Detección de residuos con modelos de visión por computadora

Ramiro Feichubuinm - Gonzalo Silva

Contexto y problema a resolver

- La acumulación de residuos en espacios públicos genera problemas ambientales y operativos.
- El monitoreo manual de la basura es costoso, ineficiente y difícil de escalar.
- Los residuos mal clasificados afectan los procesos de reciclaje y limpieza.
- Se requiere una solución automatizada para detectar residuos en imágenes reales.
- La visión por computadora permite identificar basura de forma rápida y precisa.
- Este trabajo aplica modelos de detección de objetos para abordar este desafío.

Análisis exploratorio del conjunto de datos

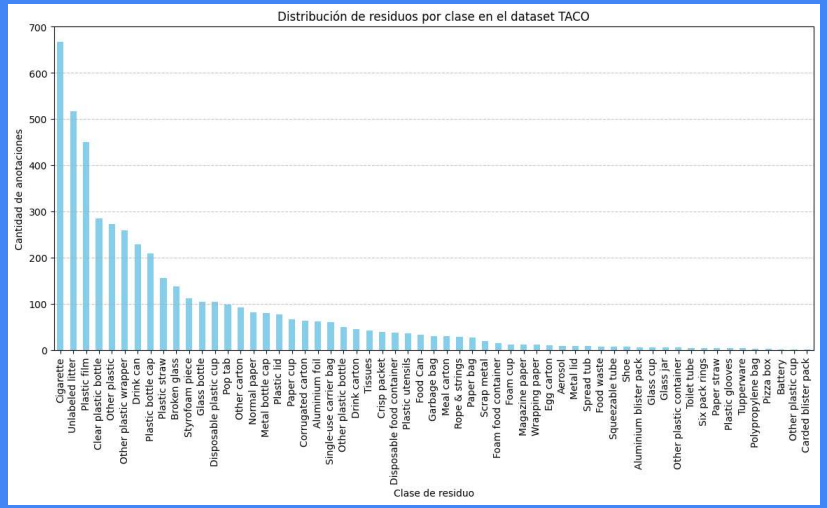
Ejemplo de clase: Aluminium foil

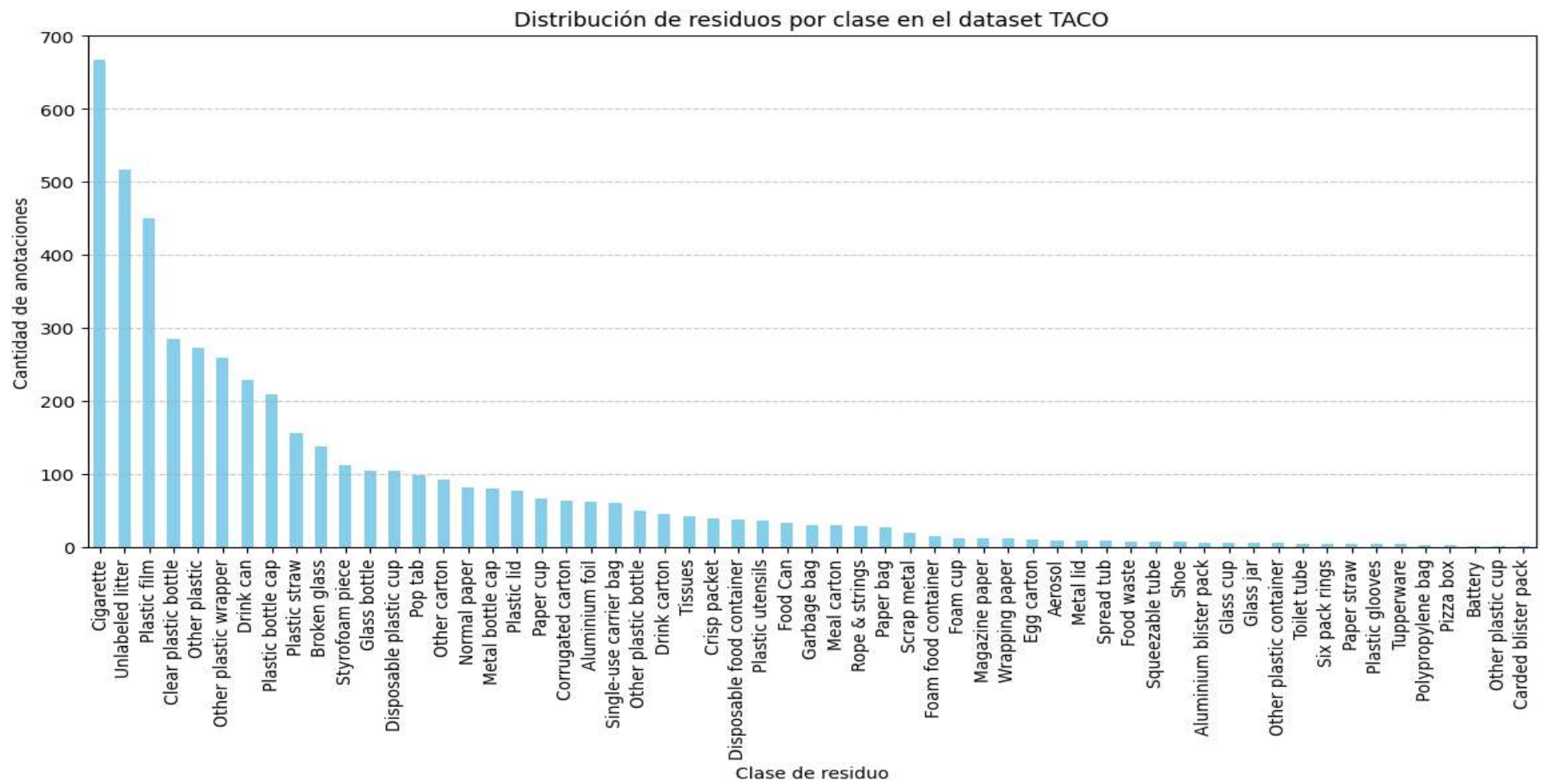
Ejemplo de clase: Clear plastic bottle

Ejemplo de clase: Other carton

Ejemplos

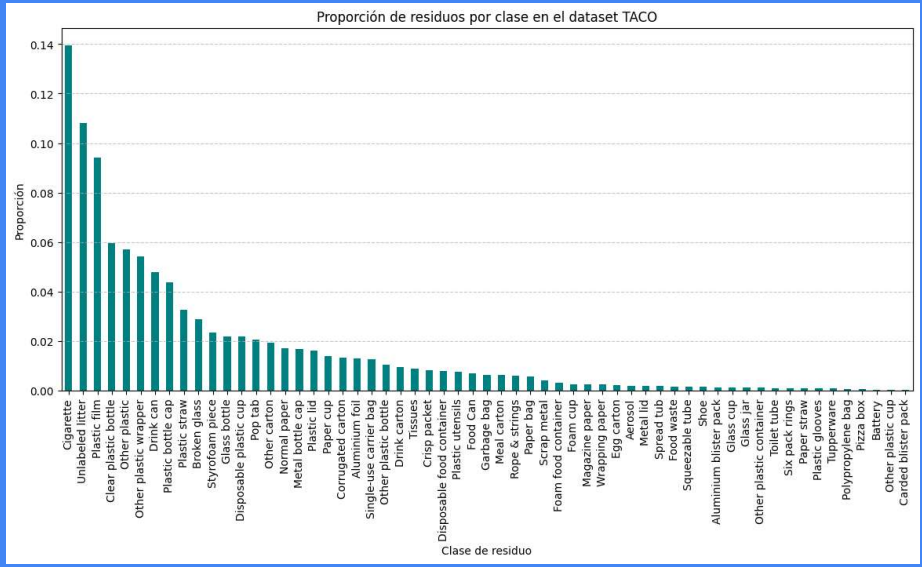
Distribución de residuos por clase

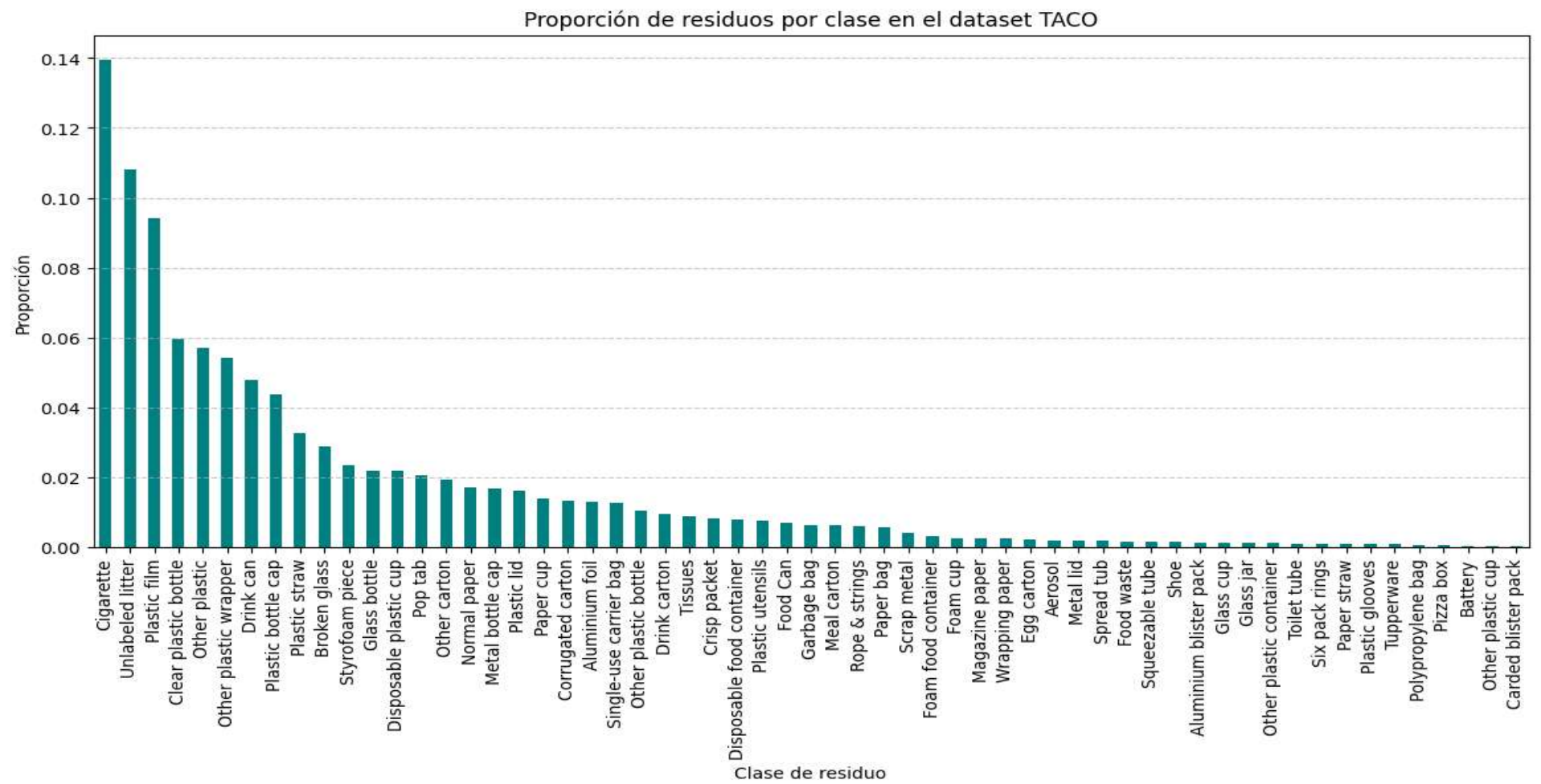




Distribución de residuos por clase

Desbalance del dataset

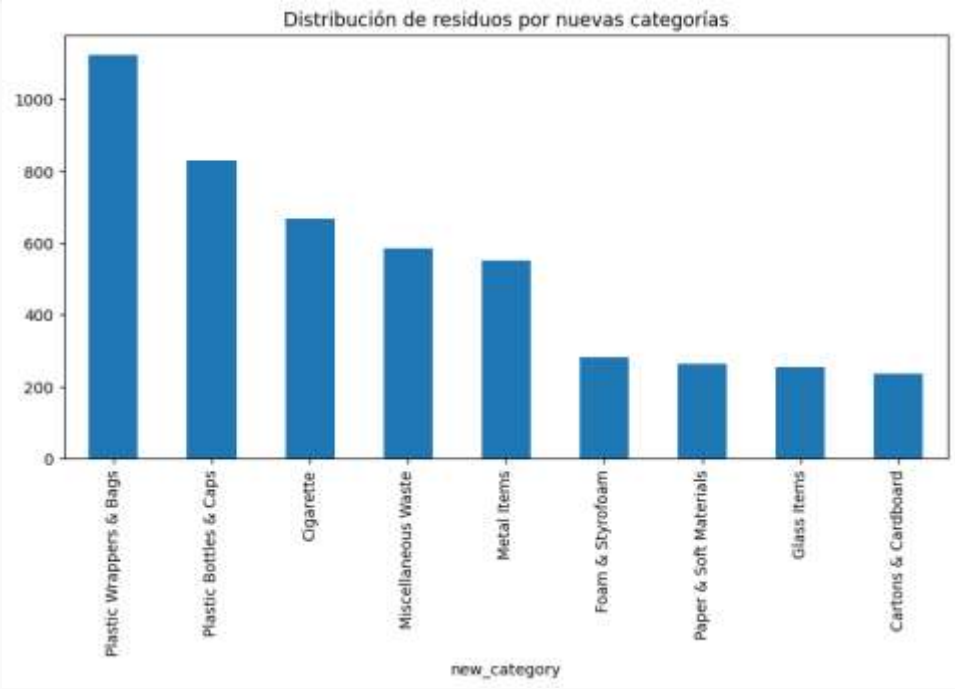




Desbalance del dataset

Distribución del dataset por nuevas categorías

- Para reducir el desbalance extremo del dataset original, se agruparon las clases en categorías generales basadas en el tipo de material (plástico, metal, cartón, etc.).
- Se definieron 9 grandes categorías: *Plastic Wrappers & Bags, Plastic Bottles & Caps, Cigarette, Miscellaneous Waste, Metal Items, Foam & Styrofoam, Paper & Soft Materials, Glass Items, Cartons & Cardboard.*

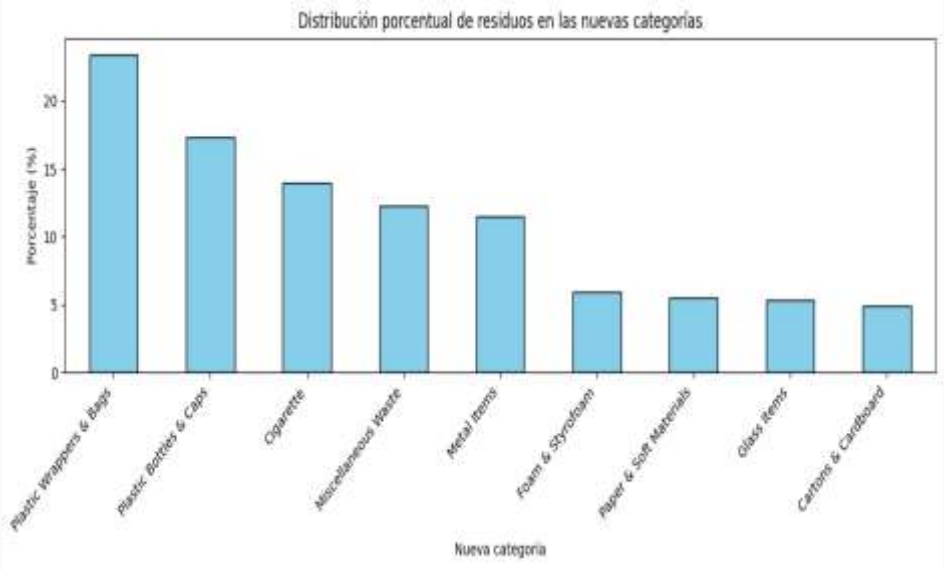


Mapeo de Clases

```
custom_mapping = {
    "Cigarette": ["Cigarette"],
    "Glass Items": ["Broken glass", "Glass bottle", "Glass cup", "Glass jar"],
    "Metal Items": ["Drink can", "Pop tab", "Metal bottle cap", "Aluminium foil", "Scrap metal",
                    "Metal lid", "Aerosol", "Food Can", "Aluminium blister pack"],
    "Plastic Bottles & Caps": ["Plastic bottle cap", "Plastic bottle", "Other plastic bottle",
                               "Plastic utensils", "Tupperware", "Polypropylene bag", "Plastic straw",
                               "Plastic lid", "Plastic glooves", "Other plastic cup", "Clear plastic bottle"],
    "Plastic Wrappers & Bags": ["Plastic film", "Other plastic wrapper", "Single-use carrier bag",
                                "Other plastic container", "Garbage bag", "Plastic bag", "Other plastic bag",
                                "Crisp packet", "Other plastic"],
    "Cartons & Cardboard": ["Other carton", "Drink carton", "Meal carton", "Pizza box", "Corrugated carton"],
    "Paper & Soft Materials": ["Normal paper", "Paper cup", "Paper bag", "Magazine paper",
                               "Wrapping paper", "Egg carton", "Toilet tube", "Paper straw", "Tissues"],
    "Foam & Styrofoam": ["Styrofoam piece", "Disposable plastic cup", "Disposable food container",
                         "Foam food container", "Foam cup"],
    "Miscellaneous Waste": ["Unlabeled litter", "Rope & strings", "Food waste", "Shoe",
                            "Squeezable tube", "Battery", "Carded blister pack", "Spread tub", "Six pack rings"]
}
```

Distribución del dataset por nuevas categorías

- La nueva distribución sigue siendo desigual, pero es más balanceada que la original.
- Esta agrupación facilita el aprendizaje del modelo y mejora la estabilidad de las métricas en clases con poca representación.
- También permite una mejor interpretación de los resultados, especialmente en contextos reales donde el tipo de material es más relevante que la clase exacta.



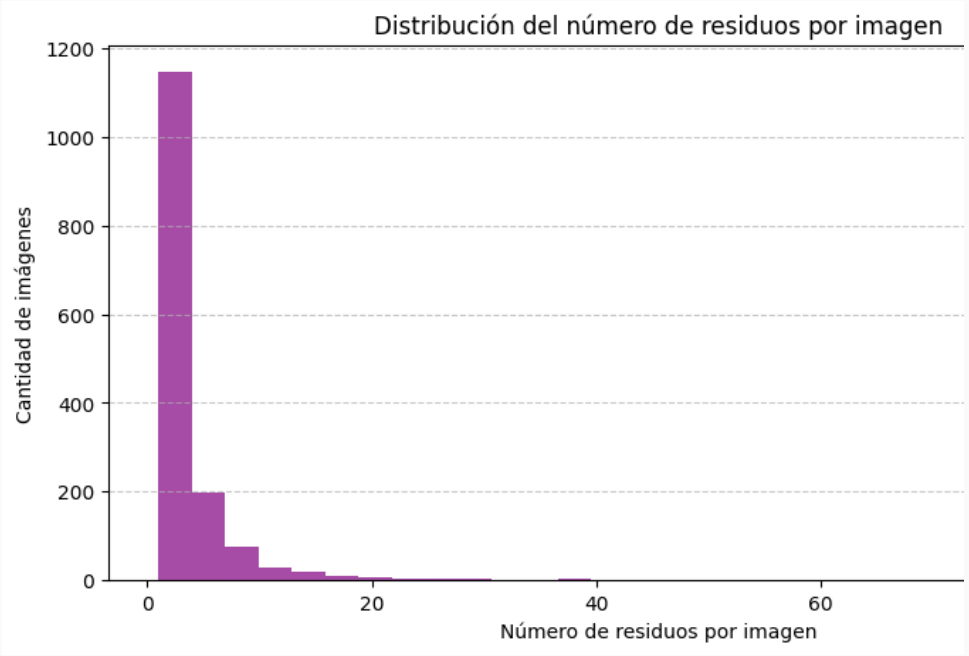
Dataframe

	id	image_id	category_id	segmentation	area	bbox	iscrowd	category_name
0	1	0	6	[[561.0, 1238.0, 568.0, 1201.0, 567.0, 1175.0,...	403954.0	[517.0, 127.0, 447.0, 1322.0]	0	Glass bottle
1	2	1	18	[[928.0, 1876.0, 938.0, 1856.0, 968.0, 1826.0,...	1071259.5	[1.0, 457.0, 1429.0, 1519.0]	0	Meal carton
2	3	1	14	[[617.0, 383.0, 703.0, 437.0, 713.0, 456.0, 72...	99583.5	[531.0, 292.0, 1006.0, 672.0]	0	Other carton
3	4	2	5	[[670.0, 993.0, 679.0, 998.0, 684.0, 1001.0, 6...	73832.5	[632.0, 987.0, 500.0, 374.0]	0	Clear plastic bottle
4	5	2	7	[[647.0, 1028.0, 650.0, 1022.0, 653.0, 1016.0,...	915.0	[632.0, 989.0, 44.0, 51.0]	0	Plastic bottle cap

	id	image_id	category_id	segmentation	area	bbox	iscrowd	category_name	new_category
0	1	0	0	[[561.0, 1238.0, 568.0, 1201.0, 567.0, 1175.0,...	4.039540e+05	[517.0, 127.0, 447.0, 1322.0]	0	Glass bottle	Glass Items
1	2	1	1	[[928.0, 1876.0, 938.0, 1856.0, 968.0, 1826.0,...	1.071260e+06	[1.0, 457.0, 1429.0, 1519.0]	0	Meal carton	Cartons & Cardboard
2	3	1	1	[[617.0, 383.0, 703.0, 437.0, 713.0, 456.0, 72...	9.958350e+04	[531.0, 292.0, 1006.0, 672.0]	0	Other carton	Cartons & Cardboard
3	4	2	2	[[670.0, 993.0, 679.0, 998.0, 684.0, 1001.0, 6...	7.383250e+04	[632.0, 987.0, 500.0, 374.0]	0	Clear plastic bottle	Plastic Bottles & Caps
4	5	2	2	[[647.0, 1028.0, 650.0, 1022.0, 653.0, 1016.0,...	9.150000e+02	[632.0, 989.0, 44.0, 51.0]	0	Plastic bottle cap	Plastic Bottles & Caps

Distribución del número de residuos por imagen

- La mayoría de las imágenes tienen muy pocos residuos anotados (entre 1 y 5).
- Solo unas pocas imágenes contienen más de 20 residuos, y casos extremos con más de 80 objetos son raros.

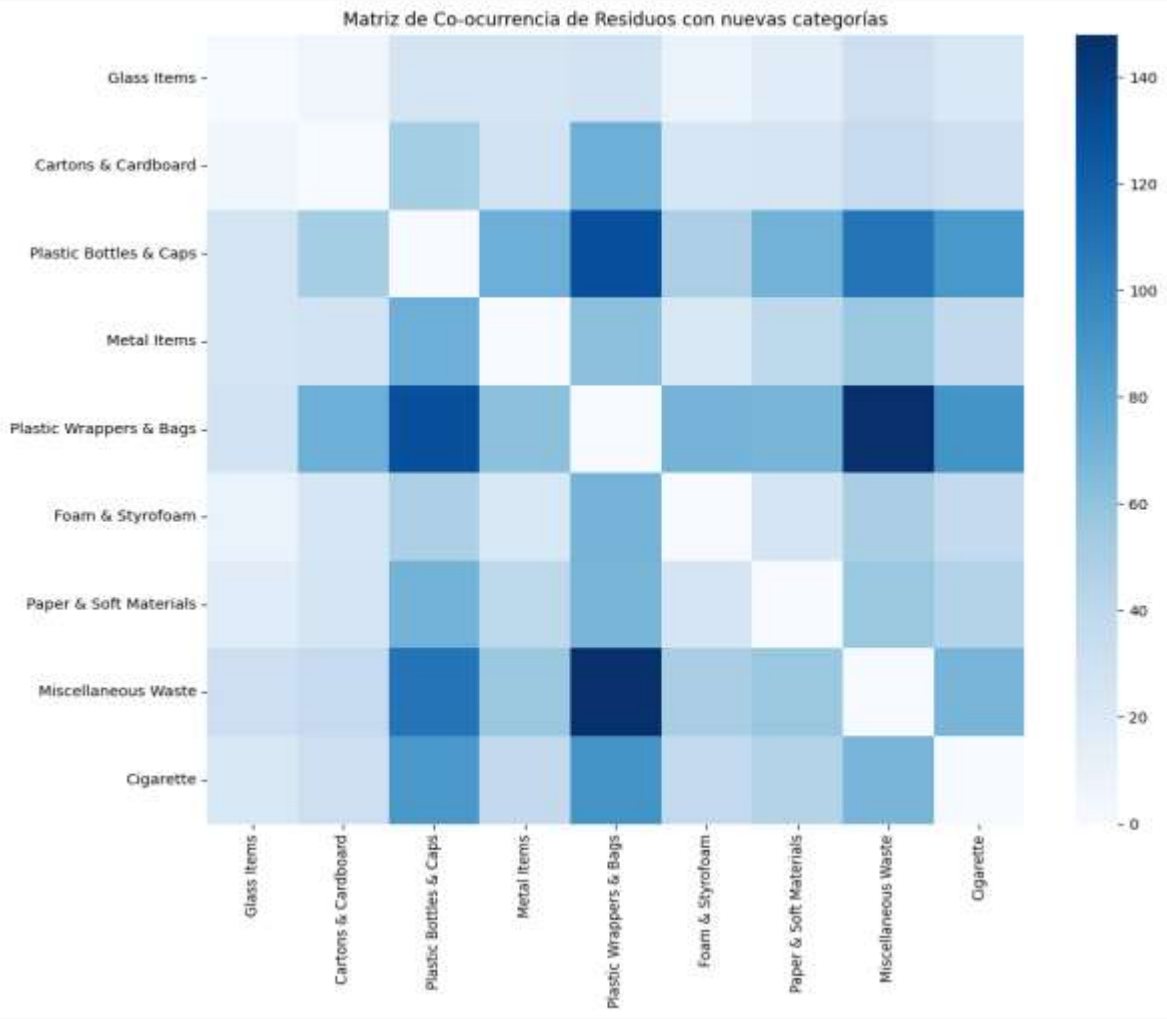


Matriz de co-ocurrencia

- Las categorías Plastic Wrappers & Bags, Plastic Bottles & Caps y Miscellaneous Waste son las que más coexisten con otras.
- Algunas combinaciones frecuentes son:

  Plastic Wrappers & Bags + Plastic Bottles & Caps

  Plastic Bottles & Caps + Miscellaneous Waste

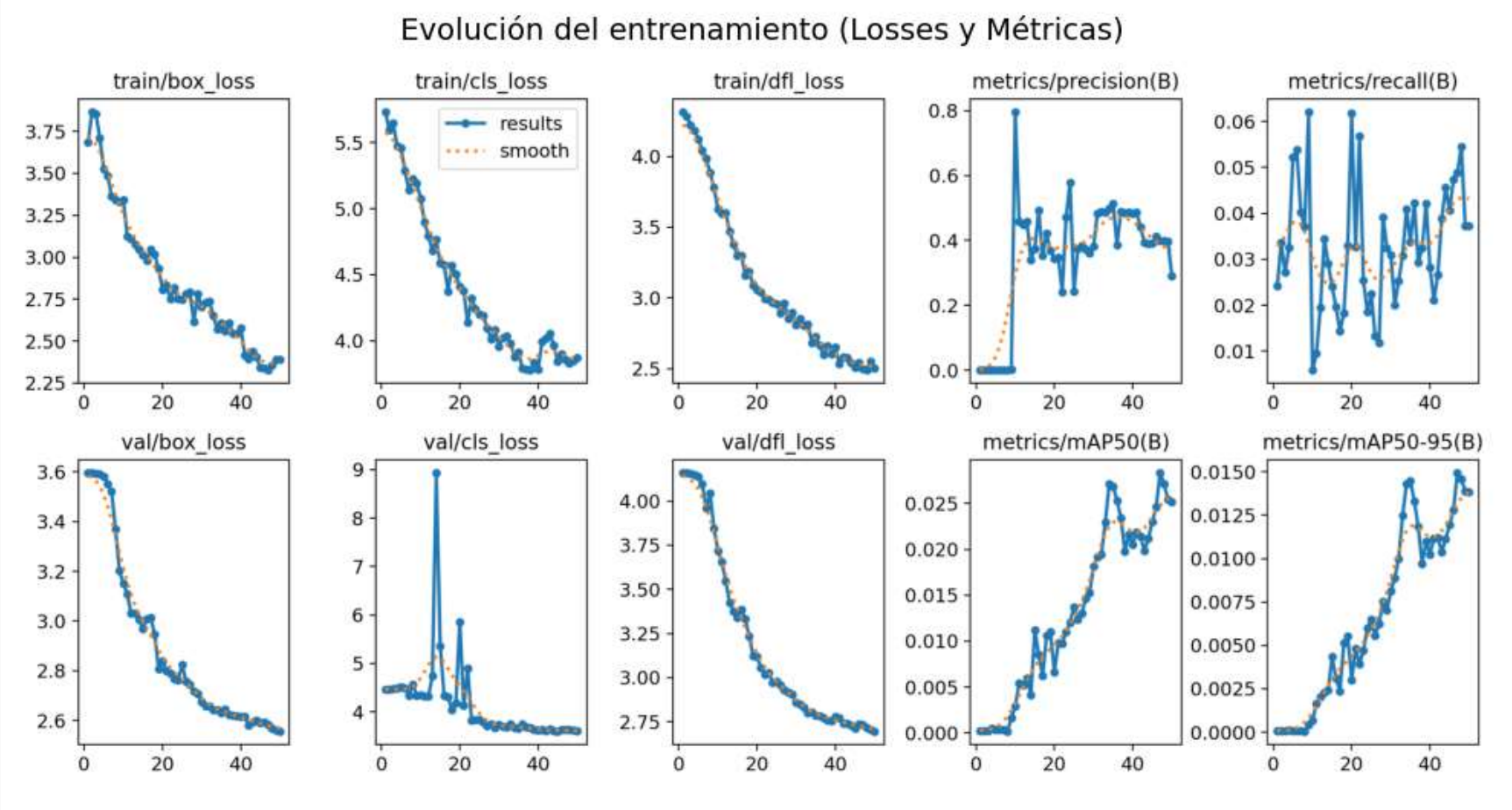


Arquitectura a utilizar

- YOLO

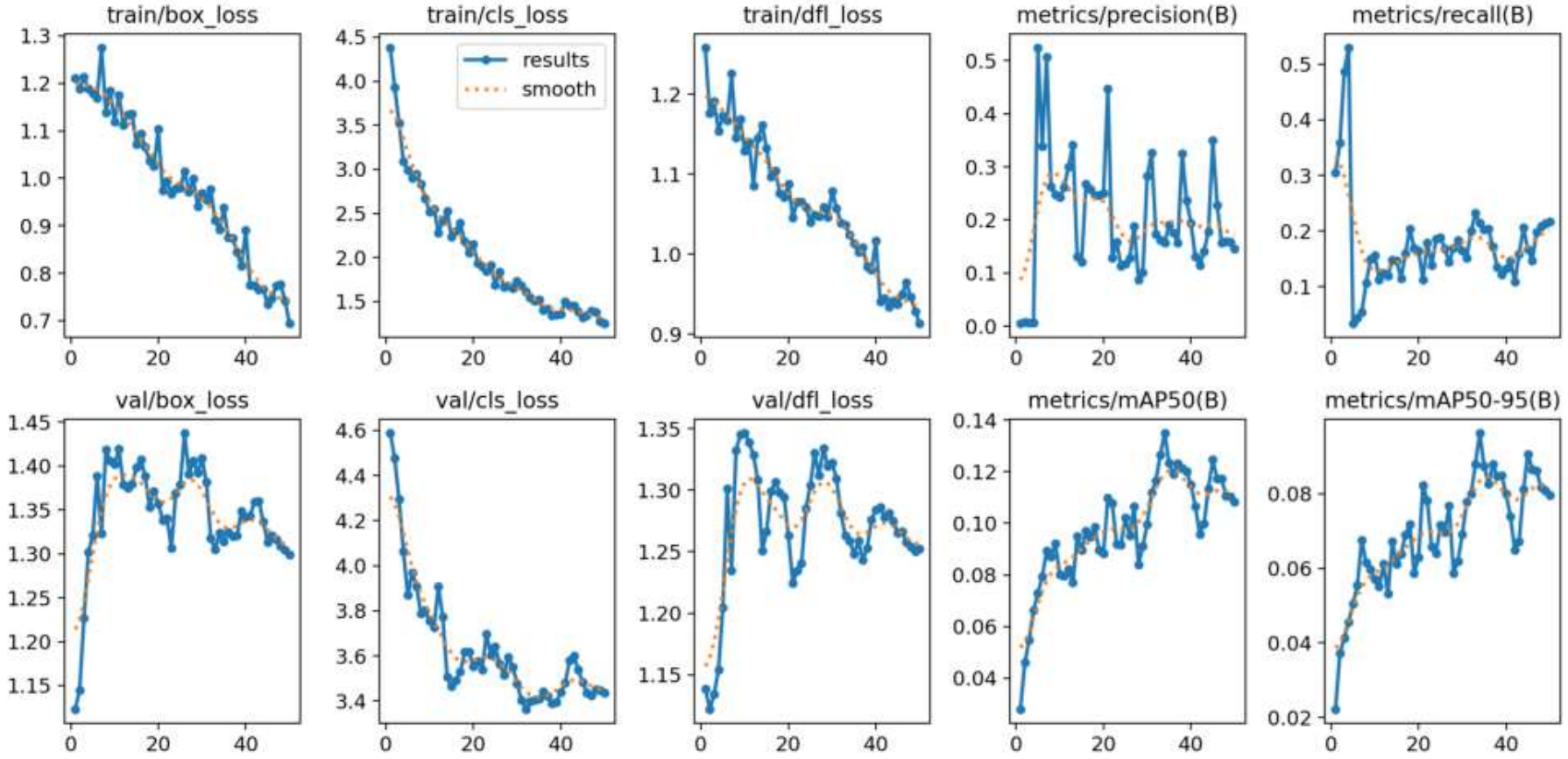
Primer entrenamiento YOLO

- Se utilizó YOLO nano
- En el primer experimento, se entrenó desde cero, sin utilizar pesos pre entrenados.
  Esto permitió evaluar la capacidad del modelo para aprender directamente desde el dataset TACO.
- Learning rate de 0.01


Evolución del entrenamiento (Losses y Métricas)
train/box_loss
train/cls_loss
results
smooth
train/dfl_loss
metrics/precision(B)
metrics/recall(B)
val/box_loss
val/cls_loss
val/dfl_loss
metrics/mAP50(B)
metrics/mAP50-95(B)

Segundo entrenamiento YOLO

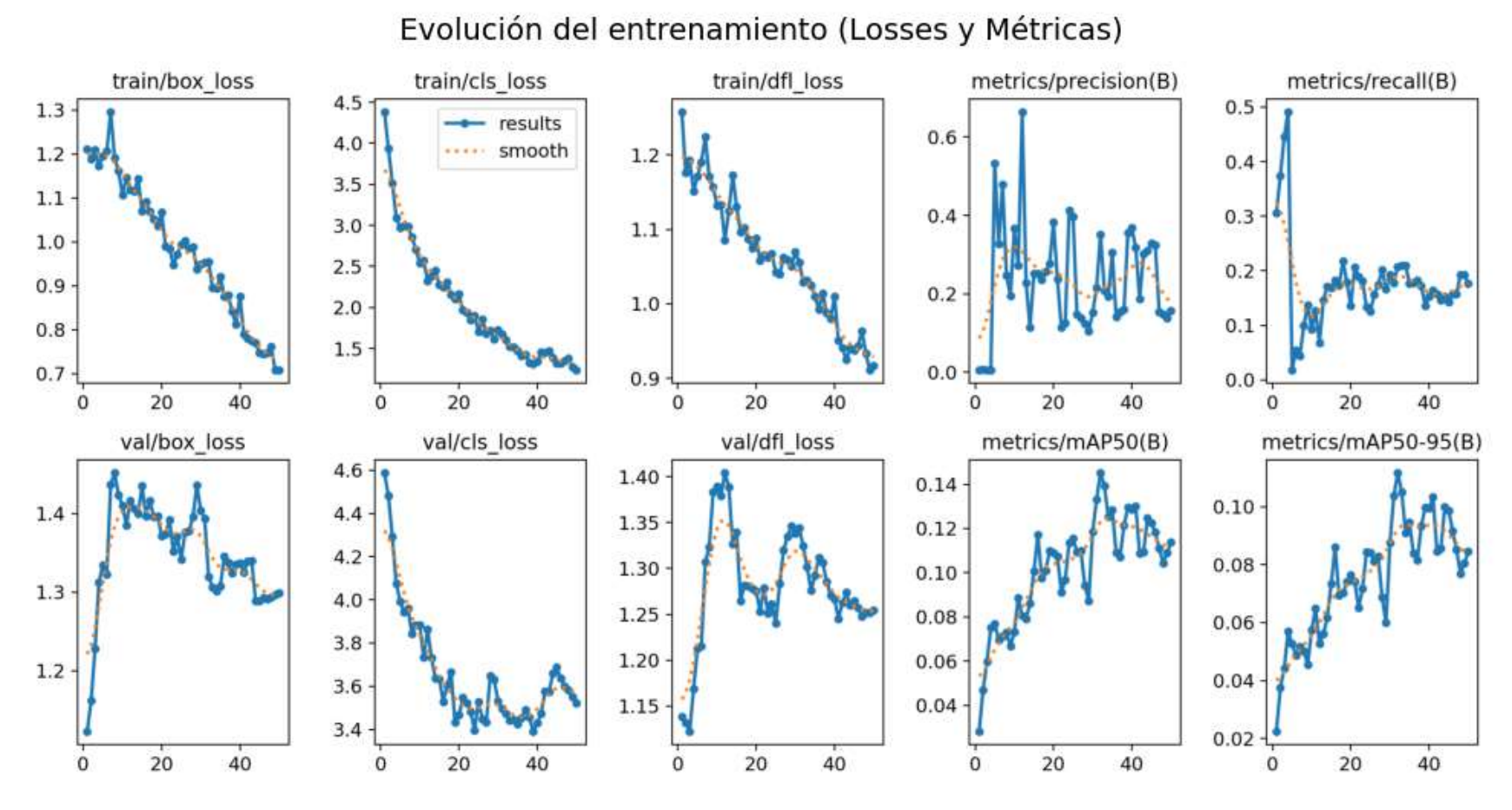
- Se utilizaron pesos pre entrenados en COCO dataset.


Evolución del entrenamiento (Losses y Métricas)
train/box_loss
train/cls_loss
results
smooth
train/dfl_loss
metrics/precision(B)
metrics/recall(B)
val/box_loss
val/cls_loss
val/dfl_loss
metrics/mAP50(B)
metrics/mAP50-95(B)

Tercer entrenamiento YOLO

- En este experimento se utilizó Transfer Learning y se ajustó el parámetro de learning rate a 0.003.

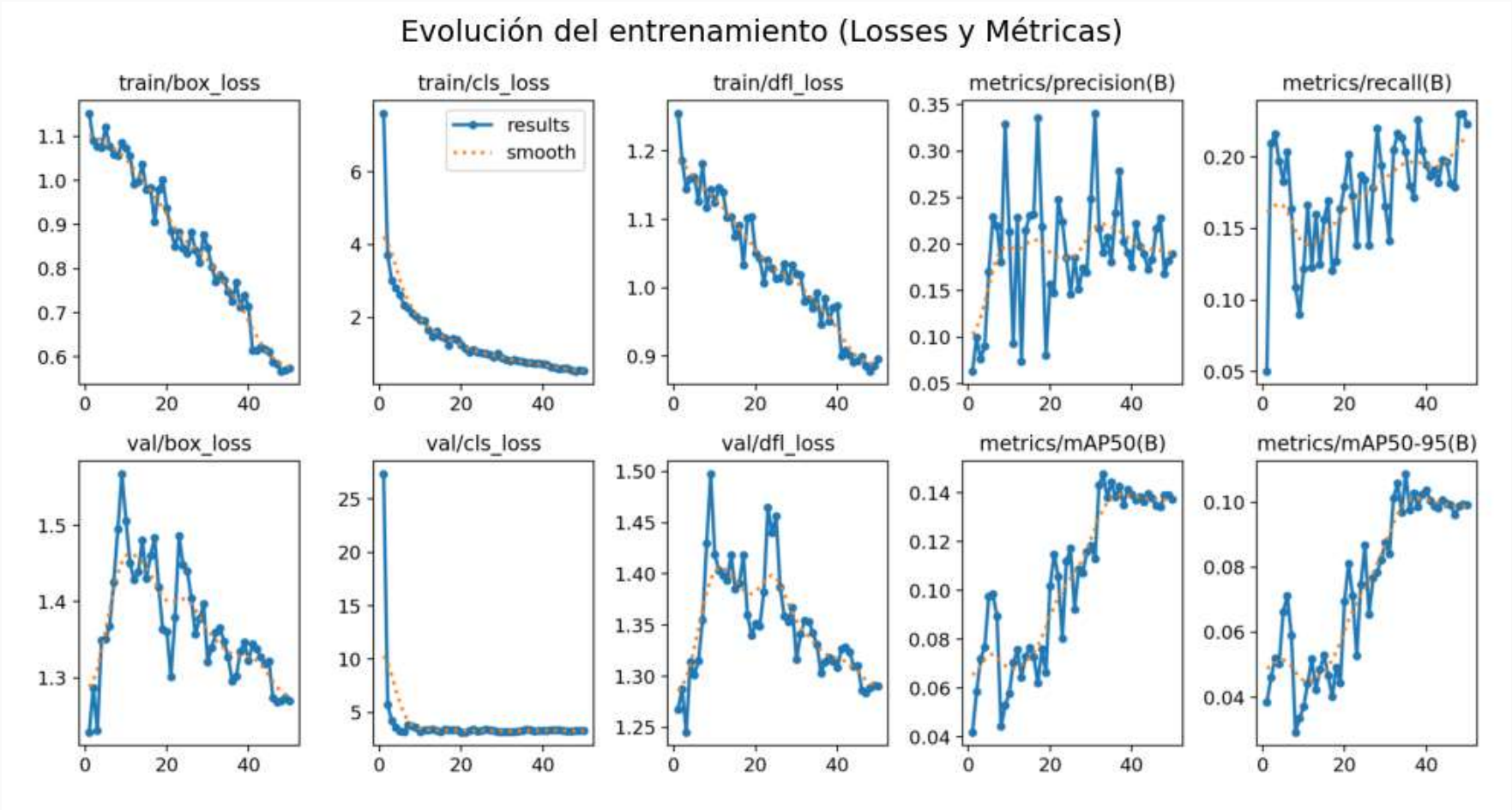
- El objetivo fue mejorar la estabilidad y precisión del modelo afinando la velocidad de aprendizaje.



Tercer entrenamiento

Cuarto entrenamiento YOLO

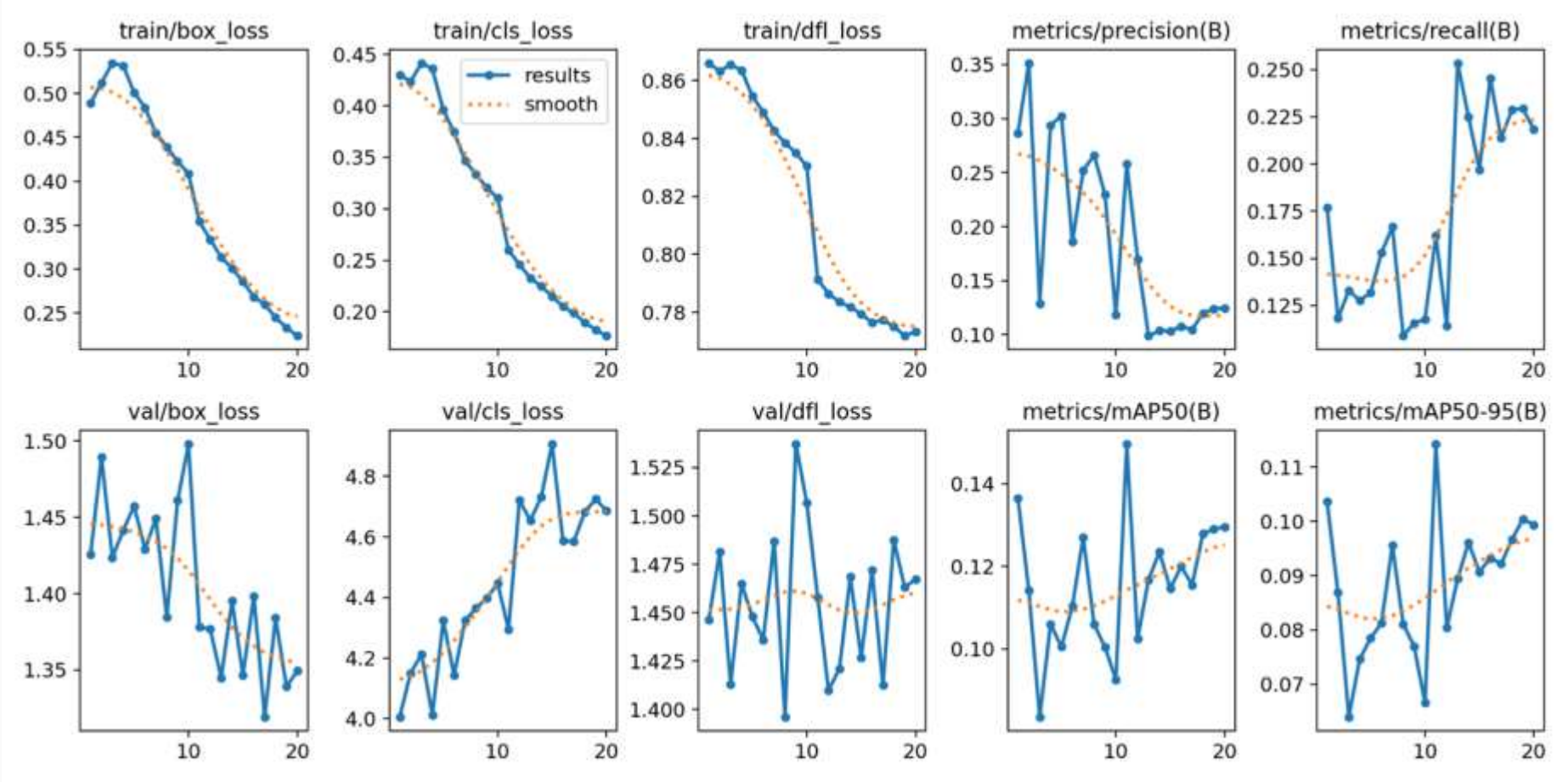
- En los entrenamientos anteriores se utilizó el modelo YOLOv8 Nano, pensada para ser extremadamente liviana pero menos precisa.
- En este cuarto experimento se cambió la arquitectura a YOLOv8 Small, que ofrece un balance entre rendimiento y velocidad.
- Se mantuvo el uso de Transfer Learning y el learning rate de 0.003.
- El modelo tardó más en entrenar que Nano, pero con mejor capacidad de aprendizaje.



Tercer entrenamiento

Quinto entrenamiento con YOLO

- En este experimento se utilizó YOLOv8 Small con Transfer Learning y learning rate de 0.003, como en el entrenamiento anterior.
- Se aplicó oversampling sobre clases poco representadas para compensar el desbalance del dataset.
- El objetivo fue mejorar el rendimiento del modelo sobre categorías minoritarias, sin afectar la precisión en clases frecuentes.
- El oversampling consistió en replicar imágenes de clases minoritarias en la carpeta de entrenamiento.


train/box_loss
train/cls_loss
results
smooth
train/dfl_loss
metrics/precision(B)
metrics/recall(B)
val/box_loss
val/cls_loss
val/dfl_loss
metrics/mAP50(B)
metrics/mAP50-95(B)

Decisión: aplicar Data Augmentation

En el este entrenamiento se aplicó oversampling para equilibrar la distribución de clases.

Aunque mejoró el recall y el mAP promedio, se observaron señales de sobreajuste:

- Pérdidas de validación inestables.
- Métricas de precisión decrecientes.
- Alta variabilidad entre épocas.

Para resolver este sobreajuste y aumentar la diversidad del conjunto de entrenamiento, se decidió implementar data augmentation:

Sexto entrenamiento con YOLO

- En este experimento se utilizó YOLOv8 Small con Transfer Learning y learning rate de 0.003.

- Se incorporó data augmentation para aumentar la variabilidad del conjunto de entrenamiento y reducir el riesgo de sobreajuste.

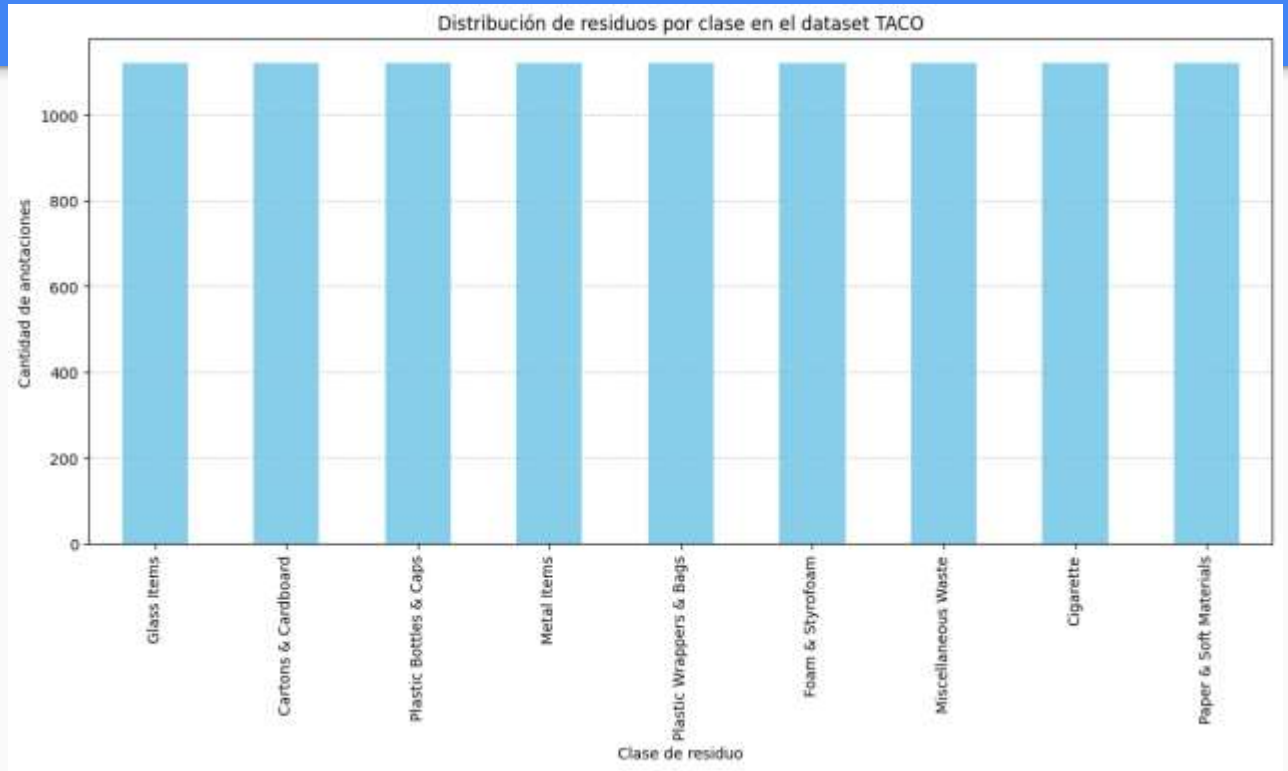
Data Augmentation

```
single_bbox_ids = new_annotation_df["image_id"].value_counts()
single_bbox_ids = single_bbox_ids[single_bbox_ids == 1].index
filtered_annotations = new_annotation_df[new_annotation_df["image_id"].isin(single_bbox_ids)]
```

- Se utilizó la biblioteca Albumentations para aplicar transformaciones aleatorias sobre las imágenes y sus respectivas bounding boxes.
- Las transformaciones incluyeron:
  - Volteo horizontal y vertical, con probabilidad del 50% cada uno.
  - Corte aleatorio redimensionado, para simular distintos encuadres y escalas.

```
transform = A.Compose([
    A.HorizontalFlip(p=0.5),    # Flip horizontal
    A.VerticalFlip(p=0.5),      # Flip vertical
    A.RandomResizedCrop(size=(224,224), scale=(0.8, 1.0), p=0.5),  # Cropping
], bbox_params=A.BboxParams(format="pascal_voc", label_fields=["category"]))
```

Distribución del dataset post-data augmentation

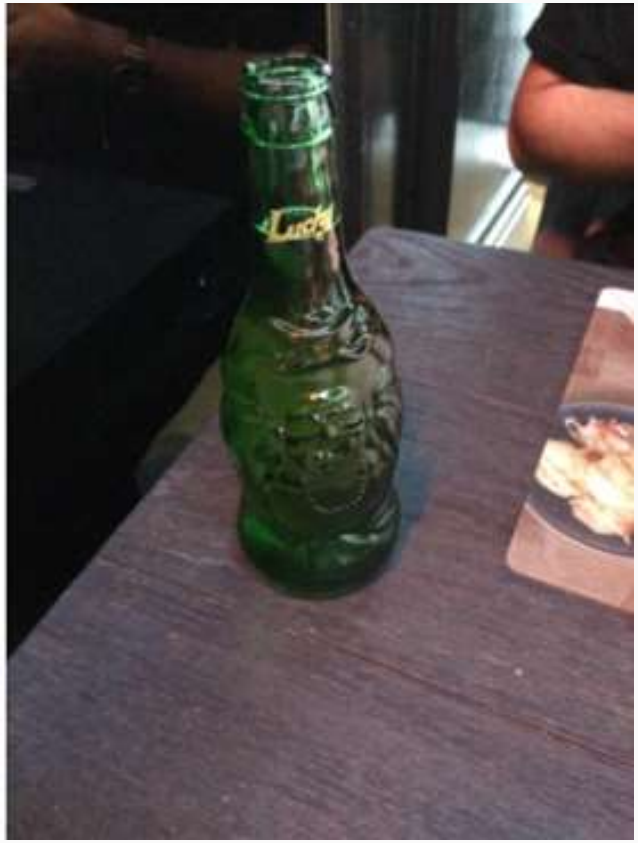


Dataframe post-data augmentation

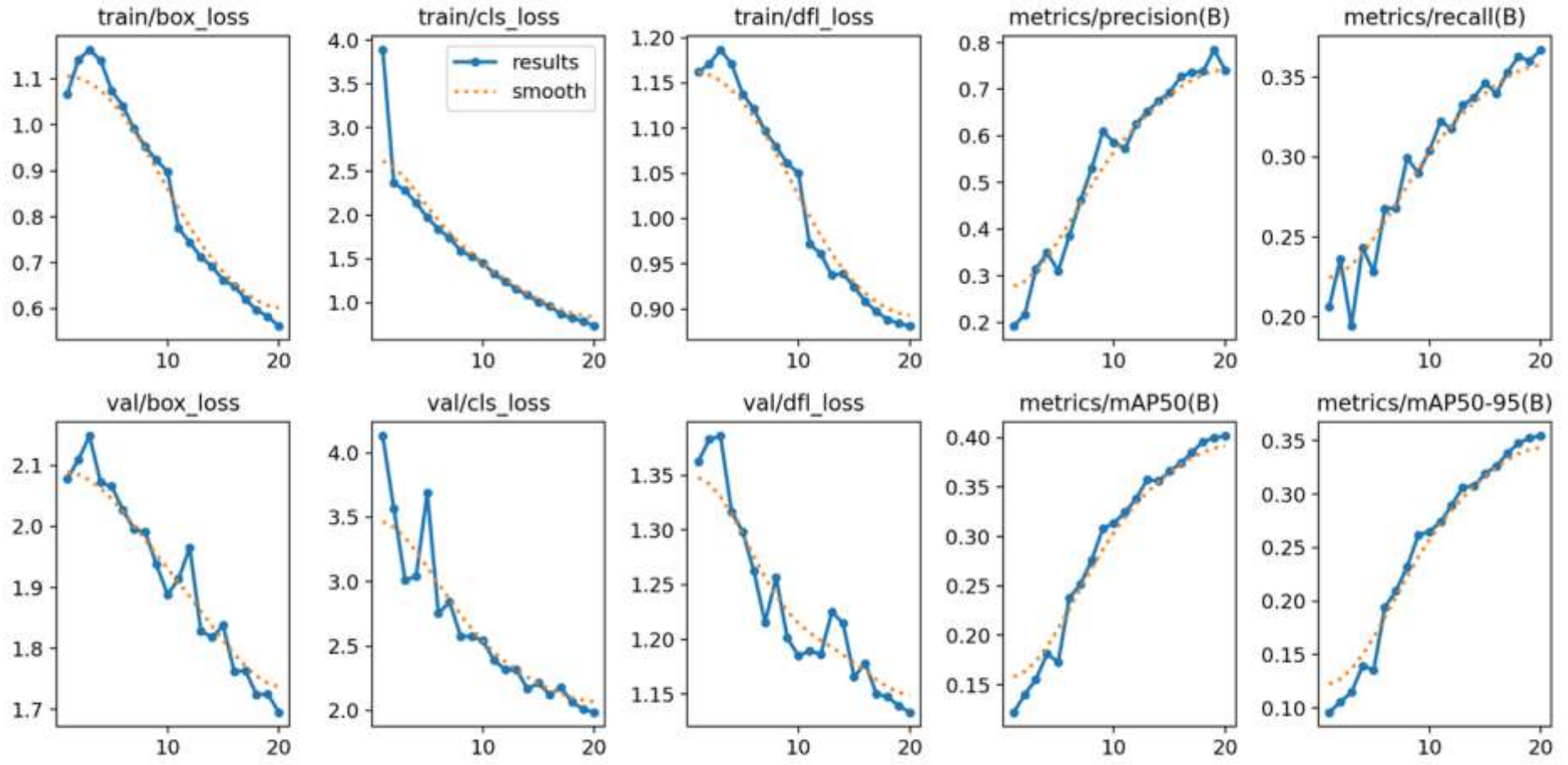
	id	image_id	category_id	segmentation	area	bbox	iscrowd	category_name	new_category	augmented
0	1	0	6	[[561.0, 1238.0, 568.0, 1201.0, 567.0, 1175.0,...	403954.0	[517.0, 127.0, 447.0, 1322.0]	0	Glass bottle	Glass Items	NaN
1	2	1	18	[[928.0, 1876.0, 938.0, 1856.0, 968.0, 1826.0,...	1071259.5	[1.0, 457.0, 1429.0, 1519.0]	0	Meal carton	Cartons & Cardboard	NaN
2	3	1	14	[[617.0, 383.0, 703.0, 437.0, 713.0, 456.0, 72...	99583.5	[531.0, 292.0, 1006.0, 672.0]	0	Other carton	Cartons & Cardboard	NaN
3	4	2	5	[[670.0, 993.0, 679.0, 998.0, 684.0, 1001.0, 6...	73832.5	[632.0, 987.0, 500.0, 374.0]	0	Clear plastic bottle	Plastic Bottles & Caps	NaN
4	5	2	7	[[647.0, 1028.0, 650.0, 1022.0, 653.0, 1016.0,...	915.0	[632.0, 989.0, 44.0, 51.0]	0	Plastic bottle cap	Plastic Bottles & Caps	NaN
...	...	...	...	...	...	...	...	...	...	...
10083	2225	aug_881_695	18	[[1163.0, 1136.0, 1236.0, 1112.0, 1270.0, 1097...	43125.0	[1134.9999961853027, 1870.9999523162842, 253.0...	0	Meal carton	Cartons & Cardboard	True
10084	2227	aug_882_697	14	[[1239.0, 1888.0, 1291.0, 1917.0, 1296.0, 1923...	98012.0	[1239.0000629425049, 1220.0000610351562, 478.9...	0	Other carton	Cartons & Cardboard	True
10085	2331	aug_883_746	14	[[1148.0, 1321.0, 1276.0, 1275.0, 1324.0, 1258...	199257.0	[73.91176509857178, 101.11111259460449, 47.970...	0	Other carton	Cartons & Cardboard	True
10086	2359	aug_884_759	14	[[1427.0, 1569.0, 1400.0, 1626.0, 1393.0, 1656...	48364.0	[1020.9999847412109, 1527.0000457763672, 311.9...	0	Other carton	Cartons & Cardboard	True
10087	2387	aug_885_774	18	[[1780.0, 933.0, 1794.0, 922.0, 1818.0, 908.0,...	57548.0	[160.2125376522813, 166.73446478547348, 28.267...	0	Meal carton	Cartons & Cardboard	True

10088 rows × 10 columns

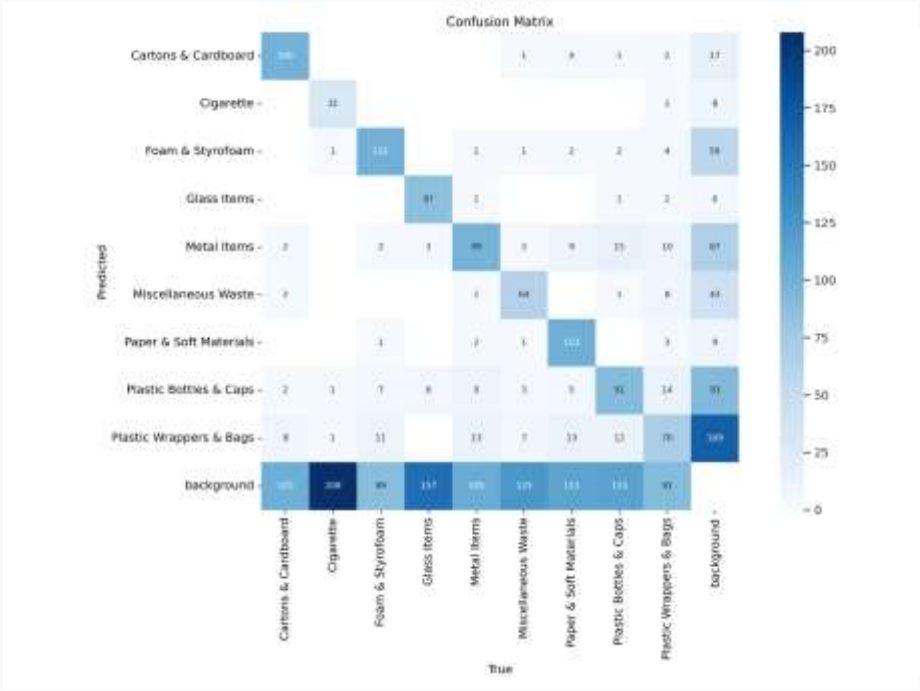
Imagenes Data Augmentation

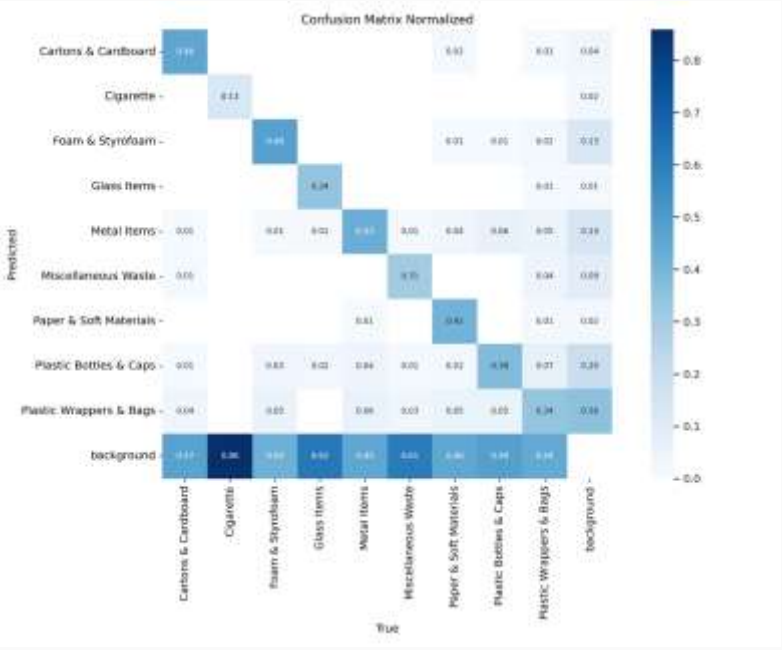


Imagenes Data Augmentation

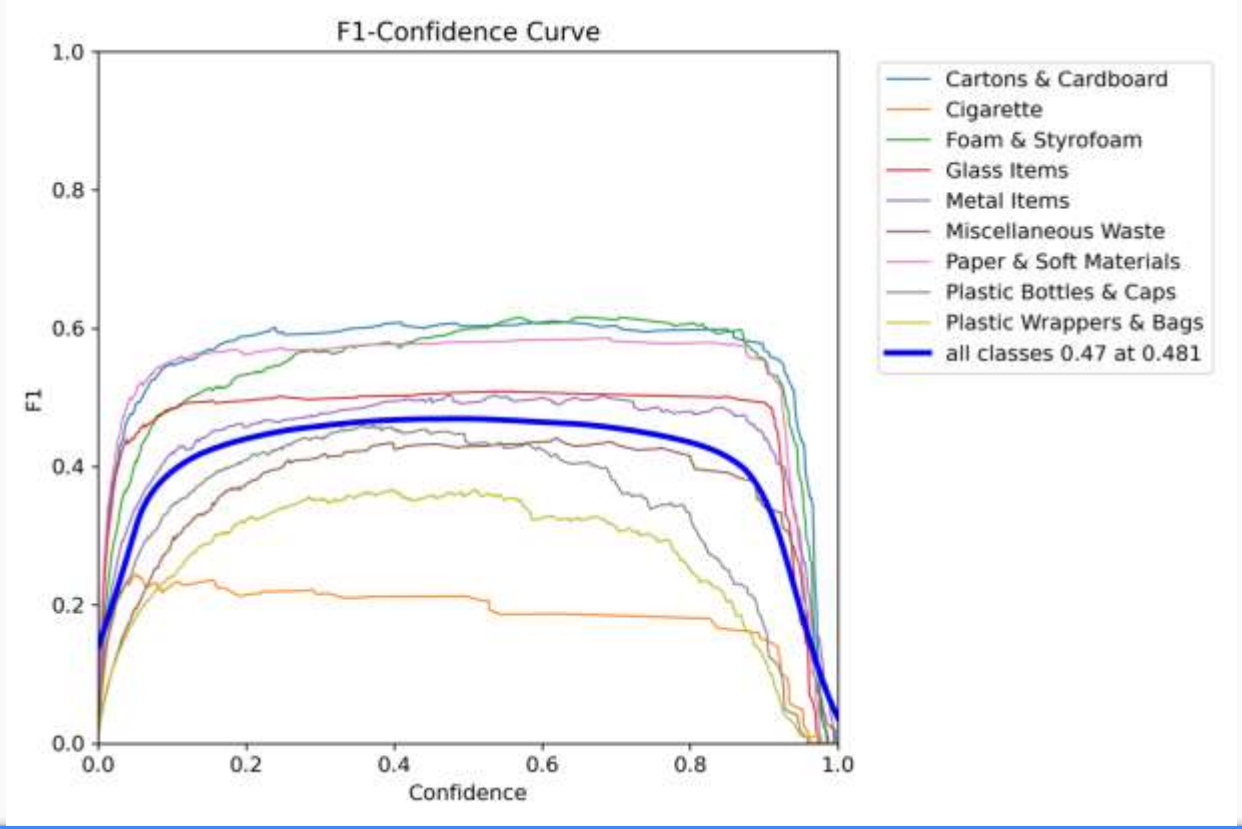


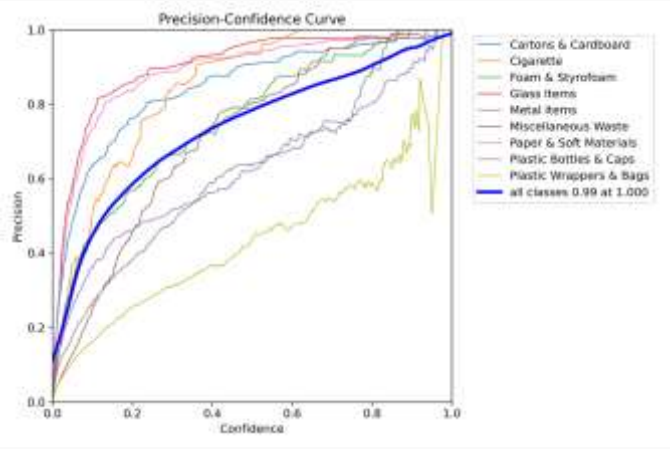
Data Augmentation

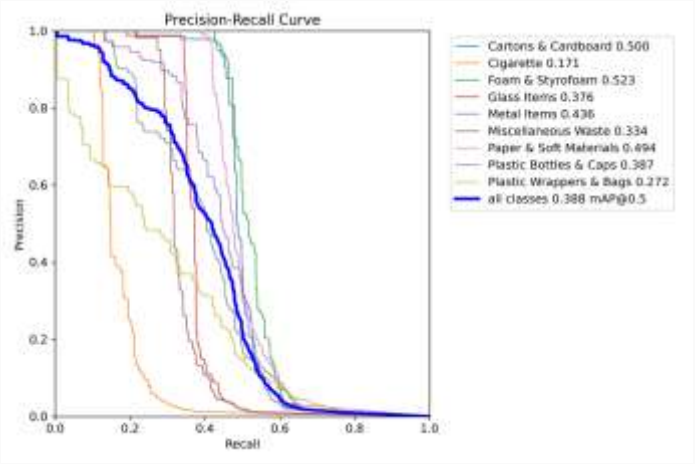

Confusion Matrix
Cartons & Cardboard
Cigarette
Foam & Styrofoam
Glass Items
Metal Items
Miscellaneous Waste
Paper & Soft Materials
Plastic Bottles & Caps
Plastic Wrappers & Bags
background
Predicted
True
200
175
150
125
100
75
50
25
0

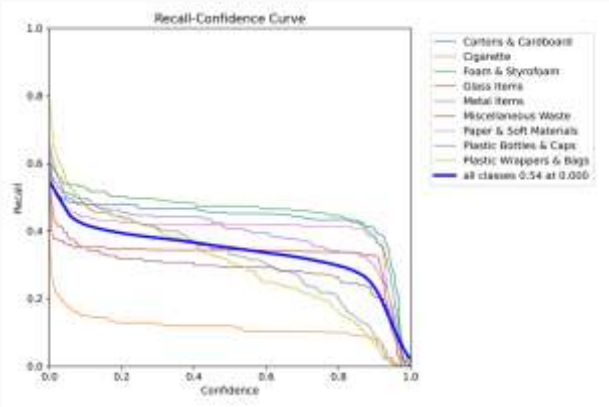

Confusion Matrix Normalized
Cartons & Cardboard
Cigarette
Foam & Styrofoam
Glass Items
Metal Items
Miscellaneous Waste
Paper & Soft Materials
Plastic Bottles & Caps
Plastic Wrappers & Bags
background
Predicted
True
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

Aún así, le falta fine tuning y/o muchas épocas de entrenamiento ya que el valor máximo de F1 Score = 0.47 con 0.481 de confidence threshold, lo cual no es adecuado para ambientes productivos.










```
Ultralytics 8.3.113 🚀 Python-3.11.12 torch-2.6.0+cu124 CPU (Intel Xeon 2.20GHz)
Model summary (fused): 72 layers, 11,129,067 parameters, 0 gradients, 28.5 GFLOPs
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultralytics/Arial.ttf'...
100%|██████████| 755k/755k [00:00<00:00, 79.6MB/s]val: Fast image access ✅ (ping: 0.0±0.0 ms, read: 1419.4±867.1 MB/s, si:

val: Scanning /content/taco_yolo_format/labels/val... 1361 images, 0 backgrounds, 0 corrupt: 100%|██████████| 1361/1361 [00

                   Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 86/86 [17:42<00:
                     all       1361       2050      0.778      0.353      0.388      0.343
   Cartons & Cardboard        213        217      0.907      0.456        0.5      0.474
             Cigarette        130        242      0.932       0.12      0.171      0.142
      Foam & Styrofoam        202        211      0.801      0.474      0.523      0.478
           Glass Items        189        253      0.962      0.344      0.376      0.355
           Metal Items        173        231      0.618      0.411      0.436      0.387
   Miscellaneous Waste        179        205       0.79      0.294      0.334      0.295
Paper & Soft Materials        233        248      0.939      0.419      0.494      0.471
Plastic Bottles & Caps        159        238      0.636      0.345      0.387      0.296
Plastic Wrappers & Bags       128        205      0.415      0.317      0.272      0.186
Speed: 5.5ms preprocess, 720.0ms inference, 0.0ms loss, 0.7ms postprocess per image
```

Ejemplos de los resultados


aug_382_batch_10_000017.jpg
Miscellaneous Waste
aug_4_batch_10_000052.jpg
Miscellaneous Waste
batch_14_000048.jpg
Plastic Bottles & Caps
Plastic Wrappers & Bags
aug_219_batch_6_000088.JPG
Metal Items
aug_100_batch_10_000064.jpg
Cigarette
aug_171_batch_10_000061.jpg
Metal Items
aug_558_bat
Metal Items
4.jpg
aug_312_batch_11_000025.jpg
Miscellaneous Waste
aug_452_batch_10_000064.jpg
Cigarette
aug_624_batch_12_000016.jpg
Foam & Styrofoam
batch_1_000004
Metal Items
Miscellaneous Waste
batch_9_000077.jpg
Plastic Bottles & Caps
aug_12_batch_12_000016.jpg
Foam & Styrofoam
aug_63_batch_6_000088.JPG
Metal Items
Plastic Bottles & Caps
Plastic Bottles & Caps

Ejemplo Data Augmentation

Ejemplo Data Augmentation

Ejemplo Data Augmentation

Conclusión

Conclusión

Se abordó la tarea de detección de objetos sobre el dataset TACO Trash, aplicando distintas arquitecturas, técnicas y estrategias de entrenamiento.

- YOLOv8 ofreció un excelente balance entre precisión y eficiencia, lo que permitió iterar más rápido.

Se exploraron distintas estrategias:

- Transfer Learning (mejoró sustancialmente las métricas).
- Ajuste del learning rate (mejor estabilidad).
- Cambio de arquitectura (de Nano a Small, mejorando la capacidad del modelo).
- Oversampling (ayudó, pero generó signos de overfitting).
- Data augmentation (implementado para mejorar generalización).

¡Gracias!