

Avance 5. Modelo Final



Análisis de Anaqueles BimboNet

Equipo 14

Integrantes del equipo:

- Giancarlo Franco Carrillo A01638108
- Luis Fernando Rivera Albarrán A01209133
- Brandon Vladimir Montoya López A01209115

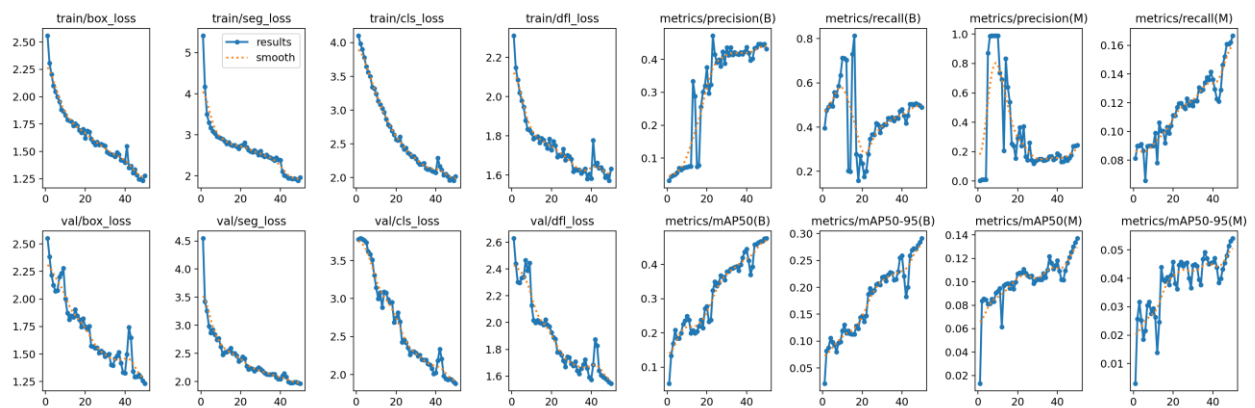
Introduccion

Para este avance, mejoramos significativamente la generación del dataset sintético, agregando físicas a los objetos y variaciones de luz entre escenas, además de incluir los productos de Barcel para generar un dataset muy variado.

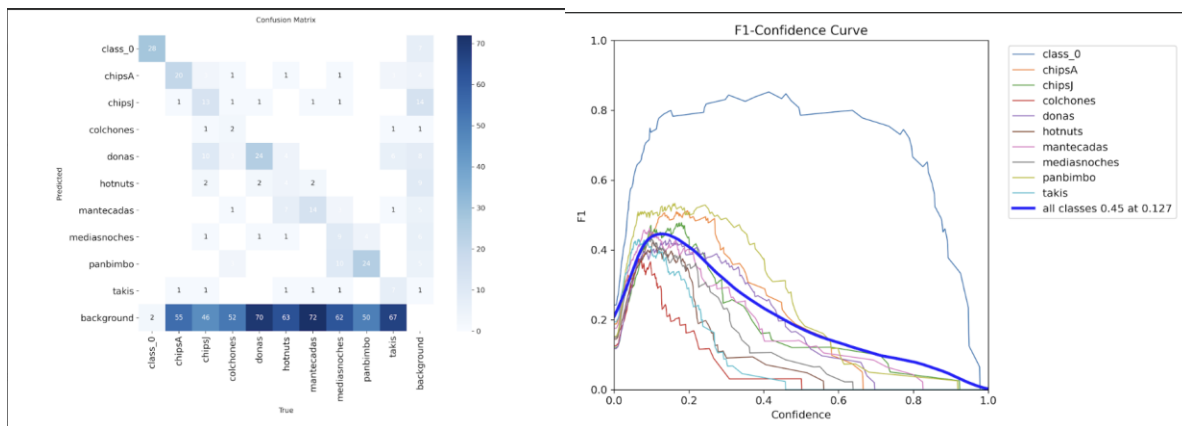
Ademas de separar en sets de entrenamiento y validación para poder entrenar un YOLOv8-Seg.

Análisis comparativo de modelos por épocas

Yolov8-Seg 50 epochs

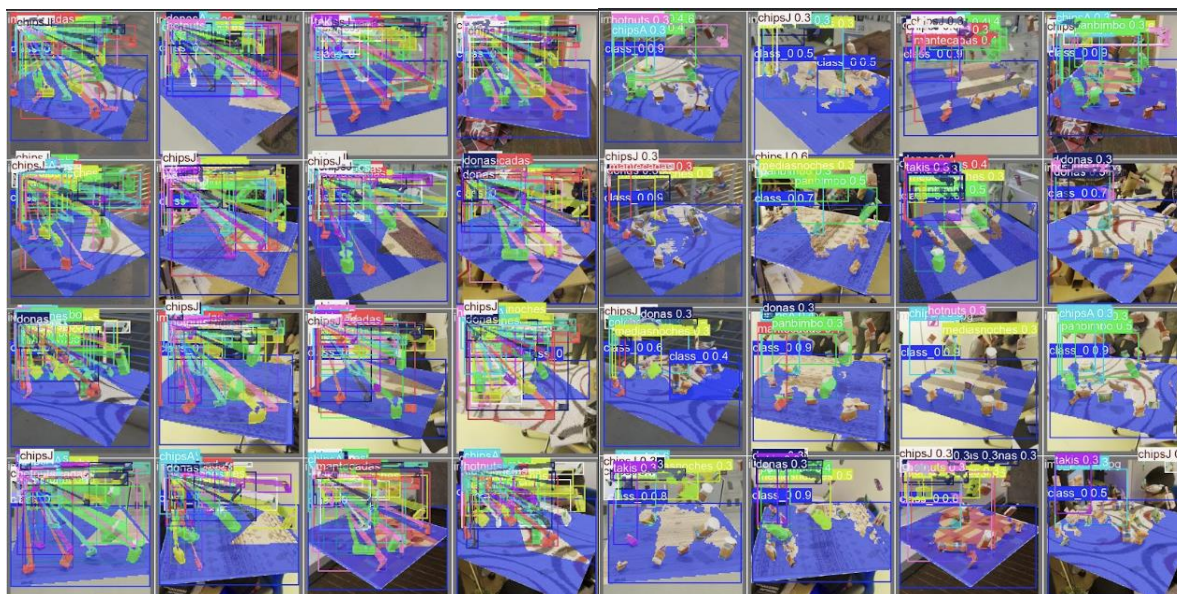


Results



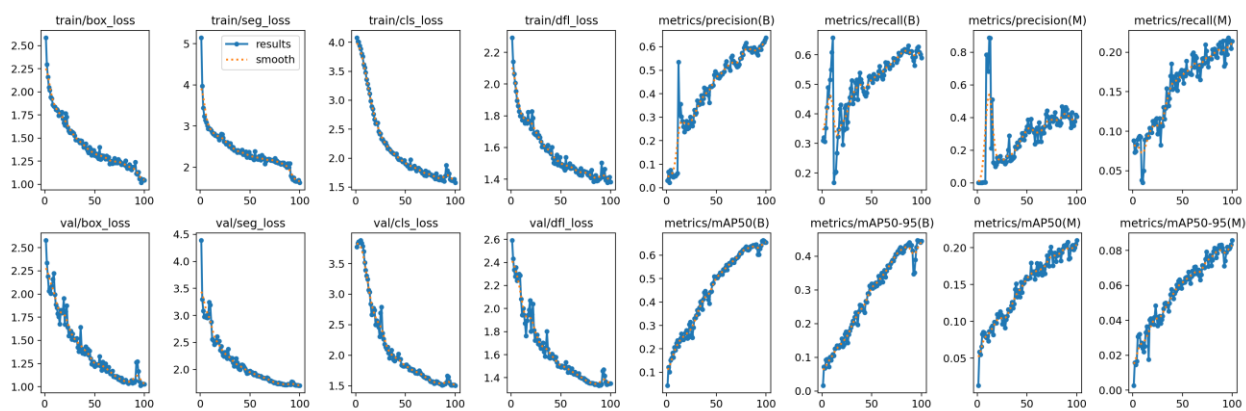
Matriz de confusion

Bbox F1 curve

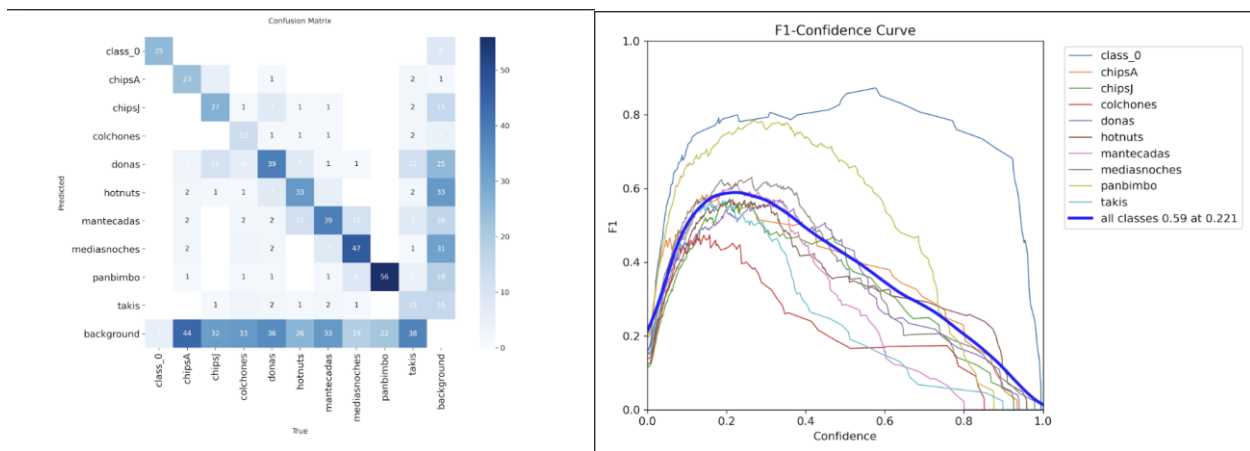


Labels vs Predictions

Yolov8-Seg 100 epochs

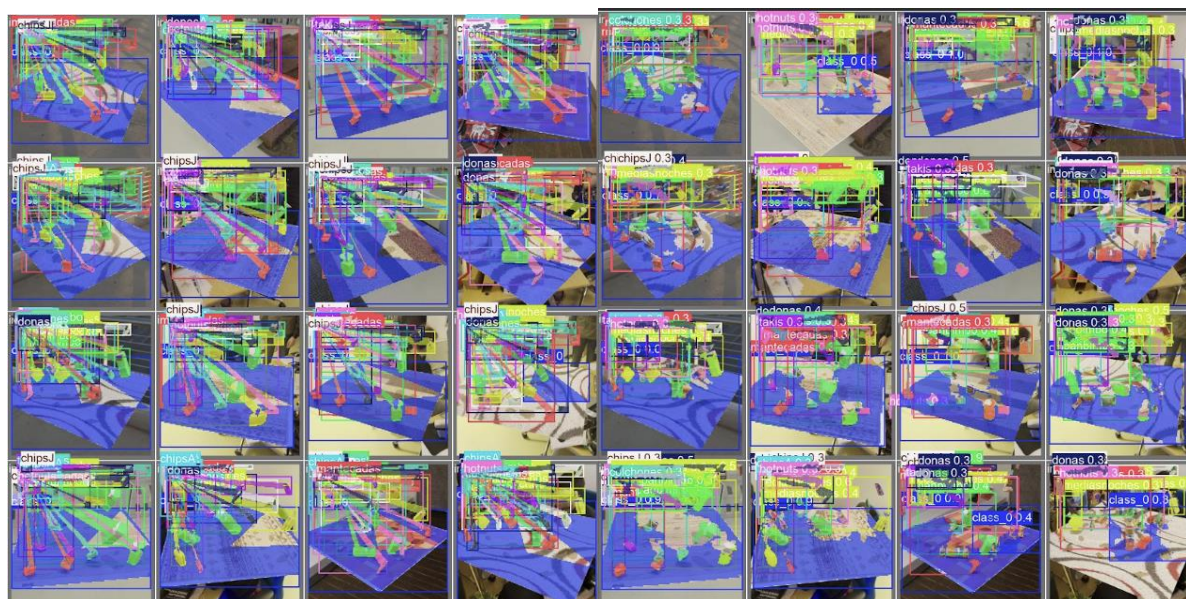


Results



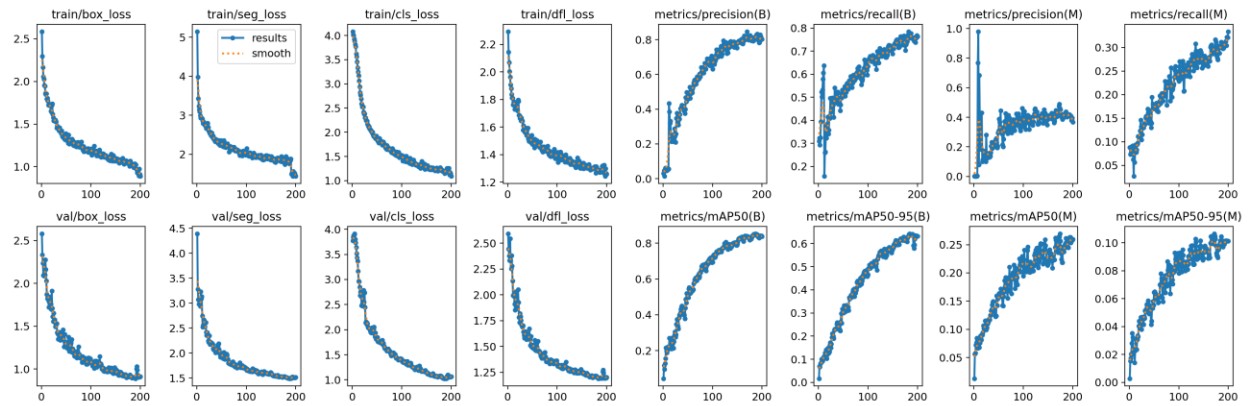
Matriz de confusion

Bbox F1 curve

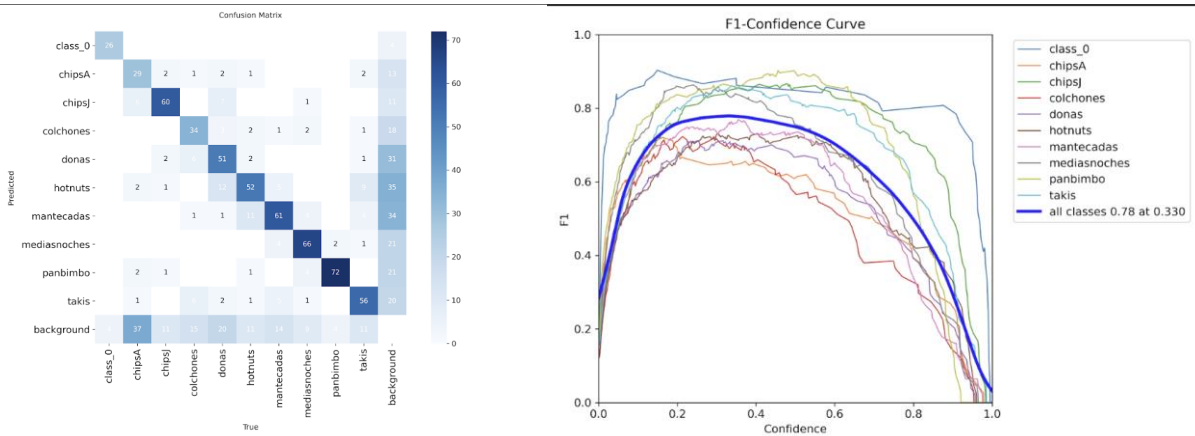


Labels vs Predictions

Yolov8-Seg 200 epochs

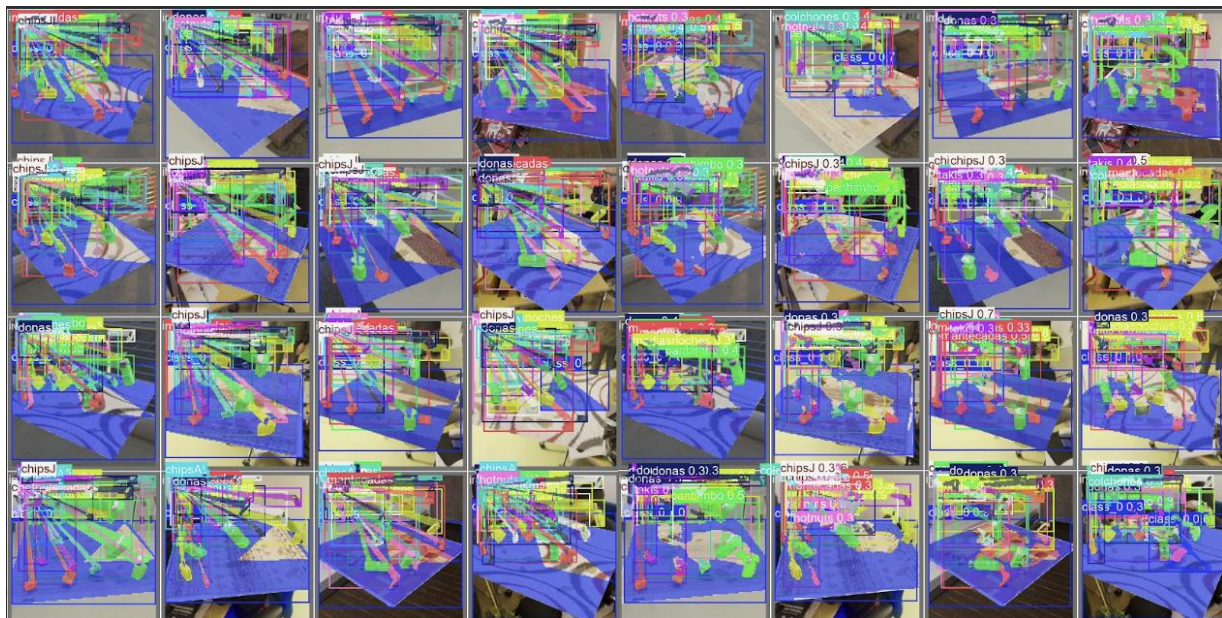


Results



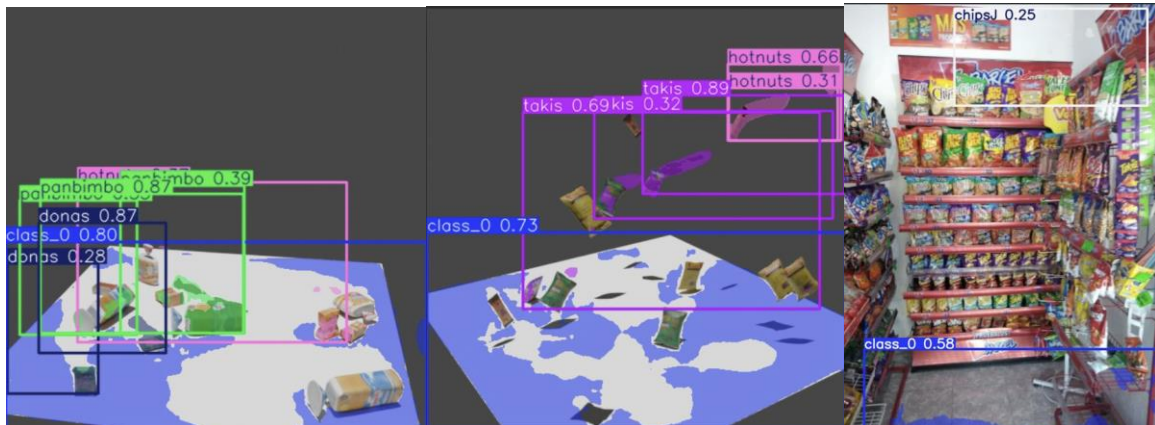
Matriz de confusion

Bbox F1 curve



Labels vs Predictions

Resultados finales



Consideramos que el modelo puede mejorar bastante con más datos ya que para entrenar estos modelos se tomaron 15 escenas con 10 frames por escena y alrededor de 2-5 instancias por clase, es decir: 150 imágenes y alrededor de 4k máscaras totales (400 máscaras por clase en el mejor escenario), con mejores modelos, más épocas y con instancias de datos reales, podría mejorar bastante.

Video demo

También generamos un video de demostración de las físicas y la generación del dataset sintético, con la finalidad de mostrar avances y las distintas técnicas que hemos usado para dar solución al problema de BimboNet, esperando que les resulte grato y nos puedan proporcionar modelos con mejor resolución que nos permitan generar un dataset más realista.

Link a los videos:

https://drive.google.com/drive/folders/1bxHH0KZ4iiewtOOQ4QZAfssQQ_X0X7gE?usp=sharing

Física en el dataset sintético

Como parte del proceso de generación de datos sintéticos mediante BlenderProc, se implementó una simulación física basada en gravedad con el objetivo de incrementar el realismo de las escenas utilizadas para entrenar modelos de clasificación de objetos.

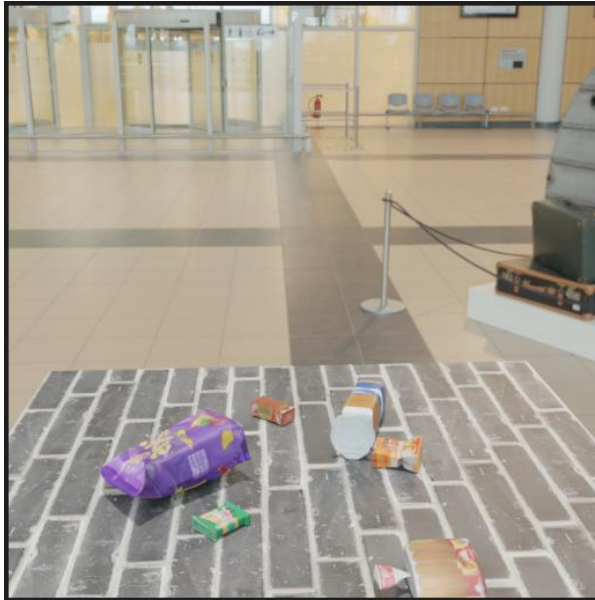
La incorporación de gravedad permite que los objetos tridimensionales interactúen de manera más verosímil con el entorno y entre sí, favoreciendo la creación de escenas que reflejan condiciones físicas del mundo real. Esto reduce la discrepancia entre los dominios sintético y real, lo cual es crucial para mejorar la generalización de modelos entrenados con datos simulados.

Proceso de implementación:

1. **Inicialización del motor físico**
Se activó el motor de simulación física de Blender mediante la función `bproc.physics.enable_physics()`, permitiendo la inclusión de dinámicas físicas realistas.
2. **Asignación de cuerpos rígidos a objetos**
Los objetos importados (en formato `.ply`) fueron configurados como cuerpos rígidos mediante `bproc.physics.set_rigidbody()`, permitiendo que fueran afectados por la gravedad y colisiones.
3. **Simulación temporal hasta reposo**
La escena fue simulada durante varios segundos utilizando `bproc.physics.simulate_and_check_stop()` para asegurar que los objetos alcanzaran una posición de reposo, es decir, sin movimiento significativo restante.
4. **Captura de datos una vez estabilizada la escena**
Tras confirmar que los objetos estaban en reposo físico, se procedió al renderizado de imágenes RGB, mapas de segmentación y etiquetas asociadas, lo que garantiza que los datos recolectados reflejen distribuciones espaciales físicamente plausibles.

Justificación

La simulación de gravedad introduce una distribución natural en la colocación de los objetos, alineada con principios físicos que serían difíciles de replicar manualmente. Esta metodología contribuye a incrementar el valor de los datos sintéticos, facilitando su uso para el entrenamiento de modelos de visión por computadora en tareas como clasificación o detección de objetos.



Link al repositorio:

https://github.com/MrLourlo/Proyecto_Integrador_Anaqueles/tree/vertex_color