

# PROYECTO DEEP LEARNING

*Data Project 4 - Grupo 2*



# CONTENIDO

01

---

Resumen del  
proyecto

02

---

Objetivos  
y metas

03

---

Cronología del  
proyecto

04

---

Pre-Procesamiento

05

---

Técnicas de Deep  
Learning

06

---

Conclusiones



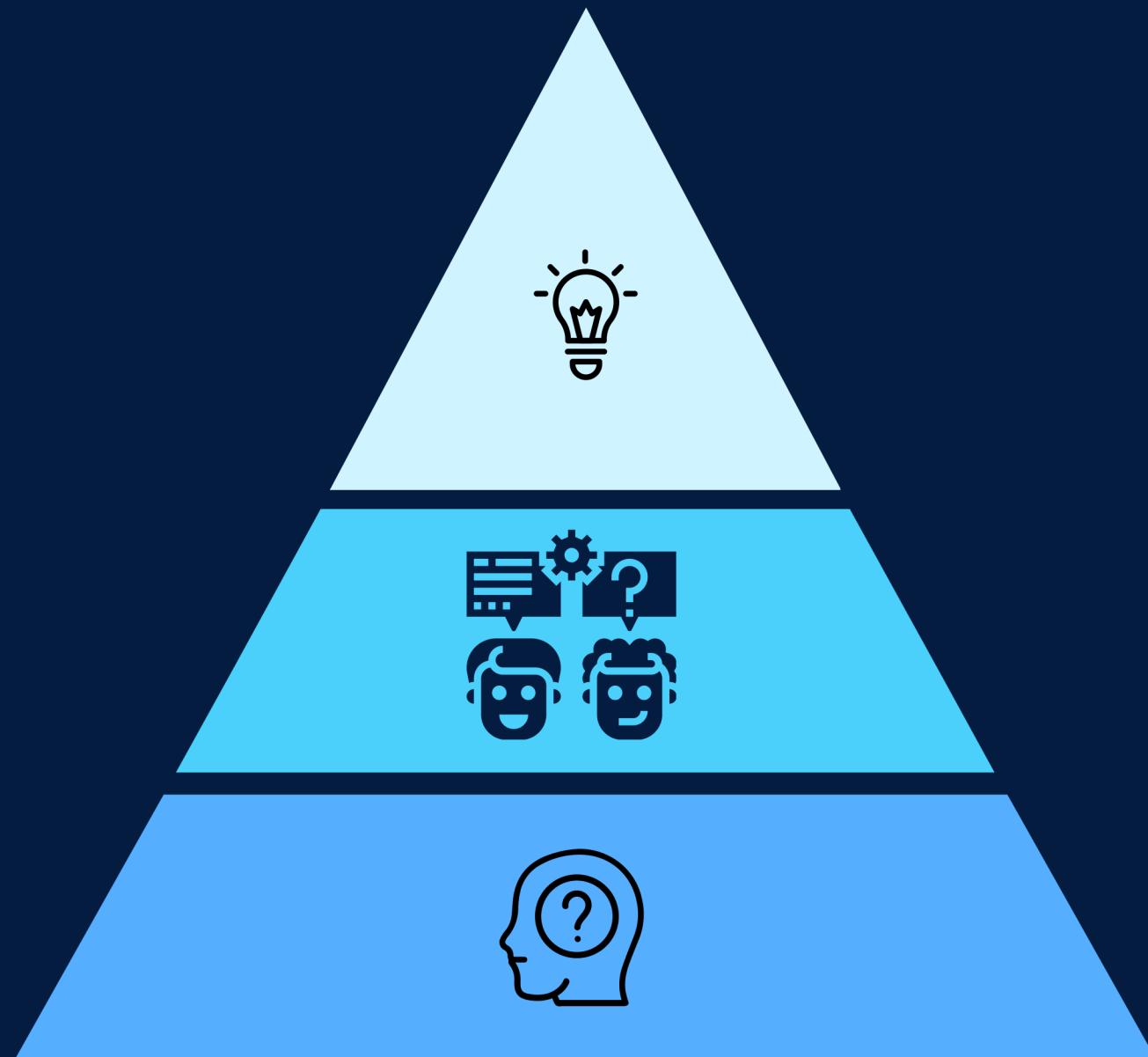
# INTRODUCCIÓN

El objetivo de este Data Project es entrenar un modelo predictivo capaz de determinar qué parte del cuerpo ha sido radiografiada utilizando técnicas de DeepLearning



# OBJETIVOS

- 01** Clasificar las imágenes en función de a que parte del cuerpo pertenecen.
- 02** Utilización de técnicas de preprocessamiento de imágenes óptimas.
- 03** Desarrollo de un modelo de apredizaje profundo con una precisión aceptable para el sector.



# CRONOLOGÍA



## Fase 1.

Análisis del problema y de sus requisitos. Reparto de tareas.

## Fase 2.

Preprocesamiento de imágenes

## Fase 3.

Entrenamiento de modelos con técnicas de Deep Learning.

## Fase 4.

Fase iterativa. Análisis de posibles mejoras en los modelos y en el procesamiento.

## Fase 5.

Ánalisis de resultados.

1

# Preprocesamiento

Procesar imágenes DICOM para mejorar el análisis y la visualización.

## 1. Preprocesamiento de una Imagen DICOM:

Función preprocess\_image(dicom)

Pasos:

Obtener la matriz de píxeles de la imagen DICOM.

Recortar las intensidades de los píxeles a los percentiles 1 y 99 para eliminar valores atípicos.

Normalizar los valores de píxeles al rango 0-255.

Convertir la imagen a un formato de 8 bits sin signo.

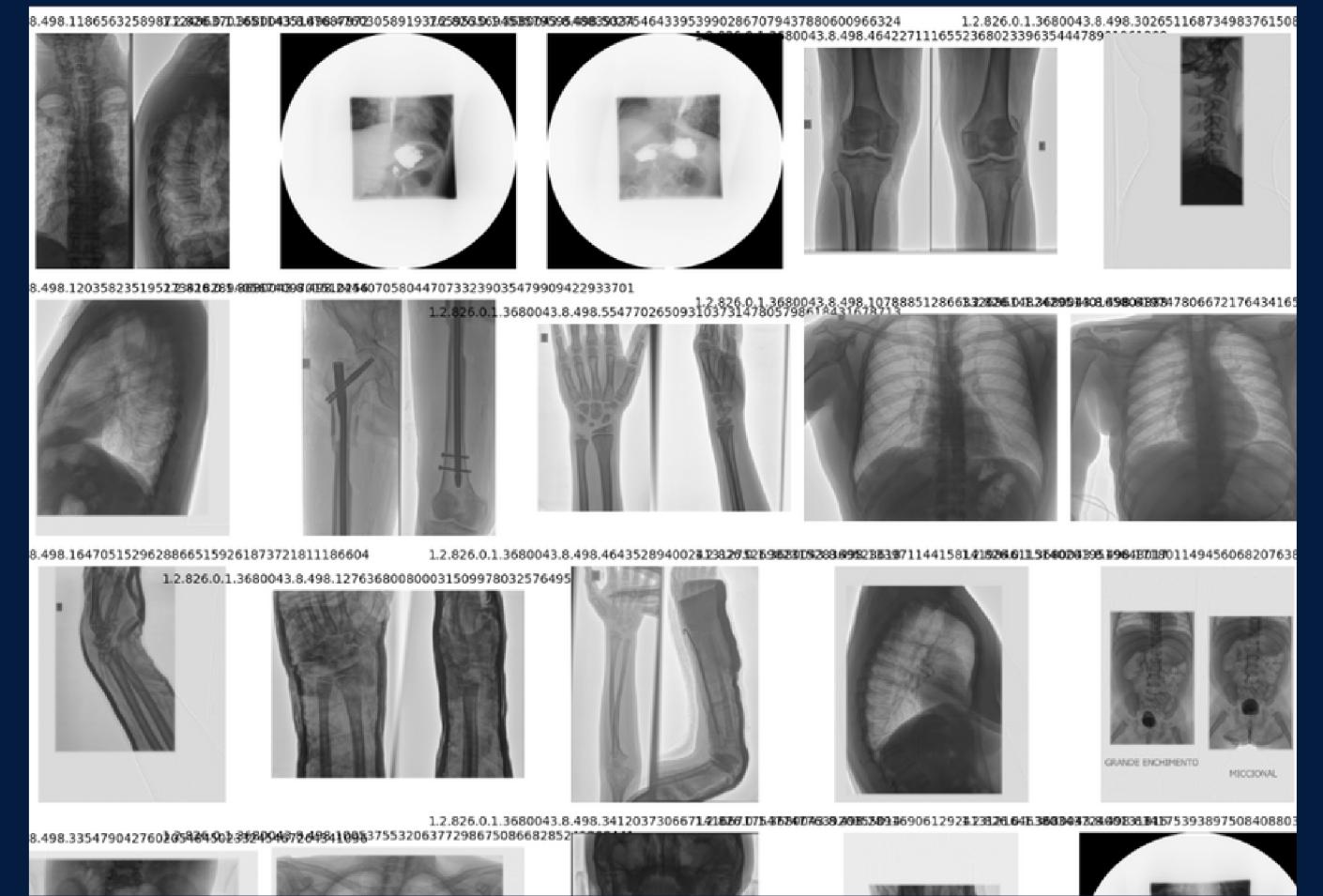
## 2. Convertir Imagen a RGB:

Función convert\_to\_rgb(img)

Pasos:

Convertir una imagen de escala de grises de un solo canal a una imagen RGB de tres canales duplicando el canal.

Para imágenes de dos canales, duplicar uno de los canales para crear un tercer canal.



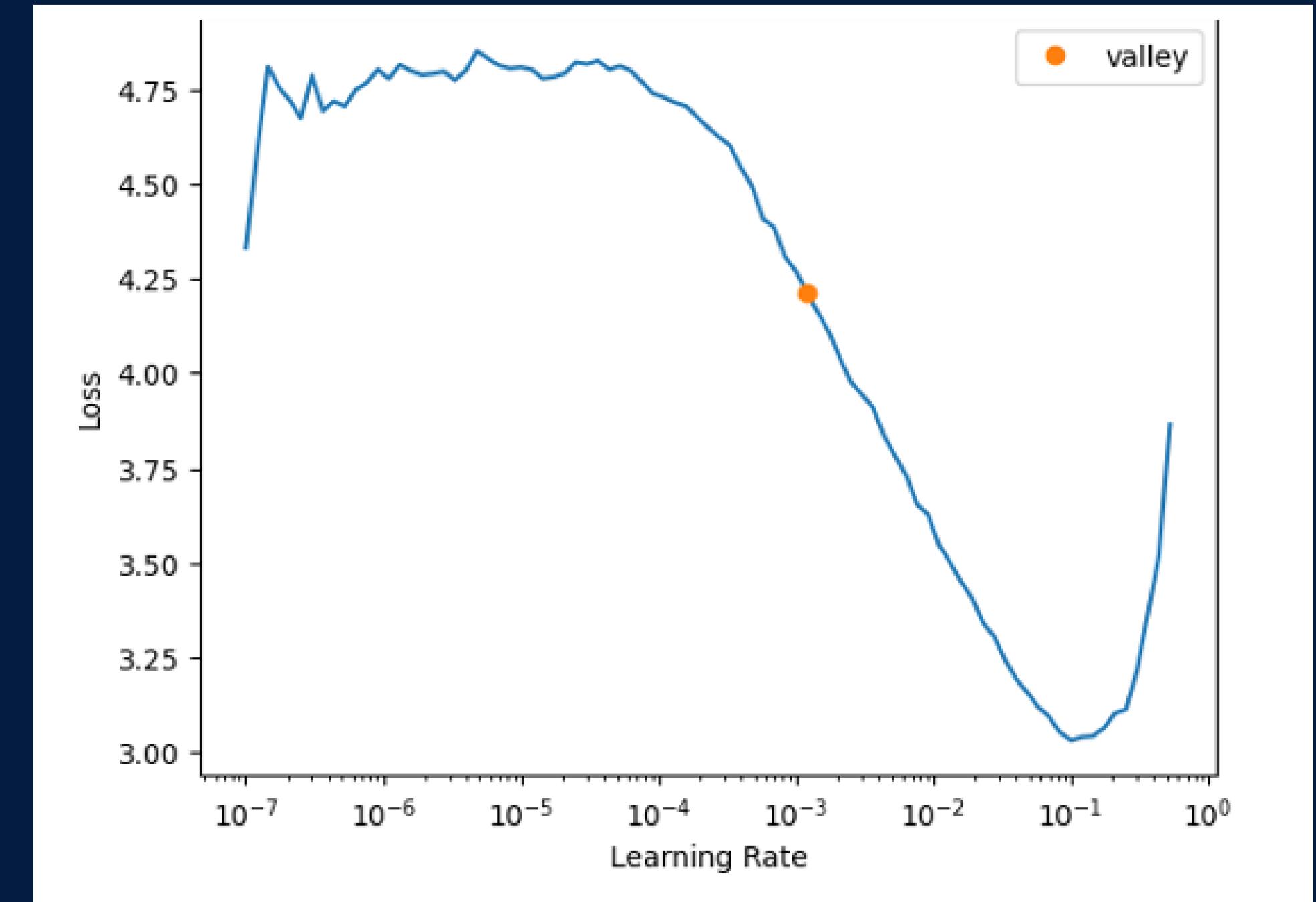
# Data Augmentation

Comprender las transformaciones de aumento de datos aplicadas durante la preparación de los modelos de entrenamiento.

- **Resize(224)**: Redimensiona las imágenes a un tamaño específico (224x224 píxeles).
- **aug\_transforms(...)**: Aplica varias transformaciones para aumentar la variabilidad de los datos:
- **flip\_vert=True**: Voltea verticalmente las imágenes con una probabilidad del 50%.
- **max\_warp=0.2**: Aplica deformaciones aleatorias con una intensidad máxima del 20%.
- **max\_rotate=30**: Gira las imágenes aleatoriamente hasta 30 grados.
- **max\_zoom=1.5**: Aplica zoom aleatorio con un factor máximo de 1.5 veces.
- **max\_lighting=0.4**: Ajusta aleatoriamente la iluminación de las imágenes hasta un 40%.
- **[RandomResizedCrop(size=(224, 224), min\_scale=0.6, max\_scale=1.4)]**: Realiza recortes redimensionados aleatorios dentro de un rango de escalas especificado.

# Optimización de hiperparámetros

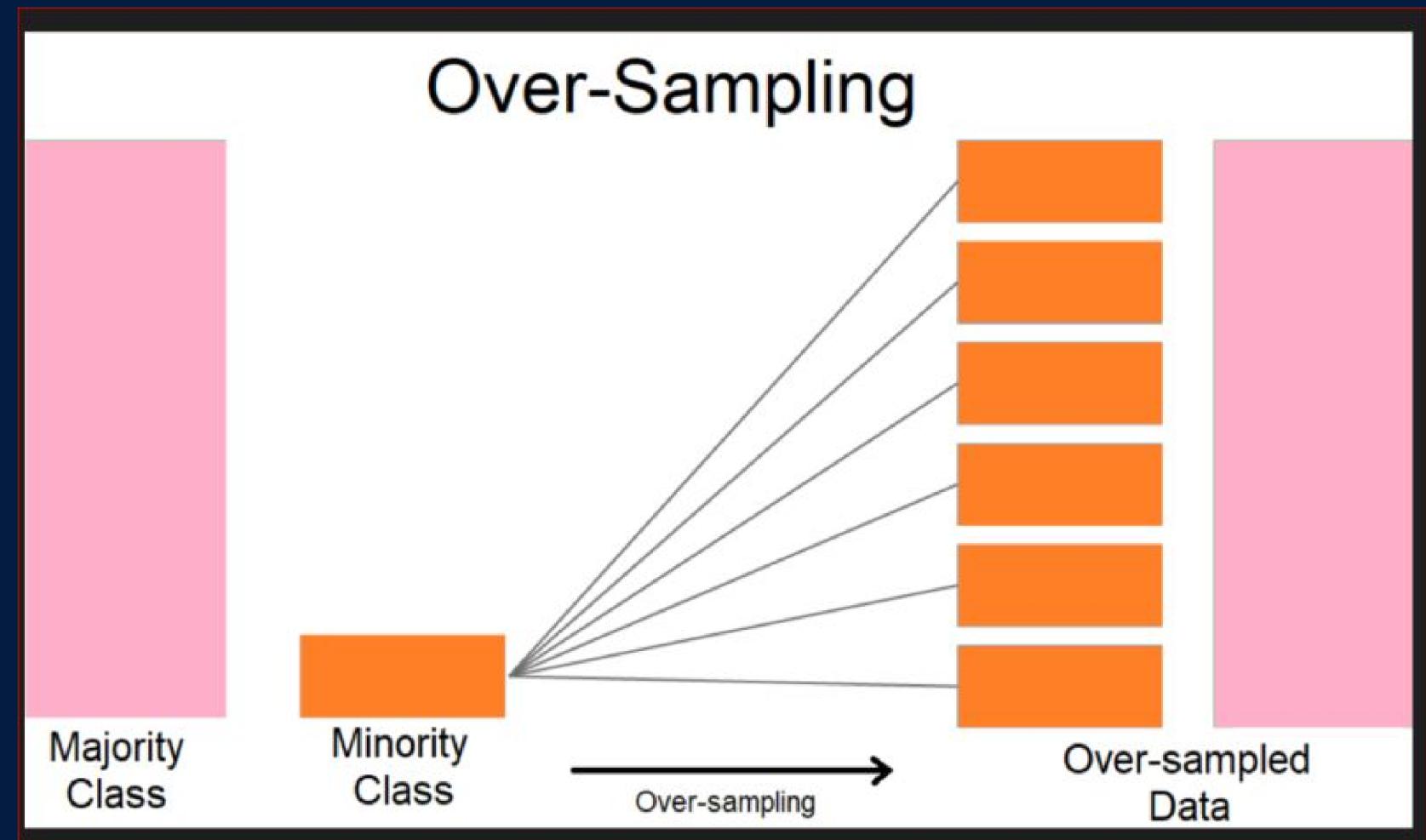
- Se crea un estudio de Optuna para mejorar la precisión del modelo.
- Se optimiza la función “objective” realizando 10 pruebas diferentes.
- Obtenemos los mejores hiperparámetros encontrados, en nuestro caso “best\_lr” y “best\_dropout”.



Visualización de los resultados obtenidos de la optimización por hiperparámetros.

# Desbalanceo de clases (Oversampling)

- Obtención de etiquetas
- Uso de RandomOverSampler para balancear las clases minoritarias.
- Genera DataLoaders para entrenamiento y validación con los datos resampleados.
- Mostramos un lote de imágenes aumentadas junto con sus etiquetas para verificar el efecto del oversampling.

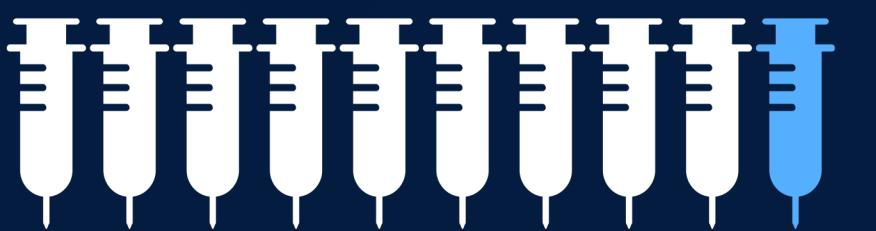


# DenseNet, FineTunning y Evaluación

- Utilización de LR para la optimización del “Learning Rate”. Se selecciona el sugerido basáandonos en el mínimo de pérdida.
- Se entrena el modelo con 3 epoch con el learning rate sugerido y se optimiza con la función “fit\_one\_cycle” el entrenamiento inicial.
- Aplicamos el método “unfreezing” para permitir que distintas capas ajusten sus pesos en el proceso de entrenamiento. Aplicamos 10 epoch para permitir a las capas este ajuste.
- Obtenemos una mejoría notable en la precisión del modelo a través de esta optimización.

# RESULTADOS

93,59%

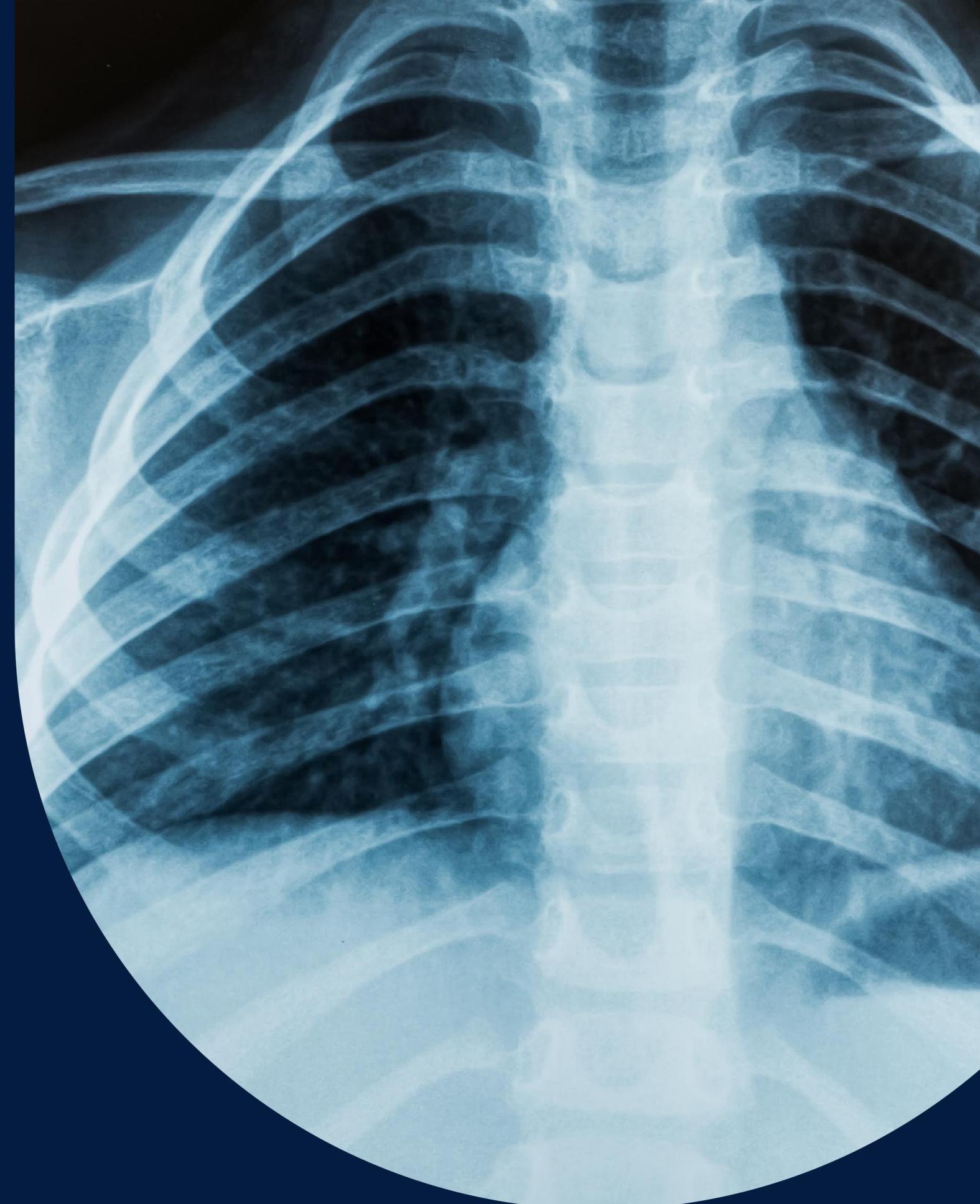


95,23%

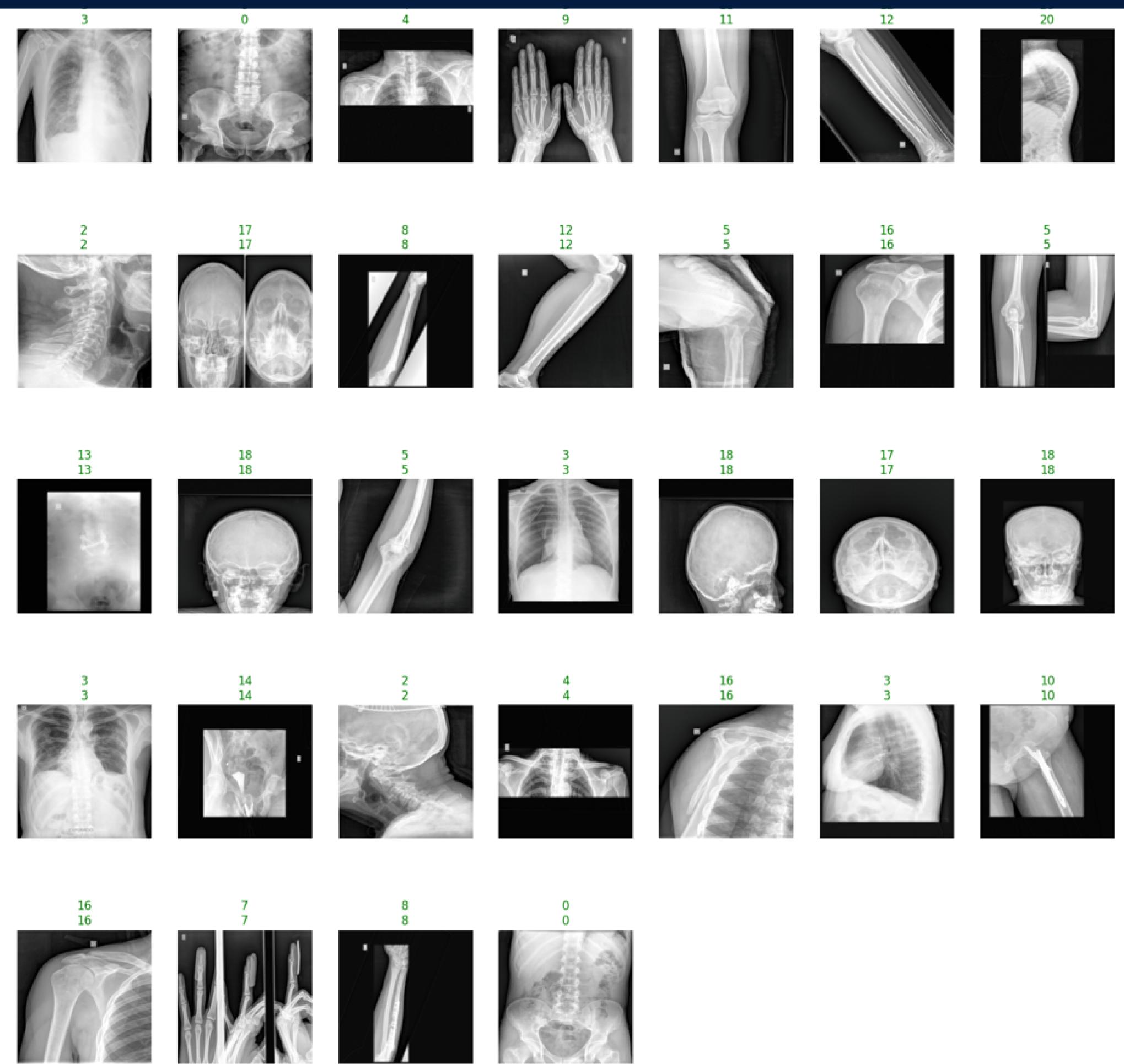


En la plataforma Kaggle, hemos obtenido un resultado máximo de F1-Score a la hora de evaluar nuestro modelo.

En nuestro entorno de evaluación, el resultado obtenido ha sido similar, demostrando la estabilidad del resultado.



Target	Frecuencia	Sección	X
3	146	Tórax	44.512195
14	25	Otros	7.621951
11	22	Rodilla	6.707317
2	17	Columna cervical	5.182927
15	16	Pelvis	4.878049
0	15	Abdomen	4.573171
21	15	Muñeca	4.573171
6	13	Pies	3.963415
1	10	Tobillo	3.048780
16	8	Hombro	2.439024
9	8	Mano	2.439024
13	6	Columna lumbar	1.829268
5	4	Codo	1.219512
4	3	Clavículas	0.914634
17	3	Senos paranasales	0.914634
18	3	Cráneo	0.914634
8	3	Antebrazo	0.914634
10	3	Cadera	0.914634
20	3	Columna torácica	0.914634
7	2	Dedos	0.609756
12	2	Pierna	0.609756
19	1	Muslo	0.304878



# CONCLUSIONES

- Vemos conveniente incrementar la cantidad de imágenes de entrenamiento a un mínimo de 10k correctamente balanceadas entre las distintas partes del cuerpo a identificar. Con esto obtendriamos un modelo mas robusto y confiable.
- Si bien el OverSampling arrojo resultados positivos, consideramos como positivo realizar una mejor selección de las radiografías de entrenamiento para consolidar la etapa de aprendizaje mejorando así la generalización del modelo.
- También observamos que la mayor cantidad de errores de estimación (68%) se centran en imágenes catalogadas como ‘otros’. El restante son errores entre abdomen y tórax, por lo que seria conveniente mejorar el entrenamiento en imágenes de estos segmentos.

# GRACIAS A TODOS

*Data Project 4 - Grupo 2*

Cristian Marty  
Eloy Martinez  
Andrés Cervera  
Lucía Esteve

