

YOLOv5_alumnos - Deteccion de objetos con YOLOv5

Asignatura: Programacion con IA | Alumno: Diego Sanchez | Fecha: 14/02/2026

1. Definicion del problema

Sistema de detección para seguridad industrial con cuatro clases: Casco, Chaleco, Persona_Sin_Equipo y Peligro. Objetivo: apoyo a monitorización preventiva y reducción de riesgo operacional.

2. Dataset

Dataset YOLO propio en dataset_final/: 138 imágenes train y 38 val; 138 instancias train y 38 val. Distribución train por clase: Casco=32, Chaleco=35, Persona_Sin_Equipo=35, Peligro=36.

3. Entrenamiento y entorno

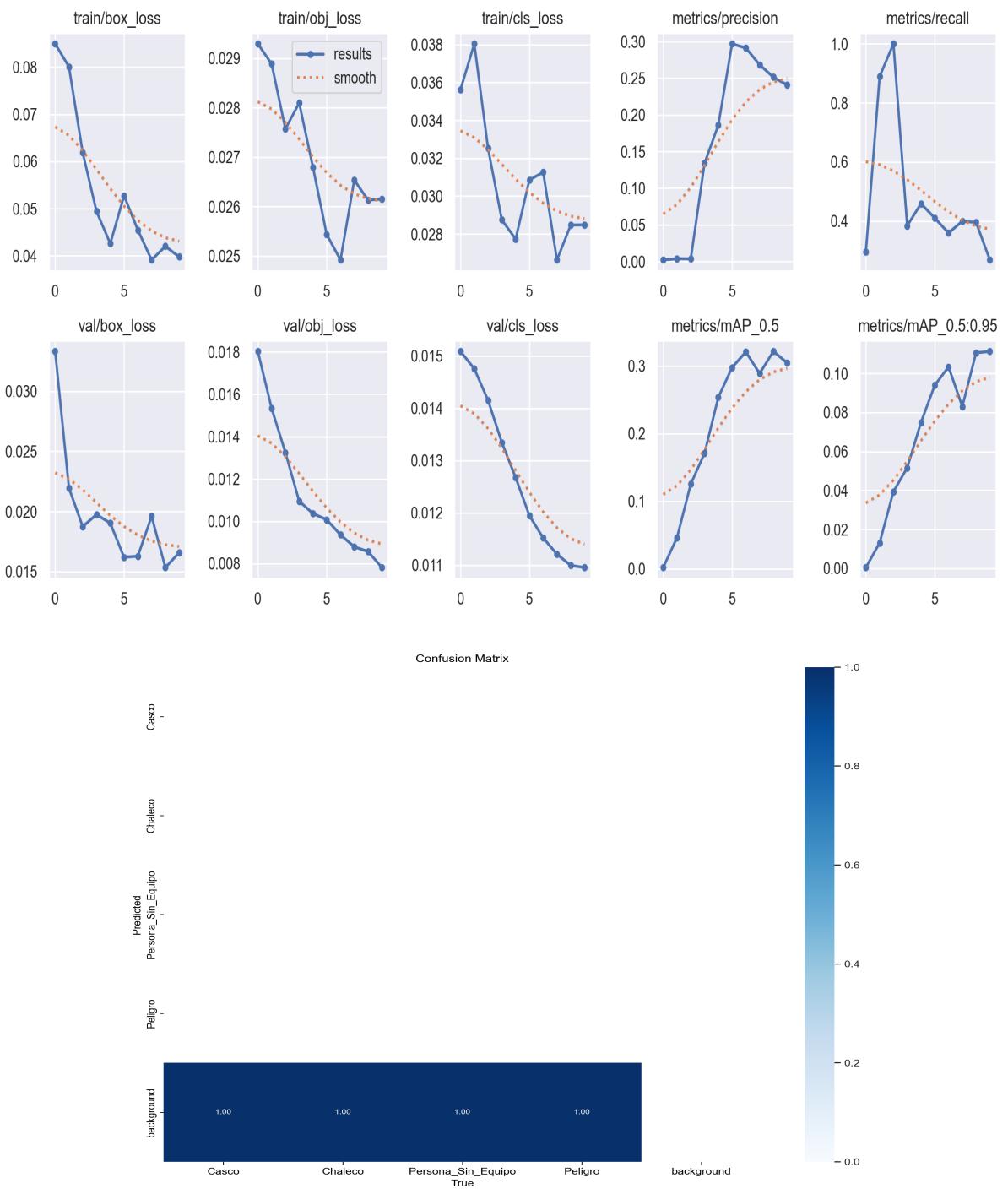
Entrenamiento ejecutado en CPU por entorno torch 2.10.0+cpu (torch.cuda.is_available=False), aunque existe GPU física RTX 2070 detectada por nvidia-smi. Comando: img=640, batch=8, epochs=10, pesos iniciales yolov5s.pt, cache activo.

4. Resultados objetivos

Mejor época (results.csv): época 8 con Precision=0.251, Recall=0.395, mAP@0.5=0.322 y mAP@0.5:0.95=0.111. Última época: mAP@0.5=0.305.

Validación por clase coherente con best.pt (val_actividad1_bs16): Casco mAP50=0.395, Chaleco=0.452, Persona_Sin_Equipo=0.102, Peligro=0.337.

Diferencia conceptual: mAP@0.5 usa IoU permisivo; mAP@0.5:0.95 promedia IoU crecientes y exige mejor localización. La brecha 0.322 vs 0.111 indica detección funcional con margen de mejora en ajuste de caja.



5. Análisis crítico

No hay evidencia fuerte de overfitting en 10 épocas: train y val loss descienden en paralelo. Si hay indicios de underfitting funcional en la clase crítica Persona_Sin_Equipo (recall bajo), probablemente por complejidad visual y pocas muestras efectivas.

Mejoras recomendadas: aumentar datos de clase débil, augmentation dirigido, y nueva corrida de 50 épocas en entorno CUDA para cerrar gap de generalización.

6. Inferencia

Se ejecuto detect.py con best.pt sobre inference_samples_val para inspeccion visual.
Metodologicamente, se recomienda complementar con set externo no visto para evitar sesgo del split de validacion.



7. Conclusiones

Entrega tecnicamente valida y trazable, con pipeline completo. Para aspirar a calificacion maxima:
reforzar evaluacion de inferencia externa y ejecutar 50 epochas en GPU para mejorar robustez de clases debiles.