

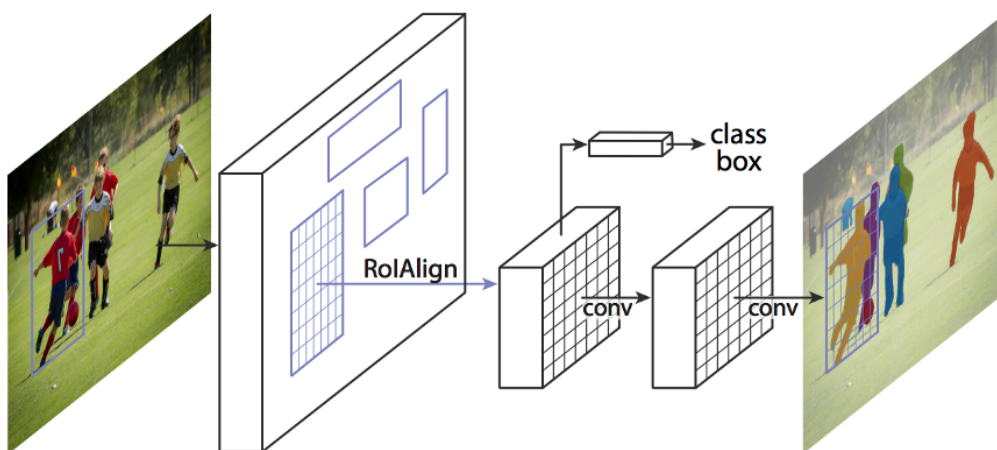
Madrid Rooftop Segmentation

Mask R-CNN para detección de tejados residenciales, piscinas y áreas deportivas

Ana Blanco Delgado | Septiembre de 2021

Este proyecto lo he realizado para el Bootcamp de Data Science realizado en The Bridge como trabajo individual de Machine Learning. Mi background relacionado con la imagen y el vídeo me ha arrastrado a tareas de *Computer Vision*, y la idea de este proyecto me aterrizó tras escuchar un muy interesante podcast de Data Stand-Up! con el entrevistado David Rey (Chief Data Officer en Idealista). En la entrevista se menciona el proyecto de Idealista Energy que ofrece información sobre los tejados y su posible aprovechamiento para la instalación de paneles solares (área del tejado, número de paneles solares a instalar, coste aproximado de su instalación, cálculo de ahorro posterior en el consumo energético...).

El objetivo del proyecto es acercarme a este servicio de Idealista Energy y detectar tejados residenciales, piscinas y áreas deportivas, en imágenes aéreas de la Comunidad de Madrid. Para ello he utilizado el framework Mask R-CNN (Mask Region Convolutional Neural Network) para la detección, clasificación y segmentación de imágenes. El framework elegido nos permite hacer segmentación de la imagen para seleccionar el área del tejado por su forma lo que nos podría permitir calcular el área de la máscara predicha. Además, esta implementación de Mask R-CNN parte de una red ResNet101 preentrenada con el dataset de imágenes MS COCO.



Mask R-CNN framework. Source: <https://arxiv.org/abs/1703.06870>

Para que el proyecto fuese viable en tiempo y recursos he centrado el entrenamiento y la validación del modelo en imágenes aéreas y cenitales de áreas rurales o urbanizaciones de casas o edificios aislados. Además las zonas concretas de Madrid estudiadas han sido: Aravaca, Cercedilla, Fuenfría, Navacerrada, Nuevo Baztán, Pozuelo, Somosaguas y Soto del Real. La fuente de las imágenes es Google Maps a un zoom de 19. La calidad de las

imágenes se queda corta sobre todo en zonas menos urbanas donde la compresión utilizada por Google es mayor.

Dataset

Como es de esperar, esta ha sido la parte más laboriosa del proyecto. La creación de un dataset de este tipo tiene dos fases: la obtención de las imágenes aéreas y el etiquetado de las instancias mediante una máscara junto con la asignación de la clase a la que pertenece cada instancia. Para obtener las imágenes aéreas he utilizado el software *Map Puzzle* que permite la descarga de imágenes de satélite de Google para unas coordenadas y zoom dados. Para el etiquetado de las imágenes utilicé *VIA Image Annotator* que permite utilizar polígonos para la segmentación de los objetos de la imagen. Esta información la guarda en un archivo de tipo json para cada dataset.

| train_set | val_set | test_set |
|--|--|--|
| Image Count: 69 Polygon Count: 1488 | Image Count: 22 Polygon Count: 545 | Image Count: 11 Polygon Count: 258 |
| Class Count: 4 0. BG 1. roof 2. pool 3. sports | Class Count: 4 0. BG 1. roof 2. pool 3. sports | Class Count: 4 0. BG 1. roof 2. pool 3. sports |



Ejemplo de imagen importada con máscaras asociadas a las 3 clases del proyecto (tejado, piscinas y canchas deportivas).

Para ver información más detallada sobre el dataset revisar el notebook:

/notebooks/inspect_data.ipynb

Primeros pasos y dificultades técnicas

Dado que la implementación de Mask R-CNN elegida para el proyecto es de 2018, he tenido que montar un entorno local con versiones anteriores de Keras y Tensorflow. Y ya que no dispongo de GPU he recurrido a Google Colab para el entrenamiento y la validación de los modelos. El tiempo gratuito de Google Colab es limitado, lo me ha obligado a entrenar por tramos de 20 epochs.

Entrenamiento

El notebook de entrenamiento en Google Colab puede verse en GitHub:

[Rooftop_detection_COLAB.ipynb](#)

Una vez heredadas las clases necesarias para cargar el dataset y configurar el modelo según especificaciones de Mask R-CNN, entrené las 69 imágenes con sus 1.488 máscaras durante 80 epochs. Evalué con el Mean Average Precision ($\text{mAP@IoU}=50$) cada epoch resultando el modelo del epoch 60 el mejor con los siguientes valores:

Evaluación modelo epoch 60:

Train_set mAP@50 : 0.720

Test_set mAP@50 : 0.624

Estos valores sugieren seguir entrenando el modelo, posiblemente con un dataset mayor, pero con tan solo 69 imágenes y 60 epochs se ha conseguido un $\text{mAP@IoU}=50$ de 0.624, así que el proyecto se presenta prometedor.

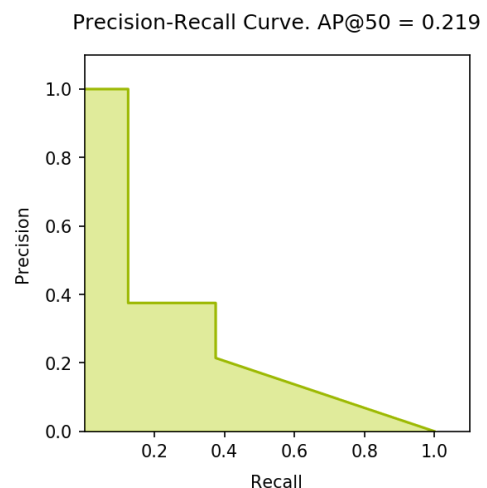
Para ver información más detallada sobre el modelo seleccionado revisar el notebook:

[/notebooks/inspect_model.ipynb](#)

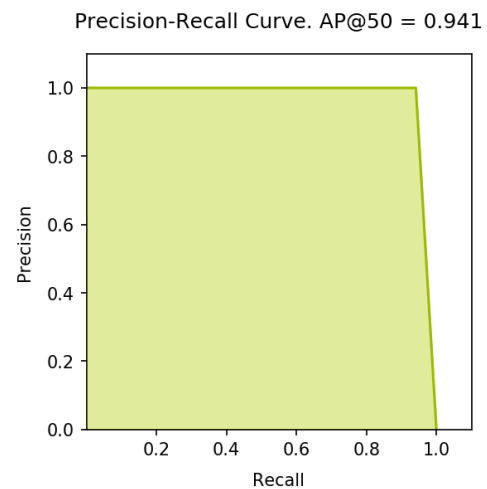
Para comprender mejor los resultados revisé las predicciones del modelo con el dataset de validación porque tenía la intuición de que el modelo tiene un buen desempeño para la segmentación de residencias aisladas pero no de adosados o edificios. A continuación muestro ambos casos:

Ejemplo de predicción y evaluación en imagen con edificios

(pozuelo 2021-09-13 200628.png)



Ejemplo de predicción y evaluación en imagen de casas aisladas (somasaguas 2021-09-13 194952.png)



Así podemos concluir que el modelo consigue unos buenos resultados con imágenes:

- En zonas de viviendas residenciales aisladas (casas y chalets)
- Y un número de instancias no muy superior a 20

Fuentes y recursos

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

Splash of Color: Instance Segmentation with Mask R-CNN and TensorFlow:
<https://engineering.matterport.com/splash-of-color-instance-segmentation-with-mask-r-cnn-and-tensorflow-7c761e238b46>

Podcast Data Stand-Up! Episodio 6, con David Rey (Chief Data Officer - Idealista):
<https://www.spreaker.com/user/bedr0o/david-rey-chief-data-officer-idealista>

How to Train an Object Detection Model with Keras:
<https://machinelearningmastery.com/how-to-train-an-object-detection-model-with-keras/>

VGG Image Annotator (VIA): <https://www.robots.ox.ac.uk/~vgg/software/via/>

Map Puzzle: <http://www.mappuzzle.se/>

Google Colab: <https://research.google.com/colaboratory/>

Paneles solares para tu casa. Autoconsumo solar fotovoltaico:

<https://www.idealista.com/energy/>

Object Detection Metrics With Worked Example:

<https://towardsdatascience.com/on-object-detection-metrics-with-worked-example-216f173ed31e>

Taming the Hyper-Parameters of Mask RCNN:

<https://medium.com/analytics-vidhya/taming-the-hyper-parameters-of-mask-rcnn-3742cb3f0e1b>

Project Zeus, or How to Detect Rooftops using Neural Networks. A Begginer's Guide:

<https://alexwhelan.medium.com/project-zeus-or-how-to-detect-rooftops-using-neural-networks-a-beginners-guide-a8efed8da28a>