

## Visió per Ordinador (VO-GIA)

### Proyecto final (PF)



#### Información:

- Entrega únicament el/los archivo/s de código con comentarios que expliquen detalladamente los distintos pasos de la solución y las distintas decisiones tomadas (métodos usados en la parte clásica, arquitectura e hiperparámetros escogidos en la parte de *deep learning*).
- Salvo que se indique lo contrario, utiliza exclusivamente las funciones vistas y trabajadas durante las sesiones de prácticas. En caso de duda, pregunta.
- Los ejercicios deben realizarse en grupo, utilizando los grupos de prácticas definidos al inicio del curso.
- La solución de estos ejercicios será parte del examen práctico, por lo que no está permitido consultar si una solución es correcta o suficiente. Los miembros de cada grupo deberán ejercer su juicio al respecto. Sin embargo, está permitido hacer preguntas sobre posibles ambigüedades en los enunciados o aspectos poco claros en la descripción de los ejercicios.
- En caso de existir dudas, el profesor podrá citar a un grupo de prácticas concreto para efectuar un breve examen práctico oral con el fin de comprobar la autoría y validez de la solución presentada. La calificación obtenida en dicho examen sustituirá a la del proyecto final. Si se observa que dos o más grupos tienen una solución sospechosamente idéntica, ambos grupos serán calificados con un cero en el proyecto final.

#### C.1

- a) Define una función que, utilizando herramientas clásicas (es decir, cualquier método que no se base en modelos entrenados), segmente los implantes en las imágenes facilitadas. En otras palabras, para cada imagen se debe generar una imagen binaria donde la región correspondiente al implante quede separada del fondo.
- b) Define una función que, nuevamente empleando herramientas clásicas (no basadas en modelos entrenados), calcule y muestre la *bounding box* del implante en las imágenes proporcionadas. Es decir, para cada imagen se debe generar una imagen en escala de grises, igual a la original, con una *bounding box* que delimite la región donde se encuentra el implante.



**Importante** → ¿Has podido desarrollar un método que funcione para todas las imágenes o, por el contrario, has tenido que ajustar la función según el tipo de imagen?

Usando las funciones desarrolladas en el ejercicio anterior, anota la base de datos facilitada. Después, separa dicha base de datos en *entrenamiento*, *test* y *validación*, utilizando un porcentaje de imágenes para cada conjunto que sea coherente. Re-entrena una arquitectura de detección de implantes y otra de identificación de implantes (puedes usar cualquiera de las arquitecturas vistas en clase).

Muestra las métricas de test y validación. Comenta si has usado *data augmentation*, de qué tipo, y, tal como se indicaba en la información al inicio del documento, explica qué arquitectura e hiperparámetros has utilizado.



**Importante** → Para anotar una base de datos, primero se debe escoger la arquitectura que se va a utilizar. Luego, se debe descargar un modelo ya existente y preentrenado con alguna base de datos extensa (e.g., COCO). Una vez hayas elegido la arquitectura, convierte los *outputs* del ejercicio anterior a un formato que dicha arquitectura entienda.

Por ejemplo, YOLO utiliza un formato concreto para definir las *bounding boxes*, por lo que deberás convertir las *bounding boxes* calculadas en el ejercicio anterior al formato YOLO. Esto no quiere decir recalcularlas, sino simplemente expresarlas en el formato requerido por la arquitectura escogida.