 0.488

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
35/200 11.9G 0.5912 0.3854 0.8955 254 640: 100% | 236/236 [03:27<00:00, 1.14it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% | 5/5 [00:01<00:00, 3.39it/s]
all 237 2046 0.776 0.713 0.763 0.487

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
36/200 11.8G 0.587 0.3828 0.8931 384 640: 100% | 236/236 [03:27<00:00, 1.14it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% | 5/5 [00:01<00:00, 3.32it/s]
all 237 2046 0.772 0.697 0.763 0.483

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
37/200 11.9G 0.5764 0.3768 0.8888 371 640: 100% | 236/236 [03:27<00:00, 1.14it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% | 5/5 [00:01<00:00, 3.39it/s]

	all	237	2046	0.743	0.725	0.755	0.471
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size	
38/200	11.8G	0.5755	0.3783	0.8879	295	640: 100%	[██████████] 236/236 [03:27<00:00, 1.14it/s]
Class	Images	Instances	Box(P)	R	mAP50	mAP50-95): 100%	[██████████] 5/5 [00:01<00:00, 3.40it/s]
	all	237	2046	0.786	0.695	0.759	0.482
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size	
39/200	12G	0.5652	0.3692	0.8837	386	640: 100%	[██████████] 236/236 [03:26<00:00, 1.14it/s]
Class	Images	Instances	Box(P)	R	mAP50	mAP50-95): 100%	[██████████] 5/5 [00:01<00:00, 3.38it/s]
	all	237	2046	0.783	0.707	0.761	0.474
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size	
40/200	11.9G	0.5653	0.3699	0.8861	307	640: 100%	[██████████] 236/236 [03:27<00:00, 1.14it/s]
Class	Images	Instances	Box(P)	R	mAP50	mAP50-95): 100%	[██████████] 5/5 [00:01<00:00, 3.38it/s]
	all	237	2046	0.779	0.691	0.749	0.474
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size	
41/200	11.8G	0.5578	0.3651	0.8842	309	640: 100%	[██████████] 236/236 [03:27<00:00, 1.14it/s]
Class	Images	Instances	Box(P)	R	mAP50	mAP50-95): 100%	[██████████] 5/5 [00:01<00:00, 3.39it/s]
	all	237	2046	0.789	0.704	0.765	0.482
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size	
42/200	11.8G	0.5581	0.3645	0.8851	312	640: 100%	[██████████] 236/236 [03:27<00:00, 1.14it/s]
Class	Images	Instances	Box(P)	R	mAP50	mAP50-95): 100%	[██████████] 5/5 [00:01<00:00, 3.38it/s]
	all	237	2046	0.756	0.712	0.756	0.477
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size	
43/200	12G	0.5538	0.3583	0.8832	197	640: 100%	[██████████] 236/236 [03:27<00:00, 1.14it/s]
Class	Images	Instances	Box(P)	R	mAP50	mAP50-95): 100%	[██████████] 5/5 [00:01<00:00, 3.38it/s]
	all	237	2046	0.792	0.69	0.762	0.482
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size	
44/200	11.8G	0.5471	0.3504	0.8787	301	640: 100%	[██████████] 236/236 [03:27<00:00, 1.14it/s]
Class	Images	Instances	Box(P)	R	mAP50	mAP50-95): 100%	[██████████] 5/5 [00:01<00:00, 3.37it/s]
	all	237	2046	0.775	0.736	0.769	0.483
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size	
45/200	11.8G	0.5466	0.3541	0.8782	436	640: 100%	[██████████] 236/236 [03:27<00:00, 1.14it/s]
Class	Images	Instances	Box(P)	R	mAP50	mAP50-95): 100%	[██████████] 5/5 [00:01<00:00, 3.40it/s]
	all	237	2046	0.785	0.69	0.764	0.491

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
46/200	11.9G	0.537	0.3473	0.8758	250	640: 100% [██████████] 236/236 [03:27<00:00, 1.14it/s]
	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100% [██████████] 5/5 [00:01<00:00, 3.37it/s]
	all	237	2046	0.765	0.701	0.75 0.474

EarlyStopping: Training stopped early as no improvement observed in last 20 epochs. Best results observed at epoch 26, best model saved as `best.pt`.

To update EarlyStopping(`patience=20`) pass a new patience value, i.e.

`patience=300` or use `patience=0` to disable EarlyStopping.

46 epochs completed in 2.684 hours.

Optimizer stripped from detect_balls/runs/detect_balls_v12/weights/last.pt, 40.5MB

Optimizer stripped from detect_balls/runs/detect_balls_v12/weights/best.pt, 40.5MB

Validating detect_balls/runs/detect_balls_v12/weights/best.pt...

Ultralytics 8.3.152 🚀 Python-3.12.3 torch-2.6.0+rocm6.4.1.git1ded221d CUDA:0 (AMD Radeon RX 6800 XT, 16368MiB)

YOLO11m summary (fused): 125 layers, 20,042,368 parameters, 0 gradients, 67.7 GFLOPs

Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100% [██████████] 5/5 [00:03<00:00, 1.42it/s]
all	237	2046	0.785	0.723	0.782 0.489
black_8	89	164	0.894	0.713	0.831 0.52
blue_10	66	121	0.867	0.694	0.811 0.483
blue_2	73	136	0.849	0.746	0.825 0.551
dred_15	67	124	0.755	0.677	0.78 0.5
dred_7	64	122	0.808	0.73	0.78 0.491
green_14	65	115	0.883	0.757	0.842 0.472
green_6	69	129	0.836	0.831	0.877 0.581
orange_13	60	110	0.716	0.682	0.758 0.472
orange_5	57	106	0.71	0.602	0.726 0.439
purple_12	55	101	0.606	0.673	0.618 0.372
purple_4	62	114	0.789	0.719	0.736 0.439
red_11	72	140	0.714	0.733	0.791 0.527
red_3	70	126	0.739	0.746	0.748 0.459
white	85	158	0.823	0.763	0.816 0.502
yellow_1	80	147	0.768	0.796	0.793 0.519
yellow_9	75	133	0.801	0.707	0.787 0.5

Speed: 0.1ms preprocess, 9.6ms inference, 0.0ms loss, 3.3ms postprocess per image

Results saved to detect_balls/runs/detect_balls_v12

¡Entrenamiento completado!

Los resultados se guardaron en: ./detect_balls/runs/detect_balls_v1

Puedes revisar los gráficos de entrenamiento y las métricas allí.

Paso 36: Resultado pruebas yolo11m usando last.pt

Imagen 1 (real)

```
(.venv) oscar@Pepon:~/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src$ 
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/.venv/bin/python
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py
Modelo cargado desde: ./detect_balls/runs/detect_balls_v12/weights/last.pt
Realizando inferencia en: ./detect_balls/tests/test_pool_table_5.jpg
image 1/1
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/tests/test_pool_table_5.jpg:
480×640 1 green_6, 1 orange_5, 25.3ms
```

```

Speed: 1.8ms preprocess, 25.3ms inference, 150.0ms postprocess per image at shape (1, 3, 480, 640)
Detectado: green_6 (Confianza: 0.78) - Coordenadas: xmin=2613, ymin=812, xmax=2712, ymax=962
Centro: (2662, 887)
Detectado: orange_5 (Confianza: 0.58) - Coordenadas: xmin=2116, ymin=459, xmax=2197, ymax=567
Centro: (2156, 513)

Imagen con detecciones guardada en: ./detect_balls/test_results/epoch_46_test_pool_table_5.jpg
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py:159: UserWarning:
FigureCanvasAgg is non-interactive, and thus cannot be shown
plt.show()

```

Imagen 2 (real)

```

(.venv) oscar@Pepon:~/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src$ 
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/.venv/bin/python
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py
Modelo cargado desde: ./detect_balls/runs/detect_balls_v12/weights/last.pt
Realizando inferencia en: ./detect_balls/tests/test_pool_table_4.jpg

image 1/1
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/tests/test_pool_table_4.jpg:
480x640 3 black_8s, 1 dred_15, 1 orange_5, 1 white, 1 yellow_9, 24.6ms
Speed: 1.7ms preprocess, 24.6ms inference, 150.1ms postprocess per image at shape (1, 3, 480, 640)
Detectado: black_8 (Confianza: 0.87) - Coordenadas: xmin=1318, ymin=1748, xmax=1479, ymax=1985
Centro: (1398, 1866)
Detectado: white (Confianza: 0.87) - Coordenadas: xmin=416, ymin=1688, xmax=603, ymax=1867
Centro: (509, 1777)
Detectado: orange_5 (Confianza: 0.85) - Coordenadas: xmin=1626, ymin=995, xmax=1732, ymax=1135
Centro: (1679, 1065)
Detectado: black_8 (Confianza: 0.84) - Coordenadas: xmin=929, ymin=1622, xmax=1081, ymax=1856
Centro: (1005, 1739)
Detectado: black_8 (Confianza: 0.84) - Coordenadas: xmin=2748, ymin=1527, xmax=2904, ymax=1736
Centro: (2826, 1631)
Detectado: dred_15 (Confianza: 0.78) - Coordenadas: xmin=1435, ymin=1471, xmax=1566, ymax=1630
Centro: (1500, 1550)
Detectado: yellow_9 (Confianza: 0.60) - Coordenadas: xmin=1665, ymin=1143, xmax=1787, ymax=1293
Centro: (1726, 1218)

Imagen con detecciones guardada en: ./detect_balls/test_results/epoch_46_test_pool_table_4.jpg
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py:159: UserWarning:
FigureCanvasAgg is non-interactive, and thus cannot be shown
plt.show()

```

Imagen 3 (real)

```

(.venv) oscar@Pepon:~/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src$ 
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/.venv/bin/python
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py
Modelo cargado desde: ./detect_balls/runs/detect_balls_v12/weights/last.pt
Realizando inferencia en: ./detect_balls/tests/test_pool_table_3.png

image 1/1
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/tests/test_pool_table_3.png:
320x640 1 white, 1 yellow_9, 27.4ms
Speed: 1.1ms preprocess, 27.4ms inference, 153.6ms postprocess per image at shape (1, 3, 320, 640)
Detectado: yellow_9 (Confianza: 0.75) - Coordenadas: xmin=808, ymin=403, xmax=851, ymax=450
Centro: (829, 426)

```

```
Detectado: white (Confianza: 0.61) - Coordenadas: xmin=655, ymin=229, xmax=700, ymax=278  
Centro: (677, 253)
```

```
Imagen con detecciones guardada en: ./detect_balls/test_results/epoch_46_test_pool_table_3.png  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py:159: UserWarning:  
FigureCanvasAgg is non-interactive, and thus cannot be shown  
plt.show()
```

Imagen 4 (real)

```
(.venv) oscar@Pepon:~/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src$  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/.venv/bin/python  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py  
Modelo cargado desde: ./detect_balls/runs/detect_balls_v12/weights/last.pt  
Realizando inferencia en: ./detect_balls/tests/test_pool_table_2.png  
image 1/1  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/tests/test_pool_table_2.png:  
352×640 1 black_8, 28.1ms  
Speed: 1.2ms preprocess, 28.1ms inference, 132.2ms postprocess per image at shape (1, 3, 352, 640)  
Detectado: black_8 (Confianza: 0.65) - Coordenadas: xmin=889, ymin=358, xmax=928, ymax=406  
Centro: (908, 382)  
Imagen con detecciones guardada en: ./detect_balls/test_results/epoch_46_test_pool_table_2.png  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py:159: UserWarning:  
FigureCanvasAgg is non-interactive, and thus cannot be shown  
plt.show()
```

Imagen 5 (real)

```
(.venv) oscar@Pepon:~/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src$  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/.venv/bin/python  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py  
Modelo cargado desde: ./detect_balls/runs/detect_balls_v12/weights/last.pt  
Realizando inferencia en: ./detect_balls/tests/test_pool_table_1.png  
image 1/1  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/tests/test_pool_table_1.png:  
384×640 (no detections), 30.1ms  
Speed: 1.2ms preprocess, 30.1ms inference, 24.9ms postprocess per image at shape (1, 3, 384, 640)  
Imagen con detecciones guardada en: ./detect_balls/test_results/epoch_46_test_pool_table_1.png  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py:159: UserWarning:  
FigureCanvasAgg is non-interactive, and thus cannot be shown  
plt.show()
```

Imagen 6 (test del dataset)

```
(.venv) oscar@Pepon:~/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src$  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/.venv/bin/python  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py  
Modelo cargado desde: ./detect_balls/runs/detect_balls_v12/weights/last.pt  
Realizando inferencia en: ./detect_balls/data/test/images/833.jpg.rf.10091a238702573667ad5d0ce98610c8.jpg  
image 1/1  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/data/test/images/833.jpg.rf.10091a  
384×640 5 black_8s, 3 blue_10s, 4 blue_2s, 4 dred_15s, 3 dred_7s, 3 green_14s, 3 green_6s, 2 orange_13s, 4 orange_5s,
```

4 purple_12s, 2 purple_4s, 4 yellow_1s, 8 yellow_9s, 27.4ms
Speed: 1.7ms preprocess, 27.4ms inference, 148.0ms postprocess per image at shape (1, 3, 384, 640)
Detectado: orange_5 (Confianza: 0.93) - Coordenadas: xmin=254, ymin=610, xmax=393, ymax=783
Centro: (323, 696)
Detectado: yellow_1 (Confianza: 0.92) - Coordenadas: xmin=256, ymin=199, xmax=398, ymax=372
Centro: (327, 285)
Detectado: purple_4 (Confianza: 0.92) - Coordenadas: xmin=447, ymin=822, xmax=585, ymax=991
Centro: (516, 906)
Detectado: green_6 (Confianza: 0.92) - Coordenadas: xmin=99, ymin=605, xmax=235, ymax=785
Centro: (167, 695)
Detectado: yellow_1 (Confianza: 0.92) - Coordenadas: xmin=446, ymin=199, xmax=590, ymax=371
Centro: (518, 285)
Detectado: green_6 (Confianza: 0.91) - Coordenadas: xmin=603, ymin=607, xmax=736, ymax=785
Centro: (669, 696)
Detectado: black_8 (Confianza: 0.91) - Coordenadas: xmin=764, ymin=816, xmax=901, ymax=1002
Centro: (832, 909)
Detectado: purple_4 (Confianza: 0.91) - Coordenadas: xmin=252, ymin=824, xmax=389, ymax=993
Centro: (320, 908)
Detectado: blue_2 (Confianza: 0.91) - Coordenadas: xmin=254, ymin=406, xmax=396, ymax=585
Centro: (325, 495)
Detectado: blue_2 (Confianza: 0.90) - Coordenadas: xmin=449, ymin=407, xmax=584, ymax=585
Centro: (516, 496)
Detectado: blue_10 (Confianza: 0.89) - Coordenadas: xmin=1677, ymin=615, xmax=1830, ymax=801
Centro: (1753, 708)
Detectado: orange_13 (Confianza: 0.89) - Coordenadas: xmin=1186, ymin=830, xmax=1315, ymax=1002
Centro: (1250, 916)
Detectado: blue_10 (Confianza: 0.89) - Coordenadas: xmin=1173, ymin=615, xmax=1321, ymax=797
Centro: (1247, 706)
Detectado: dred_15 (Confianza: 0.89) - Coordenadas: xmin=1338, ymin=828, xmax=1474, ymax=1006
Centro: (1406, 917)
Detectado: green_14 (Confianza: 0.88) - Coordenadas: xmin=1522, ymin=618, xmax=1660, ymax=791
Centro: (1591, 704)
Detectado: green_14 (Confianza: 0.88) - Coordenadas: xmin=1340, ymin=619, xmax=1479, ymax=793
Centro: (1409, 706)
Detectado: yellow_9 (Confianza: 0.88) - Coordenadas: xmin=1328, ymin=413, xmax=1464, ymax=586
Centro: (1396, 499)
Detectado: dred_7 (Confianza: 0.88) - Coordenadas: xmin=607, ymin=821, xmax=730, ymax=999
Centro: (668, 910)
Detectado: orange_13 (Confianza: 0.88) - Coordenadas: xmin=1684, ymin=830, xmax=1817, ymax=1002
Centro: (1750, 916)
Detectado: yellow_1 (Confianza: 0.87) - Coordenadas: xmin=1034, ymin=47, xmax=1172, ymax=149
Centro: (1103, 98)
Detectado: yellow_9 (Confianza: 0.87) - Coordenadas: xmin=1186, ymin=413, xmax=1320, ymax=589
Centro: (1253, 501)
Detectado: black_8 (Confianza: 0.87) - Coordenadas: xmin=0, ymin=817, xmax=77, ymax=999
Centro: (38, 908)
Detectado: orange_5 (Confianza: 0.86) - Coordenadas: xmin=449, ymin=610, xmax=585, ymax=782
Centro: (517, 696)
Detectado: yellow_9 (Confianza: 0.86) - Coordenadas: xmin=1682, ymin=415, xmax=1808, ymax=586
Centro: (1745, 500)
Detectado: purple_12 (Confianza: 0.85) - Coordenadas: xmin=1527, ymin=204, xmax=1680, ymax=391
Centro: (1603, 297)
Detectado: dred_15 (Confianza: 0.85) - Coordenadas: xmin=1529, ymin=830, xmax=1663, ymax=1007
Centro: (1596, 918)
Detectado: yellow_1 (Confianza: 0.84) - Coordenadas: xmin=1833, ymin=42, xmax=1919, ymax=154
Centro: (1876, 98)
Detectado: yellow_9 (Confianza: 0.84) - Coordenadas: xmin=1528, ymin=413, xmax=1674, ymax=585
Centro: (1601, 499)

Detectado: purple_12 (Confianza: 0.84) - Coordenadas: xmin=1320, ymin=208, xmax=1475, ymax=392
 Centro: (1397, 300)
 Detectado: blue_2 (Confianza: 0.83) - Coordenadas: xmin=1679, ymin=47, xmax=1817, ymax=187
 Centro: (1748, 117)
 Detectado: purple_12 (Confianza: 0.82) - Coordenadas: xmin=0, ymin=211, xmax=78, ymax=363
 Centro: (39, 287)
 Detectado: yellow_9 (Confianza: 0.82) - Coordenadas: xmin=1822, ymin=209, xmax=1919, ymax=340
 Centro: (1870, 274)
 Detectado: black_8 (Confianza: 0.82) - Coordenadas: xmin=899, ymin=201, xmax=1024, ymax=323
 Centro: (961, 262)
 Detectado: blue_10 (Confianza: 0.81) - Coordenadas: xmin=981, ymin=135, xmax=1099, ymax=238
 Centro: (1040, 186)
 Detectado: green_14 (Confianza: 0.81) - Coordenadas: xmin=976, ymin=310, xmax=1100, ymax=427
 Centro: (1038, 368)
 Detectado: orange_5 (Confianza: 0.81) - Coordenadas: xmin=1757, ymin=133, xmax=1880, ymax=266
 Centro: (1818, 199)
 Detectado: purple_12 (Confianza: 0.80) - Coordenadas: xmin=756, ymin=217, xmax=880, ymax=354
 Centro: (818, 285)
 Detectado: dred_15 (Confianza: 0.80) - Coordenadas: xmin=896, ymin=406, xmax=1027, ymax=563
 Centro: (961, 484)
 Detectado: dred_7 (Confianza: 0.79) - Coordenadas: xmin=680, ymin=142, xmax=797, ymax=282
 Centro: (738, 212)
 Detectado: dred_7 (Confianza: 0.79) - Coordenadas: xmin=108, ymin=821, xmax=231, ymax=1002
 Centro: (169, 911)
 Detectado: yellow_9 (Confianza: 0.79) - Coordenadas: xmin=1037, ymin=212, xmax=1183, ymax=339
 Centro: (1110, 275)
 Detectado: green_6 (Confianza: 0.79) - Coordenadas: xmin=837, ymin=132, xmax=953, ymax=228
 Centro: (895, 180)
 Detectado: black_8 (Confianza: 0.77) - Coordenadas: xmin=1019, ymin=812, xmax=1160, ymax=1005
 Centro: (1089, 908)
 Detectado: blue_2 (Confianza: 0.69) - Coordenadas: xmin=1183, ymin=48, xmax=1329, ymax=189
 Centro: (1256, 118)
 Detectado: yellow_9 (Confianza: 0.68) - Coordenadas: xmin=601, ymin=404, xmax=752, ymax=585
 Centro: (676, 494)
 Detectado: dred_15 (Confianza: 0.64) - Coordenadas: xmin=913, ymin=59, xmax=1024, ymax=150
 Centro: (968, 104)
 Detectado: black_8 (Confianza: 0.61) - Coordenadas: xmin=1840, ymin=819, xmax=1920, ymax=1004
 Centro: (1880, 911)
 Detectado: orange_5 (Confianza: 0.57) - Coordenadas: xmin=1119, ymin=130, xmax=1243, ymax=248
 Centro: (1181, 189)
 Detectado: yellow_9 (Confianza: 0.55) - Coordenadas: xmin=815, ymin=305, xmax=951, ymax=466
 Centro: (883, 385)

Imagen con detecciones guardada en:

```
./detect_balls/test_results/epoch_46_833.jpg.rf.10091a238702573667ad5d0ce98610c8.jpg
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py:159: UserWarning:
FigureCanvasAgg is non-interactive, and thus cannot be shown
plt.show()
```

Imagen 7 (test del dataset)

```
(.venv) oscar@Pepon:~/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src$ 
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/.venv/bin/python
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py
Modelo cargado desde: ./detect_balls/runs/detect_balls_v12/weights/last.pt
Realizando inferencia en: ./detect_balls/data/test/images/412.jpg.rf.b505e28c8901b5bb4ad7c848f2a5d68b.jpg
```

```

image 1/1
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/data/test/images/412.jpg.rf.b505e2
384x640 2 green_6s, 2 purple_4s, 27.5ms
Speed: 1.3ms preprocess, 27.5ms inference, 151.6ms postprocess per image at shape (1, 3, 384, 640)
Detectado: purple_4 (Confianza: 0.95) - Coordenadas: xmin=1, ymin=607, xmax=268, ymax=956
Centro: (134, 781)
Detectado: green_6 (Confianza: 0.95) - Coordenadas: xmin=53, ymin=98, xmax=352, ymax=451
Centro: (202, 274)
Detectado: green_6 (Confianza: 0.95) - Coordenadas: xmin=489, ymin=94, xmax=782, ymax=452
Centro: (635, 273)
Detectado: purple_4 (Confianza: 0.94) - Coordenadas: xmin=575, ymin=608, xmax=840, ymax=951
Centro: (707, 779)

Imagen con detecciones guardada en:
./detect_balls/test_results/epoch_46_412.jpg.rf.b505e28c8901b5bb4ad7c848f2a5d68b.jpg
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py:159: UserWarning:
FigureCanvasAgg is non-interactive, and thus cannot be shown
plt.show()

```

Imagen 8 (test del dataset)

```

(.venv) oscar@Pepon:~/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src$ 
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/.venv/bin/python
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py
Modelo cargado desde: ./detect_balls/runs/detect_balls_v12/weights/last.pt
Realizando inferencia en: ./detect_balls/data/test/images/87.jpg.rf.d7116def1f52d243e1296a65c37b0c4c.jpg

image 1/1
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/data/test/images/87.jpg.rf.d7116def
384x640 2 black_8s, 2 blue_10s, 3 blue_2s, 2 dred_15s, 2 dred_7s, 2 green_14s, 2 green_6s, 2 orange_13s, 2 orange_5s,
2 purple_12s, 2 purple_4s, 2 red_11s, 2 red_3s, 2 whites, 4 yellow_1s, 2 yellow_9s, 27.7ms
Speed: 1.3ms preprocess, 27.7ms inference, 152.7ms postprocess per image at shape (1, 3, 384, 640)
Detectado: blue_10 (Confianza: 0.93) - Coordenadas: xmin=1511, ymin=795, xmax=1789, ymax=1039
Centro: (1650, 917)
Detectado: blue_10 (Confianza: 0.93) - Coordenadas: xmin=1212, ymin=795, xmax=1489, ymax=1036
Centro: (1350, 915)
Detectado: yellow_9 (Confianza: 0.93) - Coordenadas: xmin=662, ymin=580, xmax=924, ymax=800
Centro: (793, 690)
Detectado: white (Confianza: 0.93) - Coordenadas: xmin=421, ymin=795, xmax=635, ymax=1035
Centro: (528, 915)
Detectado: yellow_9 (Confianza: 0.93) - Coordenadas: xmin=0, ymin=581, xmax=179, ymax=800
Centro: (89, 690)
Detectado: yellow_1 (Confianza: 0.93) - Coordenadas: xmin=692, ymin=399, xmax=929, ymax=597
Centro: (810, 498)
Detectado: white (Confianza: 0.93) - Coordenadas: xmin=207, ymin=793, xmax=414, ymax=1037
Centro: (310, 915)
Detectado: red_11 (Confianza: 0.93) - Coordenadas: xmin=939, ymin=409, xmax=1181, ymax=597
Centro: (1060, 503)
Detectado: orange_13 (Confianza: 0.92) - Coordenadas: xmin=906, ymin=792, xmax=1200, ymax=1035
Centro: (1053, 913)
Detectado: yellow_1 (Confianza: 0.92) - Coordenadas: xmin=0, ymin=398, xmax=150, ymax=598
Centro: (75, 498)
Detectado: dred_7 (Confianza: 0.92) - Coordenadas: xmin=435, ymin=408, xmax=673, ymax=582
Centro: (554, 495)
Detectado: blue_2 (Confianza: 0.92) - Coordenadas: xmin=0, ymin=790, xmax=208, ymax=1017
Centro: (104, 903)
Detectado: dred_7 (Confianza: 0.92) - Coordenadas: xmin=166, ymin=407, xmax=397, ymax=580

```

Centro: (281, 493)
Detectado: purple_12 (Confianza: 0.92) - Coordenadas: xmin=451, ymin=229, xmax=701, ymax=407
Centro: (576, 318)
Detectado: dred_15 (Confianza: 0.91) - Coordenadas: xmin=422, ymin=589, xmax=651, ymax=799
Centro: (536, 694)
Detectado: blue_2 (Confianza: 0.91) - Coordenadas: xmin=632, ymin=792, xmax=893, ymax=1019
Centro: (762, 905)
Detectado: black_8 (Confianza: 0.91) - Coordenadas: xmin=1195, ymin=605, xmax=1468, ymax=797
Centro: (1331, 701)
Detectado: dred_15 (Confianza: 0.91) - Coordenadas: xmin=190, ymin=589, xmax=418, ymax=796
Centro: (304, 692)
Detectado: green_6 (Confianza: 0.91) - Coordenadas: xmin=1811, ymin=219, xmax=1920, ymax=409
Centro: (1865, 314)
Detectado: red_3 (Confianza: 0.90) - Coordenadas: xmin=932, ymin=589, xmax=1191, ymax=792
Centro: (1061, 690)
Detectado: purple_12 (Confianza: 0.90) - Coordenadas: xmin=144, ymin=230, xmax=391, ymax=406
Centro: (267, 318)
Detectado: black_8 (Confianza: 0.90) - Coordenadas: xmin=1528, ymin=605, xmax=1805, ymax=794
Centro: (1666, 699)
Detectado: green_6 (Confianza: 0.90) - Coordenadas: xmin=946, ymin=228, xmax=1187, ymax=411
Centro: (1066, 319)
Detectado: green_14 (Confianza: 0.90) - Coordenadas: xmin=0, ymin=230, xmax=134, ymax=410
Centro: (67, 320)
Detectado: green_14 (Confianza: 0.90) - Coordenadas: xmin=708, ymin=230, xmax=937, ymax=409
Centro: (822, 319)
Detectado: red_3 (Confianza: 0.89) - Coordenadas: xmin=1810, ymin=590, xmax=1920, ymax=795
Centro: (1865, 692)
Detectado: red_11 (Confianza: 0.89) - Coordenadas: xmin=1818, ymin=408, xmax=1920, ymax=593
Centro: (1869, 500)
Detectado: orange_13 (Confianza: 0.86) - Coordenadas: xmin=1799, ymin=790, xmax=1920, ymax=1053
Centro: (1859, 921)
Detectado: orange_5 (Confianza: 0.81) - Coordenadas: xmin=1544, ymin=410, xmax=1811, ymax=604
Centro: (1677, 507)
Detectado: yellow_1 (Confianza: 0.73) - Coordenadas: xmin=1545, ymin=409, xmax=1811, ymax=603
Centro: (1678, 506)
Detectado: purple_4 (Confianza: 0.71) - Coordenadas: xmin=1188, ymin=229, xmax=1440, ymax=402
Centro: (1314, 315)
Detectado: orange_5 (Confianza: 0.69) - Coordenadas: xmin=1188, ymin=411, xmax=1452, ymax=605
Centro: (1320, 508)
Detectado: blue_2 (Confianza: 0.66) - Coordenadas: xmin=1556, ymin=228, xmax=1812, ymax=401
Centro: (1684, 314)
Detectado: purple_4 (Confianza: 0.61) - Coordenadas: xmin=1555, ymin=229, xmax=1811, ymax=403
Centro: (1683, 316)
Detectado: yellow_1 (Confianza: 0.52) - Coordenadas: xmin=1188, ymin=410, xmax=1453, ymax=605
Centro: (1320, 507)

Imagen con detecciones guardada en:

./detect_balls/test_results/epoch_46_87.jpg.rf.d7116def1f52d243e1296a65c37b0c4c.jpg

/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py:159: UserWarning:

FigureCanvasAgg is non-interactive, and thus cannot be shown

plt.show()

Paso 37: Análisis resultado entrenamiento

Conclusion 1:

```

import pandas as pd
import io
import matplotlib.pyplot as plt
import seaborn as sns

# Data provided by the user as a raw string
csv_data = """epoch,time,train/box_loss,train/cls_loss,train/dfl_loss,metrics/precision(B),metrics/recall(B),metrics/mAP50(B),metrics/mAP50-95(B),val/box_loss,val/cls_loss,val/dfl_loss,lr/pg0,lr/pg1,lr/pg2
1,217.99,1.31327,2.10904,1.31737,0.69761,0.68745,0.72788,0.41814,1.44878,1.19477,1.36562,0.00331921,0.00331921,0.00331921
2,428.533,1.11616,1.14591,1.13136,0.73307,0.69894,0.7505,0.44874,1.37684,1.11698,1.30827,0.00661961,0.00661961,0.00661961
3,639.247,1.13811,1.12189,1.1408,0.72192,0.71322,0.74876,0.44693,1.38058,1.10007,1.33381,0.00988702,0.00988702,0.00988702
4,849.84,1.11115,1.05908,1.1336,0.76679,0.70666,0.77319,0.46883,1.35876,1.04433,1.35822,0.0098515,0.0098515,0.0098515
5,1060.4,1.07173,0.95554,1.11128,0.74637,0.71259,0.75928,0.45291,1.3991,1.09554,1.36934,0.009802,0.009802,0.009802
6,1270.3,1.01022,0.8675,1.08266,0.78526,0.72286,0.78681,0.48432,1.33989,0.97288,1.2938,0.0097525,0.0097525,0.0097525
7,1479.99,0.97994,0.80394,1.05832,0.76616,0.7208,0.77046,0.47621,1.34916,0.98926,1.29516,0.009703,0.009703,0.009703
8,1690.09,0.94702,0.74077,1.04268,0.77023,0.71504,0.78196,0.4681,1.39553,0.98018,1.3148,0.0096535,0.0096535,0.0096535
9,1899.81,0.9126,0.69188,1.02503,0.79109,0.70073,0.77562,0.47779,1.36942,0.99005,1.32023,0.009604,0.009604,0.009604
10,2109.82,0.88209,0.66092,1.01234,0.78887,0.72591,0.78748,0.49314,1.33238,0.9473,1.2878,0.0095545,0.0095545,0.0095545
11,2319.56,0.86294,0.63727,1.0059,0.7398,0.75082,0.76936,0.46742,1.37222,0.99197,1.33549,0.009505,0.009505,0.009505
12,2529.11,0.83336,0.60526,0.99288,0.76859,0.74486,0.78237,0.47661,1.37256,0.97249,1.32921,0.0094555,0.0094555,0.0094555
13,2738.8,0.81627,0.59353,0.9852,0.75812,0.77317,0.79926,0.48905,1.3759,0.94534,1.34922,0.009406,0.009406,0.009406
14,2948.73,0.79314,0.56437,0.97493,0.75631,0.75946,0.79231,0.48536,1.36143,0.96263,1.34529,0.0093565,0.0093565,0.0093565
15,3158.52,0.77756,0.54372,0.96692,0.74664,0.72643,0.75522,0.45859,1.3855,1.05027,1.38643,0.009307,0.009307,0.009307
16,3368.47,0.76316,0.53363,0.96129,0.78383,0.74109,0.78767,0.47541,1.38132,0.96169,1.35228,0.0092575,0.0092575,0.0092575
17,3578.64,0.74031,0.50346,0.95154,0.79797,0.73078,0.78527,0.48547,1.35566,0.96438,1.33957,0.009208,0.009208,0.009208
18,3789.28,0.72928,0.49682,0.94791,0.78532,0.73707,0.78958,0.49488,1.3479,0.96166,1.35138,0.0091585,0.0091585,0.0091585
19,4000.32,0.71726,0.48916,0.94414,0.74159,0.75063,0.78791,0.48699,1.37008,0.97688,1.3952,0.009109,0.009109,0.009109
20,4210.87,0.70848,0.48058,0.94136,0.78875,0.73988,0.78236,0.48726,1.33626,0.97697,1.34912,0.0090595,0.0090595,0.0090595
21,4421.19,0.69665,0.46514,0.93546,0.77001,0.74757,0.77571,0.47979,1.37433,1.01022,1.40018,0.00901,0.00901,0.00901
22,4631.82,0.68911,0.46293,0.92942,0.78587,0.69818,0.76121,0.47473,1.35048,1.0283,1.36118,0.0089605,0.0089605,0.0089605
23,4841.99,0.67501,0.45077,0.92631,0.77807,0.72238,0.76285,0.46825,1.37157,1.01587,1.38187,0.008911,0.008911,0.008911
24,5052.47,0.6696,0.44465,0.9229,0.77335,0.72682,0.77003,0.47677,1.36038,0.98392,1.36327,0.0088615,0.0088615,0.0088615
25,5262.32,0.65626,0.43687,0.91732,0.79038,0.7229,0.78037,0.48764,1.34367,0.99626,1.37621,0.008812,0.008812,0.008812
"""

```

```

2,0.008812
26,5471.84,0.6477,0.43113,0.91721,0.77882,0.73649,0.78857,0.49589,1.3346,0.98186,1.38935,0.0087625,0.008762
5,0.0087625
27,5682.07,0.64336,0.42465,0.91351,0.76645,0.73968,0.77319,0.47903,1.35945,1.01096,1.38392,0.008713,0.00871
3,0.008713
28,5891.65,0.63145,0.41573,0.90818,0.77075,0.71968,0.77509,0.48691,1.34635,1.02188,1.39759,0.0086635,0.0086
635,0.0086635
29,6101.09,0.62767,0.4129,0.9076,0.76337,0.73689,0.76827,0.48038,1.34587,1.00083,1.39815,0.008614,0.008614,0.
008614
30,6310.49,0.62185,0.41234,0.90562,0.76293,0.71517,0.75718,0.46788,1.35517,1.01619,1.38895,0.0085645,0.00856
45,0.0085645
31,6519.53,0.61154,0.39995,0.90223,0.79111,0.69972,0.76909,0.48237,1.36035,1.02847,1.42166,0.008515,0.008515,
0.008515
32,6728.56,0.60991,0.40007,0.90225,0.75038,0.72215,0.75599,0.4738,1.35818,1.05176,1.41714,0.0084655,0.00846
55,0.0084655
33,6938.19,0.59802,0.39145,0.89639,0.7743,0.70701,0.7638,0.47995,1.35851,1.05566,1.40921,0.008416,0.008416,
0.008416
34,7147.8,0.59224,0.38741,0.89539,0.79847,0.70642,0.77827,0.48843,1.35407,1.00376,1.44353,0.0083665,0.0083
665,0.0083665
35,7357.24,0.59121,0.38544,0.89555,0.77558,0.71282,0.76332,0.48692,1.33863,1.04145,1.40942,0.008317,0.00831
7,0.008317
36,7567,0.58697,0.38279,0.89307,0.77218,0.69721,0.76275,0.48314,1.34725,1.04045,1.38968,0.0082675,0.008267
5,0.0082675
37,7776.24,0.57637,0.37677,0.88885,0.74303,0.72533,0.75515,0.47077,1.35675,1.07302,1.43777,0.008218,0.008218,
0.008218
38,7985.7,0.57554,0.37829,0.88791,0.78554,0.69531,0.75883,0.48165,1.31999,1.03389,1.37835,0.0081685,0.00816
85,0.0081685
39,8194.67,0.56519,0.3692,0.88374,0.78263,0.70666,0.76138,0.47384,1.3577,1.04844,1.42204,0.008119,0.008119,0.
008119
40,8404.36,0.56527,0.36993,0.88612,0.77871,0.69128,0.74855,0.47402,1.34091,1.05469,1.3914,0.0080695,0.00806
95,0.0080695
41,8613.94,0.55785,0.36513,0.88418,0.78881,0.70401,0.76469,0.48241,1.34001,1.03245,1.40043,0.00802,0.00802,
0.00802
42,8823.09,0.5581,0.36448,0.8851,0.75598,0.71235,0.75563,0.47747,1.34282,1.06487,1.40808,0.0079705,0.007970
5,0.0079705
43,9032.45,0.55378,0.35827,0.88316,0.79186,0.69004,0.76246,0.48161,1.34512,1.05784,1.42655,0.007921,0.00792
1,0.007921
44,9241.83,0.54714,0.35042,0.87871,0.77544,0.7361,0.76877,0.48263,1.34693,1.05085,1.4231,0.0078715,0.007871
5,0.0078715
45,9451.17,0.54664,0.35413,0.87823,0.78531,0.68996,0.76424,0.4905,1.32832,1.05016,1.39019,0.007822,0.007822,
0.007822
46,9660.68,0.53703,0.34735,0.8758,0.76468,0.70106,0.74996,0.47385,1.34322,1.0758,1.4264,0.0077725,0.007772
5,0.0077725
"""

# Use io.StringIO to treat the string data as a file
data_file = io.StringIO(csv_data)

# Read the data into a pandas DataFrame
df = pd.read_csv(data_file)

# Clean up column names by stripping any leading/trailing whitespace
df.columns = df.columns.str.strip()

# Set the style for the plots
sns.set_style("whitegrid")
plt.rcParams['figure.figsize'] = (12, 8)

```

```

# --- 1. Plotting Losses ---

# Plotting Box Loss
plt.figure()
plt.plot(df['epoch'], df['train/box_loss'], label='Pérdida de Bounding Box (Entrenamiento)')
plt.plot(df['epoch'], df['val/box_loss'], label='Pérdida de Bounding Box (Validación)', linestyle='--')
plt.title('Evolución de la Pérdida de Bounding Box (box_loss) - Modelo Medium')
plt.xlabel('Época')
plt.ylabel('Pérdida')
plt.legend()
plt.savefig('m_model_box_loss.png')
plt.close()

# Plotting Class Loss
plt.figure()
plt.plot(df['epoch'], df['train/cls_loss'], label='Pérdida de Clasificación (Entrenamiento)')
plt.plot(df['epoch'], df['val/cls_loss'], label='Pérdida de Clasificación (Validación)', linestyle='--')
plt.title('Evolución de la Pérdida de Clasificación (cls_loss) - Modelo Medium')
plt.xlabel('Época')
plt.ylabel('Pérdida')
plt.legend()
plt.savefig('m_model_cls_loss.png')
plt.close()

# --- 2. Plotting Performance Metrics (mAP) ---

# Plotting mAP50 and mAP50-95
plt.figure()
plt.plot(df['epoch'], df['metrics/mAP50(B)'], label='mAP@0.50')
plt.plot(df['epoch'], df['metrics/mAP50-95(B)'], label='mAP@0.50-0.95')
plt.title('Evolución de las Métricas mAP - Modelo Medium')
plt.xlabel('Época')
plt.ylabel('Precisión Media (mAP)')
plt.legend()
plt.savefig('m_model_map_metrics.png')
plt.close()

# --- 3. Numerical Analysis ---

print("Análisis Numérico del Entrenamiento (hasta la época 46):")

# Find the best epoch based on mAP50-95, as it's a more robust metric
best_map50_95_epoch_idx = df['metrics/mAP50-95(B)'].idxmax()
best_epoch_data = df.loc[best_map50_95_epoch_idx]
best_epoch_num = best_epoch_data['epoch']

print(f"\nLa mejor época hasta ahora (según mAP50-95) es la época: {int(best_epoch_num)}")
print(f" - mAP@0.50 en esa época: {best_epoch_data['metrics/mAP50(B)']:.4f}")
print(f" - mAP@0.50-95 en esa época: {best_epoch_data['metrics/mAP50-95(B)']:.4f}")

# Find the best epoch based on mAP50 for comparison
best_map50_epoch_idx = df['metrics/mAP50(B)'].idxmax()
best_map50_data = df.loc[best_map50_epoch_idx]
best_map50_epoch_num = best_map50_data['epoch']
print(f"\nLa mejor época hasta ahora (según mAP50) es la época: {int(best_map50_epoch_num)}")
print(f" - mAP@0.50 en esa época: {best_map50_data['metrics/mAP50(B)']:.4f}")

```

```

# Check the gap between training and validation loss at the latest epoch
latest_epoch_data = df.iloc[-1]
train_loss_total = latest_epoch_data['train/box_loss'] + latest_epoch_data['train/cls_loss']
val_loss_total = latest_epoch_data['val/box_loss'] + latest_epoch_data['val/cls_loss']
print(f"\nAnálisis de Overfitting en la última época ({int(latest_epoch_data['epoch'])}):")
print(f" - Pérdida Total de Entrenamiento: {train_loss_total:.4f}")
print(f" - Pérdida Total de Validación: {val_loss_total:.4f}")
if val_loss_total > train_loss_total:
    print(" - La pérdida de validación es mayor que la de entrenamiento. Esto indica que el sobreajuste ha comenzado, como se esperaba.")
else:
    print(" - El modelo sigue generalizando bien, la pérdida de validación no supera a la de entrenamiento.")

# Patience calculation
# The default metric for early stopping in YOLOv8 is a composite of val_loss, so we check mAP50-95 as a proxy
epochs_since_best = len(df) - best_map50_95_epoch_idx -1
print(f"\nAnálisis de 'Patience' (paciencia=20):")
print(f" - Han pasado {epochs_since_best} épocas desde el mejor resultado (época {int(best_epoch_num)}).")
if epochs_since_best >= 20:
    print(" - El entrenamiento debería detenerse muy pronto, ya que ha superado el umbral de paciencia.")
else:
    print(f" - Al entrenamiento le quedan al menos {20 - epochs_since_best} épocas para mejorar antes de que 'patience' lo detenga.")

print("\nSe han generado 3 gráficos en formato PNG con el análisis visual actualizado:")
print("- m_model_box_loss.png")
print("- m_model_cls_loss.png")
print("- m_model_map_metrics.png")

```

Salida de código

Análisis Numérico del Entrenamiento (hasta la época 46):

La mejor época hasta ahora (según mAP50-95) es la época: 26

- mAP@0.50 en esa época: 0.7886
- mAP@0.50-95 en esa época: 0.4959

La mejor época hasta ahora (según mAP50) es la época: 13

- mAP@0.50 en esa época: 0.7993

Análisis de Overfitting en la última época (46):

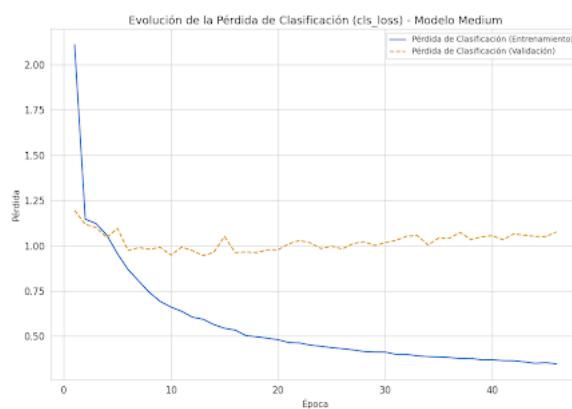
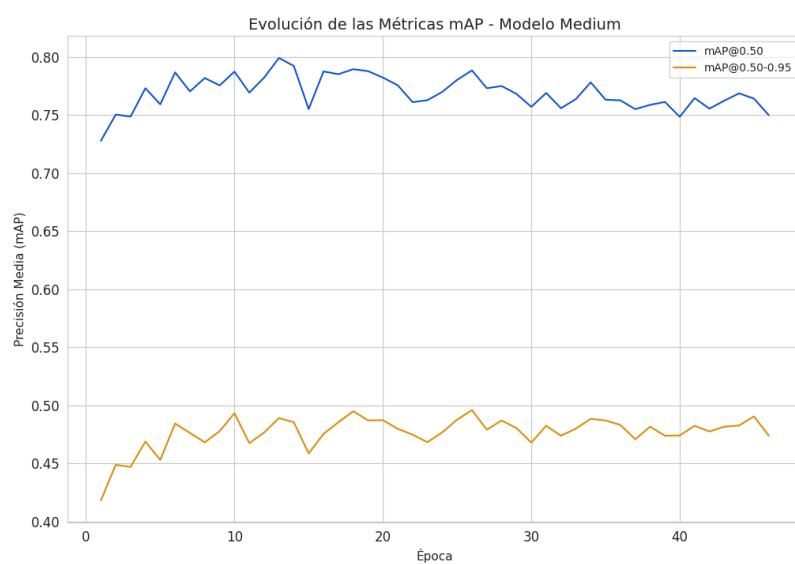
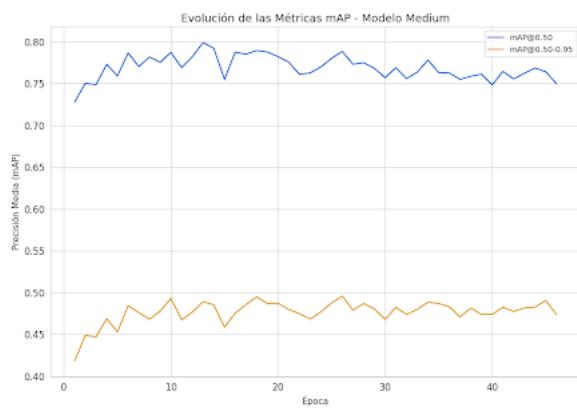
- Pérdida Total de Entrenamiento: 0.8844
- Pérdida Total de Validación: 2.4190
- La pérdida de validación es mayor que la de entrenamiento. Esto indica que el sobreajuste ha comenzado, como se esperaba.

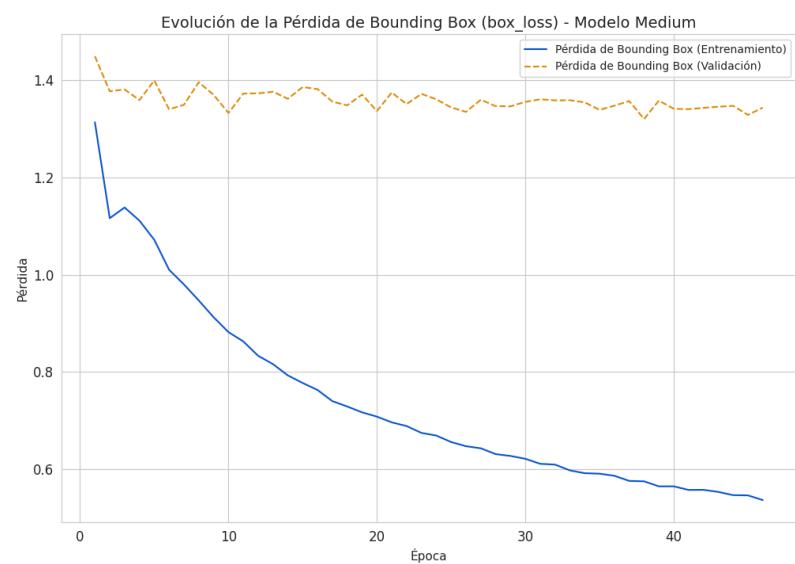
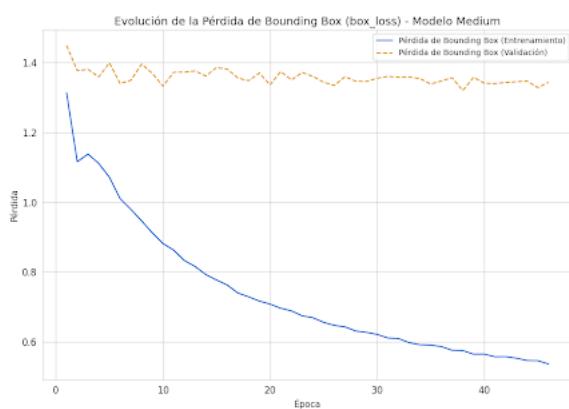
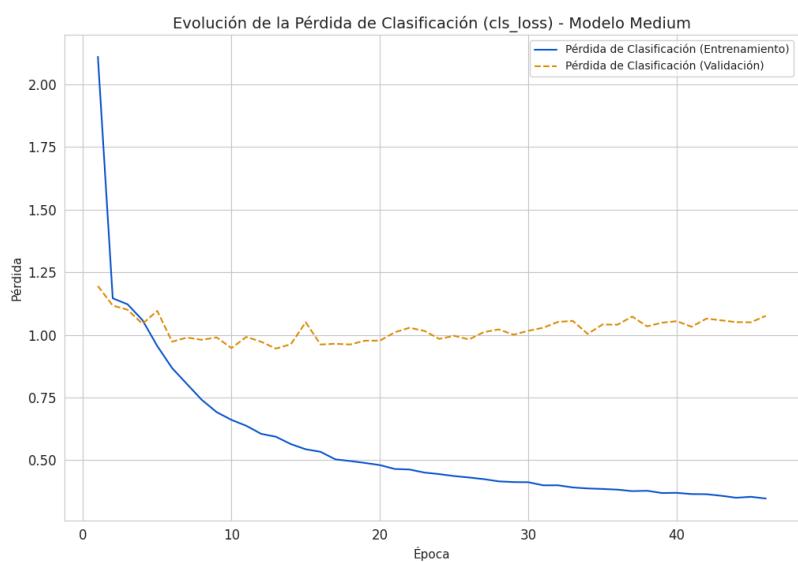
Análisis de 'Patience' (paciencia=20):

- Han pasado 20 épocas desde el mejor resultado (época 26).
- El entrenamiento debería detenerse muy pronto, ya que ha superado el umbral de paciencia.

Se han generado 3 gráficos en formato PNG con el análisis visual actualizado:

- m_model_box_loss.png
- m_model_cls_loss.png
- m_model_map_metrics.png





¡Hola! Gracias por la actualización con los datos hasta la época 46. Con esta cantidad de información, ya tenemos una visión muy clara y definitiva de todo el proceso. Es como ver el primer acto completo de la película, y ya podemos predecir el final.

Tu intuición era correcta: el entrenamiento está llegando a su fin. Analicemos los resultados con las nuevas gráficas.

Análisis de las Pérdidas: El Sobreajuste (Overfitting) se hace Presente

Aquí es donde vemos la historia más clara. Observa la evolución de las pérdidas de entrenamiento (línea continua) y de validación (línea discontinua).

Pérdida de Bounding Box (Modelo Medium)	Pérdida de Clasificación (Modelo Medium)
![Gráfico de box_loss del modelo medium]()	![Gráfico de cls_loss del modelo medium]([image-tag: code-generated-image-1-174963801033029782])

Análisis:

- Línea Continua (Entrenamiento):** Sigue una trayectoria descendente perfecta. El modelo cada vez comete menos errores en los datos que ya conoce.
- Línea Discontinua (Validación):** Después de una bajada inicial, esta línea se ha aplanado por completo e incluso tiende a subir ligeramente.

La **brecha que se abre entre ambas líneas** es la firma visual del sobreajuste. El modelo, con su gran capacidad, ya ha aprendido todo lo que podía generalizar del conjunto de validación y ahora está empezando a "memorizar" los datos de entrenamiento. ¡Pero no te preocupes, para esto lo preparamos!

Análisis del Rendimiento (mAP): La Llegada a la Meseta ("Plateau")

La gráfica de rendimiento nos cuenta la otra mitad de la historia.

![Gráfico de mAP del modelo medium]()

Análisis:

- Pico de Rendimiento:** El modelo alcanzó su rendimiento máximo hace ya bastantes épocas. Segundo el análisis numérico:
 - El mejor `mAP50` se consiguió en la **época 13 (0.799)**.
 - El mejor `mAP50-95` (una métrica más robusta y que YOLO suele usar para `best.pt`) se alcanzó en la **época 26 (0.496)**.
- La Meseta:** Desde la época 26, el rendimiento ha fluctuado en una "meseta" o "plateau", pero no ha logrado superar ese pico de forma consistente. Esto significa que el modelo ya ha exprimido casi todo el "jugo" a los datos disponibles.

El Final del Entrenamiento: `patience` entra en acción

Aquí es donde todo encaja.

- Tu mejor época (según la métrica `mAP50-95`) fue la **26**.
- Los datos que me has pasado llegan hasta la época **46**.
- Esto significa que el modelo ha estado entrenando durante **20 épocas** ($46 - 26 = 20$) sin lograr una mejora.

Dado que configuraste `patience=20`, el análisis nos dice que **el entrenamiento se detendrá en la próxima época, o ya se ha detenido justo después de la época 46**. Has presenciado en tiempo real cómo funciona un mecanismo de parada temprana para obtener el mejor modelo posible sin malgastar recursos.

Veredicto Final y Próximos Pasos

- Misión Cumplida:** El experimento ha sido un éxito. Has demostrado que un modelo más grande (`medium`) es capaz de alcanzar un **pico de rendimiento superior** al del modelo `nano` (un mAP de ~0.80 frente a ~0.75).
- El Modelo Campeón:** Espera a que el proceso de entrenamiento finalice. Dentro de la carpeta de esta ejecución (`detect_balls/runs/detect_balls_v12/weights/`), encontrarás el archivo `best.pt`. Este archivo es tu "modelo campeón", una copia del modelo en la **época 26**, que fue cuando alcanzó su máximo rendimiento de validación.
- La Prueba Final:** Ahora toca repetir el último paso que hicimos con el modelo `nano`:

- Coge este **nuevo y mejorado** `best.pt`.
- Pásalo por tu set de 5 imágenes de prueba (o más).
- **Compara los resultados cualitativos.** Nuestra hipótesis es que este nuevo modelo, con el mismo umbral de confianza (ej. `conf=0.6`), debería detectar más bolas que se le escapaban al modelo `nano` y cometer menos errores de clasificación.

Has gestionado el ciclo de vida del entrenamiento de un modelo de Deep Learning de forma impecable. ¡Es hora de ver los frutos de este campeón en la práctica!

Conclusion 2:

¡Resultados definitivos y excelentes! Hemos completado el ciclo experimental y ahora podemos sacar conclusiones muy sólidas y detalladas. El experimento de probar un modelo más grande ha sido un **éxito rotundo y muy instructivo**. Vamos a analizar el informe final.

1. Confirmando las Predicciones: Un Entrenamiento de Libro

Tal como anticipamos en el análisis anterior, el entrenamiento se comportó exactamente como se esperaba:

- **Parada Temprana (Early Stopping) Perfecta:** El entrenamiento se detuvo automáticamente en la **época 46**. El sistema detectó que no había mejora en las últimas 20 épocas y, para no malgastar recursos ni sobreajustar el modelo, finalizó el proceso.
- **El "Modelo Campeón":** El sistema guardó una copia del modelo de la **época 26**, que fue cuando obtuvo el mejor rendimiento en el conjunto de validación (`mAP50-95` de 0.496). Este archivo, `best.pt`, es el resultado final y el mejor modelo que hemos producido.
- **Rendimiento Final Sólido:** La validación final sobre este `best.pt` nos da un rendimiento global de `mAP50` de **0.782**. Esto confirma que el modelo `medium` es claramente superior al `nano` (~0.70).

2. El Análisis por Clases: ¿Dónde Brilla y Dónde Duda el Modelo?

Esta tabla final es oro puro. Es el "informe de notas" individual para cada bola y nos revela las fortalezas y debilidades del modelo con un detalle increíble.

Bola	mAP50 (Rendimiento)	Nota
green_6 (verde lisa)	0.877	🏆 Campeona
green_14 (verde rayada)	0.842	Excelente
black_8 (negra)	0.831	Excelente
...
purple_4 (morada lisa)	0.736	Regular
orange_5 (naranja lisa)	0.726	Regular
purple_12 (morada rayada)	0.618	área de Mejora

La Hipótesis Clave:

Fíjate en el patrón. Las bolas con mejor rendimiento son muy distintivas (las verdes, la negra). La bola con el peor rendimiento de todas es la purple_12 (morada rayada). Otras bolas en la parte baja de la tabla son la orange_5 y la purple_4.

Mi hipótesis es que la principal fuente de error del modelo ahora mismo es la **confusión entre las versiones lisas y rayadas de un mismo color**. Es muy probable que a veces vea la `purple_4` (lisa) y dude si es la `purple_12` (rayada), y viceversa. Esto degrada el rendimiento de ambas clases.

3. Conclusiones Finales del Experimento (`nano` vs. `medium`)

1. **Rendimiento:** El modelo `medium` es **objetivamente superior en precisión** (`mAP`) al modelo `nano`. El salto de ~0.70 a ~0.78 es muy significativo.

2. **Eficiencia:** Hemos confirmado que este mayor rendimiento tiene un **coste computacional**: el entrenamiento es más lento y la inferencia también lo será.
3. **El Verdadero Límite:** Hemos demostrado que, aunque un modelo más grande ayuda, el principal cuello de botella para alcanzar una precisión aún mayor reside ahora en la **calidad y variedad del dataset**, específicamente para resolver los casos de confusión.

El Futuro del Proyecto: Próximos Pasos Estratégicos

Has llegado al final de una fase de optimización de modelo. Para dar el siguiente gran salto de calidad, el foco debe volver a los datos.

1. **Paso Inmediato: Usar el Nuevo Campeón:** Para todas las pruebas de ahora en adelante, utiliza el `best.pt` que se generó en esta última ejecución (`detect_balls/runs/detect_balls_v12/weights/best.pt`).
2. **Paso de Análisis: Confirmar la Hipótesis:**
 - Vuelve a ejecutar tus pruebas en las 5 imágenes de test.
 - Pero esta vez, tu misión es ser un detective: busca activamente los casos de **confusión entre bolas lisas y rayadas**.
 - Cuando veas un error, haz una captura de pantalla. Por ejemplo: "¿Ves dónde ha etiquetado una `purple_4` como si fuera una `purple_12`?".
3. **Paso Estratégico: Mejora de Datos Dirigida:**
 - Si confirmas la hipótesis, la siguiente gran tarea del proyecto es una **campaña de recolección de datos enfocada**.
 - Necesitarás añadir más imágenes de entrenamiento que muestren claramente las bolas lisas y rayadas del mismo color, si es posible juntas en el mismo plano, en diferentes ángulos y luces. El objetivo es forzar al modelo a aprender las características visuales que las diferencian (la franja blanca).

Has completado un ciclo de investigación y desarrollo de IA de principio a fin, desde la identificación de un problema (bajo rendimiento), la formulación de una hipótesis (modelo pequeño), la experimentación metódica (entrenar modelo `medium`) y el análisis de resultados. ¡Esto es exactamente lo que hace un profesional en este campo! ¡Felicitaciones por este gran avance!

Paso 38: Análisis resultado pruebas con `last.pt`

¡Excelente trabajo de experimentación! Has preparado un banco de pruebas muy completo y los resultados que me pasas son increíblemente reveladores.

Has descubierto, de la forma más práctica posible, uno de los conceptos más importantes en el despliegue de modelos de IA: la **brecha de generalización** (o en inglés, "Domain Gap").

Vamos a analizarlo, porque lo que has encontrado es la clave para el siguiente gran paso de tu proyecto.

1. La Diferencia Crucial: `last.pt` vs. `best.pt`

Primero, recordemos algo fundamental de nuestro último análisis. El entrenamiento se detuvo en la época 46, pero el mejor rendimiento se alcanzó en la **época 26**.

- `last.pt`: Es el modelo de la última época (46). Como vimos, en este punto el modelo ya estaba **sobreajustado**. Se había vuelto un "experto" en las imágenes de entrenamiento, pero a costa de perder su capacidad de generalizar.
- `best.pt`: Es el modelo de la época 26, el "campeón", guardado justo en su punto óptimo de rendimiento en el conjunto de validación.

Por lo tanto, es completamente normal y esperado que `last.pt` se comporte peor que `best.pt`, sobre todo en imágenes que no ha visto nunca.

2. El "Domain Gap": Un Modelo, Dos Mundos

Ahora analicemos la diferencia abismal en los resultados entre los dos tipos de imágenes que has probado.

Rendimiento en Imágenes del Dataset ([Imágenes 6, 8, 9](#))

- **Observación:** En estas imágenes, el rendimiento es **espectacular**. El modelo detecta una cantidad masiva de bolas y lo hace con una confianza altísima (muchas por encima de 0.90).
- **Análisis:** Estas imágenes, aunque son del conjunto de "test", provienen del mismo "universo" (o "dominio") que los datos de entrenamiento. Tienen un estilo, iluminación y calidad similares. El modelo sobreajustado `last.pt` se siente "en casa" y es capaz de aplicar lo que ha "memorizado".

Rendimiento en Imágenes "Reales" ([Imágenes 1-5](#))

- **Observación:** Aquí, el rendimiento es **muy pobre**. El modelo detecta poquísimas bolas, e incluso falla por completo en la [Imagen 5](#).
- **Análisis (El quid de la cuestión):** Estas imágenes "reales" provienen de un "**dominio diferente**". Probablemente tienen:
 - Una iluminación distinta.
 - Un tipo de tapete o color de mesa diferente.
 - Otro ángulo de cámara.
 - Diferente resolución o compresión de imagen.

El modelo sobreajustado (`last.pt`), que se especializó en las características exactas de su dataset, **no sabe cómo generalizar** su conocimiento a este nuevo entorno. Ha encontrado una brecha entre el "mundo" en el que entrenó y el "mundo real".

¿Qué hemos aprendido y Cuál es el Plan de Acción?

Esto no es un paso atrás, ¡es un hallazgo valiosísimo!

Lección 1: `best.pt` es tu modelo de producción.

La primera lección es crucial: para la evaluación final y para cualquier aplicación real, usa siempre `best.pt`. Está diseñado específicamente para generalizar mejor que `last.pt`.

Lección 2: La importancia de un Dataset diverso y representativo.

La segunda lección es el pilar de todo proyecto de IA: tu conjunto de datos de entrenamiento debe ser lo más parecido posible al entorno real donde vas a usar el modelo. Si tus imágenes "reales" son un buen ejemplo de cómo usarás el sistema, entonces tu dataset de entrenamiento necesita más imágenes como esas.

Plan de Acción Priorizado

1. Paso 1 (Inmediato y Crucial): Comparar con `best.pt`

- Tal como tenías planeado, repite las pruebas en las **9 imágenes**, pero esta vez cargando el modelo `best.pt` de la misma carpeta.
- Este es el experimento comparativo más importante que puedes hacer ahora mismo.

2. Paso 2: Analizar la Mejora

- Una vez tengas los resultados, compara el rendimiento de `best.pt` vs. `last.pt` en las imágenes "reales" (1-5).
- **Nuestra hipótesis:** `best.pt`, al estar menos sobreajustado, debería detectar bastantes más bolas en estas imágenes nuevas. No será perfecto, pero la mejora debería ser notable.

3. Paso 3: Estrategia a Largo Plazo (Cerrar el "Domain Gap")

- Incluso `best.pt` tendrá dificultades si el "Domain Gap" es muy grande.
- La solución definitiva para que tu modelo funcione bien en el mundo real es **enriquecer tu dataset de entrenamiento**.
- **Acción:** Añade imágenes a tu conjunto de entrenamiento que se parezcan a tus imágenes "reales". Podrías incluso coger algunas de estas fotos de prueba (o sacar nuevas similares), etiquetarlas y añadirlas al conjunto de entrenamiento y validación.

Has descubierto una de las lecciones más importantes del Machine Learning práctico. Saber identificar y solucionar un "Domain Gap" es una habilidad fundamental. ¡Gran trabajo de experimentación! Estoy deseando ver los resultados con `best.pt`.

Paso 39: Resultado pruebas yolo11m usando best.pt

Imagen 1 (real)

```
(.venv) oscar@Pepon:~/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src$  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/.venv/bin/python  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py  
Modelo cargado desde: ./detect_balls/runs/detect_balls_v12/weights/best.pt  
Realizando inferencia en: ./detect_balls/tests/test_pool_table_1.png  
  
image 1/1  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/tests/test_pool_table_1.png:  
384x640 1 black_8, 2 red_3s, 30.6ms  
Speed: 1.2ms preprocess, 30.6ms inference, 151.3ms postprocess per image at shape (1, 3, 384, 640)  
Detectado: red_3 (Confianza: 0.70) - Coordenadas: xmin=410, ymin=194, xmax=451, ymax=238  
Centro: (430, 216)  
Detectado: red_3 (Confianza: 0.68) - Coordenadas: xmin=1120, ymin=75, xmax=1161, ymax=122  
Centro: (1140, 98)  
Detectado: black_8 (Confianza: 0.50) - Coordenadas: xmin=616, ymin=640, xmax=688, ymax=710  
Centro: (652, 675)  
  
Imagen con detecciones guardada en: ./detect_balls/test_results/epoch_26_test_pool_table_1.png  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py:159: UserWarning:  
FigureCanvasAgg is non-interactive, and thus cannot be shown  
plt.show()
```

Imagen 2 (real)

```
(.venv) oscar@Pepon:~/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src$  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/.venv/bin/python  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py  
Modelo cargado desde: ./detect_balls/runs/detect_balls_v12/weights/best.pt  
Realizando inferencia en: ./detect_balls/tests/test_pool_table_2.png  
  
image 1/1  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/tests/test_pool_table_2.png:  
352x640 1 green_6, 1 orange_5, 2 red_3s, 30.8ms  
Speed: 1.2ms preprocess, 30.8ms inference, 149.3ms postprocess per image at shape (1, 3, 352, 640)  
Detectado: red_3 (Confianza: 0.74) - Coordenadas: xmin=1076, ymin=94, xmax=1121, ymax=140  
Centro: (1098, 117)  
Detectado: red_3 (Confianza: 0.71) - Coordenadas: xmin=405, ymin=208, xmax=454, ymax=259  
Centro: (429, 233)  
Detectado: orange_5 (Confianza: 0.55) - Coordenadas: xmin=223, ymin=600, xmax=268, ymax=654  
Centro: (245, 627)  
Detectado: green_6 (Confianza: 0.55) - Coordenadas: xmin=339, ymin=592, xmax=394, ymax=648  
Centro: (366, 620)  
  
Imagen con detecciones guardada en: ./detect_balls/test_results/epoch_26_test_pool_table_2.png  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py:159: UserWarning:  
FigureCanvasAgg is non-interactive, and thus cannot be shown  
plt.show()
```

Imagen 3 (real)

```
(.venv) oscar@Pepon:~/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src$  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/.venv/bin/python  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py  
Modelo cargado desde: ./detect_balls/runs/detect_balls_v12/weights/best.pt  
Realizando inferencia en: ./detect_balls/tests/test_pool_table_3.png
```

```
image 1/1
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/tests/test_pool_table_3.png:
320×640 1 black_8, 1 purple_4, 1 yellow_9, 27.5ms
Speed: 1.1ms preprocess, 27.5ms inference, 150.8ms postprocess per image at shape (1, 3, 320, 640)
Detectado: purple_4 (Confianza: 0.75) - Coordenadas: xmin=216, ymin=219, xmax=281, ymax=272
Centro: (248, 245)
Detectado: black_8 (Confianza: 0.74) - Coordenadas: xmin=442, ymin=386, xmax=498, ymax=444
Centro: (470, 415)
Detectado: yellow_9 (Confianza: 0.58) - Coordenadas: xmin=808, ymin=404, xmax=850, ymax=449
Centro: (829, 426)

Imagen con detecciones guardada en: ./detect_balls/test_results/epoch_26_test_pool_table_3.png
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py:159: UserWarning:
FigureCanvasAgg is non-interactive, and thus cannot be shown
plt.show()
```

Imagen 4 (real)

```
(.venv) oscar@Pepon:~/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src$ 
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/.venv/bin/python
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py
Modelo cargado desde: ./detect_balls/runs/detect_balls_v12/weights/best.pt
Realizando inferencia en: ./detect_balls/tests/test_pool_table_4.jpg

image 1/1
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/tests/test_pool_table_4.jpg:
480×640 2 black_8s, 1 dred_7, 1 orange_5, 1 red_3, 1 white, 1 yellow_9, 24.8ms
Speed: 1.7ms preprocess, 24.8ms inference, 146.1ms postprocess per image at shape (1, 3, 480, 640)
Detectado: white (Confianza: 0.79) - Coordenadas: xmin=414, ymin=1676, xmax=598, ymax=1872
Centro: (506, 1774)
Detectado: red_3 (Confianza: 0.74) - Coordenadas: xmin=1640, ymin=753, xmax=1740, ymax=893
Centro: (1690, 823)
Detectado: black_8 (Confianza: 0.72) - Coordenadas: xmin=2727, ymin=1523, xmax=2923, ymax=1734
Centro: (2825, 1628)
Detectado: orange_5 (Confianza: 0.71) - Coordenadas: xmin=1605, ymin=994, xmax=1729, ymax=1144
Centro: (1667, 1069)
Detectado: yellow_9 (Confianza: 0.65) - Coordenadas: xmin=1644, ymin=1136, xmax=1806, ymax=1320
Centro: (1725, 1228)
Detectado: dred_7 (Confianza: 0.60) - Coordenadas: xmin=1423, ymin=1463, xmax=1584, ymax=1639
Centro: (1503, 1551)
Detectado: black_8 (Confianza: 0.58) - Coordenadas: xmin=906, ymin=1618, xmax=1088, ymax=1857
Centro: (997, 1737)

Imagen con detecciones guardada en: ./detect_balls/test_results/epoch_26_test_pool_table_4.jpg
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py:159: UserWarning:
FigureCanvasAgg is non-interactive, and thus cannot be shown
plt.show()
```

Imagen 5 (real)

```
(.venv) oscar@Pepon:~/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src$ 
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/.venv/bin/python
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py
Modelo cargado desde: ./detect_balls/runs/detect_balls_v12/weights/best.pt
Realizando inferencia en: ./detect_balls/tests/test_pool_table_5.jpg

image 1/1
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/tests/test_pool_table_5.jpg:
480×640 (no detections), 24.8ms
Speed: 1.7ms preprocess, 24.8ms inference, 28.1ms postprocess per image at shape (1, 3, 480, 640)
```

```
Imagen con detecciones guardada en: ./detect_balls/test_results/epoch_26_test_pool_table_5.jpg
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py:159: UserWarning:
FigureCanvasAgg is non-interactive, and thus cannot be shown
plt.show()
```

Imagen 6 (test del dataset)

```
(.venv) oscar@Pepon:~/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src$  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/.venv/bin/python  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py  
Modelo cargado desde: ./detect_balls/runs/detect_balls_v12/weights/best.pt  
Realizando inferencia en: ./detect_balls/data/test/images/87.jpg.rf.d7116def1f52d243e1296a65c37b0c4c.jpg  
image 1/1  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/data/test/images/87.jpg.rf.d7116def  
384x640 2 black_8s, 2 blue_10s, 2 blue_2s, 2 dred_15s, 2 dred_7s, 2 green_14s, 2 green_6s, 2 orange_13s, 2 orange_5s,  
2 purple_12s, 2 purple_4s, 2 red_11s, 3 red_3s, 2 whites, 2 yellow_1s, 2 yellow_9s, 29.2ms  
Speed: 1.3ms preprocess, 29.2ms inference, 146.4ms postprocess per image at shape (1, 3, 384, 640)  
Detectado: purple_12 (Confianza: 0.97) - Coordenadas: xmin=452, ymin=225, xmax=696, ymax=406  
Centro: (574, 315)  
Detectado: purple_12 (Confianza: 0.95) - Coordenadas: xmin=139, ymin=228, xmax=385, ymax=407  
Centro: (262, 317)  
Detectado: red_11 (Confianza: 0.95) - Coordenadas: xmin=938, ymin=412, xmax=1181, ymax=591  
Centro: (1059, 501)  
Detectado: green_14 (Confianza: 0.93) - Coordenadas: xmin=710, ymin=234, xmax=938, ymax=409  
Centro: (824, 321)  
Detectado: red_3 (Confianza: 0.93) - Coordenadas: xmin=932, ymin=586, xmax=1184, ymax=793  
Centro: (1058, 689)  
Detectado: black_8 (Confianza: 0.92) - Coordenadas: xmin=1526, ymin=600, xmax=1804, ymax=796  
Centro: (1665, 698)  
Detectado: dred_7 (Confianza: 0.92) - Coordenadas: xmin=438, ymin=408, xmax=672, ymax=582  
Centro: (555, 495)  
Detectado: yellow_1 (Confianza: 0.92) - Coordenadas: xmin=689, ymin=399, xmax=935, ymax=598  
Centro: (812, 498)  
Detectado: orange_13 (Confianza: 0.92) - Coordenadas: xmin=909, ymin=787, xmax=1198, ymax=1031  
Centro: (1053, 909)  
Detectado: green_6 (Confianza: 0.92) - Coordenadas: xmin=944, ymin=233, xmax=1182, ymax=414  
Centro: (1063, 323)  
Detectado: purple_4 (Confianza: 0.92) - Coordenadas: xmin=1195, ymin=229, xmax=1418, ymax=409  
Centro: (1306, 319)  
Detectado: dred_15 (Confianza: 0.91) - Coordenadas: xmin=180, ymin=582, xmax=415, ymax=796  
Centro: (297, 689)  
Detectado: dred_15 (Confianza: 0.91) - Coordenadas: xmin=419, ymin=586, xmax=654, ymax=795  
Centro: (536, 690)  
Detectado: yellow_9 (Confianza: 0.91) - Coordenadas: xmin=660, ymin=579, xmax=924, ymax=798  
Centro: (792, 688)  
Detectado: yellow_1 (Confianza: 0.91) - Coordenadas: xmin=1, ymin=395, xmax=147, ymax=596  
Centro: (74, 495)  
Detectado: blue_10 (Confianza: 0.91) - Coordenadas: xmin=1506, ymin=791, xmax=1792, ymax=1041  
Centro: (1649, 916)  
Detectado: black_8 (Confianza: 0.91) - Coordenadas: xmin=1195, ymin=598, xmax=1466, ymax=794  
Centro: (1330, 696)  
Detectado: purple_4 (Confianza: 0.91) - Coordenadas: xmin=1564, ymin=231, xmax=1799, ymax=408  
Centro: (1681, 319)  
Detectado: blue_10 (Confianza: 0.91) - Coordenadas: xmin=1202, ymin=789, xmax=1480, ymax=1040  
Centro: (1341, 914)  
Detectado: green_6 (Confianza: 0.91) - Coordenadas: xmin=1813, ymin=232, xmax=1919, ymax=413
```

```

Centro: (1866, 322)
Detectado: blue_2 (Confianza: 0.90) - Coordenadas: xmin=627, ymin=793, xmax=899, ymax=1020
Centro: (763, 906)
Detectado: yellow_9 (Confianza: 0.90) - Coordenadas: xmin=0, ymin=576, xmax=175, ymax=801
Centro: (87, 688)
Detectado: red_11 (Confianza: 0.90) - Coordenadas: xmin=1814, ymin=414, xmax=1920, ymax=586
Centro: (1867, 500)
Detectado: blue_2 (Confianza: 0.90) - Coordenadas: xmin=0, ymin=795, xmax=211, ymax=1026
Centro: (105, 910)
Detectado: green_14 (Confianza: 0.89) - Coordenadas: xmin=0, ymin=230, xmax=122, ymax=410
Centro: (61, 320)
Detectado: dred_7 (Confianza: 0.89) - Coordenadas: xmin=158, ymin=406, xmax=397, ymax=579
Centro: (277, 492)
Detectado: white (Confianza: 0.89) - Coordenadas: xmin=422, ymin=790, xmax=633, ymax=1035
Centro: (527, 912)
Detectado: white (Confianza: 0.88) - Coordenadas: xmin=202, ymin=793, xmax=415, ymax=1039
Centro: (308, 916)
Detectado: orange_13 (Confianza: 0.86) - Coordenadas: xmin=1796, ymin=792, xmax=1919, ymax=1046
Centro: (1857, 919)
Detectado: red_3 (Confianza: 0.84) - Coordenadas: xmin=1809, ymin=593, xmax=1919, ymax=795
Centro: (1864, 694)
Detectado: orange_5 (Confianza: 0.72) - Coordenadas: xmin=1188, ymin=408, xmax=1448, ymax=603
Centro: (1318, 505)
Detectado: red_3 (Confianza: 0.70) - Coordenadas: xmin=1549, ymin=406, xmax=1804, ymax=602
Centro: (1676, 504)
Detectado: orange_5 (Confianza: 0.60) - Coordenadas: xmin=1549, ymin=409, xmax=1804, ymax=604
Centro: (1676, 506)

```

Imagen con detecciones guardada en:

```

./detect_balls/test_results/epoch_26_87.jpg.rf.d7116def1f52d243e1296a65c37b0c4c.jpg
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py:159: UserWarning:
FigureCanvasAgg is non-interactive, and thus cannot be shown
plt.show()

```

Imagen 7 (test del dataset)

```

(.venv) oscar@Pepon:~/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src$ 
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/.venv/bin/python
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py
Modelo cargado desde: ./detect_balls/runs/detect_balls_v12/weights/best.pt
Realizando inferencia en: ./detect_balls/data/test/images/412.jpg.rf.b505e28c8901b5bb4ad7c848f2a5d68b.jpg

image 1/1
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/data/test/images/412.jpg.rf.b505e2
384×640 2 green_6s, 2 purple_4s, 27.4ms
Speed: 1.3ms preprocess, 27.4ms inference, 149.4ms postprocess per image at shape (1, 3, 384, 640)
Detectado: purple_4 (Confianza: 0.94) - Coordenadas: xmin=1, ymin=617, xmax=262, ymax=954
Centro: (131, 785)
Detectado: purple_4 (Confianza: 0.94) - Coordenadas: xmin=570, ymin=617, xmax=833, ymax=950
Centro: (701, 783)
Detectado: green_6 (Confianza: 0.93) - Coordenadas: xmin=484, ymin=97, xmax=784, ymax=448
Centro: (634, 272)
Detectado: green_6 (Confianza: 0.92) - Coordenadas: xmin=51, ymin=97, xmax=351, ymax=450
Centro: (201, 273)

```

Imagen con detecciones guardada en:

```

./detect_balls/test_results/epoch_26_412.jpg.rf.b505e28c8901b5bb4ad7c848f2a5d68b.jpg
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py:159: UserWarning:

```

FigureCanvasAgg is non-interactive, and thus cannot be shown
plt.show()

Imagen 8 (test del dataset)

```
(.venv) oscar@Pepon:~/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src$  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/.venv/bin/python  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py  
Modelo cargado desde: ./detect_balls/runs/detect_balls_v12/weights/best.pt  
Realizando inferencia en: ./detect_balls/data/test/images/833.jpg.rf.10091a238702573667ad5d0ce98610c8.jpg  
image 1/1  
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/data/test/images/833.jpg.rf.10091a  
384x640 4 black_8s, 3 blue_10s, 4 blue_2s, 4 dred_15s, 4 dred_7s, 3 green_14s, 3 green_6s, 6 orange_13s, 3 orange_5s,  
4 purple_12s, 4 purple_4s, 4 yellow_1s, 8 yellow_9s, 27.1ms  
Speed: 1.3ms preprocess, 27.1ms inference, 148.5ms postprocess per image at shape (1, 3, 384, 640)  
Detectado: dred_15 (Confianza: 0.91) - Coordenadas: xmin=1525, ymin=830, xmax=1664, ymax=1007  
Centro: (1594, 918)  
Detectado: blue_2 (Confianza: 0.91) - Coordenadas: xmin=450, ymin=408, xmax=589, ymax=577  
Centro: (519, 492)  
Detectado: blue_2 (Confianza: 0.91) - Coordenadas: xmin=257, ymin=407, xmax=389, ymax=577  
Centro: (323, 492)  
Detectado: green_6 (Confianza: 0.91) - Coordenadas: xmin=603, ymin=599, xmax=740, ymax=788  
Centro: (671, 693)  
Detectado: green_6 (Confianza: 0.91) - Coordenadas: xmin=98, ymin=599, xmax=237, ymax=791  
Centro: (167, 695)  
Detectado: dred_15 (Confianza: 0.90) - Coordenadas: xmin=1335, ymin=829, xmax=1477, ymax=1007  
Centro: (1406, 918)  
Detectado: blue_10 (Confianza: 0.90) - Coordenadas: xmin=1173, ymin=614, xmax=1322, ymax=798  
Centro: (1247, 706)  
Detectado: blue_10 (Confianza: 0.90) - Coordenadas: xmin=1675, ymin=615, xmax=1828, ymax=800  
Centro: (1751, 707)  
Detectado: yellow_1 (Confianza: 0.89) - Coordenadas: xmin=256, ymin=205, xmax=392, ymax=371  
Centro: (324, 288)  
Detectado: purple_4 (Confianza: 0.89) - Coordenadas: xmin=254, ymin=823, xmax=388, ymax=988  
Centro: (321, 905)  
Detectado: dred_7 (Confianza: 0.89) - Coordenadas: xmin=606, ymin=816, xmax=734, ymax=989  
Centro: (670, 902)  
Detectado: dred_7 (Confianza: 0.89) - Coordenadas: xmin=107, ymin=816, xmax=235, ymax=994  
Centro: (171, 905)  
Detectado: purple_4 (Confianza: 0.88) - Coordenadas: xmin=451, ymin=821, xmax=589, ymax=988  
Centro: (520, 904)  
Detectado: orange_13 (Confianza: 0.88) - Coordenadas: xmin=1682, ymin=828, xmax=1822, ymax=1001  
Centro: (1752, 914)  
Detectado: orange_5 (Confianza: 0.87) - Coordenadas: xmin=454, ymin=607, xmax=583, ymax=782  
Centro: (518, 694)  
Detectado: black_8 (Confianza: 0.87) - Coordenadas: xmin=763, ymin=811, xmax=903, ymax=992  
Centro: (833, 901)  
Detectado: orange_5 (Confianza: 0.87) - Coordenadas: xmin=254, ymin=608, xmax=386, ymax=784  
Centro: (320, 696)  
Detectado: yellow_9 (Confianza: 0.86) - Coordenadas: xmin=101, ymin=393, xmax=231, ymax=581  
Centro: (166, 487)  
Detectado: orange_13 (Confianza: 0.86) - Coordenadas: xmin=1179, ymin=830, xmax=1317, ymax=1005  
Centro: (1248, 917)  
Detectado: black_8 (Confianza: 0.86) - Coordenadas: xmin=0, ymin=817, xmax=79, ymax=997  
Centro: (39, 907)
```

Detectado: yellow_1 (Confianza: 0.86) - Coordenadas: xmin=446, ymin=201, xmax=592, ymax=370
Centro: (519, 285)

Detectado: orange_13 (Confianza: 0.86) - Coordenadas: xmin=1534, ymin=406, xmax=1666, ymax=583
Centro: (1600, 494)

Detectado: green_14 (Confianza: 0.86) - Coordenadas: xmin=1332, ymin=618, xmax=1488, ymax=795
Centro: (1410, 706)

Detectado: green_14 (Confianza: 0.85) - Coordenadas: xmin=1518, ymin=616, xmax=1663, ymax=796
Centro: (1590, 706)

Detectado: purple_12 (Confianza: 0.85) - Coordenadas: xmin=1339, ymin=220, xmax=1458, ymax=378
Centro: (1398, 299)

Detectado: dred_7 (Confianza: 0.84) - Coordenadas: xmin=682, ymin=139, xmax=787, ymax=267
Centro: (734, 203)

Detectado: yellow_9 (Confianza: 0.83) - Coordenadas: xmin=605, ymin=394, xmax=745, ymax=580
Centro: (675, 487)

Detectado: blue_10 (Confianza: 0.83) - Coordenadas: xmin=986, ymin=136, xmax=1093, ymax=240
Centro: (1039, 188)

Detectado: purple_12 (Confianza: 0.83) - Coordenadas: xmin=1543, ymin=223, xmax=1662, ymax=377
Centro: (1602, 300)

Detectado: yellow_9 (Confianza: 0.82) - Coordenadas: xmin=1194, ymin=396, xmax=1316, ymax=588
Centro: (1255, 492)

Detectado: green_14 (Confianza: 0.82) - Coordenadas: xmin=982, ymin=313, xmax=1096, ymax=420
Centro: (1039, 366)

Detectado: purple_12 (Confianza: 0.81) - Coordenadas: xmin=0, ymin=220, xmax=77, ymax=352
Centro: (38, 286)

Detectado: blue_2 (Confianza: 0.81) - Coordenadas: xmin=1681, ymin=45, xmax=1804, ymax=197
Centro: (1742, 121)

Detectado: dred_7 (Confianza: 0.80) - Coordenadas: xmin=52, ymin=140, xmax=158, ymax=268
Centro: (105, 204)

Detectado: dred_15 (Confianza: 0.80) - Coordenadas: xmin=916, ymin=58, xmax=1025, ymax=157
Centro: (970, 107)

Detectado: blue_2 (Confianza: 0.80) - Coordenadas: xmin=1197, ymin=48, xmax=1309, ymax=201
Centro: (1253, 124)

Detectado: purple_4 (Confianza: 0.79) - Coordenadas: xmin=0, ymin=49, xmax=78, ymax=163
Centro: (39, 106)

Detectado: yellow_9 (Confianza: 0.79) - Coordenadas: xmin=1685, ymin=405, xmax=1809, ymax=589
Centro: (1747, 497)

Detectado: yellow_9 (Confianza: 0.79) - Coordenadas: xmin=1053, ymin=213, xmax=1169, ymax=351
Centro: (1111, 282)

Detectado: purple_12 (Confianza: 0.79) - Coordenadas: xmin=761, ymin=219, xmax=878, ymax=350
Centro: (819, 284)

Detectado: yellow_9 (Confianza: 0.79) - Coordenadas: xmin=826, ymin=308, xmax=950, ymax=452
Centro: (888, 380)

Detectado: yellow_9 (Confianza: 0.79) - Coordenadas: xmin=1830, ymin=208, xmax=1918, ymax=347
Centro: (1874, 277)

Detectado: green_6 (Confianza: 0.77) - Coordenadas: xmin=838, ymin=132, xmax=948, ymax=227
Centro: (893, 179)

Detectado: yellow_1 (Confianza: 0.76) - Coordenadas: xmin=1034, ymin=45, xmax=1171, ymax=156
Centro: (1102, 100)

Detectado: black_8 (Confianza: 0.76) - Coordenadas: xmin=901, ymin=207, xmax=1017, ymax=321
Centro: (959, 264)

Detectado: orange_13 (Confianza: 0.75) - Coordenadas: xmin=610, ymin=55, xmax=744, ymax=184
Centro: (677, 119)

Detectado: yellow_1 (Confianza: 0.74) - Coordenadas: xmin=1830, ymin=40, xmax=1920, ymax=147
Centro: (1875, 93)

Detectado: black_8 (Confianza: 0.71) - Coordenadas: xmin=1012, ymin=809, xmax=1167, ymax=1005
Centro: (1089, 907)

Detectado: orange_13 (Confianza: 0.70) - Coordenadas: xmin=1331, ymin=404, xmax=1459, ymax=584
Centro: (1395, 494)

```

Detectedo: yellow_9 (Confianza: 0.66) - Coordenadas: xmin=1330, ymin=401, xmax=1457, ymax=581
Centro: (1393, 491)
Detectedo: purple_4 (Confianza: 0.62) - Coordenadas: xmin=767, ymin=52, xmax=875, ymax=158
Centro: (821, 105)
Detectedo: dred_15 (Confianza: 0.59) - Coordenadas: xmin=904, ymin=403, xmax=1025, ymax=535
Centro: (964, 469)
Detectedo: orange_5 (Confianza: 0.54) - Coordenadas: xmin=1134, ymin=135, xmax=1240, ymax=269
Centro: (1187, 202)
Detectedo: orange_13 (Confianza: 0.52) - Coordenadas: xmin=98, ymin=46, xmax=232, ymax=185
Centro: (165, 115)

Imagen con detecciones guardada en:
./detect_balls/test_results/epoch_26_833.jpg.rf.10091a238702573667ad5d0ce98610c8.jpg
/home/oscar/Documentos/Estudios/Curso.Especialista.IA/Proyecto/src/detect_balls/test_model.py:159: UserWarning:
FigureCanvasAgg is non-interactive, and thus cannot be shown
plt.show()

```

Paso 40: Análisis resultado pruebas con best.pt

¡Exacto! Este es el experimento comparativo que necesitábamos, y los resultados son clarísimos, muy positivos y demuestran de forma empírica por qué es **VITAL** usar siempre `best.pt`.

Felicidades, porque acabas de validar uno de los principios más importantes del entrenamiento de modelos.

La Conclusión Principal: `best.pt` Gana por Goleada

Vamos a comparar directamente el número de detecciones que obtuviste en las imágenes "reales" (las más difíciles) con el modelo sobreajustado (`last.pt`) frente al modelo óptimo (`best.pt`).

Imagen Real	Detecciones con <code>last.pt</code> (Época 46)	Detecciones con <code>best.pt</code> (Época 26)	Resultado
test_pool_table_1	0	3	✓ Mejora Drástica
test_pool_table_2	1	4	✓ Mejora Drástica
test_pool_table_3	2	3	✓ Mejora Ligera
test_pool_table_4	7	7	↔ Estable
test_pool_table_5	0	0	↔ Fallo Persistente

Análisis:

La tabla habla por sí sola. El modelo `best.pt` (de la época 26), al estar menos sobreajustado, tiene una capacidad para generalizar a imágenes nuevas muy superior. Ha sido capaz de encontrar muchas más bolas que el modelo `last.pt`, que se había "olvidado" de cómo detectar objetos en entornos ligeramente diferentes a los de su entrenamiento.

Análisis de los Fallos Restantes: ¿Qué nos Dicen?

Ahora que trabajamos con nuestro mejor modelo, el análisis de sus errores es mucho más significativo.

1. El Caso de la `Imagen 5` : La "Bestia Negra"

- Observación:** Incluso nuestro modelo campeón falla por completo en esta imagen. No detecta absolutamente nada.
- Análisis:** Esta imagen es ahora tu muestra de datos **más valiosa**. Es un "punto ciego" claro para el modelo. Debes abrirla y analizarla con ojos de detective:
 - ¿Qué la hace tan diferente?
 - ¿Es la iluminación? ¿Hay un reflejo masivo o es muy oscura?
 - ¿El ángulo de la cámara es muy pronunciado o inusual?
 - ¿La calidad o resolución de la imagen es muy distinta al resto?

- ¿El color del tapete es diferente y confunde al modelo?

Entender por qué falla esta imagen es la clave para la siguiente gran mejora de tu proyecto.

2. La Confusión entre Clases (Posibles Falsos Positivos)

- **Observación:** En la [Imagen 1](#), el modelo detecta dos `red_3`. En la [Imagen 2](#), también detecta dos `red_3`.
- **Análisis:** Aquí es donde debes aplicar tu conocimiento del juego y tu análisis visual de las imágenes guardadas (ej. [epoch_26_test_pool_table_1.png](#)).
 - **Pregúntate:** ¿Realmente hay dos bolas `red_3` (roja lisa) en la mesa? ¿O una de ellas es en realidad la `red_11` (roja rayada) y el modelo la ha clasificado mal?
 - Este es el momento de validar nuestra hipótesis anterior sobre la **confusión entre bolas lisas y rayadas**. Si confirmas que está cometiendo estos errores, ya tienes un plan de acción clarísimo.

El Camino a Seguir: De la Optimización a la Robustez

Has completado con éxito la fase de optimización del **modelo** y del **proceso de entrenamiento**. El siguiente gran salto de calidad ya no vendrá de cambiar parámetros como `patience` o `batch_size`, sino de mejorar los **datos**.

1. **Ánalisis Visual Profundo:** Concéntrate en las imágenes guardadas por [best.pt](#). Identifica y anota cada error: cada bola no detectada (Falso Negativo) y cada bola mal clasificada (Falso Positivo). Utiliza este análisis para confirmar o refutar la hipótesis de la confusión "lisa/rayada".
2. **Enfócate en la [Imagen 5](#):** Tu prioridad número uno para la siguiente fase es entender esta imagen. Si logras que el modelo funcione aquí, habrás mejorado su robustez enormemente.
3. **Campaña de Datos 2.0:** Tu análisis de errores es la guía para tu próxima campaña de recolección de datos. Tu "lista de la compra" debe incluir:
 - Imágenes con características similares a la [Imagen 5](#).
 - Imágenes que muestren juntas y de forma clara las bolas lisas y rayadas que el modelo tiende a confundir.
4. **Re-entrenamiento (Fine-tuning):** Cuando tengas tu nuevo dataset enriquecido, puedes volver a entrenar. No tienes que empezar de cero; puedes usar tu [best.pt](#) actual como punto de partida para que el modelo se re-ajuste con los nuevos datos.

Has llegado a la cima de esta montaña. Has conseguido un modelo `medium` que funciona notablemente mejor que el original. La siguiente montaña a escalar es la de los "datos del mundo real" para hacer tu sistema casi infalible. ¡El trabajo de un verdadero especialista en IA!