

APRENDIZAJE PROFUNDO

En este proyecto aprenderás a generar modelos entrenados mediante aprendizaje profundo que puedan clasificar 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. En nuestro caso, vamos a utilizarla para clasificar imágenes, que es un problema más sencillo.

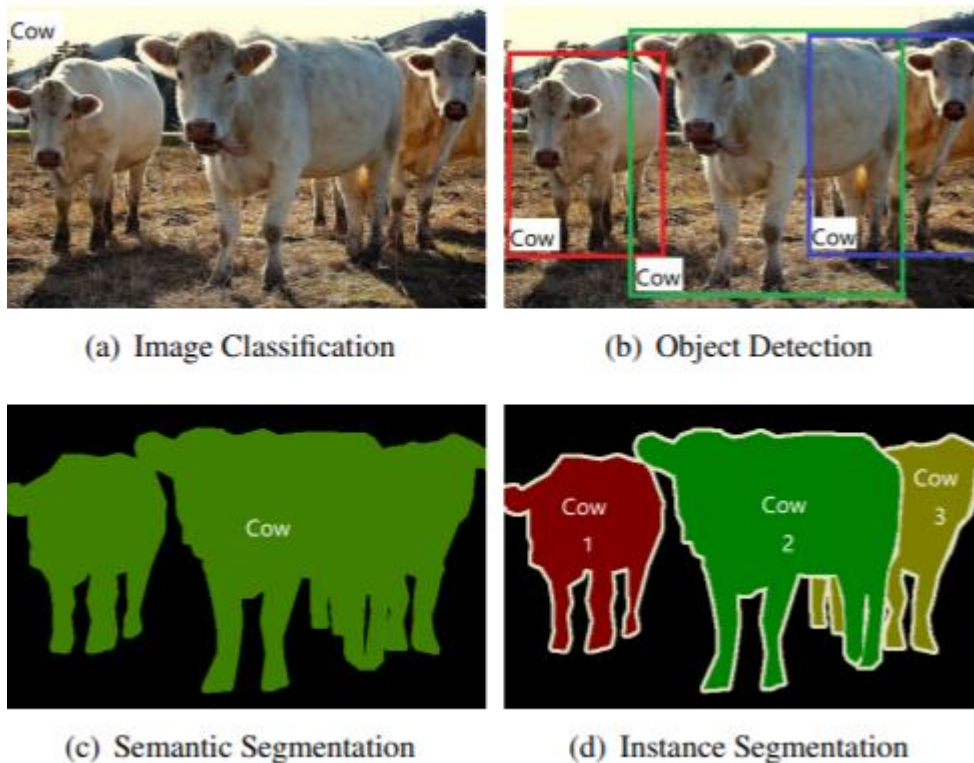


Fig 1. Diferencias entre clasificación, detección y segmentación. [Imagen tomada de <https://www.macnica.co.jp/en/business/ai/blog/142067>]

Se usará el modelo pre-entrenado más reciente para clasificar imágenes:

<https://docs.ultralytics.com/es/models/>

<https://docs.ultralytics.com/es/tasks/classify/>

En esta práctica vamos a utilizar el conjunto de datos ISIC 2019, que trata sobre clasificación de lesiones cutáneas: melanoma, nevus, queratosis, etc.

<https://challenge.isic-archive.com/data/#2019>

<https://challenge.isic-archive.com/landing/2019/>

Por simplicidad, trabajaremos un subconjunto del conjunto de datos de entrenamiento:

https://isic-challenge-data.s3.amazonaws.com/2019/ISIC_2019_Training_Input.zip

Y usaremos las etiquetas de dichas imágenes:

https://isic-challenge-data.s3.amazonaws.com/2019/ISIC_2019_Training_GroundTruth.csv

Se pide desarrollar un código (archivo setup.ipynb) que descargue el conjunto de datos de entrenamiento ISIC 2019 y sus etiquetas, y que genere dos subconjuntos de datos a partir de dicho conjunto: el primero (será el conjunto de entrenamiento de nuestro modelo) contendrá 100 imágenes de cada clase, mientras que el segundo (será el conjunto de test de nuestro modelo) contendrá 10 imágenes de cada clase.

Se pide desarrollar un código (archivo train.ipynb) que entrene y guarde los modelos entrenados para las versiones n, s y m de YOLO. Guardar gráficas sobre el proceso de entrenamiento con las que analizar su comportamiento y así establecer hipótesis como si se produce sobreajuste o subajuste.

Se pide desarrollar un código (archivo eval.ipynb) que, para cada modelo, 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.