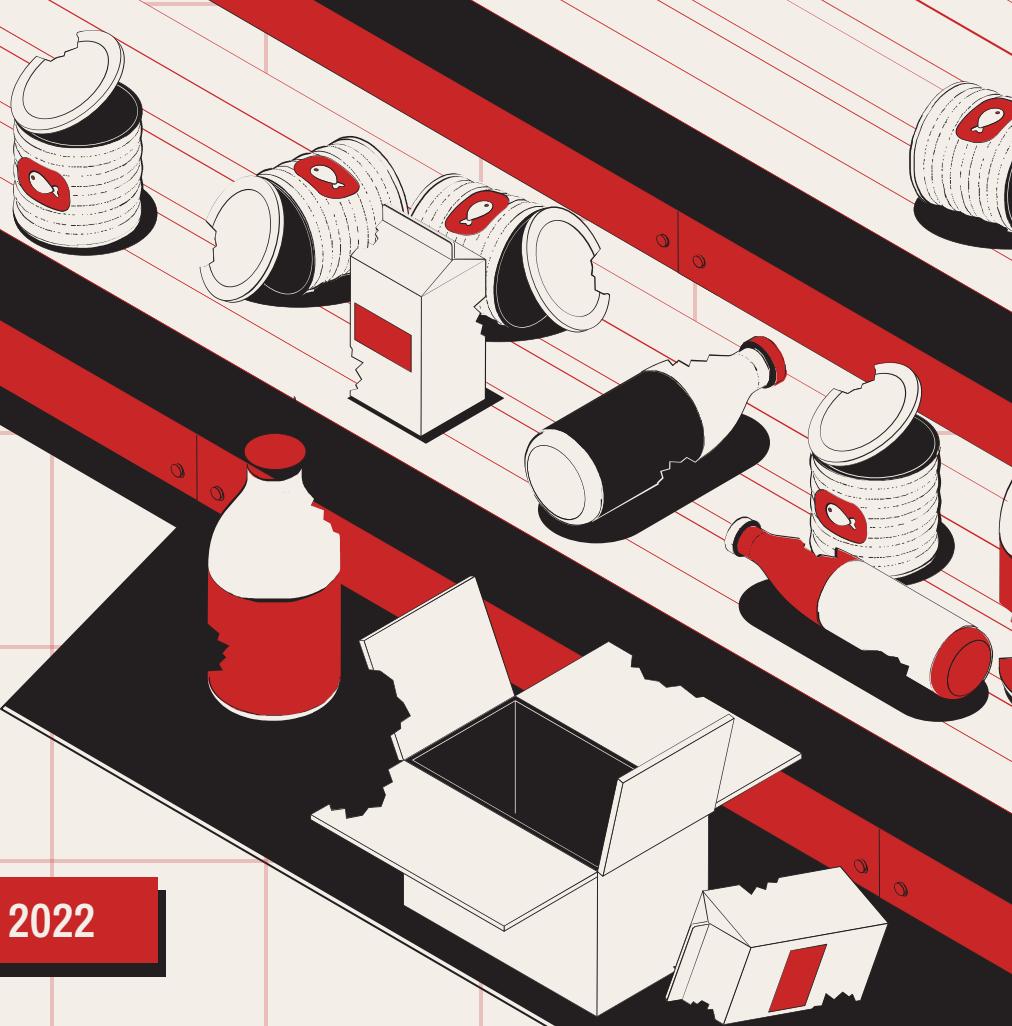


# CV para Detección de residuos

Lucas Rivela - Ariadna Garmendia | Oct. 2022



# Etapas del trabajo

**01**

**Selección del  
Dataset**

Selección del dataset y problema a resolver. Análisis exploratorio

**02**

**Identificación y armado  
de modelos**

Selección de una y dos etapas. Armado del código. Identificación de la plataforma para el entrenamiento.

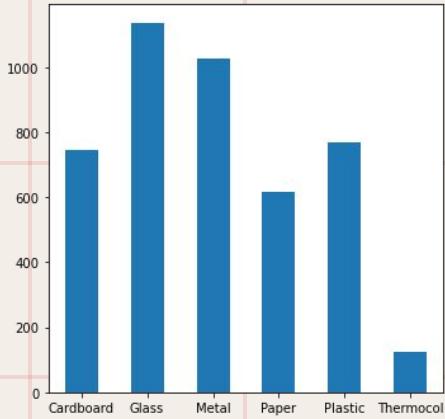
**03**

**Entrenamiento y  
evaluación**

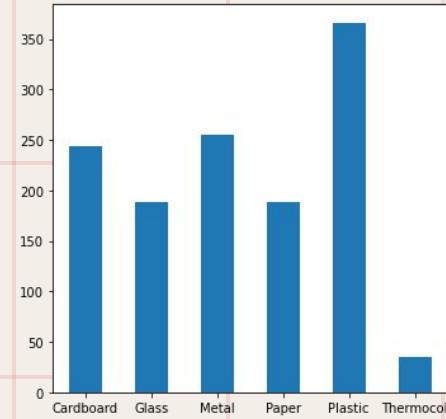
Entrenamiento, inferencia, desarrollo y evaluación de métricas. Iteración. Evaluación de resultados.



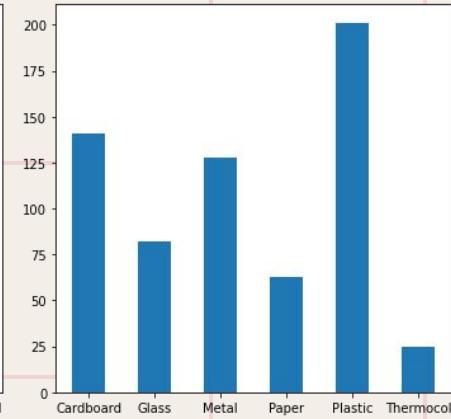
Distr. de clases en el dataset de Train



Distr. de clases en el dataset de Valid



Distr. de clases en el dataset de Test



# Dataset

Tamaño de Imágenes: 640 x 640

Train: 4076 | Validación: 1173 | Test: 568

Total: 5817

6 Clases: Cardboard, Glass, Metal, Paper, Plastic, Thermocol

Roboflow download:

<https://universe.roboflow.com/dark-mqa4m/waste-segregation-3ykjs>



# Redes utilizadas

01

## Faster R-CNN

Transfer learning (default freeze)  
Augmentations (in-line)

02

## YOLOv7

Fine tuning.  
Con y sin Augmentations (offline)

03

## SSD

Transfer learning (default freeze)  
Sin Augmentations

04

## RetinaNet

Transfer learning c/ Backbone  
freeze. Sin Augmentations

# Métricas de performance

	Faster R-CNN Augmented (inline)	YOLOv7 Augmented (offline)	YOLOv7	RetinaNet	SSD
<b>Tiempo de entrenamiento (*)</b>	14min/epoch	25 hs (25 min/epoch) (60 epochs)	25hs	3 hs (19 min/epoch 10 epochs)	2.5 hs (3.5 min/epoch) 40 epochs
<b>Tiempo de Inferencia (568 imágenes)</b>	203 seg	38.4 seg	23 seg	279 seg	81 seg
<b>mAP_50</b>	0.406	0.476	0.697	0.136	0.261
<b>mAP_75</b>	0.329	-	-	0.134	0.221
<b>Avg. Precision</b>	0.595	0.508	0.698	0.178	0.435
<b>Avg. Recall</b>	0.549	0.442	0.636	0.150	0.367

(\*) Con todo el backbone freezado. Batch size = 8 (8 GB GPU)



# mAP por clase

	Faster R-CNN	YOLOv7 Augmented	YOLOv7	RetinaNet	SSD
<b>1: Cardboard</b>	0.351	0.464	0.764	0.1286	0.207
<b>2: Glass</b>	0.383	0.623	0.786	0.097	0.171
<b>3: Metal</b>	0.464	0.656	0.881	0.122	0.289
<b>4: Paper</b>	0.419	0.589	0.784	0.361	0.423
<b>5: Plastic</b>	0.153	0.41	0.552	0.017	0.022
<b>6: Thermocol</b>	0	0.112	0.417	0	0



(\*) Con todo el backbone freezado. Batch size = 8 (8 GB GPU)

(\*) IoU threshold=0.5

# Comparación Faster R CNN

Faster R-CNN	Augmented Default Freeze	Augmented Complete Freeze	Augmented No Freeze	No Aug Default Freeze
mAP _50	0.406	0.349	0.415	0.402
mAP _75	0.329	0.291	0.338	0.337
Avg. Precision	0.595	0.525	0.652	0.632
Avg. Recall	0.549	0.487	0.620	0.600

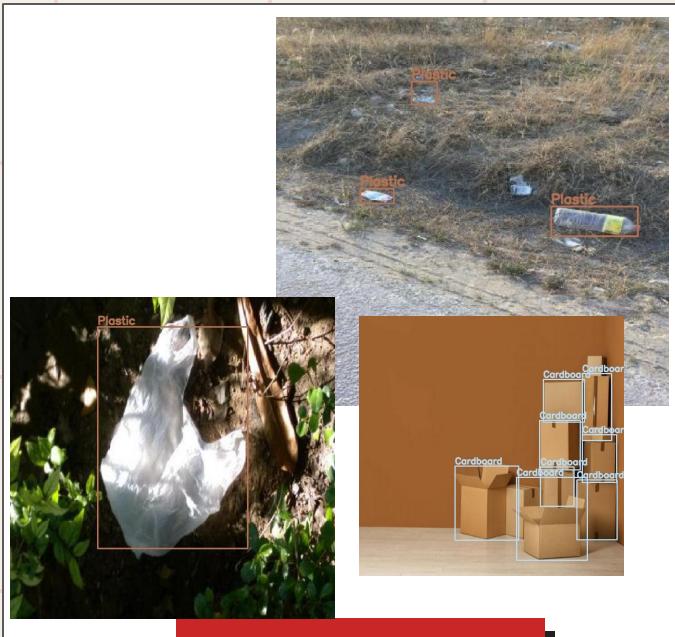


# Comparación SSD

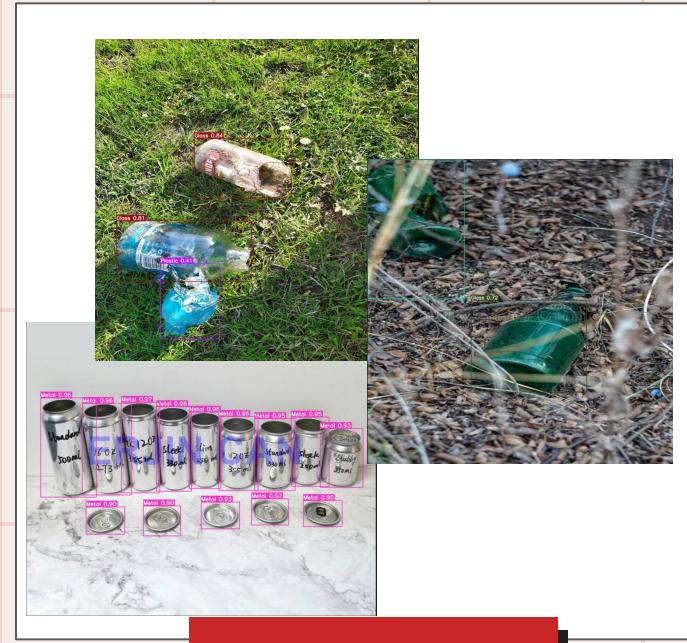
SSD	Non-Augmented Default Freeze (trainable bb layers=4)	Non-Augmented Complete Freeze
Tiempo de entr. (40 epochs)	3.7 hs (5.6 min/epoch)	2.5 hs (3.5 min/epoch)
Tiempo de inferencia (568 im.)	81 seg	
mAP _50	0.280	0.261
mAP _75	0.222	0.221
Avg. Precision	0.457	0.435
Avg. Recall	0.395	0.367



# Detecciones

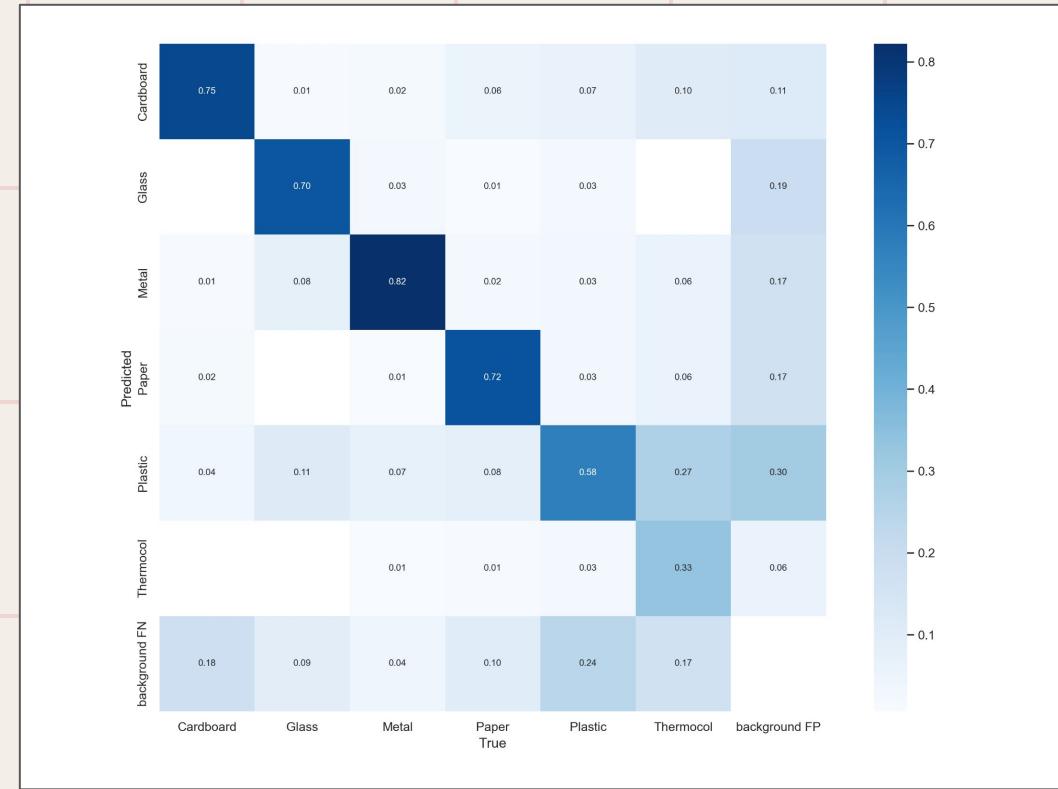


Faster R-CNN



YOLO

# Matriz Confusión YoloV7 (Sin Aug)



# Detecciones



# Conclusiones

- Yolo presentó la mejor performance y parece ser menos sensible al desbalance de clases (es la única que detectó la clase minoritaria)
- SSD fue alrededor de 4 veces más rápida al momento del entrenamiento, pero no mostró buena performance (overfitting)
- Faster R-CNN presenta bounding boxes notablemente precisos.
- En la inferencia, las redes de una etapa(\*) completaron la tarea entre 3 y 5 veces más rápido que Faster R-CNN
- Augmentations (online con Albumentations y offline via Roboflow) no produjeron buenos resultados, posiblemente porque las transformaciones no se ajustan bien a las utilizadas en el modelo pre-entrenado
- Desafíos:
  - Escasa documentación de Pytorch (sobre todo para Transfer Learning)
  - Procesamiento requerido de targets y predicciones para la evaluación de métricas
  - Recursos computacionales (GPU) limitados

(\*) Excepto RetinaNet



# Gracias!

## Preguntas?

