

Análisis Comparativo modelo RF Detr

Puntos clave:

Modelo actual:

- Todavía hay algunos errores
- Buena anotación en la mayoría de clases
- Recortes un poco salidos
- El modelo no se equivoca a la hora de identificar clases, o muy poco el problema es en los recortes
- YOLOv11 Nano es el modelo más ligero con solo 0.91M parámetros, ideal para sistemas embebidos en drones.

Modelo RF Detr:

- No puede correr con polígonos (instance segmentation) y todo nuestro proyecto es con polígonos
- Rf detr corre únicamente en proyectos de bounding boxes o object detection
- Rf detr es por naturaleza mas pesado para un embebido o una jetson
- RF Detr supera a YOLOv11 con mas de 60 AP de precisión en coco, YOLOv11 llega de 54 a 57

Conclusion:

Rf detr actualmente solo puede utilizarse en modelos de Bounding boxes u Object detection, por lo que en nuestro proyecto actual (instance segmentation o polígonos), no podremos correr el modelo de rf detr.

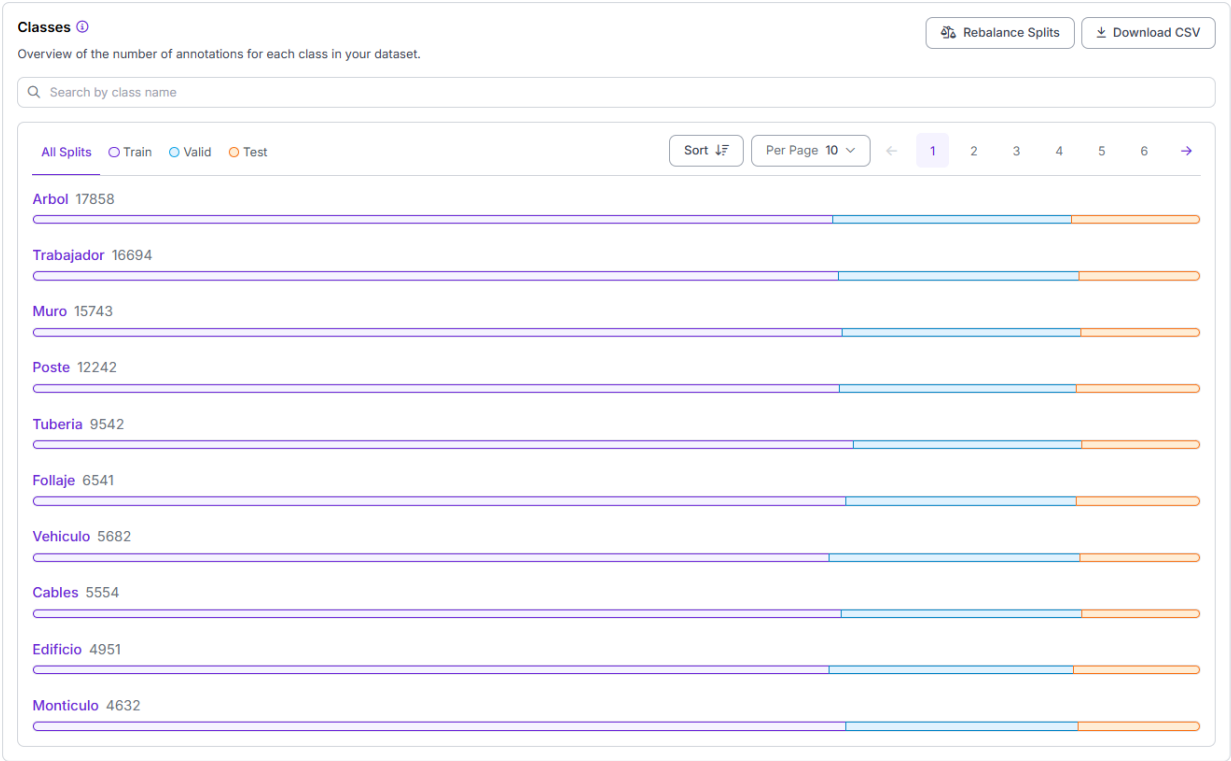
La precisión de rf detr no rebasa mucho a la de yoloV11, por lo que no hay mucha necesidad de actualizar, rf detr es mas pesado para un embebido.

<u>Modelo</u>	<u>Precisión (mAP)</u>	<u>Tamaño del modelo</u>
<u>RF-DETR</u>	60+ AP en COCO	<u>Modelo mas pesado</u>
<u>YOLOv11</u>	54-57 AP en COCO	<u>Modelo más ligero</u>

Modelo	Precisión (mAP)	Velocidad (FPS)	Tamaño del modelo
RF-DETR Base	60+ AP	Inferencia más lenta	29M parámetros
RF-DETR Large	60+ AP	Inferencia más lenta	129M parámetros
YOLOv11 Nano	54-57 AP	Inferencia rápida	0.91M parámetros
YOLOv11 Small	54-57 AP	Inferencia rápida	9M parámetros
YOLOv11 Medium	54-57 AP	Inferencia rápida	25M parámetros
YOLOv11 Large	54-57 AP	Inferencia rápida	54M parámetros

Modelo anterior

Análisis de clases mas utilizadas:



Análisis del dataset:

Dataset Analytics

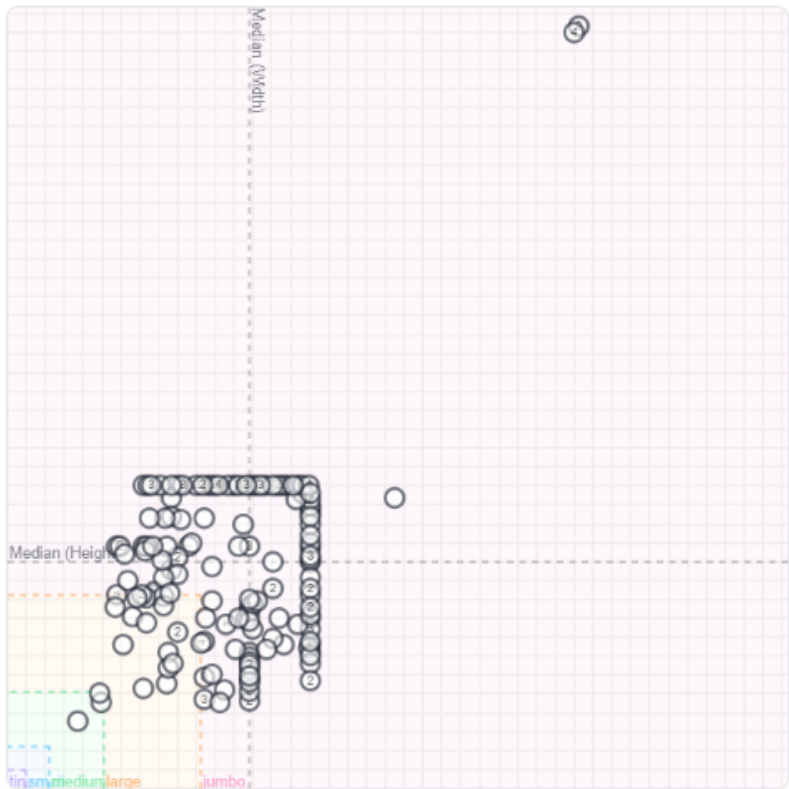
Generated on March 07, 2025 at 1:15 pm
 [Regenerate](#)

<div>Number of Images</div> <div>9793</div> <div> 0 missing annotations 6 null examples </div>	<div>Number of Annotations</div> <div>140757</div> <div> 14.4 per image (average) </> Across 59 classes </div>	<div>Average Image Size</div> <div>1.44 mp</div> <div> from 0.13 mp to 12.19 mp </div>	<div>Median Image Ratio</div> <div>1280x1200</div> <div>4:3 wide</div>
---	---	---	--

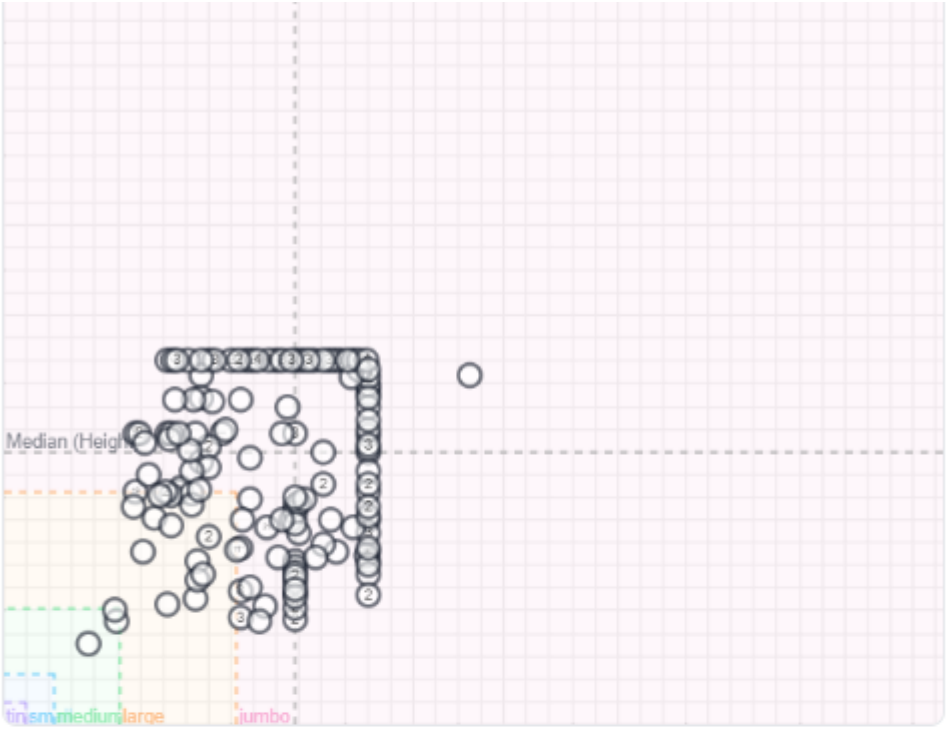
Dimension Insights ⓘ

Overview of the sizes and aspect ratios of the images in your dataset.

The dashed lines represent the **median width** (1280 px) and **median height** (1200 px) of images in your dataset.



Images

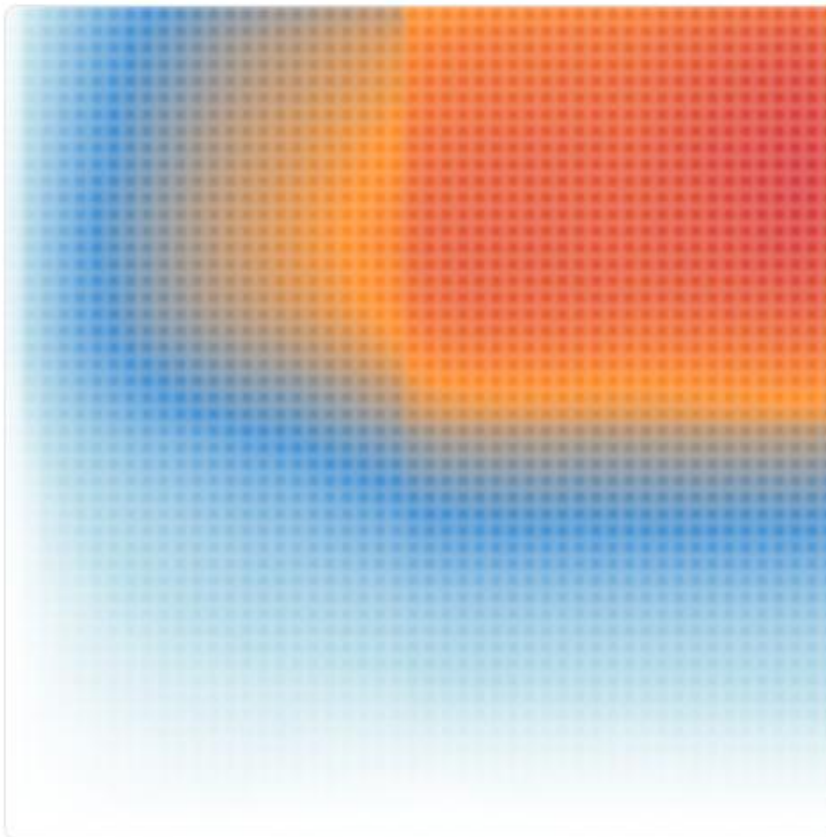


Sizes		9793 total images
jumbo	9653	98.6%
large	137	1.4%
medium	3	0.0%

Aspect Ratios	
wide	4079 41.7%
tall	3733 38.1%
very wide	1934 19.7%
very tall	33 0.3%
extra wide	6 0.1%
square	8 0.1%

Annotation Heat Map ⓘ

Shows you where most of your annotations are. Color gradients signify the number of annotations per grid cell.



Images

Legend (# of Annotations Per Grid) ⓘ

Min (0) Q1 (567) Median (2542) Q3 (6325) Max (9929)

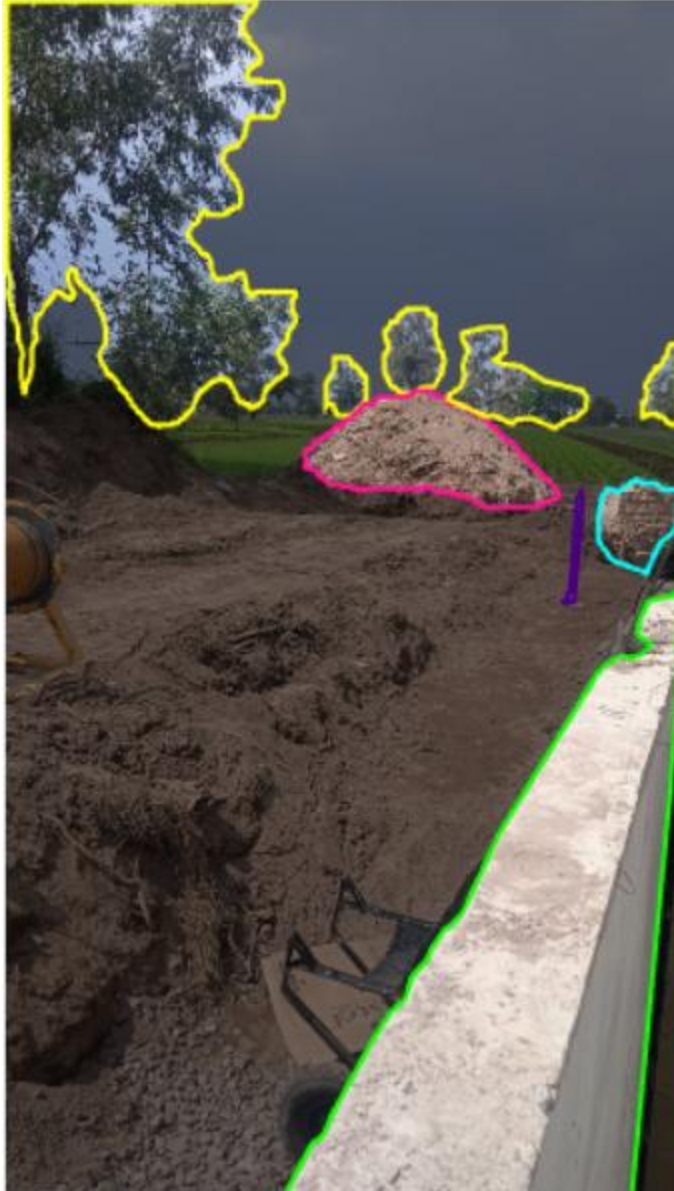


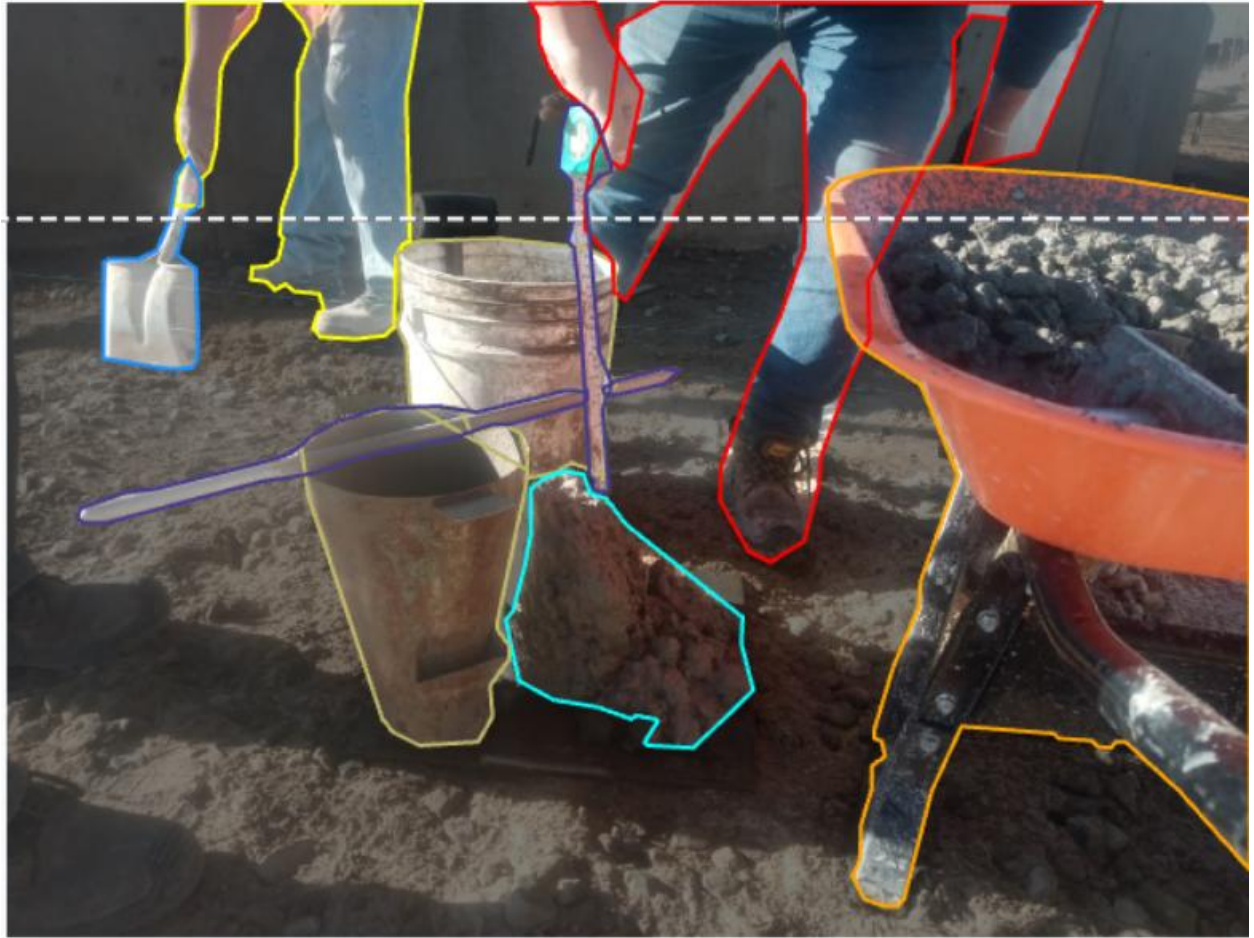
Grid Size 50 × 50 ▾



Show Boxes

Ejemplos de imágenes:



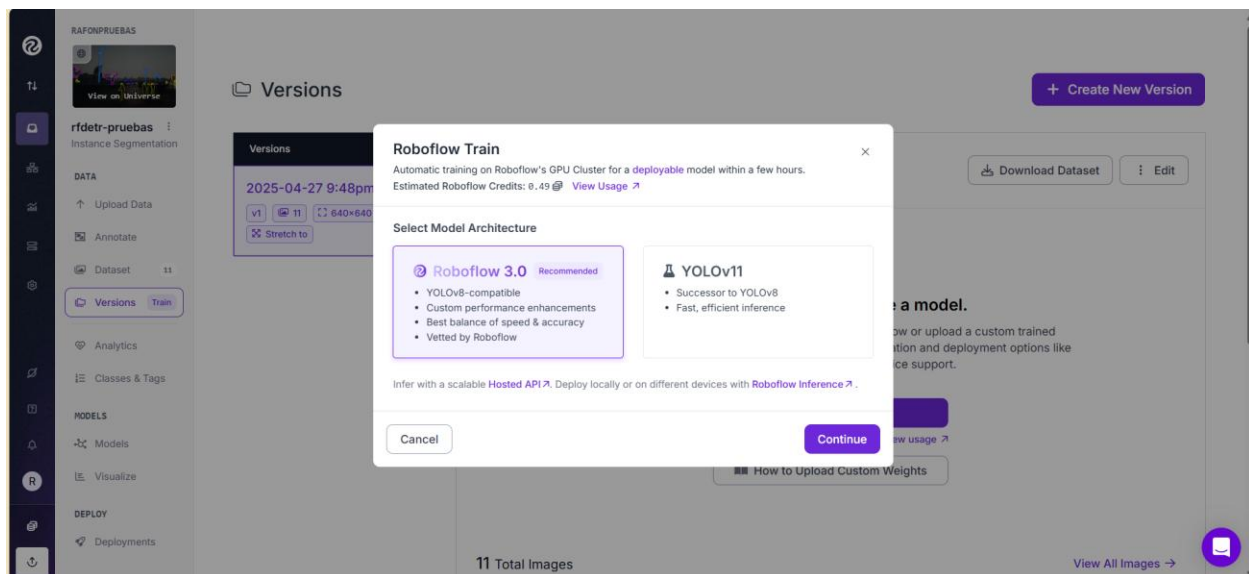




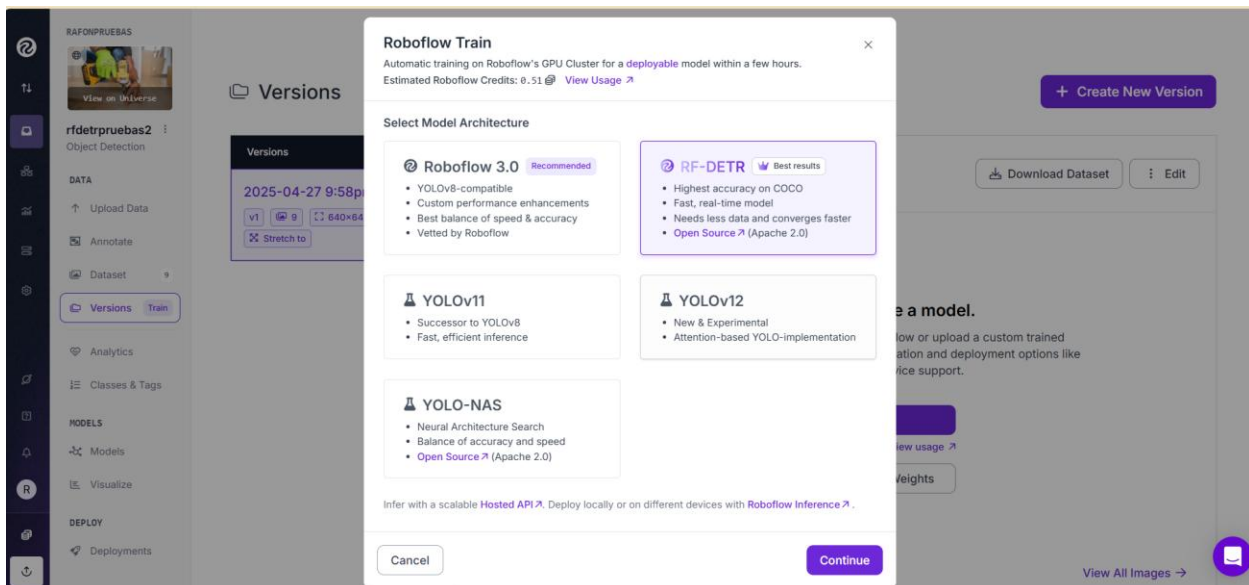
Puntos de feedback:

- Todavía hay algunos errores
- Buena anotación en la mayoría de clases
- Recortes un poco salidos
- El modelo no se equivoca a la hora de identificar clases, o muy poco el problema es en los recortes

Nuevo modelo RF Detr



RF detr no esta disponible para nuestro tipo de proyecto actual mente por el uso de poligonos (Instance segmentation)



Roboflow Train



Automatic training on Roboflow's GPU Cluster for a [deployable](#) model within a few hours.

Estimated Roboflow Credits: 0.51 [View Usage](#)

Select Model Size


When training a model, there is always a trade off between inference speed and accuracy. [Learn More](#)

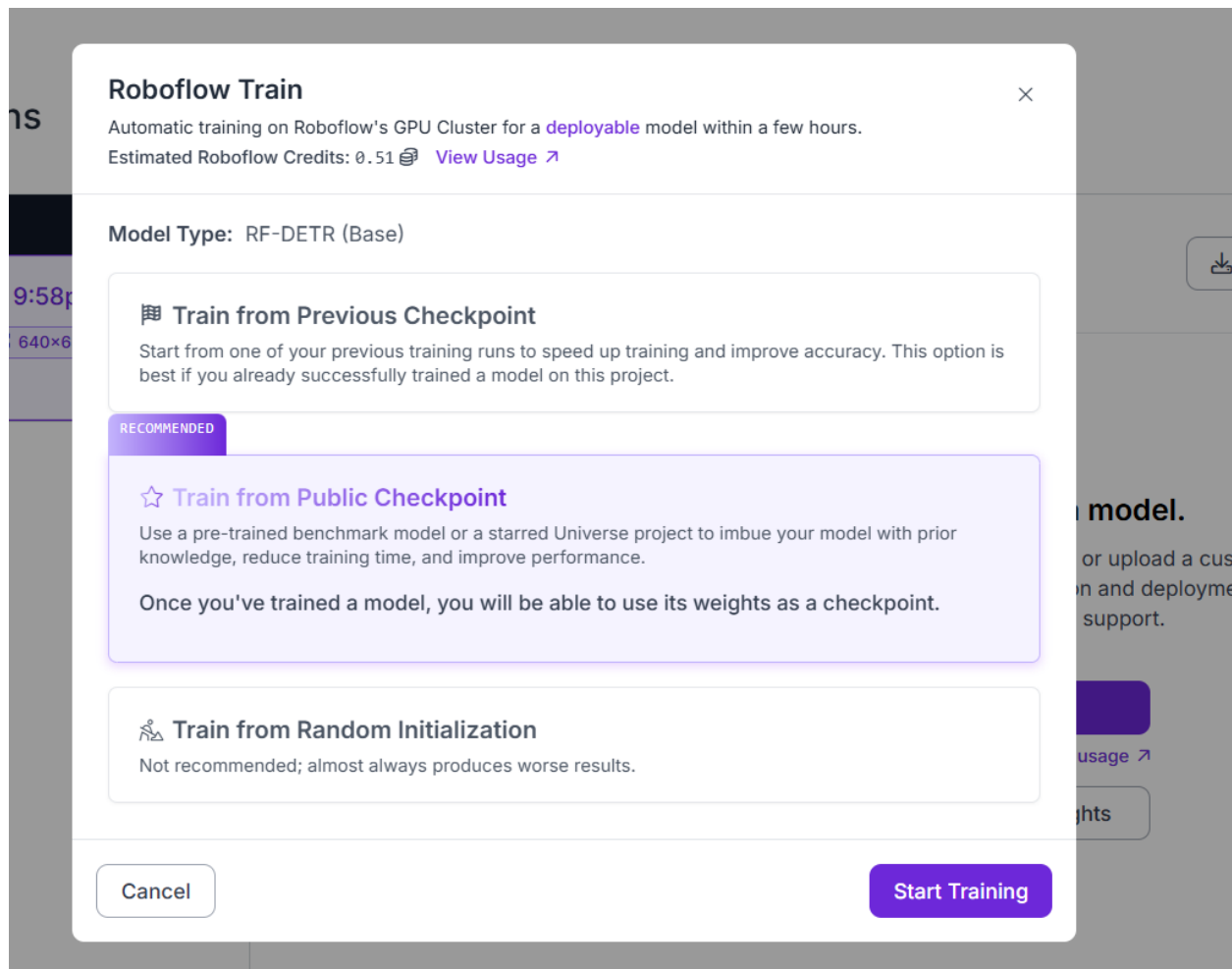
Base

- New State-of-the-Art model by Roboflow
- Size comparable to YOLO medium

Cancel

Continue

 How to Upload Custom Weights



Si lo cambiamos a object detection (con bounding boxes) ahora si nos permite usar rf detr

☒ rfdetrpruebas2 2 View Model →

Model URL:	Checkpoint:
rfdetrpruebas2/2	-
Updated On:	Model Type:
4/27/25, 10:13 PM	RF-DETR (Base)

Metrics ⓘ
mAP@50
75.2%


Detailed Model Evaluation UPGRADE FOR ACCESS
Understand model performance and how to improve it.

Confusion Matrix Vector Analysis

En mi caso mi modelo tuvo una precisión del 75.2% pero fluctúa dependiendo la imagen

Preview model

Samples from Test Set



[View Test Set →](#)

Upload Image or Video File


Drop file here or

[Select File](#)

Image URL

[🔗](#)

[</> Try On My Machine](#)



1 object detected

Confidence Threshold: **50%**

0% 100%

Overlap Threshold: **50%**

0% 100%

Opacity Threshold: **%**

0% 100%

Label Display Mode:

[Draw Confidence](#)

```
{
  "predictions": [
    {
      "x": 1545.459,
      "y": 1058.853,
      "width": 2178.194,
      "height": 2117.705,
      "confidence": 0.627,
      "class": "trabajador",
      "class_id": 6,
      "detection_id": "b7b9309"
    }
  ]
}
```

[Copy](#)

Classes ⓘ

Overview of the number of annotations for each class in your dataset.

[🔄 Rebalance Splits](#)

[📄 Download CSV](#)

[All Splits](#) ☐ Train ☐ Valid ☐ Test

[Sort ↓](#)

trabajador 12

edificio 6

grua 5

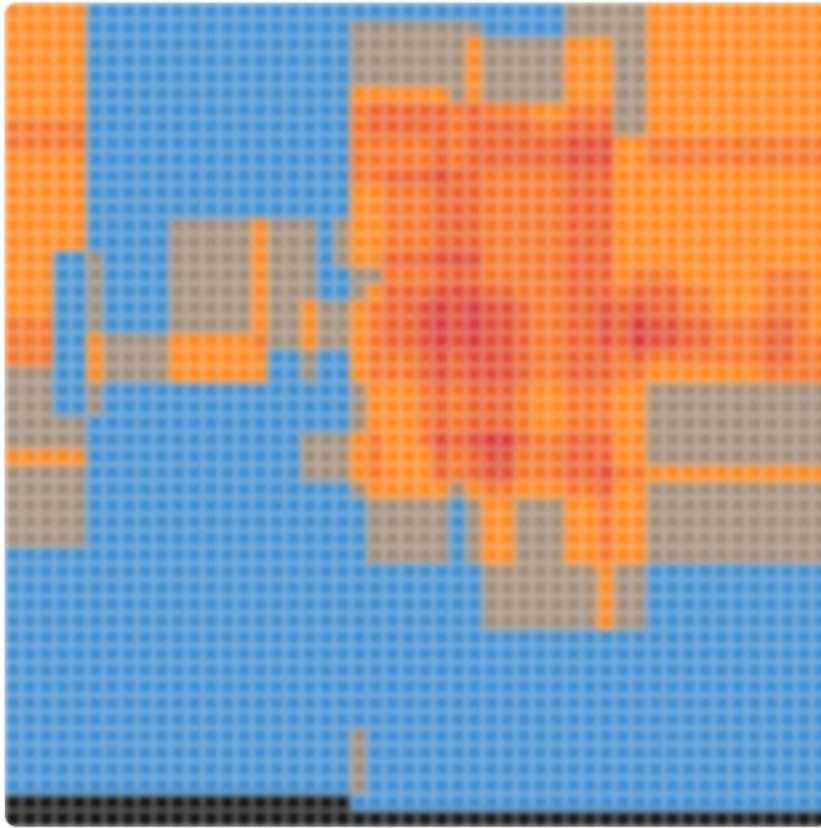
excavadora 4

construccion 3

casco 3

Annotation Heat Map ⓘ

Shows you where most of your annotations are. Color gradients signify the number of annotations per grid cell.



Images

Drag over a

Conclusiones modelo rf detr:

- No puede correr con polígonos (instance segmentation) y todo nuestro proyecto es con polígonos
- Rf detr corre únicamente en proyectos de bounding boxes o object detection
- Rf detr es por naturaleza mas pesado para un embebido o una jetson
- RF Detr supera a YOLOv11 con mas de 60 AP de precisión en coco, YOLOv11 llega de 54 a 57
- Mucho más preciso