

Trabajo práctico integrador: Clasificación de Nódulos Pulmonares en Tomografías de Tórax

Profesores

Juan Ignacio Cornet - juanignaciocornet@gmail.com

Alfonso Rafel - alfonsorafel93@gmail.com

Seyed Pakdaman - khodadad.pakdaman@gmail.com

Alumnos

Ing. Pablo Martin Gomez Verdini - <u>gomezpablo86@gmail.com</u>
Ing. Diego Paciotti lacchelli - <u>diegopaciotti@gmail.com</u>
Ing. Joaquin Gonzalez - <u>joagonzalez@gmail.com</u>

• Repositorio Github

Motivación

Las enfermedades pulmonares, en particular el cáncer de pulmón, siguen siendo una de las principales causas de muerte a nivel global.

La detección temprana de nódulos pulmonares mediante tomografías computarizadas (CT-Scans) es fundamental para mejorar el pronóstico y la supervivencia de los pacientes.

Sin embargo, este proceso diagnóstico es complejo, demanda tiempo y depende fuertemente de la experiencia del profesional médico.

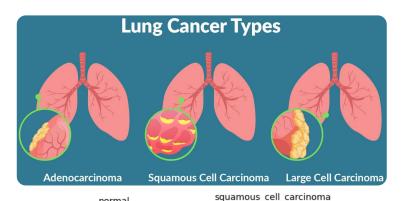


Objetivo

Este proyecto tiene como objetivo explorar el desempeño de modelos de aprendizaje profundo en la detección automática de nódulos pulmonares en imágenes de tomografía computarizada de tórax.

Utilizando el dataset público Chest CT-Scan Images Dataset de Kaggle, clasificaremos imágenes según la presencia o ausencia de nódulos.

Durante el desarrollo se evaluarán diferentes arquitecturas de redes neuronales, estrategias de data augmentation y el impacto del uso de transfer learning, con un enfoque exploratorio orientado a entender las capacidades y limitaciones actuales de estos modelos en el contexto médico.



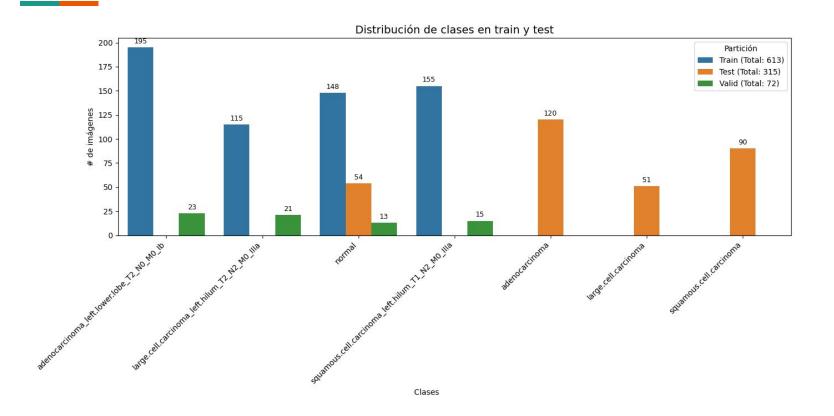
normal



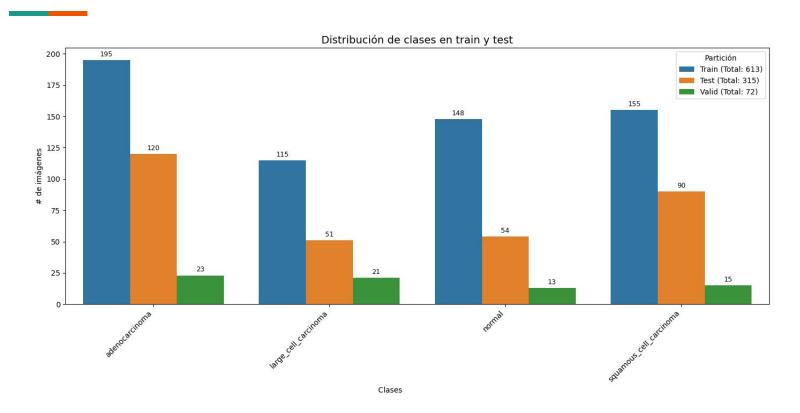




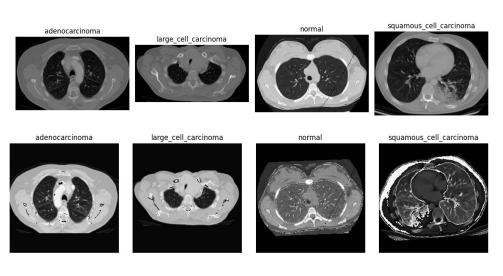
Análisis Exploratorio: Raw data



Análisis Exploratorio: Normalización de clases

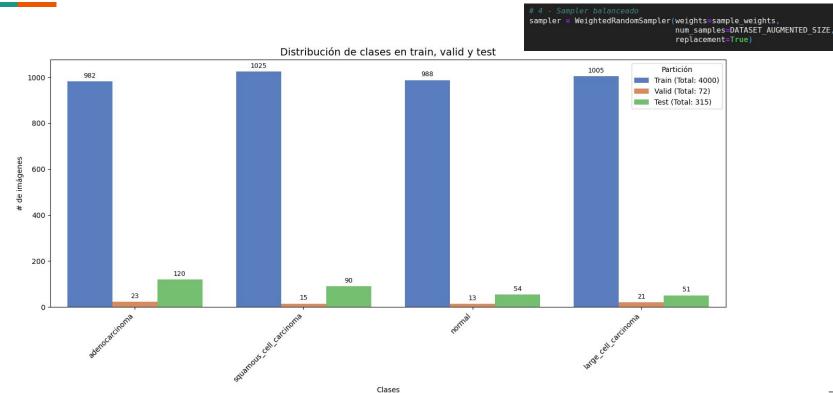


Análisis Exploratorio: Data Augmentation



```
train transform = transforms.Compose([
   transforms.Grayscale(),
   ResizeWithAspectRatioPadding(image size),
   transforms.RandomHorizontalFlip(p=0.5),
   transforms.RandomRotation(10),
   transforms ColorJitter(brightness=0.1, contrast=0.1), # Augmentations suaves
   transforms.ToTensor(),
   transforms.Normalize(mean=[0.5], std=[0.5])
valid transform = transforms.Compose([
   transforms.Grayscale(),
   ResizeWithAspectRatioPadding(image size),
   transforms.ToTensor(),
   transforms.Normalize(mean=[0.5], std=[0.5])
test transform = transforms.Compose([
   transforms.Grayscale(),
   ResizeWithAspectRatioPadding(image size),
   transforms.ToTensor(),
   transforms.Normalize(mean=[0.5], std=[0.5])
```

Análisis Exploratorio: Data Augmentation y balanceo

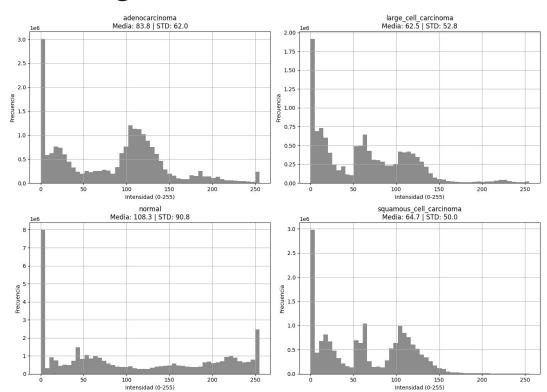


Análisis Exploratorio: Histogramas de intensidad

Se puede observar que las imágenes correspondientes a casos de cáncer presentan picos bien definidos en los histogramas de intensidad, con patrones distintivos entre subtipos.

Estas diferencias podrían reflejar variaciones estructurales en el tejido afectado.

En contraste, las imágenes de pacientes normales muestran una distribución más uniforme, con menor presencia de concentraciones marcadas en rangos específicos de intensidad.



Clasificación: Uso de CNNs

Arquitectura con 5 bloques conv. y activación ReLU

Batch Norm. y Max Pooling

Dropout2D para lidiar con overfitting

Arq. progresiva de filtros para capturar de patrones más simples a más complejos (32-512)

Capas densas para explotar features de capas conv. para clasificación

No se agregan más capas ya que se tuvieron que implementar técnicas de normalización para evitar overfitting.

Training sale por Early Stopping

```
(conv1): Sequential(
  (0): Conv2d(1, 32, kernel size=(3, 3), stride=(1, 1), padding=same)
 (1): BatchNorm2d(32, eps=le-05, momentum=0.1, affine=True, track running stats=True)
 (2): ReLU(inplace=True)
 (3): MaxPool2d(kernel size=2, stride=2, padding=0, dilation=1, ceil mode=False)
(conv2): Sequential(
  (0): Conv2d(32, 64, kernel size=(3, 3), stride=(1, 1), padding=same)
 (1): BatchNorm2d(64, eps=le-05, momentum=0.1, affine=True, track running stats=True)
 (2): ReLU(inplace=True)
  (3): MaxPool2d(kernel size=2, stride=2, padding=0, dilation=1, ceil mode=False)
(conv3): Seguential(
  (0): Conv2d(64, 128, kernel size=(3, 3), stride=(1, 1), padding=same)
 (1): BatchNorm2d(128, eps=le-05, momentum=0.1, affine=True, track running stats=True)
  (2): ReLU(inplace=True)
 (3): MaxPool2d(kernel size=2, stride=2, padding=0, dilation=1, ceil mode=False)
(conv4): Sequential(
  (0): Conv2d(128, 256, kernel size=(3, 3), stride=(1, 1), padding=same)
 (1): BatchNorm2d(256, eps=le-05, momentum=0.1, affine=True, track running stats=True)
  (2): ReLU(inplace=True)
 (3): MaxPool2d(kernel size=2, stride=2, padding=0, dilation=1, ceil mode=False)
(conv5): Sequential(
  (0): Conv2d(256, 512, kernel size=(3, 3), stride=(1, 1), padding=same)
 (1): BatchNorm2d(512, eps=le-05, momentum=0.1, affine=True, track running stats=True)
 (2): ReLU(inplace=True)
  (3): MaxPool2d(kernel size=2, stride=2, padding=0, dilation=1, ceil mode=False)
(dropout): Dropout2d(p=0.15, inplace=False)
(fc): Sequential(
  (0): Flatten(start dim=1, end dim=-1)
  (1): Linear(in features=25088, out features=512, bias=True)
  (2): ReLU()
 (3): Dropout(p=0.4, inplace=False)
 (4): Linear(in features=512, out features=256, bias=True)
  (5): ReLU()
 (6): Dropout(p=0.25, inplace=False)
 (7): Linear(in features=256, out features=4, bias=True)
```

Clasificación: Uso de CNNs

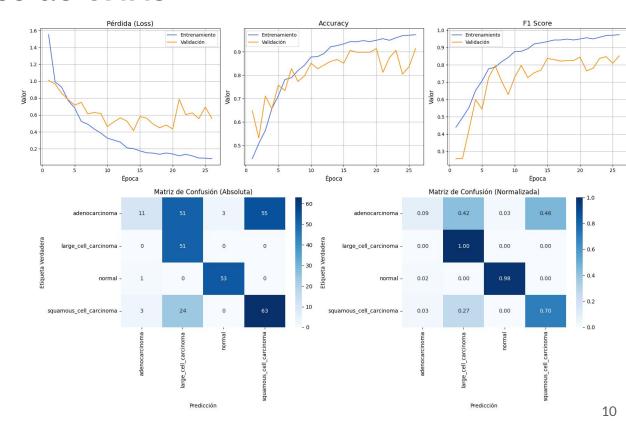
Entrenamiento con 224x224 grayscale images

La red performa bien en 3 de 4 clases, que por su baja performance tirá el score para abajo

Parece haber problemas en diferenciar adenocarcinomas de las otras variantes

La validación se estabiliza en época 15, ya no se mejora en generalización

A pesar de convergencia en entrenamiento no se mejora en separación de clases de carcinomas



Clasificación: Uso de CNNs

Histogramas muy similares, dificultades para separar clases

Buen modelo para clasificación binaria

Indicios de modelo saturado, se trataran arquitecturas más complejas con transfer learning

Se toma como baseline para las próximas pruebas

	precision	recall	fl-score	support
adenocarcinoma	0.73	0.09	0.16	120
large cell carcinoma	0.40	1.00	0.58	51
normal	0.95	0.98	0.96	54
squamous_cell_carcinoma	0.53	0.70	0.61	90
accuracy			0.57	315
macro avg	0.65	0.69	0.58	315
weighted avg	0.66	0.57	0.49	315
Accuracy: 0.5651				
F1 Score (weighted): 0.4 Loss: 1.4835	937			

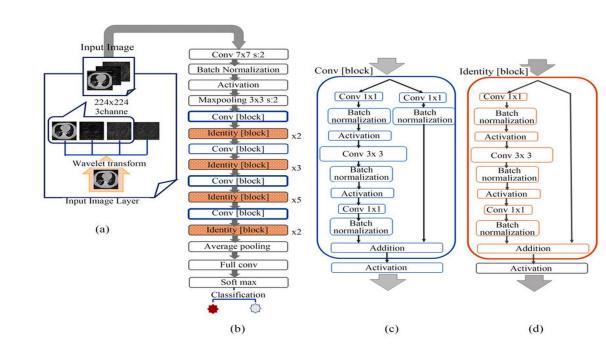
Clasificación: Transfer Learning Resnet-50

ResNet-50 mejora a una CNN tradicional al permitir entrenar redes mucho más profundas sin degradar el rendimiento, gracias a sus conexiones residuales que evitan el desvanecimiento del gradiente.

Estas conexiones residuales, o skip connections, permiten el flujo directo del gradiente entre capas, lo que estabiliza el entrenamiento y facilita la optimización de redes muy profundas.

Compuesta por dos tipos de bloques: Identity blocks y Convolutional blocks.

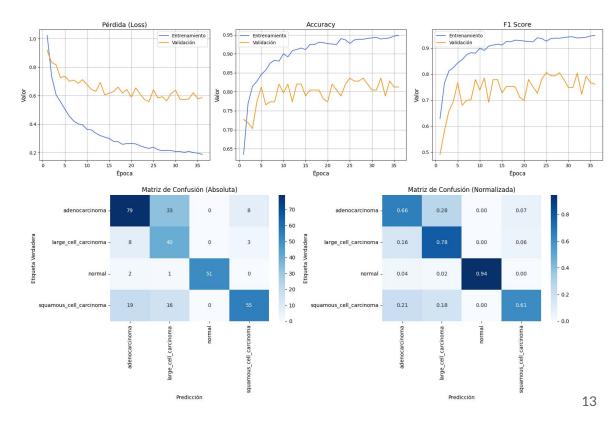
Modelo pre-entrenado con ImageNet (1.2 millones de imágenes y 1000 clases)



Resnet-50: Fine tuning capa FC

Para el primer Fine-tuning se modificó la última capa FC para para poder clasificar a los 4 tipos de cánceres, ya que el modelo original predice 1000 clases de Imagenet.

El modelo mejoró el weighted average f1-score respecto al baseline de la CNN en un 63% solo cambiando la última capa de clasificación.

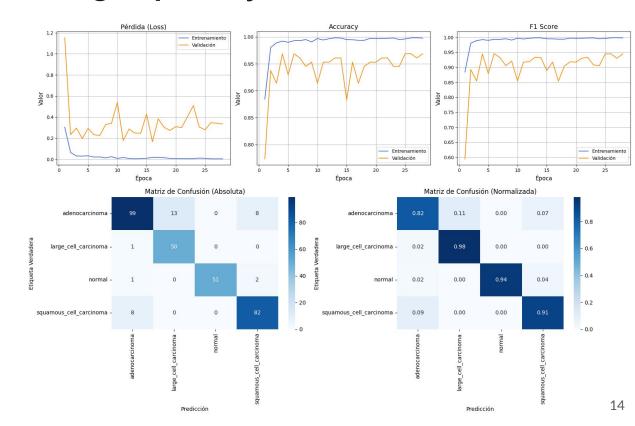


Resnet-50: Fine tuning capa FC y ult. conv.

El segundo fine tuning se realizó modificando, no sólo la última capa FC, sino también la última capa convolucional.

Esto produjo el mejor modelo al momento con un weighted average f1-score de **0.895** que para el caso, representa una mejora adicional del **11%** (~70% con respecto a baseline).

Esto se debe a que las últimas capas son las que capturan los detalles globales de la imagen (mayor campo receptivo).

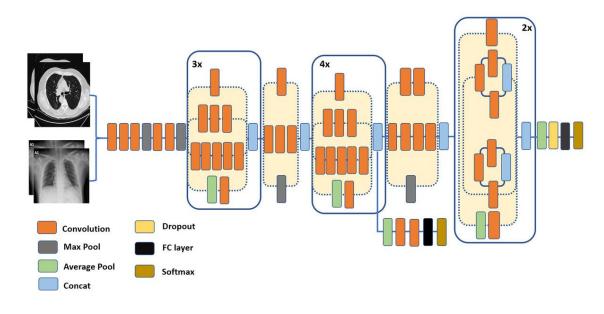


Clasificación: Transfer Learning Inception-V3

Se eligió Inception-V3 debido a su capacidad para capturar patrones a múltiples escalas gracias a sus módulos Inception, que combinan convoluciones de diferentes tamaños en paralelo.

Esta arquitectura permite que el modelo extraiga tanto detalles finos como estructuras globales en las imágenes médicas, mejorando su capacidad de discriminación entre clases con diferencias morfológicas sutiles.

Además, Inception-V3 incorpora técnicas como la factorización de convoluciones y la normalización auxiliar, lo que reduce el costo computacional sin sacrificar profundidad ni rendimiento.



https://peerj.com/articles/cs-303/

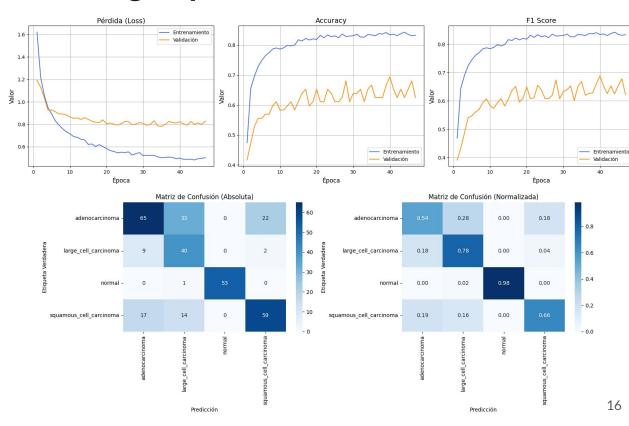
Inception-V3: Fine tuning capa FC

Para el caso de **Inception-v3**, al igual que en **Resnet-50**, se realizó un fine tuning de la última capa FC para poder clasificar los 4 tipos de cánceres.

Inception logró una mejora del **47**% respecto al baseline de CNN.

Resnet-50 = 1.63 * CNN

Inception-V3 = 1.47 * CNN

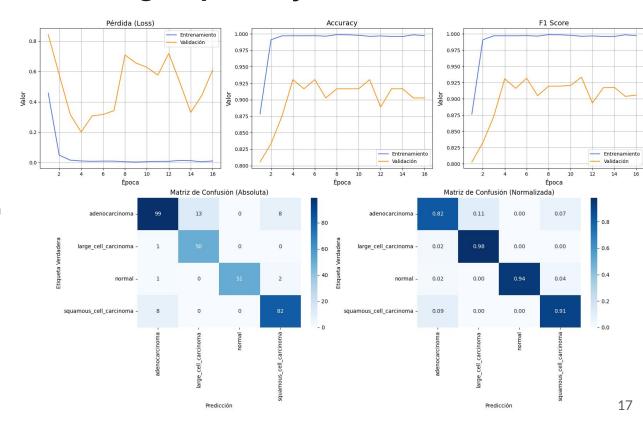


Inception-V3: Fine tuning capa FC y ult. conv.

El segundo fine tuning se realizó modificando, no sólo la última capa FC, sino también la última capa convolucional.

Esto produjo el mejor modelo al momento, con un weighted avg F1-score de 0.87, que para el caso representa una mejora adicional del 20% respecto al primer fine tuning (que alcanzaba solo un 0.72 en validación).

Esto se debe a que las últimas capas son las que capturan los detalles globales de la imagen (mayor campo receptivo), lo cual resulta clave para distinguir estructuras sutiles entre clases difíciles como adenocarcinoma y squamous cell carcinoma.



Conclusiones

En nuestro estudio, se evaluaron diferentes arquitecturas bajo un esquema de fine-tuning completo, utilizando el mismo dataset y configuración experimental para asegurar una comparación justa. Los resultados obtenidos son los siguientes:

- ResNet-50 obtuvo el mejor desempeño, con un F1-Score promedio de 0.895.
- Inception-V3 se posicionó en segundo lugar, con un F1-Score promedio de 0.876.
- Por último, la **CNN simple** alcanzó un F1-Score promedio de 0.493, mostrando claras limitaciones frente a modelos más profundos y pre entrenados.

Estos resultados son consistentes con la capacidad de representación de cada modelo, relacionada con su número de parámetros entrenables. Aunque Inception-V3 tiene una mayor cantidad de parámetros que ResNet-50, esta última mostró una mejor capacidad de generalización, lo cual podría atribuirse a su diseño con conexiones residuales que facilitan el entrenamiento profundo.

Modelo	Parámetros Entrenables	F1-Score Promedio	
ResNet-50	25,557,032	0.895	
Inception-V3	27,161,264	0.876	
CNN Simple	14,547,908	0.493	

Muchas gracias por su atención!

