

## APRENDIZAJE PROFUNDO

En este proyecto aprenderás a generar modelos entrenados mediante aprendizaje profundo que puedan detectar objetos en imágenes.

Vamos a trabajar con YOLO, que es una red neuronal convolucional (CNN) utilizada principalmente para la detección de objetos en tiempo real. También se puede utilizar para segmentar imágenes (que es un problema más complejo) o clasificar imágenes (que es un problema más sencillo).

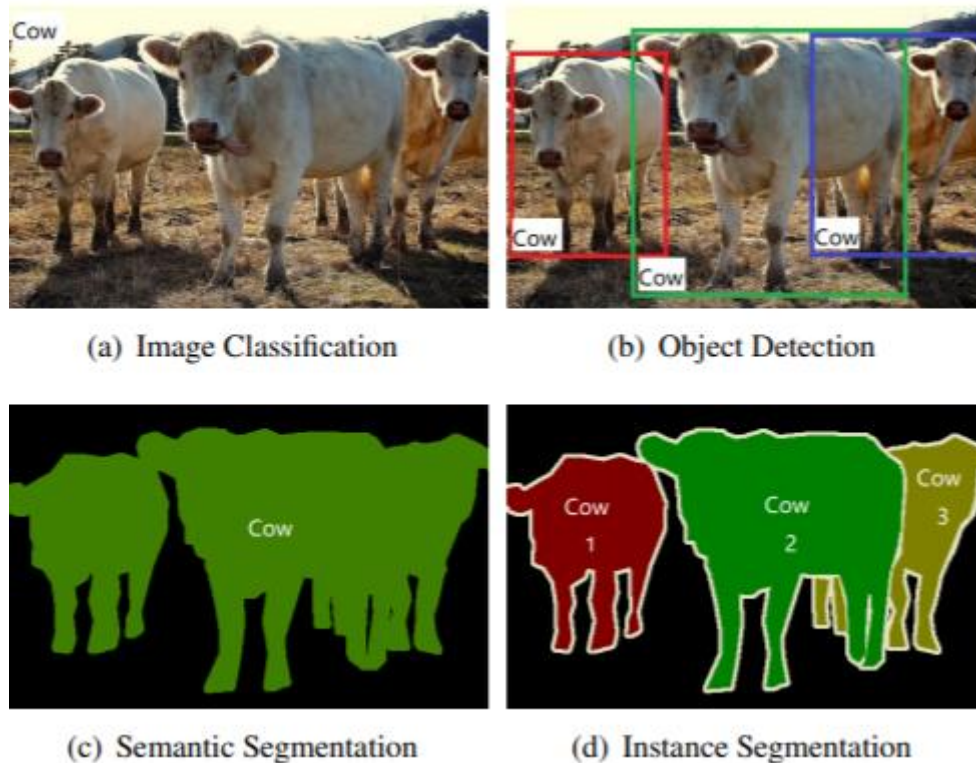


Fig 1. Diferencias entre clasificación, detección y segmentación. [Imagen tomada de <https://www.macnica.co.jp/en/business/ai/blog/142067>]

Se usará el modelo pre-entrenado más reciente para detectar objetos en imágenes:  
<https://docs.ultralytics.com/es/models/>

Vamos a utilizar el conjunto de datos COCO, que es uno de los más conocidos ya que cuenta con multitud de imágenes etiquetadas y considera muy diversas clases. En particular, se va a utilizar el conjunto COCO 2017, únicamente el subconjunto de validación:

<http://images.cocodataset.org/zips/val2017.zip>

[http://images.cocodataset.org/annotations/annotations\\_trainval2017.zip](http://images.cocodataset.org/annotations/annotations_trainval2017.zip)

Por simplicidad, trabajaremos un subconjunto de este conjunto, así que se tomarán únicamente las 100 primeras imágenes.

Se pide desarrollar un código (archivo tutorial.ipynb) que visualice la máscara de verdad (también denominada ground-truth) de la imagen 139 del conjunto de datos de validación COCO 2017, es decir, que dibujes el bounding-box y la clase de cada objeto de dicha imagen utilizando OpenCV. Tras esto, para cada modelo de YOLO (n, s, m, l, x), haz la predicción y guárdala. Para el modelo que más objetos detecte en esa imagen, extrae las clases detectadas, coordenadas de los bounding boxes predichos y confianza. Dibuja las detecciones y la ground-truth utilizando OpenCV, y guarda este resultado. Usa como métricas el mAP o F1, y compara el rendimiento entre los distintos modelos.

Se pide desarrollar un código (archivo setup.ipynb) que descargue el conjunto de datos de validación COCO 2017 y sus etiquetas, y que genere un subconjunto formado por las 100 primeras imágenes.

Se pide desarrollar un código (archivo eval.ipynb) que, para cada modelo (n, s, m, l, x), realice las predicciones de los tests y genere su rendimiento.

Se pide desarrollar un código (archivo results.ipynb) que cargue el rendimiento de cada modelo y genere tablas y figuras con las que compararlos fácilmente.

Finalmente, se tiene que redactar un informe con los resultados obtenidos y las conclusiones que se pueden extraer. El informe tiene que estar escrito en Latex usando la plantilla LNCS.

Se tiene que entregar un único archivo comprimido que contenga los siguientes archivos:

- El informe en formato pdf generado por el proyecto Latex.
- Archivo comprimido (.rar o .zip) con todos los archivos del proyecto Latex.
- El código (en formato ipynb) que se ha desarrollado.
- Un video (en formato mp4) con la ejecución del código.
- Declaración explícita en la que se asuma la originalidad del trabajo entregado.