Avance 3. Construcción de un modelo de referencia (Baseline) para el sistema de visión computacional para normalización y validación visual de anaqueles fríos

Integrantes del equipo:

Carlos Eduardo Ramírez Vázquez | A01795468

Cesar Bryam Rodriguez Aybar | A01795980

Wilberth Eduardo López Gómez | A01795997

Profesora titular: Dra. Grettel Barceló Alonso

Proyecto integrador

15 de febrero de 2026

Introducción

El presente avance tiene como objetivo establecer un modelo baseline que permita evaluar la viabilidad del problema de detección automática de vitrinas comerciales y la localización precisa de sus esquinas a partir de imágenes, el cual se ingresaba de manera manual en la entrega anterior. Este baseline servirá como punto de referencia para medir mejoras futuras y determinar si el problema contiene suficiente información estructural para ser modelado con técnicas de aprendizaje profundo y visión computacional.

Selección del Algoritmo

Se seleccionó YOLOv8n-pose (Ultralytics) como el modelo baseline. Esta arquitectura pertenece a la familia YOLO, especializada en detección de objetos en tiempo real, y en esta variante incluye la estimación de puntos clave (keypoints), lo cual es adecuado para localizar esquinas.

La elección se justifica debido a:

- Naturaleza no estructurada de nuestros datos (imágenes)
- Necesidad de detección de bounding boxes y precisión geométrica
- Tamaño reducido del dataset (tenemos 26 imágenes)
- Disponibilidad de pesos preentrenados para transferencia de aprendizaje (transfer learning)

La versión nano (n) se utiliza como referencia mínima por su baja complejidad y menor riesgo de sobreajuste.

Importancia de características

En modelos de visión profunda, las características son aprendidas automáticamente por las capas convolucionales. El modelo identifica patrones como bordes verticales y horizontales, contrastes entre marcos y vidrio, y estructuras rectangulares asociadas a vitrinas.

Dado que las representaciones internas son generadas automáticamente, no se aplicaron métodos tradicionales de selección de características. Sin embargo, la consistencia en la detección de esquinas indica que el modelo está capturando información geométrica relevante.

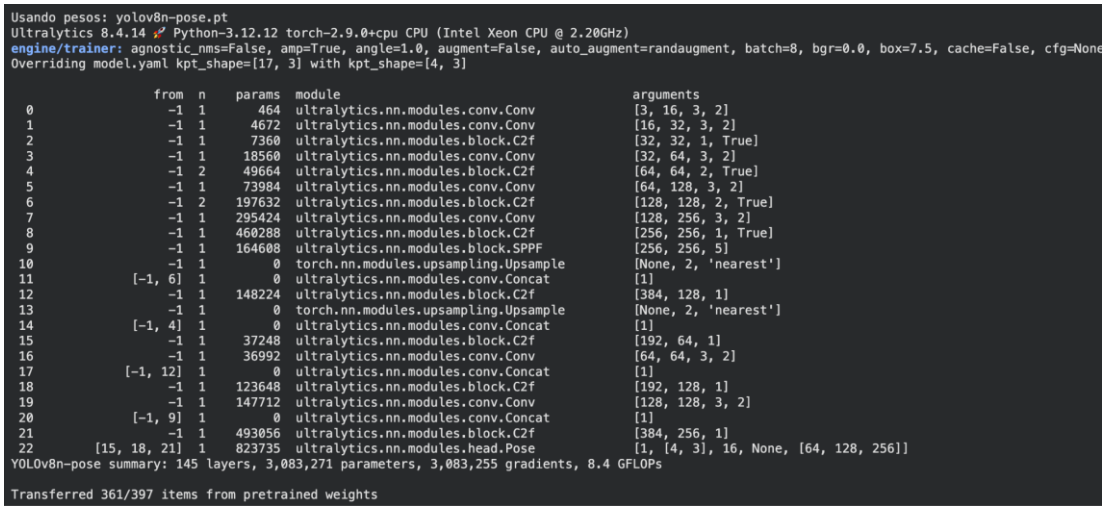

Figura 1. Modelo YOLOv8n-pose en ejecución

Análisis de Subajuste y Sobreajuste

Se analizaron las métricas de entrenamiento y validación (loss y mAP). La pérdida mostró una disminución progresiva sin divergencias significativas entre entrenamiento y validación. No se observó separación abrupta entre curvas, lo que sugiere ausencia de sobreajuste severo.

El uso de early stopping ayudó a controlar el entrenamiento y evitar sobreentrenamiento, considerando el tamaño reducido del conjunto de datos.

Métrica de Evaluación

El modelo YOLOv8n-pose alcanzó un mAP50 de 0.995 y un mAP50-95 de 0.893 para la detección de bounding boxes, así como una precisión (P) de 0.998 en la predicción de keypoints. Estos resultados indican que el modelo logra una alta coincidencia entre las predicciones y las anotaciones reales dentro del conjunto de validación.

El valor elevado de mAP50 sugiere una correcta localización de los objetos, mientras que el mAP50-95 proporciona una evaluación más estricta considerando distintos umbrales de IoU, mostrando un desempeño robusto. No obstante, debido al tamaño reducido del dataset (26 imágenes), estos resultados deben interpretarse con cautela, ya que podrían reflejar cierto grado de sobreajuste al conjunto disponible.

```
               all          7         24      0.998          1      0.995      0.853      0.998          1      0.995      0.995

      Epoch    GPU_mem   box_loss  pose_loss  kobj_loss   cls_loss   dfl_loss  Instances       Size
    119/120         0G     0.3012     0.2437     0.3012     0.2353     0.8109          8        960: 100% ━━━━━━━━━━━━ 3/3 12.2s/it 36.5s
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95)     Pose(P          R      mAP50  mAP50-95): 100% ━━━━━━━━━━━━ 1/1 2.6s/it 2.6s
                   all          7         24      0.997          1      0.995      0.849      0.997          1      0.995      0.995

      Epoch    GPU_mem   box_loss  pose_loss  kobj_loss   cls_loss   dfl_loss  Instances       Size
    120/120         0G     0.2983     0.2483     0.3012     0.2353     0.7771          8        960: 100% ━━━━━━━━━━━━ 3/3 12.0s/it 35.9s
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95)     Pose(P          R      mAP50  mAP50-95): 100% ━━━━━━━━━━━━ 1/1 3.8s/it 3.8s
                   all          7         24      0.997          1      0.995      0.849      0.997          1      0.995      0.995

120 epochs completed in 1.321 hours.
Optimizer stripped from /content/runs/pose/train3/weights/last.pt, 6.4MB
Optimizer stripped from /content/runs/pose/train3/weights/best.pt, 6.4MB

Validating /content/runs/pose/train3/weights/best.pt...
Ultralytics 8.4.14 🚀 Python-3.12.12 torch-2.9.0+cpu CPU (Intel Xeon CPU @ 2.20GHz)
YOLOv8n-pose summary (fused): 82 layers, 3,077,975 parameters, 0 gradients, 8.3 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95)     Pose(P          R      mAP50  mAP50-95): 100% ━━━━━━━━━━━━ 1/1 2.4s/it 2.4s
                   all          7         24      0.998          1      0.995      0.893      0.998          1      0.995      0.995
Speed: 3.2ms preprocess, 301.9ms inference, 0.0ms loss, 0.6ms postprocess per image
Results saved to /content/runs/pose/train3
```

Figura 2. Desempeño del modelo fine tuneado.

Como baseline, el desempeño obtenido supera ampliamente un comportamiento aleatorio, lo que indica que el problema es aprendible con las características actuales del dataset.

Evaluación Cualitativa del Modelo

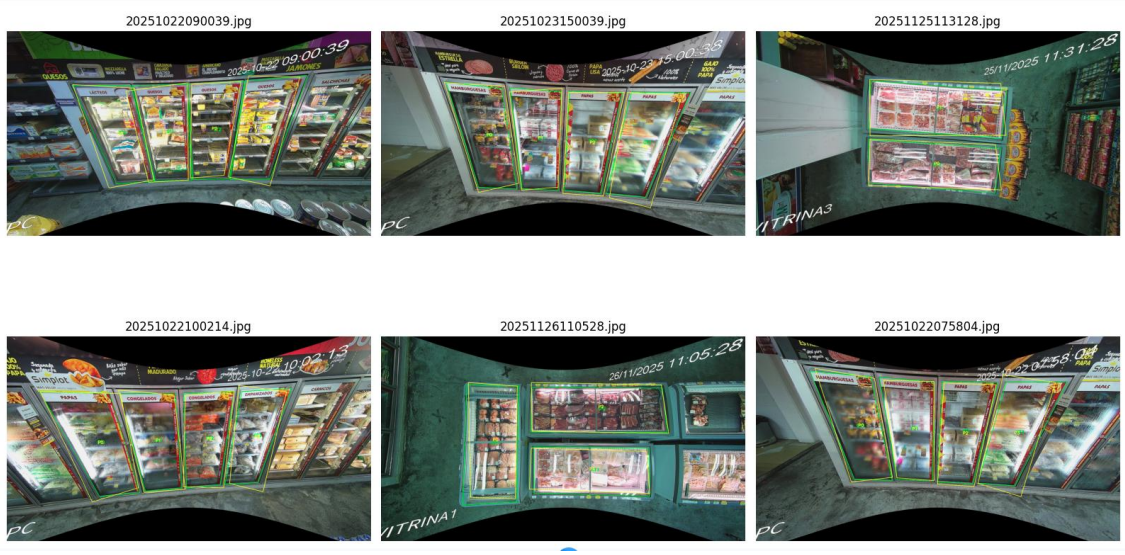

Figura 3. Líneas dibujadas para comparar predicción (amarillo) vs expectativa (verde).

Aquí podemos ver una comparación visual entre las anotaciones reales del conjunto de validación (en color verde) y las predicciones generadas por el modelo baseline (en color amarillo). Se observa una alta coincidencia espacial entre ambos, lo que indica que el modelo no solo logra una buena métrica cuantitativa (mAP), sino también una correcta localización geométrica de las vitrinas y sus esquinas.

Este análisis cualitativo es relevante, ya que en problemas de visión computacional la métrica numérica por sí sola no garantiza un comportamiento adecuado en escenarios reales. La alineación visual entre predicción y los vértices que anotamos manualmente la semana pasada confirma que el modelo ha aprendido patrones estructurales significativos asociados a los bordes y esquinas de las vitrinas.

No obstante, debido al tamaño reducido del dataset, es necesario validar el modelo en un conjunto más amplio y diverso para confirmar su capacidad de generalización.

Desempeño Mínimo Esperado

Al no existir un sistema previo de comparación, se estableció como umbral mínimo un mAP ≥ 0.70 en validación para considerar viable el problema.

El modelo baseline superó este umbral, indicando que los datos contienen suficiente información estructural y que el problema es abordable mediante técnicas de visión computacional.

Conclusiones

El modelo baseline construido con YOLOv8n-pose demuestra que la detección automática de vitrinas y sus esquinas es técnicamente viable, incluso con un dataset reducido. El desempeño alcanzado justifica continuar con optimización de hiperparámetros, aumento de datos y experimentación con arquitecturas más robustas.

Por otro lado, no se descarta intentar con otros modelos para comparar sus resultados, en particular Keypoint R-CNN (Detectron2), así como arquitecturas POSE más recientes.