Madrid Rooftops Image Segmentation

Implementación Mask R-CNN para la segmentación de tejados residenciales, piscinas y canchas deportivas en la Comunidad de Madrid

Ana Blanco Delgado
Septiembre 2021

Inspiración

idealista/energy

Calle Prim, 11:

https://www.idealista.com/energy/calculadora-de-ahorro-solar/#ref=1350216VK4715A0001FW&lat=40.422026&lng=-3.6937241

293 € **ahorro anual en la factura de la luz por vecino**

23.009 € coste de la instalación de 61 paneles solares (15,6 KWp de potencia)

5 años tardaría la instalación en amortizarse.

190.746 € ahorro total durante 25 años de vida útil de la instalación

Ver detalles del cálculo de ahorro y financiación

Instalación solar óptima para el tejado de Calle Prim 11, Madrid

- 164 m² disponibles
- 3.043 horas de sol al año
- 38% de inclinación
- 61 paneles solares
- 15,6 KWp de potencia
- 6 tn de CO2 menos al año

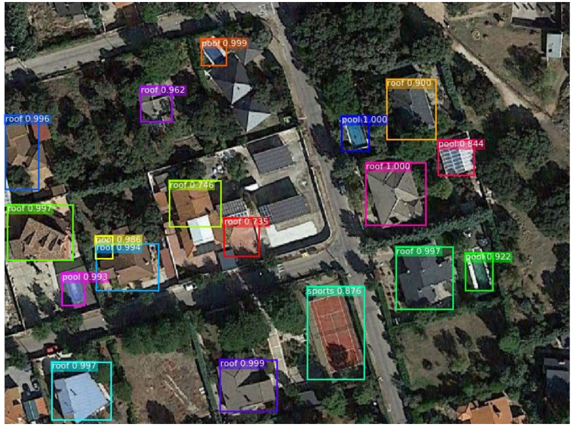
El ahorro total contempla la pérdida de rendimiento de los paneles. Los cálculos están basados en un comportamiento regular de la instalación y son siempre orientativos y no vinculantes.

Ana Blanco Delgado
Septiembre 2021

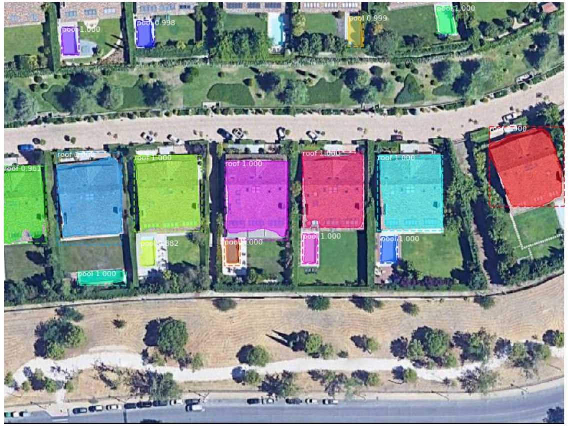
Computer vision tasks

Clasificación

Detección



Segmentación



Ana Blanco Delgado
Septiembre 2021

Mask R-CNN

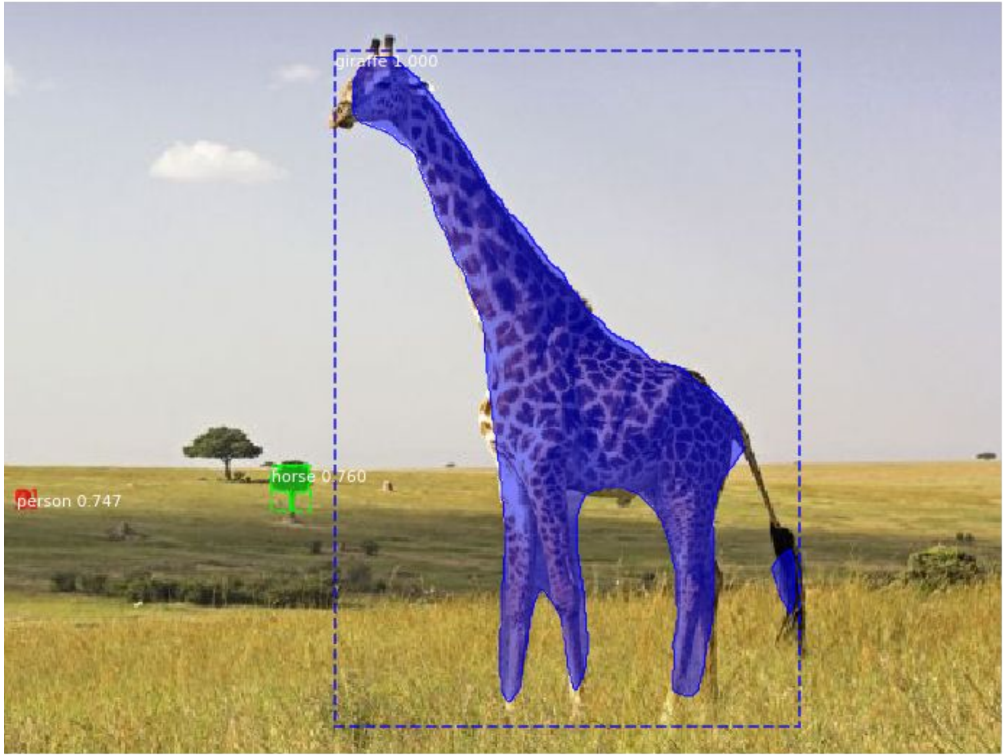
Mask Region Convolutional Neural Network

Mask R-CNN paper oficial
https://arxiv.org/abs/1703.06870

Mask R-CNN for Object Detection and Segmentation
(repositorio open-source de Matterplot)
https://github.com/matterport/Mask_RCNN

Mask R-CNN

COCO Dataset Classes: ['BG', 'person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus', 'train', 'truck', 'boat', 'traffic light', 'fire hydrant', 'stop sign', 'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow', 'elephant', 'bear', 'zebra', 'giraffe', 'backpack', 'umbrella', 'handbag', 'tie', 'suitcase', 'frisbee', 'skis', 'snowboard', 'sports ball', 'kite', 'baseball bat', 'baseball glove', 'skateboard', 'surfboard', 'tennis racket', 'bottle', 'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl', 'banana', 'apple', 'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza', 'donut', 'cake', 'chair', 'couch', 'potted plant', 'bed', 'dining table', 'toilet', 'tv', 'laptop', 'mouse', 'remote', 'keyboard', 'cell phone', 'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'book', 'clock', 'vase', 'scissors', 'teddy bear', 'hair drier', 'toothbrush']



Implementation selected

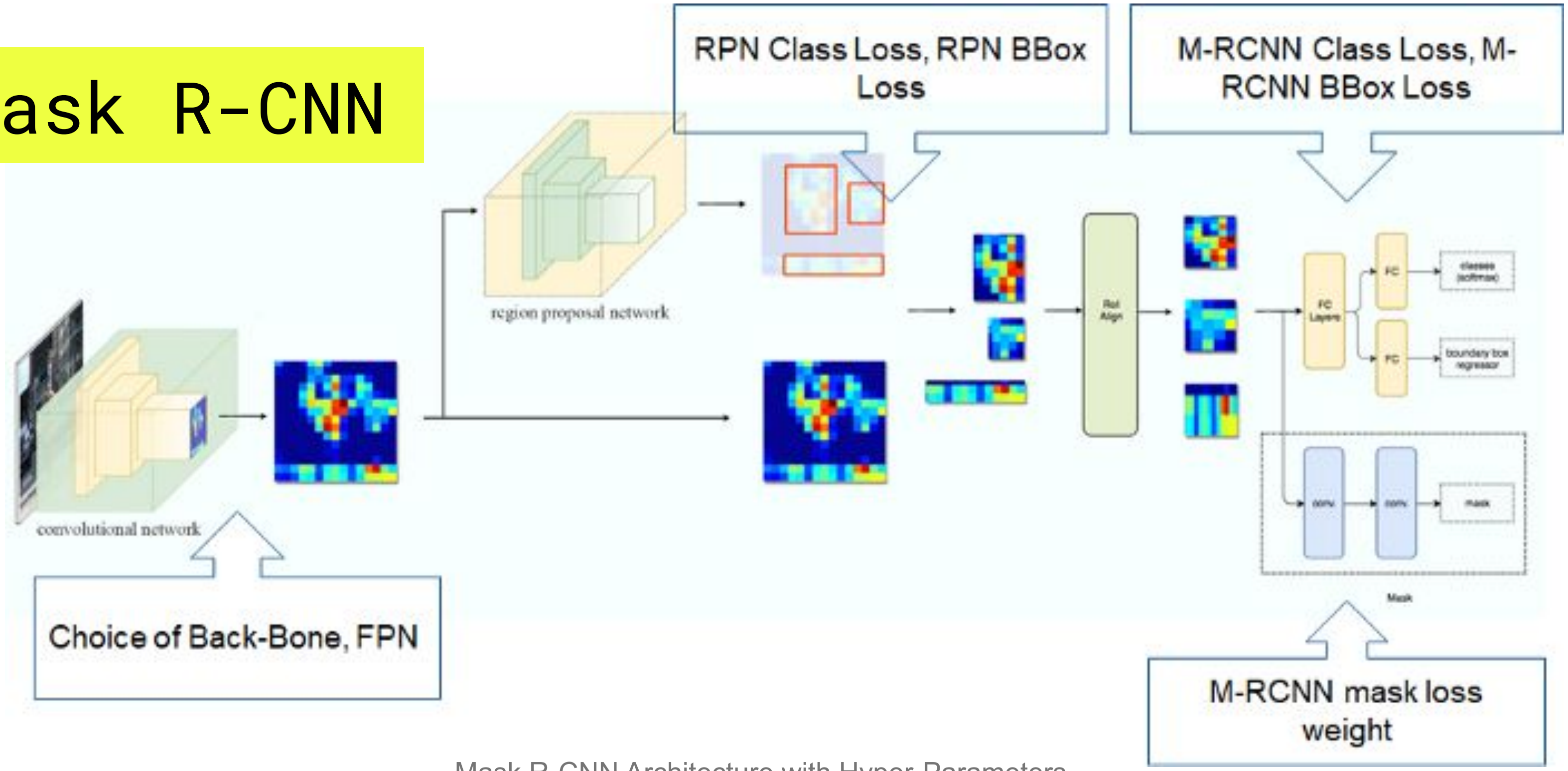
Mask R-CNN

Implementation selected

Mask R-CNN

Implementation selected

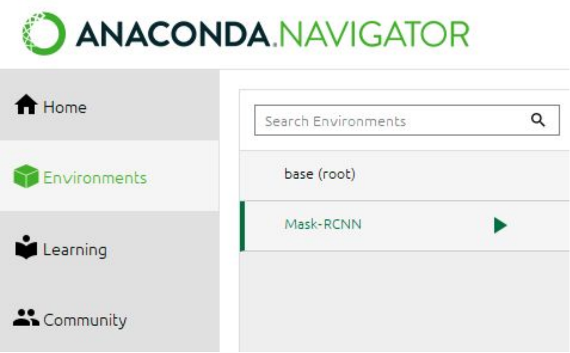
Mask R-CNN

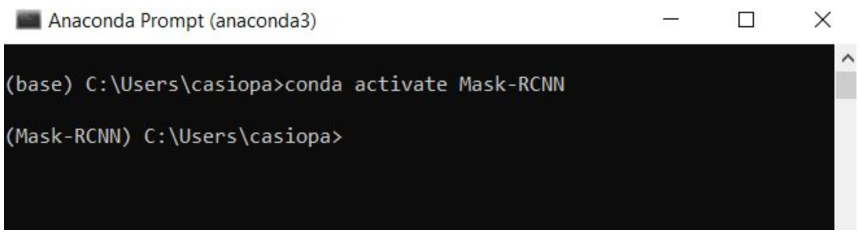


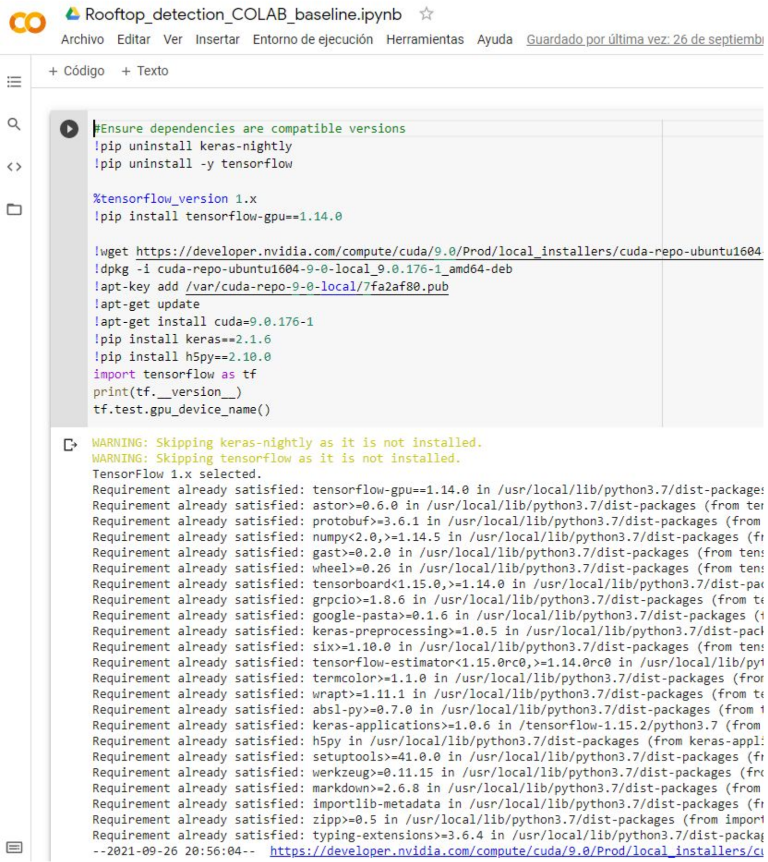
Mask R-CNN Architecture with Hyper-Parameters

Entorno de trabajo local y en Google Colab

Mask R-CNN







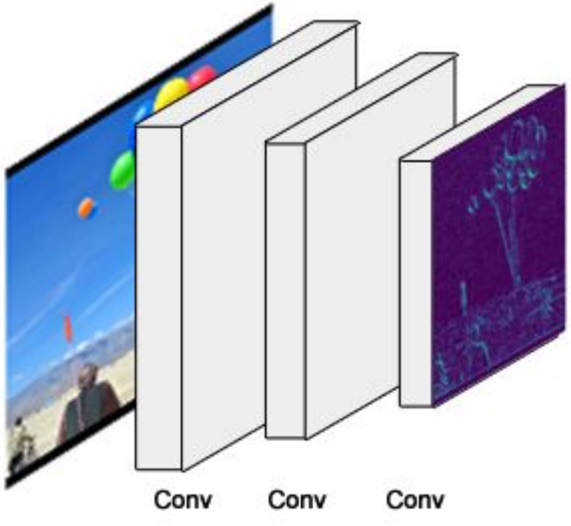
Mask R-CNN modules

1.Backbone | 2.RPN | 3.ROI-Clf/BB-Regr | 4.Segmentation

CNN ResNet101

`model.resnet_graph()`

Input image
1024 x 1024 x 3



Features
32 x 32 x 2048

1.Backbone 2.RPN 3.ROI-Clf/BB-Regr 4.Segmentation

Feature Pyramid Networks (FPN)

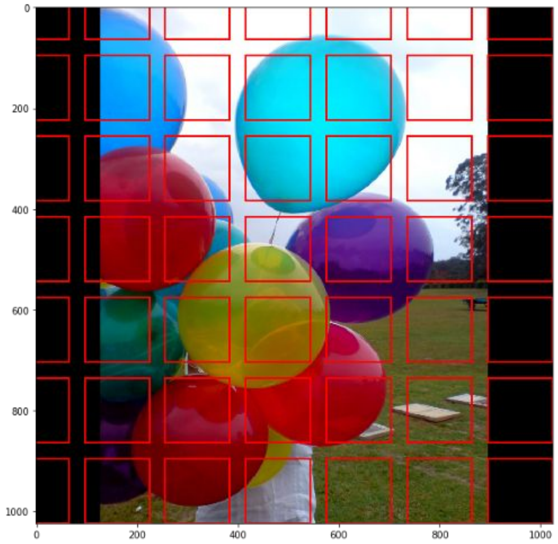
model.build()

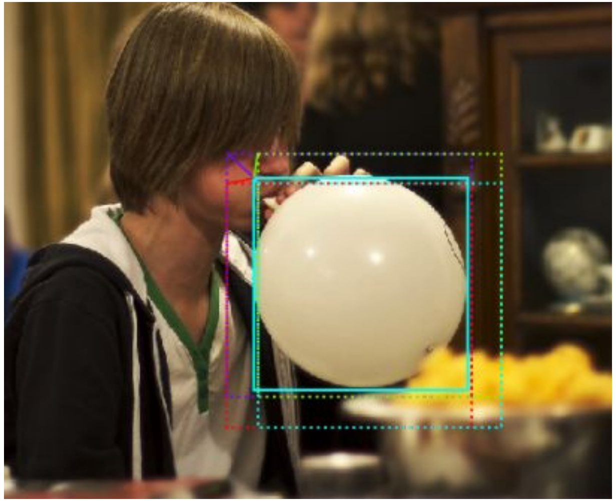
Mask R-CNN modules

1.Backbone | 2.RPN | 3.ROI-Clf/BB-Regr | 4.Segmentation

Region Proposal Network (RPN)

model.rpn_graph()



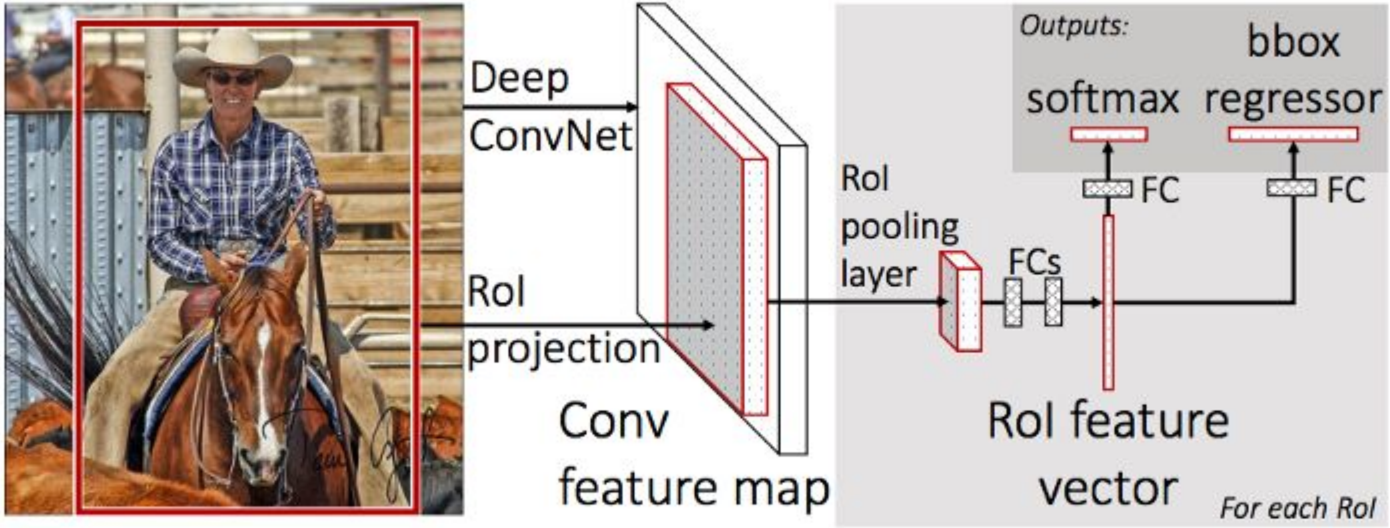


Mask R-CNN modules

1.Backbone | 2.RPN | **3.ROI-Clf/BB-Regr** | 4.Segmentation

ROI Classifier & Bounding Box Regressor

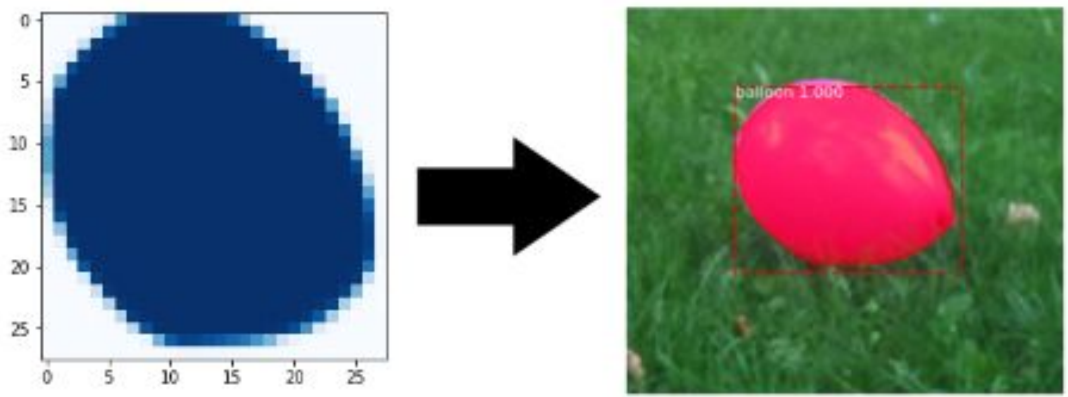
`model.fpn_classifier_graph()`



Mask R-CNN modules

Segmentation Masks

`model.build_fpn_mask_graph()`



28x28 Soft Mask

Resized Binary Mask

Dataset

train_set	val_set
Image Count: 69 Polygon Count: 1488 Class Count: 4 0. BG 1. roof 2. pool 3. sports	Image Count: 22 Polygon Count: 545 Class Count: 4 0. BG 1. roof 2. pool 3. sports

Image Count: 91
Polygon Count: 2033

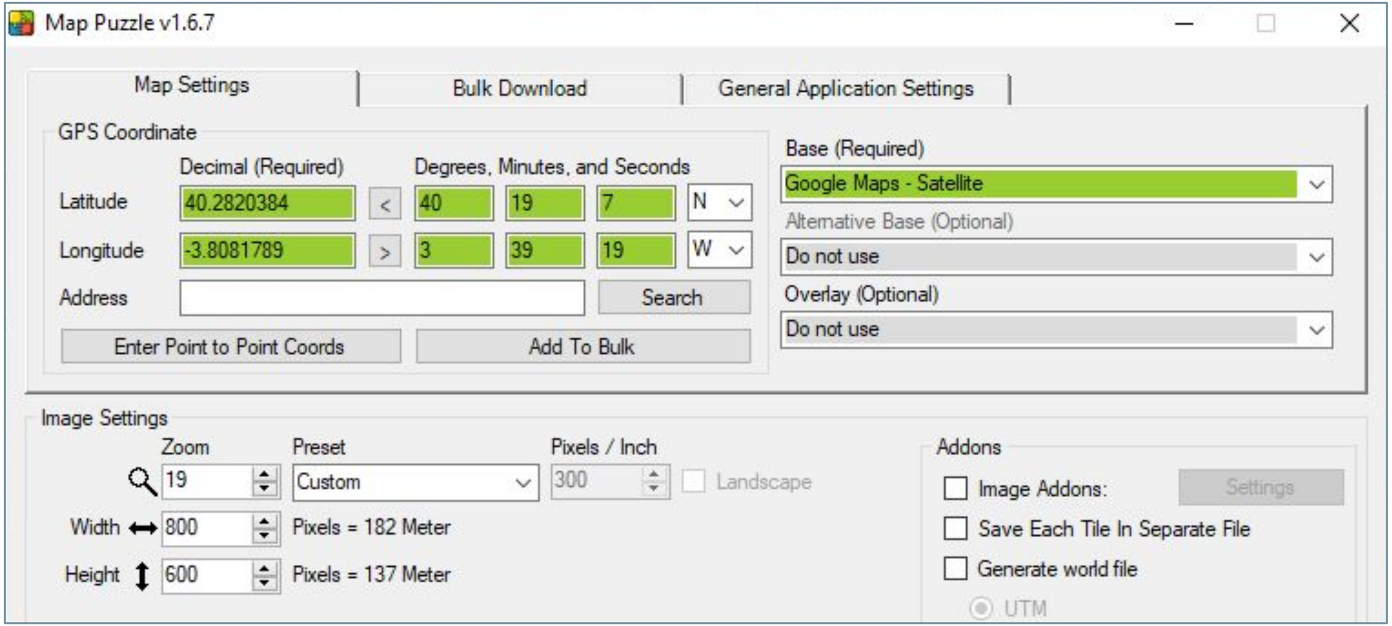
Aravaca, Cercedilla, Fuenfría, Navacerrada, Nuevo Baztán, Pozuelo, Somosaguas y Soto del Real

Dataset

1. Imágenes

2. Annotations

Map Puzzle

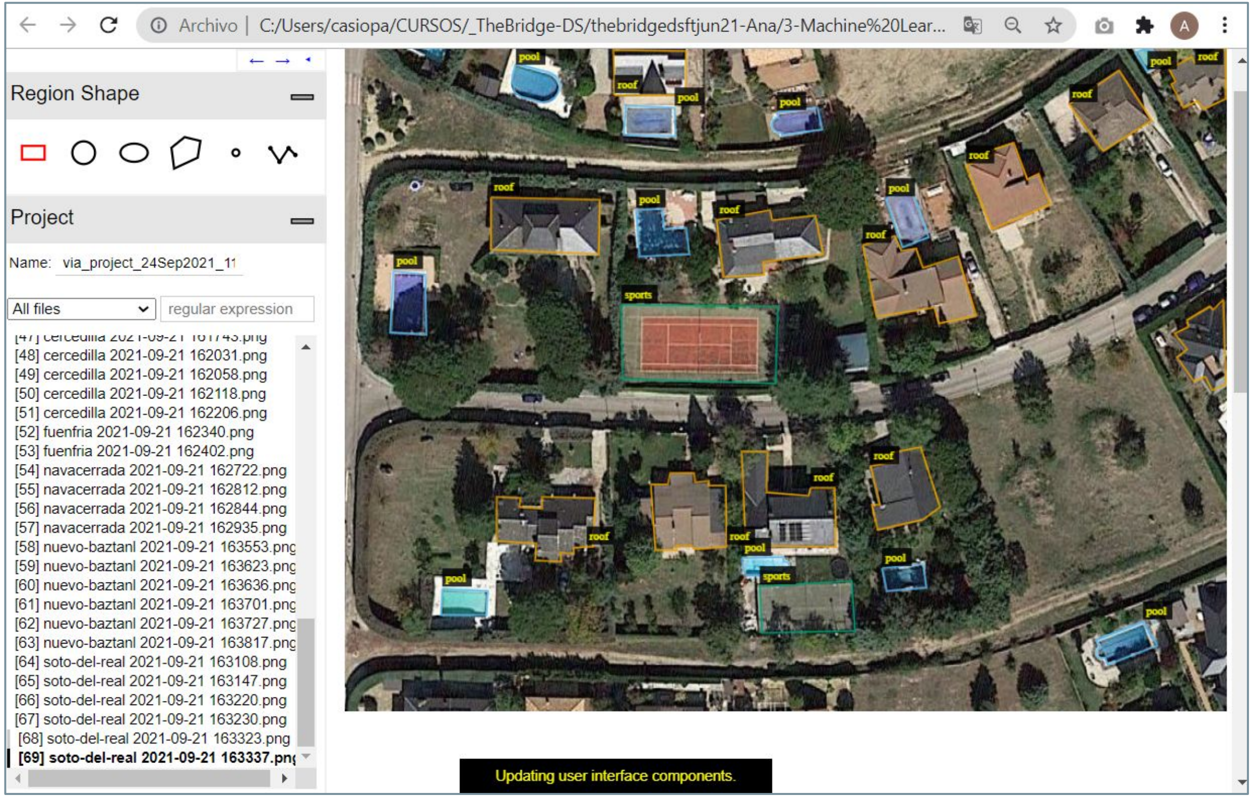


Dataset

1.Imágenes

2.Annotations

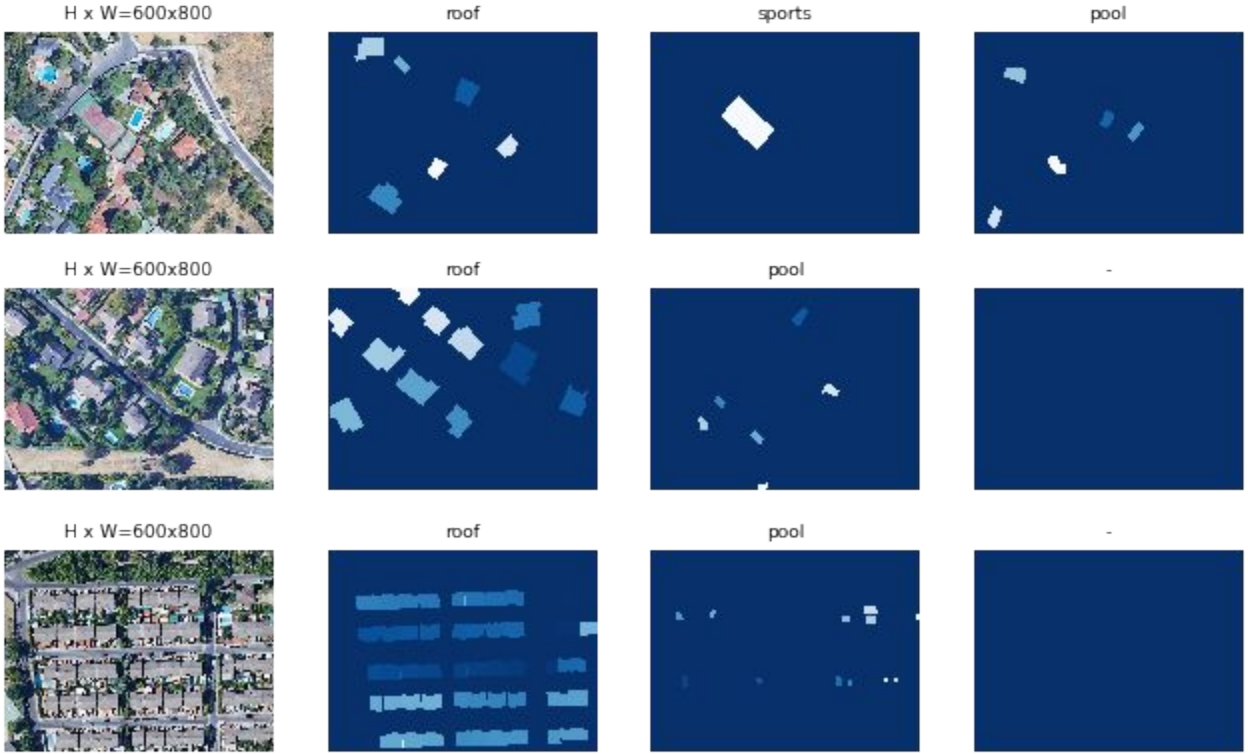
VGG Image Annotator (VIA)



Dataset

1. Imágenes

2. Annotations



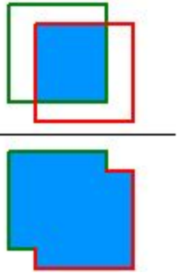
Object detection metrics

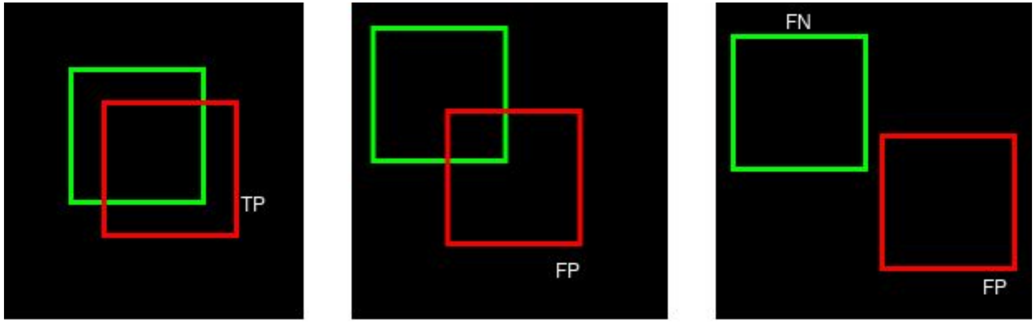
1. AP

2. mAP

Intersect over Union (IoU)

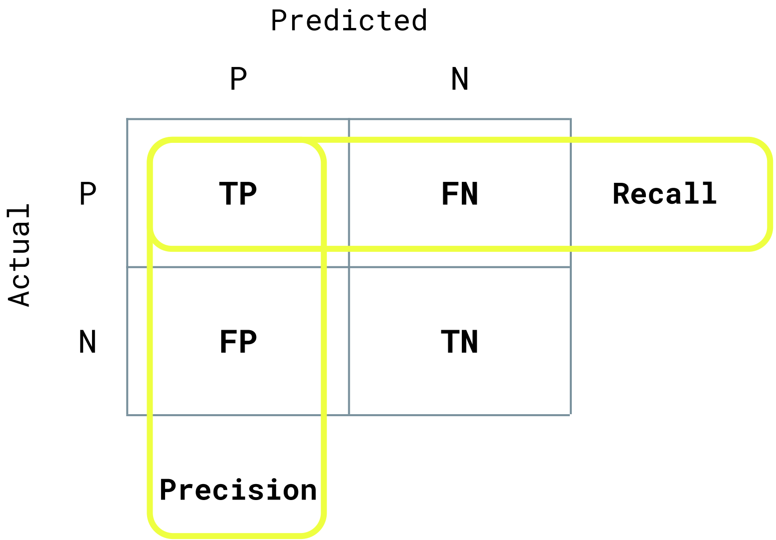
$$IOU = \frac{\text{area of overlap}}{\text{area of union}} =$$





IoU = 0.86 IoU = 0.24 IoU = 0

Precision/Recall



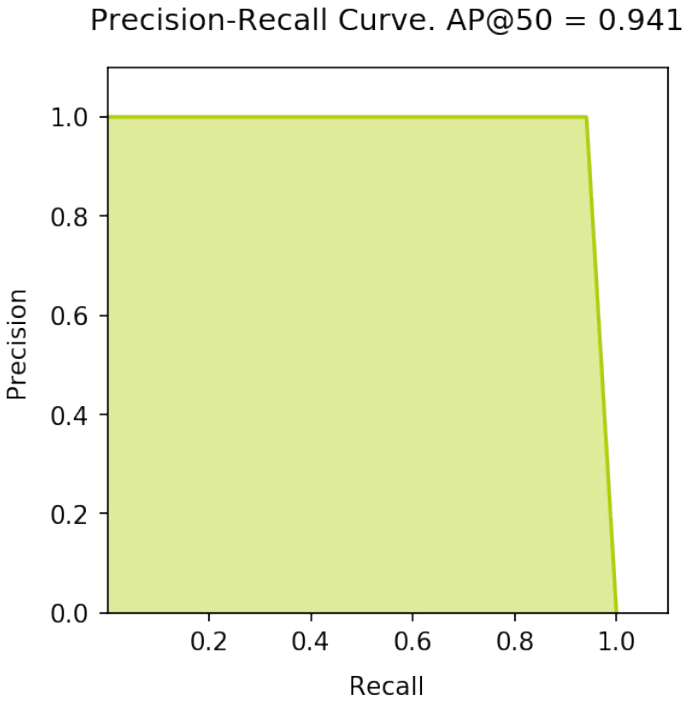
Object detection metrics

1.AP

2.mAP

Área bajo la curva
Precision/Recall
AUC-PR

Average Precision
AP



Mean Average Precision Formula

$$\mathrm{mAP}@\alpha = \frac{1}{n}\sum_{i=1}^{n} \mathrm{AP}_i$$

1.Baseline model

Learning Rate: 80 epochs: 0.001

Loss Weights: 'rpn_class_loss': 1.0
'rpn_bbox_loss': 1.0
'mrcnn_class_loss': 1.0
'mrcnn_bbox_loss': 1.0
'mrcnn_mask_loss': 1.0

Best model 60 epochs:
Train mAP@50: 0.720
Val mAP@50: 0.527

2. Hyperparameter tuning

Learning Rate: 10 epochs: 0.001
+10 epochs: 0.0005

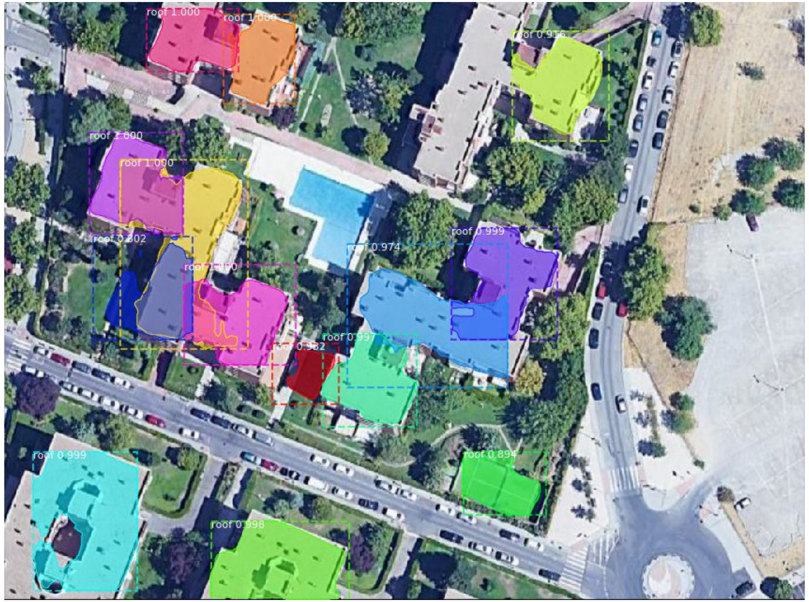
Loss Weights: 'rpn_class_loss': 1.0
'rpn_bbox_loss': 0.8
'mrcnn_class_loss': 6.0
'mrcnn_bbox_loss': 6.0
'mrcnn_mask_loss': 6.0

Augmentation 50%: Fliplr(0.5)
Blur [1, 5]

Train mAP@50: 0.562
Val mAP@50: 0.469

Test_set prediction image

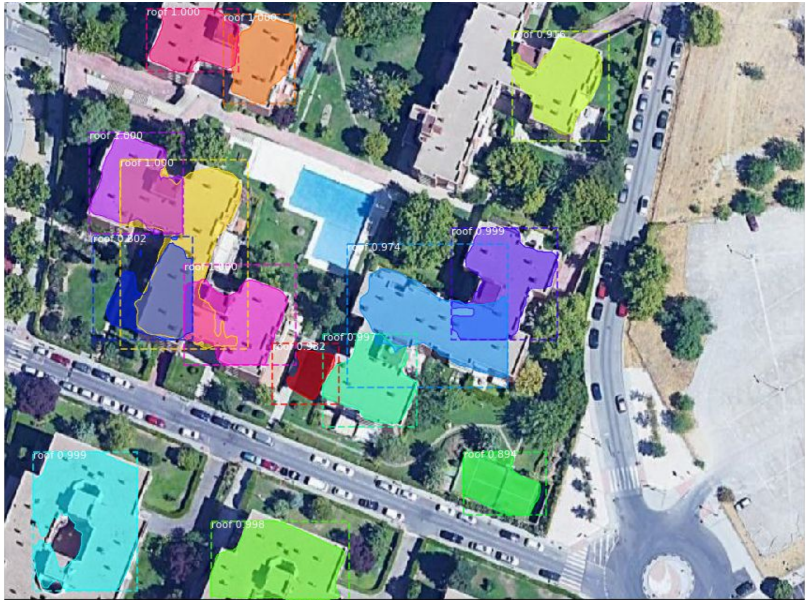
1.Edificios 2.Casas



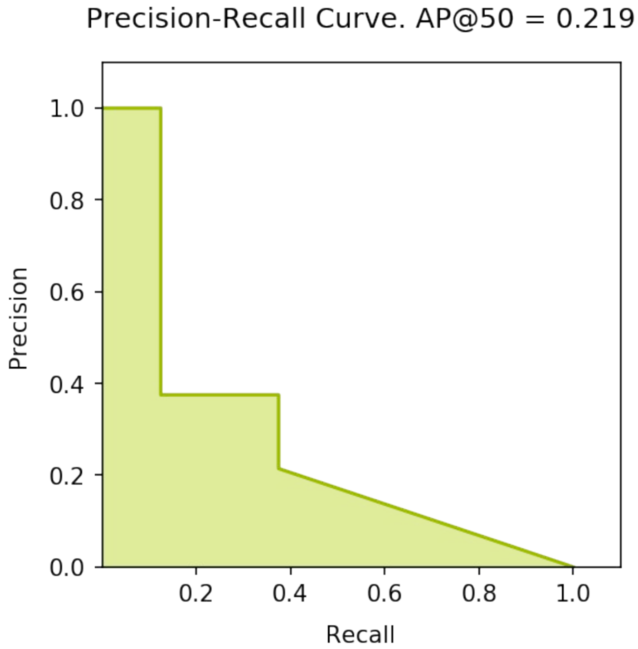
pozuelo 2021-09-13 200628.png

Test_set prediction image

1.Edificios 2.Casas



pozuelo 2021-09-13 200628.png

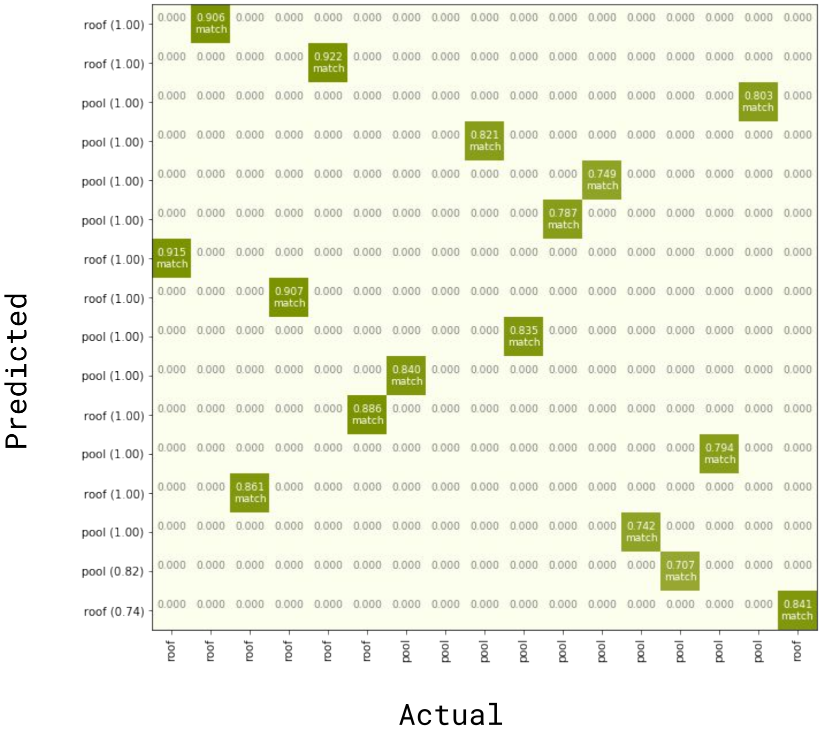


Test_set prediction image

1.Edificios

2.Casas

somosaguas 2021-09-13 194952.png



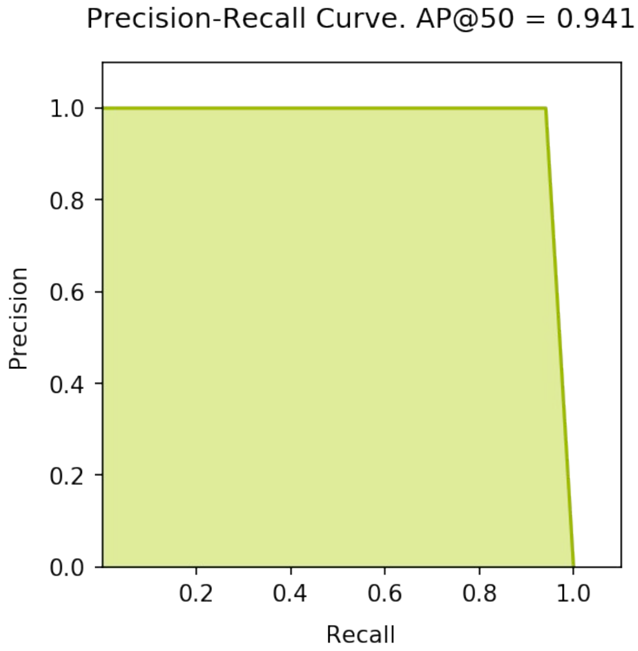
Predicted
roof (1.00) 0.000 0.906 match 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
roof (1.00) 0.000 0.000 0.000 0.000 0.922 match 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
pool (1.00) 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.803 match 0.000
pool (1.00) 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.821 match 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
pool (1.00) 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.749 match 0.000 0.000 0.000 0.000 0.000
pool (1.00) 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.787 match 0.000 0.000 0.000 0.000 0.000 0.000
roof (1.00) 0.915 match 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
roof (1.00) 0.000 0.000 0.000 0.907 match 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
pool (1.00) 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.835 match 0.000 0.000 0.000 0.000 0.000 0.000 0.000
pool (1.00) 0.000 0.000 0.000 0.000 0.000 0.000 0.840 match 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
roof (1.00) 0.000 0.000 0.000 0.000 0.000 0.886 match 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
pool (1.00) 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.794 match 0.000 0.000
roof (1.00) 0.000 0.000 0.861 match 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
pool (1.00) 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.742 match 0.000 0.000 0.000 0.000
pool (0.82) 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.707 match 0.000 0.000 0.000
roof (0.74) 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.841 match
roof roof roof roof roof roof pool pool pool pool pool pool pool pool pool pool roof
Actual

Test_set prediction image

1.Edificios

2.Casas

somosaguas 2021-09-13 194952.png



Test final

Modelo orientado para segmentar:

- Viviendas residenciales aisladas (casas, chalets)
- Instancias por imagen < 20
- Imágenes con buena resolución

Madrid Rooftops Image Segmentation project

Septiembre 2021

https://github.com/casiopa/Madrid_Rooftops