

# PRÁCTICA II

Pablo Chantada Saborido & José Romero Conde

---

## Introducción

## Clasificación Ship/No-ship

### Modelo base

Describimos ahora nuestro modelo baseline, que nos valió para iterar y comparar resultados. El modelo se compone de los siguientes elementos:

- **Efficientnet.** La usamos como CNN de partida. Tiene como requerimiento que las imágenes sean de tamaño (244, 244, 3), para ello implementamos un recortador automático de *el cuadrado más grande* que llamamos desde los `DataLoader` de entrenamiento y test. Aunque esta forma es claramente subóptima nos pareció adecuada para una primera aproximación. Mas adelante se comentarán los cambios
- **Aumento de datos.** El aumento de datos en esta fase esencialmente consta de dos partes, por un lado, las transformaciones de `torchvision.transforms`. Nosotros usamos: volteos horizontales, fluctuaciones leves en el color, también leves transformaciones afines y conversión a blanco y negro con baja probabilidad. El segundo punto es que hemos recortado manualmente las imágenes de barcos de forma que queden centrados y sin fondo que estorbe, se han añadido estas imágenes al conjunto de datos cuando se especificaba aumento de datos.
- Un **MLP** de tres capas, sobre la salida de la red. La salida de la última capa `nn.Softmax()` tiene 2 o 3 neuronas según si se quería clasificar en {barco, no barco} o en {barco, barco no amarrado, barco amarrado}.

### Entrenamiento y evaluación

## Clasificación Docked/Undocked

## Resultados

### Evaluación de modelos

### Análisis de resultados

## Conclusión

## Cambios para el segundo ejercicio