

IMPORTAR LIBRERÍAS Y CONJUNTO DE DATOS

```
import tensorflow as tf
import numpy as np
from tensorflow.keras import layers, optimizers
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.layers import Input, Add, Dense, Activation, ZeroPadding2D, BatchNormalization, Flatten, Conv2D
from tensorflow.keras.models import Model, load_model
from tensorflow.keras import backend as K
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.callbacks import ReduceLROnPlateau, EarlyStopping, ModelCheckpoint, LearningRateScheduler
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

In []:

```
# Directorio del conjunto de entrenamiento
XRay_Directory = "A:/Google Drive/Proyectos IA/O4_Proyectos_Machine_Learning/Detección enfermedades pulmonares"
```

In []:

```
# Listado de los archivos de dicho directorio
os.listdir(XRay_Directory)
```

Out[]:

```
[\"'1\", \"0\", \"3\", \"2\"]
```

In []:

```
# Se utiliza el generador de imágenes para generar datos de imágenes de tensores y normalizarlos
# Utilizar el 20% de los datos para la validación cruzada posterior
image_generator = ImageDataGenerator(rescale=1./255, validation_split = 0.2)
```

In []:

```
# El número total de imágenes es 133 * 4 = 532 imágenes
# El entrenamiento usará las 428 (80%) y la validación usará 104 (20%)
# Se crea muestra aleatoria y cambio del tamaño en las imágenes
train_generator = image_generator.flow_from_directory(batch_size = 40, directory = XRay_Directory, shuffle = True, target_size = (256, 256), class_mode = "categorical", su
```

Found 428 images belonging to 4 classes.

In []:

```
train_generator = image_generator.flow_from_directory(batch_size = 40, directory = XRay_Directory, shuffle = True, target_size = (256, 256), class_mode = "categorical", su
```

Found 104 images belonging to 4 classes.

In []:

```
# Se genera un lote de 40 imágenes y etiquetas
train_images, train_labels = next(train_generator)
```

In []:

```
train_images.shape
```

Out[]:

```
(40, 256, 256, 3)
```

In []:

```
train_labels.shape
```

Out[]:

```
(40, 4)
```

In []:

```
train_labels
```

Out[]:

```
array([[1., 0., 0., 0.],
       [0., 1., 1., 0.],
       [0., 1., 0., 0.],
       [0., 1., 0., 0.],
       [0., 1., 0., 0.],
       [1., 0., 0., 0.],
       [1., 0., 0., 0.],
       [0., 0., 0., 1.],
       [0., 0., 0., 1.],
       [0., 0., 0., 1.],
       [1., 0., 0., 0.],
       [1., 0., 0., 0.],
       [1., 0., 0., 0.],
       [0., 0., 0., 1.],
       [0., 0., 0., 1.],
       [0., 0., 0., 1.],
       [0., 0., 0., 1.],
       [0., 1., 0., 0.],
       [0., 1., 0., 0.],
       [0., 1., 0., 0.],
       [0., 1., 0., 0.],
       [0., 0., 0., 1.],
       [0., 0., 0., 1.],
       [0., 0., 0., 1.],
       [0., 0., 0., 1.],
       [0., 0., 0., 1.],
       [0., 0., 0., 1.],
       [0., 0., 0., 1.],
       [0., 0., 0., 1.],
       [0., 0., 0., 1.],
       [0., 0., 0., 1.],
       [0., 0., 0., 1.]]) dtype=float32)
```

In []:

```
# Traducción de cada Etiqueta
label_names = {0: "COVID-19", 1: "Normal", 2: "Pneumonia Virica", 3: "Pneumonia Bacteriana"}
```

VISUALIZACIÓN DEL DATASET

In []:

```
# Se crea una matriz de 36 imágenes junto con sus etiquetas correspondientes
L = 6
W = 6

fig, axes = plt.subplots(L, W, figsize = (12,12))
axes = axes.ravel()

for i in np.arange(0, L*W):
    axes[i].imshow(train_images[i])
    axes[i].set_title(label_names[np.argmax(train_labels[i])])
    axes[i].axis('off')
```

plt.subplots_adjust(wspace = 0.5)

The visualization shows a grid of chest X-ray images categorized by disease type. The labels are COVID-19, Pneumonia Virica, Normal, and Pneumonia Bacteriana. Some images show clear lung fields (Normal), while others show characteristic patterns like consolidation or ground-glass opacities associated with pneumonia or COVID-19.

| | | |
|-------------------------------|---------------------------|------------------|
| conv1_pad (ZeroPadding2D) | (None, 262, 262, 3) 0 | input_1[0][0] |
| conv1_conv (Conv2D) | (None, 128, 128, 64) 9472 | conv1_pad[0][0] |
| conv1_bn (BatchNormalization) | (None, 128, 128, 64) 256 | conv1_conv[0][0] |
| conv1_relu (Activation) | (None, 128, 128, 64) 0 | conv1_bn[0][0] |

```

pool1_pool (ZeroPadding2D) (None, 128, 128, 64) 0 conv1_relu[0][0]
pool1_pool (MaxPooling2D) (None, 64, 64, 64) 0 pool1_pool[0][0]
conv2_block1_1_conv (Conv2D) (None, 64, 64, 64) 4160 conv1_pool[0][0]
conv2_block1_1_bn (BatchNormal) (None, 64, 64, 64) 256 conv2_block1_1_conv[0][0]
conv2_block1_1_relu (Activation) (None, 64, 64, 64) 0 conv2_block1_1_bn[0][0]
conv2_block1_2_conv (Conv2D) (None, 64, 64, 64) 36928 conv2_block1_1_relu[0][0]
conv2_block1_2_bn (BatchNormal) (None, 64, 64, 64) 256 conv2_block1_2_conv[0][0]
conv2_block1_2_relu (Activation) (None, 64, 64, 64) 0 conv2_block1_2_bn[0][0]
conv2_block1_3_conv (Conv2D) (None, 64, 64, 256) 16640 pool1_pool[0][0]
conv2_block1_3_bn (BatchNormal) (None, 64, 64, 256) 1024 conv2_block1_3_conv[0][0]
conv2_block1_add (Add) (None, 64, 64, 256) 0 conv2_block1_3_bn[0][0]
conv2_block1_out (Activation) (None, 64, 64, 256) 0 conv2_block1_add[0][0]
conv2_block2_1_conv (Conv2D) (None, 64, 64, 64) 16448 conv2_block1_out[0][0]
conv2_block2_1_bn (BatchNormal) (None, 64, 64, 64) 256 conv2_block2_1_conv[0][0]
conv2_block2_2_relu (Activation) (None, 64, 64, 64) 0 conv2_block2_1_bn[0][0]
conv2_block2_2_conv (Conv2D) (None, 64, 64, 64) 36928 conv2_block2_2_relu[0][0]
conv2_block2_2_bn (BatchNormal) (None, 64, 64, 64) 256 conv2_block2_2_conv[0][0]
conv2_block2_3_conv (Conv2D) (None, 64, 64, 64) 0 conv2_block2_2_bn[0][0]
conv2_block2_3_bn (BatchNormal) (None, 64, 64, 256) 1024 conv2_block2_3_conv[0][0]
conv2_block2_add (Add) (None, 64, 64, 256) 0 conv2_block2_3_bn[0][0]
conv2_block2_out (Activation) (None, 64, 64, 256) 0 conv2_block2_add[0][0]
conv2_block3_1_conv (Conv2D) (None, 64, 64, 64) 16448 conv2_block2_out[0][0]
conv2_block3_1_bn (BatchNormal) (None, 64, 64, 64) 256 conv2_block3_1_conv[0][0]
conv2_block3_2_relu (Activation) (None, 64, 64, 64) 0 conv2_block3_1_bn[0][0]
conv2_block3_2_conv (Conv2D) (None, 64, 64, 64) 36928 conv2_block3_2_relu[0][0]
conv2_block3_2_bn (BatchNormal) (None, 64, 64, 64) 256 conv2_block3_2_conv[0][0]
conv2_block3_3_relu (Activation) (None, 64, 64, 64) 0 conv2_block3_2_bn[0][0]
conv2_block3_3_conv (Conv2D) (None, 64, 64, 256) 16640 conv2_block3_3_relu[0][0]
conv2_block3_3_bn (BatchNormal) (None, 64, 64, 256) 1024 conv2_block3_3_conv[0][0]
conv2_block3_add (Add) (None, 64, 64, 256) 0 conv2_block3_3_bn[0][0]
conv2_block3_out (Activation) (None, 64, 64, 256) 0 conv2_block3_add[0][0]
conv3_block1_1_conv (Conv2D) (None, 32, 32, 128) 32896 conv2_block3_out[0][0]
conv3_block1_1_bn (BatchNormal) (None, 32, 32, 128) 512 conv3_block1_1_conv[0][0]
conv3_block1_2_relu (Activation) (None, 32, 32, 128) 0 conv3_block1_1_bn[0][0]
conv3_block1_2_conv (Conv2D) (None, 32, 32, 128) 147584 conv3_block1_2_relu[0][0]
conv3_block1_2_bn (BatchNormal) (None, 32, 32, 128) 512 conv3_block1_2_conv[0][0]
conv3_block1_3_relu (Activation) (None, 32, 32, 128) 0 conv3_block1_2_bn[0][0]
conv3_block1_3_conv (Conv2D) (None, 32, 32, 512) 131584 conv3_block1_3_relu[0][0]
conv3_block1_3_bn (BatchNormal) (None, 32, 32, 512) 66408 conv3_block1_3_conv[0][0]
conv3_block1_add (Add) (None, 32, 32, 512) 0 conv3_block1_3_bn[0][0]
conv3_block1_out (Activation) (None, 32, 32, 512) 0 conv3_block1_add[0][0]
conv3_block2_1_conv (Conv2D) (None, 32, 32, 128) 65664 conv3_block1_out[0][0]
conv3_block2_1_bn (BatchNormal) (None, 32, 32, 128) 512 conv3_block2_1_conv[0][0]
conv3_block2_2_relu (Activation) (None, 32, 32, 128) 0 conv3_block2_1_bn[0][0]
conv3_block2_2_conv (Conv2D) (None, 32, 32, 128) 147584 conv3_block2_2_relu[0][0]
conv3_block2_2_bn (BatchNormal) (None, 32, 32, 128) 512 conv3_block2_2_conv[0][0]
conv3_block2_3_conv (Conv2D) (None, 32, 32, 512) 66408 conv3_block2_3_conv[0][0]
conv3_block2_3_bn (BatchNormal) (None, 32, 32, 512) 2048 conv3_block2_3_bn[0][0]
conv3_block2_add (Add) (None, 32, 32, 512) 0 conv3_block2_3_bn[0][0]
conv3_block2_out (Activation) (None, 32, 32, 512) 0 conv3_block2_add[0][0]
conv3_block3_1_conv (Conv2D) (None, 32, 32, 128) 65664 conv3_block2_out[0][0]
conv3_block3_1_bn (BatchNormal) (None, 32, 32, 128) 512 conv3_block3_1_conv[0][0]
conv3_block3_2_relu (Activation) (None, 32, 32, 128) 0 conv3_block3_1_bn[0][0]
conv3_block3_2_conv (Conv2D) (None, 32, 32, 128) 147584 conv3_block3_2_relu[0][0]
conv3_block3_2_bn (BatchNormal) (None, 32, 32, 128) 512 conv3_block3_2_conv[0][0]
conv3_block3_3_relu (Activation) (None, 32, 32, 128) 0 conv3_block3_2_bn[0][0]
conv3_block3_3_conv (Conv2D) (None, 32, 32, 512) 66408 conv3_block3_3_relu[0][0]
conv3_block3_3_bn (BatchNormal) (None, 32, 32, 512) 2048 conv3_block3_3_conv[0][0]
conv3_block3_add (Add) (None, 32, 32, 512) 0 conv3_block3_3_bn[0][0]
conv3_block3_out (Activation) (None, 32, 32, 512) 0 conv3_block3_add[0][0]
conv3_block4_1_conv (Conv2D) (None, 32, 32, 128) 65664 conv3_block3_out[0][0]
conv3_block4_1_bn (BatchNormal) (None, 32, 32, 128) 512 conv3_block4_1_conv[0][0]
conv3_block4_2_relu (Activation) (None, 32, 32, 128) 0 conv3_block4_1_bn[0][0]
conv3_block4_2_conv (Conv2D) (None, 32, 32, 128) 147584 conv3_block4_2_relu[0][0]
conv3_block4_2_bn (BatchNormal) (None, 32, 32, 128) 512 conv3_block4_2_conv[0][0]
conv3_block4_3_conv (Conv2D) (None, 32, 32, 512) 66408 conv3_block4_3_conv[0][0]
conv3_block4_3_bn (BatchNormal) (None, 32, 32, 512) 2048 conv3_block4_3_bn[0][0]
conv3_block4_add (Add) (None, 32, 32, 512) 0 conv3_block4_3_bn[0][0]
conv3_block4_out (Activation) (None, 32, 32, 512) 0 conv3_block4_add[0][0]
conv4_block1_1_conv (Conv2D) (None, 16, 16, 512) 131328 conv3_block4_out[0][0]
conv4_block1_1_bn (BatchNormal) (None, 16, 16, 512) 1024 conv4_block1_1_conv[0][0]
conv4_block1_2_relu (Activation) (None, 16, 16, 512) 0 conv4_block1_1_bn[0][0]
conv4_block1_2_conv (Conv2D) (None, 16, 16, 256) 590080 conv4_block1_2_relu[0][0]
conv4_block1_2_bn (BatchNormal) (None, 16, 16, 256) 1024 conv4_block1_2_conv[0][0]
conv4_block1_3_relu (Activation) (None, 16, 16, 256) 0 conv4_block1_2_bn[0][0]
conv4_block1_3_conv (Conv2D) (None, 16, 16, 1024) 263168 conv4_block1_3_relu[0][0]
conv4_block1_3_bn (BatchNormal) (None, 16, 16, 1024) 4096 conv4_block1_3_conv[0][0]
conv4_block1_add (Add) (None, 16, 16, 1024) 0 conv4_block1_3_bn[0][0]
conv4_block1_out (Activation) (None, 16, 16, 1024) 0 conv4_block1_add[0][0]
conv4_block2_1_conv (Conv2D) (None, 16, 16, 256) 262400 conv4_block1_out[0][0]
conv4_block2_1_bn (BatchNormal) (None, 16, 16, 256) 1024 conv4_block2_1_conv[0][0]
conv4_block2_2_relu (Activation) (None, 16, 16, 256) 0 conv4_block2_1_bn[0][0]
conv4_block2_2_conv (Conv2D) (None, 16, 16, 256) 590080 conv4_block2_2_relu[0][0]
conv4_block2_2_bn (BatchNormal) (None, 16, 16, 256) 1024 conv4_block2_2_conv[0][0]
conv4_block2_3_conv (Conv2D) (None, 16, 16, 1024) 263168 conv4_block2_3_conv[0][0]
conv4_block2_3_bn (BatchNormal) (None, 16, 16, 1024) 4096 conv4_block2_3_bn[0][0]
conv4_block2_add (Add) (None, 16, 16, 1024) 0 conv4_block2_3_bn[0][0]
conv4_block2_out (Activation) (None, 16, 16, 1024) 0 conv4_block2_add[0][0]
conv4_block3_1_conv (Conv2D) (None, 16, 16, 256) 262400 conv4_block2_out[0][0]
conv4_block3_1_bn (BatchNormal) (None, 16, 16, 256) 1024 conv4_block3_1_conv[0][0]
conv4_block3_2_relu (Activation) (None, 16, 16, 256) 0 conv4_block3_1_bn[0][0]
conv4_block3_2_conv (Conv2D) (None, 16, 16, 256) 590080 conv4_block3_2_relu[0][0]
conv4_block3_2_bn (BatchNormal) (None, 16, 16, 256) 1024 conv4_block3_2_conv[0][0]
conv4_block3_3_conv (Conv2D) (None, 16, 16, 1024) 263168 conv4_block3_3_conv[0][0]
conv4_block3_3_bn (BatchNormal) (None, 16, 16
```

```
Epoch 00004: val_loss didnot improve from 0.86877
107/107 [=====] - 11s 10ms/step - loss: 0.8107 - accuracy: 0.6963 - val_
- val_accuracy: 0.6346
Epoch 5/25
107/107 [=====] - ETA: 0s - loss: 0.7743 - accuracy: 0.6916
Epoch 00005: val_loss improved from 0.86877 to 0.72603, saving model to weights.hdf5
107/107 [=====] - 11s 10ms/step - loss: 0.7260 - accuracy: 0.6916 - val_
```

```

val_accuracy: 0.6635
Epoch 6/25
107/107 [=====] - ETA: 0s - loss: 0.7394 - accuracy: 0.7103
Epoch 0006: val_loss improved from 0.72603 to 0.72148, saving model to weights.hdf5
107/107 [=====] - 11s 10ms/step - loss: 0.7394 - accuracy: 0.7103 - val_loss: 0.7215
- val_accuracy: 0.7212
Epoch 7/25
106/107 [=====] - ETA: 0s - loss: 0.6882 - accuracy: 0.7406
Epoch 0007: val_loss improved from 0.72148 to 0.68642, saving model to weights.hdf5
107/107 [=====] - 11s 10ms/step - loss: 0.6889 - accuracy: 0.7407 - val_loss: 0.6864
- val_accuracy: 0.7308
Epoch 8/25
107/107 [=====] - ETA: 0s - loss: 0.6493 - accuracy: 0.7547
Epoch 0008: val_loss did not improve from 0.68642
107/107 [=====] - 11s 10ms/step - loss: 0.6493 - accuracy: 0.7547 - val_loss: 0.7639
- val_accuracy: 0.7315
Epoch 9/25
107/107 [=====] - ETA: 0s - loss: 0.5552 - accuracy: 0.7874
Epoch 0009: val_loss did not improve from 0.68642
107/107 [=====] - 11s 9ms/step - loss: 0.5552 - accuracy: 0.7874 - val_loss: 0.7398
- val_accuracy: 0.7115
Epoch 10/25
107/107 [=====] - ETA: 0s - loss: 0.5436 - accuracy: 0.8154
Epoch 0010: val_loss did not improve from 0.68642
107/107 [=====] - 11s 10ms/step - loss: 0.5436 - accuracy: 0.8154 - val_loss: 1.2915
- val_accuracy: 0.5577
Epoch 11/25
107/107 [=====] - ETA: 0s - loss: 0.5401 - accuracy: 0.8131
Epoch 0011: val_loss did not improve from 0.68642
107/107 [=====] - 11s 10ms/step - loss: 0.5401 - accuracy: 0.8131 - val_loss: 0.6889
- val_accuracy: 0.7596
Epoch 12/25
106/107 [=====] - ETA: 0s - loss: 0.5150 - accuracy: 0.8373
Epoch 0012: val_loss did not improve from 0.68642
107/107 [=====] - 11s 10ms/step - loss: 0.5159 - accuracy: 0.8341 - val_loss: 2.0237
- val_accuracy: 0.6442
Epoch 13/25
107/107 [=====] - ETA: 0s - loss: 0.4757 - accuracy: 0.8341
Epoch 0013: val_loss did not improve from 0.68642
107/107 [=====] - 11s 10ms/step - loss: 0.4757 - accuracy: 0.8341 - val_loss: 0.7399
- val_accuracy: 0.7885
Epoch 14/25
106/107 [=====] - ETA: 0s - loss: 0.4935 - accuracy: 0.8302
Epoch 0014: val_loss did not improve from 0.68642
107/107 [=====] - 11s 10ms/step - loss: 0.4899 - accuracy: 0.8318 - val_loss: 2.4922
- val_accuracy: 0.6200
Epoch 15/25
106/107 [=====] - ETA: 0s - loss: 0.4180 - accuracy: 0.8561
Epoch 0015: val_loss did not improve from 0.68642
107/107 [=====] - 11s 10ms/step - loss: 0.4264 - accuracy: 0.8551 - val_loss: 1.0930
- val_accuracy: 0.7308
Epoch 16/25
107/107 [=====] - ETA: 0s - loss: 0.4676 - accuracy: 0.8505
Epoch 0016: val_loss did not improve from 0.68642
107/107 [=====] - 11s 10ms/step - loss: 0.4676 - accuracy: 0.8505 - val_loss: 1.1367
- val_accuracy: 0.7019
Epoch 17/25
107/107 [=====] - ETA: 0s - loss: 0.4983 - accuracy: 0.8435
Epoch 0017: val_loss did not improve from 0.68642
107/107 [=====] - 11s 102ms/step - loss: 0.4983 - accuracy: 0.8435 - val_loss: 1.7773
- val_accuracy: 0.5962
Epoch 18/25
107/107 [=====] - ETA: 0s - loss: 0.4135 - accuracy: 0.8481
Epoch 0018: val_loss did not improve from 0.68642
107/107 [=====] - 11s 10ms/step - loss: 0.4135 - accuracy: 0.8481 - val_loss: 0.8396
- val_accuracy: 0.7981
Epoch 19/25
107/107 [=====] - ETA: 0s - loss: 0.3833 - accuracy: 0.8575
Epoch 0019: val_loss did not improve from 0.68642
107/107 [=====] - 11s 9ms/step - loss: 0.3835 - accuracy: 0.8575 - val_loss: 0.7528
- val_accuracy: 0.7596
Epoch 20/25
107/107 [=====] - ETA: 0s - loss: 0.3824 - accuracy: 0.8668
Epoch 0020: val_loss did not improve from 0.68642
107/107 [=====] - 11s 10ms/step - loss: 0.3824 - accuracy: 0.8668 - val_loss: 1.3665
- val_accuracy: 0.6538
Epoch 21/25
106/107 [=====] - ETA: 0s - loss: 0.4035 - accuracy: 0.8491
Epoch 0021: val_loss improved from 0.68642 to 0.63674, saving model to weights.hdf5
107/107 [=====] - 11s 10ms/step - loss: 0.3998 - accuracy: 0.8505 - val_loss: 0.6367
- val_accuracy: 0.8077
Epoch 22/25
107/107 [=====] - ETA: 0s - loss: 0.3910 - accuracy: 0.8645
Epoch 0022: val_loss did not improve from 0.63674
107/107 [=====] - 11s 10ms/step - loss: 0.3910 - accuracy: 0.8645 - val_loss: 1.9714
- val_accuracy: 0.6538
Epoch 23/25
106/107 [=====] - ETA: 0s - loss: 0.3688 - accuracy: 0.8892
Epoch 0023: val_loss did not improve from 0.63674
107/107 [=====] - 11s 10ms/step - loss: 0.3655 - accuracy: 0.8902 - val_loss: 0.6917
- val_accuracy: 0.7885
Epoch 24/25
107/107 [=====] - ETA: 0s - loss: 0.3199 - accuracy: 0.8832
Epoch 0024: val_loss improved from 0.63674 to 0.58392, saving model to weights.hdf5
107/107 [=====] - 11s 10ms/step - loss: 0.3199 - accuracy: 0.8832 - val_loss: 0.5839
- val_accuracy: 0.8654
Epoch 25/25
107/107 [=====] - ETA: 0s - loss: 0.4139 - accuracy: 0.8528
Epoch 0025: val_loss did not improve from 0.58392
107/107 [=====] - 11s 102ms/step - loss: 0.4139 - accuracy: 0.8528 - val_loss: 1.1470
- val_accuracy: 0.6635

```

EVALUAR EL MODELO DE DEEP LEARNING ENTRENADO

```

In [ ]: history.history.keys()

Out[ ]: dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])

In [ ]: plt.plot(history.history['accuracy'])
plt.plot(history.history['loss'])

plt.title('Perdida y Precision en la fase de Entrenamiento del Modelo')
plt.xlabel('Epoch')
plt.ylabel('Precision y Perdida')
plt.legend(['Precision en Entrenamiento', 'Perdida en Entrenamiento'])

Out[ ]: Matplotlib.Legend.Legend at 0x7fc052288980

```

```

In [ ]: plt.plot(history.history['val_loss'])

plt.title('Perdida y Precision en la fase de Validacion Cruzada del Modelo')
plt.xlabel('Epoch')
plt.ylabel('Precision y Perdida en Validacion')
plt.legend(['Perdida en Validacion'])

Out[ ]: Matplotlib.Legend.Legend at 0x7fc0505f4900

```

```
plt.plot(history.history['val_accuracy'])
```

```
plt.title("Precisión en la fase de Validación Cruzada del Modelo")
plt.xlabel("Epoch")
plt.ylabel("Precision en Validation")
plt.legend(("Precisión en Validación"))
```

```
Out [ ] : <matplotlib.legend.Legend at 0x7fc0d59eb278>
```

```
In [ ] : test_directory = "A:/A_Google Drive/Proyectos IA/O4_Proyectos Machine Learning/Detección enfermedades pulmonares"
```

```
In [ ] : test_gen = ImageDataGenerator(rescale = 1./255)

test_generator = test_gen.flow_from_directory(batch_size=40, directory=test_directory, shuffle=True, target_size=(256, 256))

evaluate = model.evaluate_generator(test_generator, steps = test_generator.n // 4, verbose = 1)

print("Precisión en la fase de Test : {}".format(evaluate[1]))
```

Found 40 images belonging to 4 classes.
WARNING:tensorflow:From <python-input-29-9da2758a80b1>:5: Model.evaluation_generator (from tensorflow.python.keras.engine.training) is deprecated and will be removed in a future version.
Instructions for updating:
Please use Model.evaluate, which supports generators.
1/10 [=>.....] - ETA: 0s - loss: 1.9450 - accuracy: 0.6250
WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at least `steps_per_epoch * epochs` batches (in this case, 10 batches). You may need to use the repeat() function when building your dataset.
1/10 [=>.....] - 0s 40ms/step - loss: 1.9450 - accuracy: 0.6250
Precisión en la fase de Test : 0.625

```
In [ ] : from sklearn.metrics import confusion_matrix, classification_report, accuracy_score

prediction = []
original = []
image = []

for i in range(len(os.listdir(test_directory))):
    for item in os.listdir(os.path.join(test_directory, str(i)))
        img = cv2.imread(os.path.join(test_directory, str(i), item))
        img = cv2.resize(img, (256, 256))
        image.append(img)
        img = img/255
        img = img.reshape(-1, 256, 256, 3)
        predict = model.predict(img)
        predict = np.argmax(predict)
        prediction.append(predict)
        original.append(i)
```

```
In [ ] : len(original)
```

```
Out [ ] : 40
```

```
In [ ] : score = accuracy_score(original, prediction)
print("Eficacia de la predicción {}".format(score))
```

Eficacia de la predicción 0.525

```
In [ ] : L = 8
W = 5

fig, axes = plt.subplots(L, W, figsize = (12,12))
axes = axes.ravel()

for i in np.arange(0, L*W):
    axes[i].imshow(image[i])
    axes[i].set_title("Pred={} | Verd={}".format(str(label_names[prediction[i]]), str(label_names[original[i]])))
    axes[i].axis('off')
```

plt.subplots_adjust(wspace = 1.2, hspace=1)

| Pred=COVID-19 Verd=COVID-19 | Pred=Pneumonia Bacteriana Verd=COVID-19 | Pred=Pneumonia Bacteriana Verd=COVID-19 | Pred=COVID-19 Verd=COVID-19 | Pred=COVID-19 Verd=COVID-19 |
|--|--|--|--|--|
| | | | | |
| Pred=Pneumonia Bacteriana Verd=COVID-19 | Pred=COVID-19 Verd=COVID-19 | Pred=Pneumonia Bacteriana Verd=COVID-19 | Pred=COVID-19 Verd=COVID-19 | Pred=COVID-19 Verd=COVID-19 |
| Pred=Pneumonia Bacteriana Verd=Normal | Pred=Normal Verd=Normal | Pred=Normal Verd=Normal | Pred=Pneumonia Bacteriana Verd=Normal | Pred=Pneumonia Bacteriana Verd=Normal |
| Pred=Normal Verd=Normal | Pred=Pneumonia Bacteriana Verd=Normal | Pred=Pneumonia Bacteriana Verd=Normal | Pred=Pneumonia Bacteriana Verd=Normal | Pred=Pneumonia Bacteriana Verd=Normal |
| Pred=Pneumonia Virica Verd=Pneumonia Virica | Pred=Pneumonia Bacteriana Verd=Pneumonia Virica | Pred=Pneumonia Bacteriana Verd=Pneumonia Virica | Pred=Pneumonia Bacteriana Verd=Pneumonia Virica | Pred=Pneumonia Bacteriana Verd=Pneumonia Virica |
| Pred=Pneumonia Bacteriana Verd=Pneumonia Virica | Pred=Pneumonia Bacteriana Verd=Pneumonia Virica | Pred=Pneumonia Bacteriana Verd=Pneumonia Virica | Pred=Pneumonia Bacteriana Verd=Pneumonia Virica | Pred=Pneumonia Bacteriana Verd=Pneumonia Virica |
| Pred=Pneumonia Bacteriana Verd=Pneumonia Bacteriana | Pred=Pneumonia Bacteriana Verd=Pneumonia Bacteriana | Pred=Pneumonia Bacteriana Verd=Pneumonia Bacteriana | Pred=Pneumonia Bacteriana Verd=Pneumonia Bacteriana | Pred=Pneumonia Bacteriana Verd=Pneumonia Bacteriana |
| Pred=Pneumonia Bacteriana Verd=Pneumonia Bacteriana | Pred=Pneumonia Bacteriana Verd=Pneumonia Bacteriana | Pred=Pneumonia Bacteriana Verd=Pneumonia Bacteriana | Pred=Pneumonia Bacteriana Verd=Pneumonia Bacteriana | Pred=Pneumonia Bacteriana Verd=Pneumonia Bacteriana |

```
In [ ] : print(classification_report(np.asarray(original), np.asarray(prediction)))
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 1.00 | 0.60 | 0.75 | 10 |
| 1 | 1.00 | 0.20 | 0.33 | 10 |
| 2 | 1.00 | 0.30 | 0.46 | 10 |
| 3 | 0.34 | 1.00 | 0.51 | 10 |
| accuracy | | | 0.53 | 40 |
| macro avg | 0.84 | 0.53 | 0.51 | 40 |
| weighted avg | 0.84 | 0.53 | 0.51 | 40 |

