actividad-transferlearning

October 17, 2023

0.1 # Actividad - Transfer Learning

Diego Sú Gómez - A01620476

Vanessa Mendez - A01639925

Javier Sánchez Panduro - A01639832

Estefanía Pérez Yeo - A01639270

0.2 ## Importación de librerías

```
[ ]: import matplotlib.pyplot as plt
     import numpy as np
     import pandas as pd
     import tensorflow as tf
     import tensorflow_datasets as tfds
     from tensorflow.keras import layers
     import zipfile
```

0.3 ## Creación del dataset y preprocesamiento de datos

```
[ ]: #Generar los datasets de entrenamiento y validación junto con los labels de ambas clases
     _url = "/content/Transferlearning.zip"
     with zipfile.ZipFile(_url, 'r') as zip_ref:
       zip_ref.extractall('/content')
```

```
[ ]: #Generar los conjuntos de entrenamiento y validación
     train_dataset = tf.keras.utils.image_dataset_from_directory('/content/Transferlearning/train', shuffle=True, batch_size=32, image_size=(224,224))
     test_dataset = tf.keras.utils.image_dataset_from_directory('/content/Transferlearning/test', shuffle=True, batch_size=32, image_size=(224,224))
     validation_dataset = tf.keras.utils.image_dataset_from_directory('/content/Transferlearning/valid', shuffle=True, batch_size=32, image_size=(224,224))
```

```
Found 2115 files belonging to 3 classes.
Found 29 files belonging to 3 classes.
Found 60 files belonging to 3 classes.
```

```
[ ]: class_names = train_dataset.class_names

     plt.figure(figsize=(10,10))
     for image, labels in train_dataset.take(1):
       for i in range(9):
         ax = plt.subplot(3,3,i+1)
         plt.imshow(image[i].numpy().astype("uint8"))
         plt.title(class_names[labels[i]])
         plt.axis("off")
```

```
[ ]: #Utilizar el GPU para procesar mejor las imágenes y mejorar la eficiencia
     AUTOTUNE = tf.data.AUTOTUNE
```

```
train_dataset = train_dataset.prefetch(buffer_size=AUTOTUNE)
validation_dataset = validation_dataset.prefetch(buffer_size=AUTOTUNE)
test_dataset = test_dataset.prefetch(buffer_size=AUTOTUNE)
```

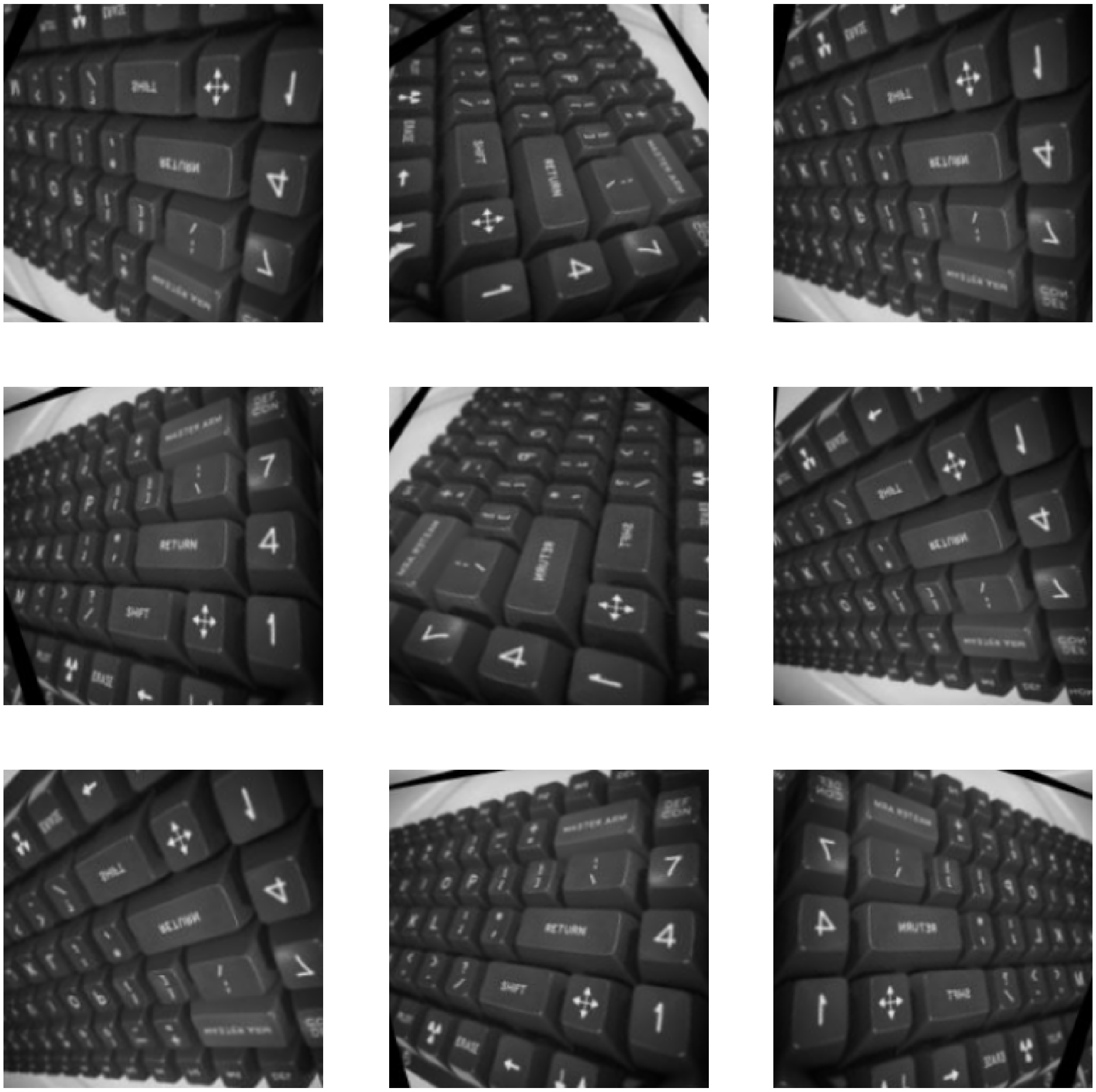
0.4 ## Entrenamiento de la red neuronal

[ ]:

```
#Capa de aumentación de los datos
data_augmentation = tf.keras.Sequential(
    [tf.keras.layers.RandomFlip("horizontal_and_vertical"),
    tf.keras.layers.RandomRotation(0.2),
    tf.keras.layers.RandomZoom(height_factor = (-0.05, -0.2), width_factor=(-0.
 ↪05,-0.2)),
     ])
```

[ ]:

```
for image, _ in train_dataset.take(1):
  plt.figure(figsize=(10,10))
  first_image = image[0]
  for i in range(9):
    ax = plt.subplot(3,3,i+1)
    augmented_img = data_augmentation(tf.expand_dims(first_image,0))
    plt.imshow(augmented_img[0]/255.0)
    plt.axis("off")
```



[ ]:

```
#Reescalado y preprocesamiento de las imágenes
rescale = tf.keras.layers.Rescaling(1./127.5, offset=-1)
preprocess_input = tf.keras.applications.mobilenet_v2.preprocess_input
```

[ ]:

```
#Obtener la red neuronal pre-entrenada (ResNet50)
base_model = tf.keras.applications.ResNet50(input_shape=(224,224,3),include_top= False, weights="imagenet")
```

[ ]:

```
#Congelar las capas de la ResNet para no entrenarlas
base_model.trainable = False
base_model.summary()
```

```
Model: "resnet50"
```

```
--------------------------------------------------------------------------------
------------------
 Layer (type)                Output Shape                 Param #   Connected to
================================================================================
==================
 input_1 (InputLayer)        [(None, 224, 224, 3)]        0         []

 conv1_pad (ZeroPadding2D)   (None, 230, 230, 3)          0
['input_1[0][0]']

 conv1_conv (Conv2D)         (None, 112, 112, 64)         9472
['conv1_pad[0][0]']

 conv1_bn (BatchNormalizati  (None, 112, 112, 64)         256
['conv1_conv[0][0]']
 on)

 conv1_relu (Activation)     (None, 112, 112, 64)         0
['conv1_bn[0][0]']

 pool1_pad (ZeroPadding2D)   (None, 114, 114, 64)         0
['conv1_relu[0][0]']

 pool1_pool (MaxPooling2D)   (None, 56, 56, 64)           0
['pool1_pad[0][0]']

 conv2_block1_1_conv (Conv2  (None, 56, 56, 64)           4160
['pool1_pool[0][0]']
 D)

 conv2_block1_1_bn (BatchNo  (None, 56, 56, 64)           256
['conv2_block1_1_conv[0][0]']
 rmalization)

 conv2_block1_1_relu (Activ  (None, 56, 56, 64)           0
['conv2_block1_1_bn[0][0]']
 ation)

 conv2_block1_2_conv (Conv2  (None, 56, 56, 64)           36928
['conv2_block1_1_relu[0][0]']
 D)

 conv2_block1_2_bn (BatchNo  (None, 56, 56, 64)           256
['conv2_block1_2_conv[0][0]']
 rmalization)

 conv2_block1_2_relu (Activ  (None, 56, 56, 64)           0
['conv2_block1_2_bn[0][0]']
```

```
 ation)

 conv2_block1_0_conv (Conv2  (None, 56, 56, 256)          16640
['pool1_pool[0][0]']
 D)

 conv2_block1_3_conv (Conv2  (None, 56, 56, 256)          16640
['conv2_block1_2_relu[0][0]']
 D)

 conv2_block1_0_bn (BatchNo  (None, 56, 56, 256)          1024
['conv2_block1_0_conv[0][0]']
 rmalization)

 conv2_block1_3_bn (BatchNo  (None, 56, 56, 256)          1024
['conv2_block1_3_conv[0][0]']
 rmalization)

 conv2_block1_add (Add)      (None, 56, 56, 256)          0
['conv2_block1_0_bn[0][0]',
'conv2_block1_3_bn[0][0]']

 conv2_block1_out (Activati  (None, 56, 56, 256)          0
['conv2_block1_add[0][0]']
 on)

 conv2_block2_1_conv (Conv2  (None, 56, 56, 64)           16448
['conv2_block1_out[0][0]']
 D)

 conv2_block2_1_bn (BatchNo  (None, 56, 56, 64)           256
['conv2_block2_1_conv[0][0]']
 rmalization)

 conv2_block2_1_relu (Activ  (None, 56, 56, 64)           0
['conv2_block2_1_bn[0][0]']
 ation)

 conv2_block2_2_conv (Conv2  (None, 56, 56, 64)           36928
['conv2_block2_1_relu[0][0]']
 D)

 conv2_block2_2_bn (BatchNo  (None, 56, 56, 64)           256
['conv2_block2_2_conv[0][0]']
 rmalization)

 conv2_block2_2_relu (Activ  (None, 56, 56, 64)           0
['conv2_block2_2_bn[0][0]']
```

```
 ation)

 conv2_block2_3_conv (Conv2  (None, 56, 56, 256)          16640
['conv2_block2_2_relu[0][0]']
 D)

 conv2_block2_3_bn (BatchNo  (None, 56, 56, 256)          1024
['conv2_block2_3_conv[0][0]']
 rmalization)

 conv2_block2_add (Add)      (None, 56, 56, 256)          0
['conv2_block1_out[0][0]',
'conv2_block2_3_bn[0][0]']

 conv2_block2_out (Activati  (None, 56, 56, 256)          0
['conv2_block2_add[0][0]']
 on)

 conv2_block3_1_conv (Conv2  (None, 56, 56, 64)           16448
['conv2_block2_out[0][0]']
 D)

 conv2_block3_1_bn (BatchNo  (None, 56, 56, 64)           256
['conv2_block3_1_conv[0][0]']
 rmalization)

 conv2_block3_1_relu (Activ  (None, 56, 56, 64)           0
['conv2_block3_1_bn[0][0]']
 ation)

 conv2_block3_2_conv (Conv2  (None, 56, 56, 64)           36928
['conv2_block3_1_relu[0][0]']
 D)

 conv2_block3_2_bn (BatchNo  (None, 56, 56, 64)           256
['conv2_block3_2_conv[0][0]']
 rmalization)

 conv2_block3_2_relu (Activ  (None, 56, 56, 64)           0
['conv2_block3_2_bn[0][0]']
 ation)

 conv2_block3_3_conv (Conv2  (None, 56, 56, 256)          16640
['conv2_block3_2_relu[0][0]']
 D)

 conv2_block3_3_bn (BatchNo  (None, 56, 56, 256)          1024
['conv2_block3_3_conv[0][0]']
```

```
 rmalization)

 conv2_block3_add (Add)      (None, 56, 56, 256)          0
['conv2_block2_out[0][0]',
'conv2_block3_3_bn[0][0]']

 conv2_block3_out (Activati  (None, 56, 56, 256)          0
['conv2_block3_add[0][0]']
 on)

 conv3_block1_1_conv (Conv2  (None, 28, 28, 128)          32896
['conv2_block3_out[0][0]']
 D)

 conv3_block1_1_bn (BatchNo  (None, 28, 28, 128)          512
['conv3_block1_1_conv[0][0]']
 rmalization)

 conv3_block1_1_relu (Activ  (None, 28, 28, 128)          0
['conv3_block1_1_bn[0][0]']
 ation)

 conv3_block1_2_conv (Conv2  (None, 28, 28, 128)          147584
['conv3_block1_1_relu[0][0]']
 D)

 conv3_block1_2_bn (BatchNo  (None, 28, 28, 128)          512
['conv3_block1_2_conv[0][0]']
 rmalization)

 conv3_block1_2_relu (Activ  (None, 28, 28, 128)          0
['conv3_block1_2_bn[0][0]']
 ation)

 conv3_block1_0_conv (Conv2  (None, 28, 28, 512)          131584
['conv2_block3_out[0][0]']
 D)

 conv3_block1_3_conv (Conv2  (None, 28, 28, 512)          66048
['conv3_block1_2_relu[0][0]']
 D)

 conv3_block1_0_bn (BatchNo  (None, 28, 28, 512)          2048
['conv3_block1_0_conv[0][0]']
 rmalization)

 conv3_block1_3_bn (BatchNo  (None, 28, 28, 512)          2048
['conv3_block1_3_conv[0][0]']
```

```
 rmalization)

 conv3_block1_add (Add)      (None, 28, 28, 512)          0
['conv3_block1_0_bn[0][0]',
'conv3_block1_3_bn[0][0]']

 conv3_block1_out (Activati  (None, 28, 28, 512)          0
['conv3_block1_add[0][0]']
 on)

 conv3_block2_1_conv (Conv2  (None, 28, 28, 128)          65664
['conv3_block1_out[0][0]']
 D)

 conv3_block2_1_bn (BatchNo  (None, 28, 28, 128)          512
['conv3_block2_1_conv[0][0]']
 rmalization)

 conv3_block2_1_relu (Activ  (None, 28, 28, 128)          0
['conv3_block2_1_bn[0][0]']
 ation)

 conv3_block2_2_conv (Conv2  (None, 28, 28, 128)          147584
['conv3_block2_1_relu[0][0]']
 D)

 conv3_block2_2_bn (BatchNo  (None, 28, 28, 128)          512
['conv3_block2_2_conv[0][0]']
 rmalization)

 conv3_block2_2_relu (Activ  (None, 28, 28, 128)          0
['conv3_block2_2_bn[0][0]']
 ation)

 conv3_block2_3_conv (Conv2  (None, 28, 28, 512)          66048
['conv3_block2_2_relu[0][0]']
 D)

 conv3_block2_3_bn (BatchNo  (None, 28, 28, 512)          2048
['conv3_block2_3_conv[0][0]']
 rmalization)

 conv3_block2_add (Add)      (None, 28, 28, 512)          0
['conv3_block1_out[0][0]',
'conv3_block2_3_bn[0][0]']

 conv3_block2_out (Activati  (None, 28, 28, 512)          0
['conv3_block2_add[0][0]']
```

```
 on)

 conv3_block3_1_conv (Conv2  (None, 28, 28, 128)          65664
['conv3_block2_out[0][0]']
 D)

 conv3_block3_1_bn (BatchNo  (None, 28, 28, 128)          512
['conv3_block3_1_conv[0][0]']
 rmalization)

 conv3_block3_1_relu (Activ  (None, 28, 28, 128)          0
['conv3_block3_1_bn[0][0]']
 ation)

 conv3_block3_2_conv (Conv2  (None, 28, 28, 128)          147584
['conv3_block3_1_relu[0][0]']
 D)

 conv3_block3_2_bn (BatchNo  (None, 28, 28, 128)          512
['conv3_block3_2_conv[0][0]']
 rmalization)

 conv3_block3_2_relu (Activ  (None, 28, 28, 128)          0
['conv3_block3_2_bn[0][0]']
 ation)

 conv3_block3_3_conv (Conv2  (None, 28, 28, 512)          66048
['conv3_block3_2_relu[0][0]']
 D)

 conv3_block3_3_bn (BatchNo  (None, 28, 28, 512)          2048
['conv3_block3_3_conv[0][0]']
 rmalization)

 conv3_block3_add (Add)      (None, 28, 28, 512)          0
['conv3_block2_out[0][0]',
'conv3_block3_3_bn[0][0]']

 conv3_block3_out (Activati  (None, 28, 28, 512)          0
['conv3_block3_add[0][0]']
 on)

 conv3_block4_1_conv (Conv2  (None, 28, 28, 128)          65664
['conv3_block3_out[0][0]']
 D)

 conv3_block4_1_bn (BatchNo  (None, 28, 28, 128)          512
['conv3_block4_1_conv[0][0]']
```

```
 rmalization)

 conv3_block4_1_relu (Activ  (None, 28, 28, 128)          0
['conv3_block4_1_bn[0][0]']
 ation)

 conv3_block4_2_conv (Conv2  (None, 28, 28, 128)          147584
['conv3_block4_1_relu[0][0]']
 D)

 conv3_block4_2_bn (BatchNo  (None, 28, 28, 128)          512
['conv3_block4_2_conv[0][0]']
 rmalization)

 conv3_block4_2_relu (Activ  (None, 28, 28, 128)          0
['conv3_block4_2_bn[0][0]']
 ation)

 conv3_block4_3_conv (Conv2  (None, 28, 28, 512)          66048
['conv3_block4_2_relu[0][0]']
 D)

 conv3_block4_3_bn (BatchNo  (None, 28, 28, 512)          2048
['conv3_block4_3_conv[0][0]']
 rmalization)

 conv3_block4_add (Add)      (None, 28, 28, 512)          0
['conv3_block3_out[0][0]',
'conv3_block4_3_bn[0][0]']

 conv3_block4_out (Activati  (None, 28, 28, 512)          0
['conv3_block4_add[0][0]']
 on)

 conv4_block1_1_conv (Conv2  (None, 14, 14, 256)          131328
['conv3_block4_out[0][0]']
 D)

 conv4_block1_1_bn (BatchNo  (None, 14, 14, 256)          1024
['conv4_block1_1_conv[0][0]']
 rmalization)

 conv4_block1_1_relu (Activ  (None, 14, 14, 256)          0
['conv4_block1_1_bn[0][0]']
 ation)

 conv4_block1_2_conv (Conv2  (None, 14, 14, 256)          590080
['conv4_block1_1_relu[0][0]']
```

```
 D)

 conv4_block1_2_bn (BatchNo  (None, 14, 14, 256)         1024
['conv4_block1_2_conv[0][0]']
 rmalization)

 conv4_block1_2_relu (Activ  (None, 14, 14, 256)         0
['conv4_block1_2_bn[0][0]']
 ation)

 conv4_block1_0_conv (Conv2  (None, 14, 14, 1024)        525312
['conv3_block4_out[0][0]']
 D)

 conv4_block1_3_conv (Conv2  (None, 14, 14, 1024)        263168
['conv4_block1_2_relu[0][0]']
 D)

 conv4_block1_0_bn (BatchNo  (None, 14, 14, 1024)        4096
['conv4_block1_0_conv[0][0]']
 rmalization)

 conv4_block1_3_bn (BatchNo  (None, 14, 14, 1024)        4096
['conv4_block1_3_conv[0][0]']
 rmalization)

 conv4_block1_add (Add)      (None, 14, 14, 1024)        0
['conv4_block1_0_bn[0][0]',
'conv4_block1_3_bn[0][0]']

 conv4_block1_out (Activati  (None, 14, 14, 1024)        0
['conv4_block1_add[0][0]']
 on)

 conv4_block2_1_conv (Conv2  (None, 14, 14, 256)         262400
['conv4_block1_out[0][0]']
 D)

 conv4_block2_1_bn (BatchNo  (None, 14, 14, 256)         1024
['conv4_block2_1_conv[0][0]']
 rmalization)

 conv4_block2_1_relu (Activ  (None, 14, 14, 256)         0
['conv4_block2_1_bn[0][0]']
 ation)

 conv4_block2_2_conv (Conv2  (None, 14, 14, 256)         590080
['conv4_block2_1_relu[0][0]']
```

```
 D)

 conv4_block2_2_bn (BatchNo  (None, 14, 14, 256)          1024
['conv4_block2_2_conv[0][0]']
 rmalization)

 conv4_block2_2_relu (Activ  (None, 14, 14, 256)          0
['conv4_block2_2_bn[0][0]']
 ation)

 conv4_block2_3_conv (Conv2  (None, 14, 14, 1024)         263168
['conv4_block2_2_relu[0][0]']
 D)

 conv4_block2_3_bn (BatchNo  (None, 14, 14, 1024)         4096
['conv4_block2_3_conv[0][0]']
 rmalization)

 conv4_block2_add (Add)      (None, 14, 14, 1024)         0
['conv4_block1_out[0][0]',
'conv4_block2_3_bn[0][0]']

 conv4_block2_out (Activati  (None, 14, 14, 1024)         0
['conv4_block2_add[0][0]']
 on)

 conv4_block3_1_conv (Conv2  (None, 14, 14, 256)          262400
['conv4_block2_out[0][0]']
 D)

 conv4_block3_1_bn (BatchNo  (None, 14, 14, 256)          1024
['conv4_block3_1_conv[0][0]']
 rmalization)

 conv4_block3_1_relu (Activ  (None, 14, 14, 256)          0
['conv4_block3_1_bn[0][0]']
 ation)

 conv4_block3_2_conv (Conv2  (None, 14, 14, 256)          590080
['conv4_block3_1_relu[0][0]']
 D)

 conv4_block3_2_bn (BatchNo  (None, 14, 14, 256)          1024
['conv4_block3_2_conv[0][0]']
 rmalization)

 conv4_block3_2_relu (Activ  (None, 14, 14, 256)          0
['conv4_block3_2_bn[0][0]']
```

```
 ation)

 conv4_block3_3_conv (Conv2  (None, 14, 14, 1024)         263168
['conv4_block3_2_relu[0][0]']
 D)

 conv4_block3_3_bn (BatchNo  (None, 14, 14, 1024)         4096
['conv4_block3_3_conv[0][0]']
 rmalization)

 conv4_block3_add (Add)      (None, 14, 14, 1024)         0
['conv4_block2_out[0][0]',
'conv4_block3_3_bn[0][0]']

 conv4_block3_out (Activati  (None, 14, 14, 1024)         0
['conv4_block3_add[0][0]']
 on)

 conv4_block4_1_conv (Conv2  (None, 14, 14, 256)          262400
['conv4_block3_out[0][0]']
 D)

 conv4_block4_1_bn (BatchNo  (None, 14, 14, 256)          1024
['conv4_block4_1_conv[0][0]']
 rmalization)

 conv4_block4_1_relu (Activ  (None, 14, 14, 256)          0
['conv4_block4_1_bn[0][0]']
 ation)

 conv4_block4_2_conv (Conv2  (None, 14, 14, 256)          590080
['conv4_block4_1_relu[0][0]']
 D)

 conv4_block4_2_bn (BatchNo  (None, 14, 14, 256)          1024
['conv4_block4_2_conv[0][0]']
 rmalization)

 conv4_block4_2_relu (Activ  (None, 14, 14, 256)          0
['conv4_block4_2_bn[0][0]']
 ation)

 conv4_block4_3_conv (Conv2  (None, 14, 14, 1024)         263168
['conv4_block4_2_relu[0][0]']
 D)

 conv4_block4_3_bn (BatchNo  (None, 14, 14, 1024)         4096
['conv4_block4_3_conv[0][0]']
```

```
 rmalization)

 conv4_block4_add (Add)      (None, 14, 14, 1024)        0
['conv4_block3_out[0][0]',
'conv4_block4_3_bn[0][0]']

 conv4_block4_out (Activati  (None, 14, 14, 1024)        0
['conv4_block4_add[0][0]']
 on)

 conv4_block5_1_conv (Conv2  (None, 14, 14, 256)         262400
['conv4_block4_out[0][0]']
 D)

 conv4_block5_1_bn (BatchNo  (None, 14, 14, 256)         1024
['conv4_block5_1_conv[0][0]']
 rmalization)

 conv4_block5_1_relu (Activ  (None, 14, 14, 256)         0
['conv4_block5_1_bn[0][0]']
 ation)

 conv4_block5_2_conv (Conv2  (None, 14, 14, 256)         590080
['conv4_block5_1_relu[0][0]']
 D)

 conv4_block5_2_bn (BatchNo  (None, 14, 14, 256)         1024
['conv4_block5_2_conv[0][0]']
 rmalization)

 conv4_block5_2_relu (Activ  (None, 14, 14, 256)         0
['conv4_block5_2_bn[0][0]']
 ation)

 conv4_block5_3_conv (Conv2  (None, 14, 14, 1024)        263168
['conv4_block5_2_relu[0][0]']
 D)

 conv4_block5_3_bn (BatchNo  (None, 14, 14, 1024)        4096
['conv4_block5_3_conv[0][0]']
 rmalization)

 conv4_block5_add (Add)      (None, 14, 14, 1024)        0
['conv4_block4_out[0][0]',
'conv4_block5_3_bn[0][0]']

 conv4_block5_out (Activati  (None, 14, 14, 1024)        0
['conv4_block5_add[0][0]']
```

```
 on)

 conv4_block6_1_conv (Conv2  (None, 14, 14, 256)          262400
['conv4_block5_out[0][0]']
 D)

 conv4_block6_1_bn (BatchNo  (None, 14, 14, 256)          1024
['conv4_block6_1_conv[0][0]']
 rmalization)

 conv4_block6_1_relu (Activ  (None, 14, 14, 256)          0
['conv4_block6_1_bn[0][0]']
 ation)

 conv4_block6_2_conv (Conv2  (None, 14, 14, 256)          590080
['conv4_block6_1_relu[0][0]']
 D)

 conv4_block6_2_bn (BatchNo  (None, 14, 14, 256)          1024
['conv4_block6_2_conv[0][0]']
 rmalization)

 conv4_block6_2_relu (Activ  (None, 14, 14, 256)          0
['conv4_block6_2_bn[0][0]']
 ation)

 conv4_block6_3_conv (Conv2  (None, 14, 14, 1024)         263168
['conv4_block6_2_relu[0][0]']
 D)

 conv4_block6_3_bn (BatchNo  (None, 14, 14, 1024)         4096
['conv4_block6_3_conv[0][0]']
 rmalization)

 conv4_block6_add (Add)      (None, 14, 14, 1024)         0
['conv4_block5_out[0][0]',
'conv4_block6_3_bn[0][0]']

 conv4_block6_out (Activati  (None, 14, 14, 1024)         0
['conv4_block6_add[0][0]']
 on)

 conv5_block1_1_conv (Conv2  (None, 7, 7, 512)            524800
['conv4_block6_out[0][0]']
 D)

 conv5_block1_1_bn (BatchNo  (None, 7, 7, 512)            2048
['conv5_block1_1_conv[0][0]']
```

```
 rmalization)

 conv5_block1_1_relu (Activ  (None, 7, 7, 512)            0
['conv5_block1_1_bn[0][0]']
 ation)

 conv5_block1_2_conv (Conv2  (None, 7, 7, 512)            2359808
['conv5_block1_1_relu[0][0]']
 D)

 conv5_block1_2_bn (BatchNo  (None, 7, 7, 512)            2048
['conv5_block1_2_conv[0][0]']
 rmalization)

 conv5_block1_2_relu (Activ  (None, 7, 7, 512)            0
['conv5_block1_2_bn[0][0]']
 ation)

 conv5_block1_0_conv (Conv2  (None, 7, 7, 2048)           2099200
['conv4_block6_out[0][0]']
 D)

 conv5_block1_3_conv (Conv2  (None, 7, 7, 2048)           1050624
['conv5_block1_2_relu[0][0]']
 D)

 conv5_block1_0_bn (BatchNo  (None, 7, 7, 2048)           8192
['conv5_block1_0_conv[0][0]']
 rmalization)

 conv5_block1_3_bn (BatchNo  (None, 7, 7, 2048)           8192
['conv5_block1_3_conv[0][0]']
 rmalization)

 conv5_block1_add (Add)      (None, 7, 7, 2048)           0
['conv5_block1_0_bn[0][0]',
'conv5_block1_3_bn[0][0]']

 conv5_block1_out (Activati  (None, 7, 7, 2048)           0
['conv5_block1_add[0][0]']
 on)

 conv5_block2_1_conv (Conv2  (None, 7, 7, 512)            1049088
['conv5_block1_out[0][0]']
 D)

 conv5_block2_1_bn (BatchNo  (None, 7, 7, 512)            2048
['conv5_block2_1_conv[0][0]']
```

```
 rmalization)

 conv5_block2_1_relu (Activ  (None, 7, 7, 512)            0
['conv5_block2_1_bn[0][0]']
 ation)

 conv5_block2_2_conv (Conv2  (None, 7, 7, 512)            2359808
['conv5_block2_1_relu[0][0]']
 D)

 conv5_block2_2_bn (BatchNo  (None, 7, 7, 512)            2048
['conv5_block2_2_conv[0][0]']
 rmalization)

 conv5_block2_2_relu (Activ  (None, 7, 7, 512)            0
['conv5_block2_2_bn[0][0]']
 ation)

 conv5_block2_3_conv (Conv2  (None, 7, 7, 2048)           1050624
['conv5_block2_2_relu[0][0]']
 D)

 conv5_block2_3_bn (BatchNo  (None, 7, 7, 2048)           8192
['conv5_block2_3_conv[0][0]']
 rmalization)

 conv5_block2_add (Add)      (None, 7, 7, 2048)           0
['conv5_block1_out[0][0]',
'conv5_block2_3_bn[0][0]']

 conv5_block2_out (Activati  (None, 7, 7, 2048)           0
['conv5_block2_add[0][0]']
 on)

 conv5_block3_1_conv (Conv2  (None, 7, 7, 512)            1049088
['conv5_block2_out[0][0]']
 D)

 conv5_block3_1_bn (BatchNo  (None, 7, 7, 512)            2048
['conv5_block3_1_conv[0][0]']
 rmalization)

 conv5_block3_1_relu (Activ  (None, 7, 7, 512)            0
['conv5_block3_1_bn[0][0]']
 ation)

 conv5_block3_2_conv (Conv2  (None, 7, 7, 512)            2359808
['conv5_block3_1_relu[0][0]']
```

```
 D)

 conv5_block3_2_bn (BatchNo  (None, 7, 7, 512)            2048
['conv5_block3_2_conv[0][0]']
 rmalization)

 conv5_block3_2_relu (Activ  (None, 7, 7, 512)            0
['conv5_block3_2_bn[0][0]']
 ation)

 conv5_block3_3_conv (Conv2  (None, 7, 7, 2048)           1050624
['conv5_block3_2_relu[0][0]']
 D)

 conv5_block3_3_bn (BatchNo  (None, 7, 7, 2048)           8192
['conv5_block3_3_conv[0][0]']
 rmalization)

 conv5_block3_add (Add)      (None, 7, 7, 2048)           0
['conv5_block2_out[0][0]',
'conv5_block3_3_bn[0][0]']

 conv5_block3_out (Activati  (None, 7, 7, 2048)           0
['conv5_block3_add[0][0]']
 on)

================================================================================
==================
Total params: 23587712 (89.98 MB)
Trainable params: 0 (0.00 Byte)
Non-trainable params: 23587712 (89.98 MB)
________________________________________________________________________________
__________________
```

[ ]:

```
#Añadir una global average layer
global_average_layer = tf.keras.layers.GlobalAveragePooling2D()
```

[ ]:

```
#Añadir la capa de predicción
prediction_layer = tf.keras.layers.Dense(3) #Mouse, Teclado, Monitor
```

[ ]:

```
#Unir todo el modelo para probarlo
inputs = tf.keras.Input(shape=(224,224,3))
x = data_augmentation(inputs)
x = preprocess_input(x)
x = base_model(x, training=False)
x = global_average_layer(x)
x = tf.keras.layers.Dropout(0.2)(x)
```

```
outputs = prediction_layer(x)
model = tf.keras.Model(inputs, outputs)
```

[ ]:
```
#Compilar el modelo con Categorical Cross Entropy al ser 3 clases diferentes
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.0001), loss=tf.
 ↪keras.losses.SparseCategoricalCrossentropy(from_logits=True),␣
 ↪metrics=["accuracy"])
```

[ ]:
```
model.summary()
```

```
Model: "model"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 224, 224, 3)]     0

 sequential (Sequential)     (None, 224, 224, 3)       0

 tf.math.truediv (TFOpLambd  (None, 224, 224, 3)       0
 a)

 tf.math.subtract (TFOpLamb  (None, 224, 224, 3)       0
 da)

 resnet50 (Functional)       (None, 7, 7, 2048)        23587712

 global_average_pooling2d (  (None, 2048)              0
 GlobalAveragePooling2D)

 dropout (Dropout)           (None, 2048)              0

 dense (Dense)               (None, 3)                 6147

=================================================================
Total params: 23593859 (90.00 MB)
Trainable params: 6147 (24.01 KB)
Non-trainable params: 23587712 (89.98 MB)
_________________________________________________________________
```

[ ]:
```
#Evaluar el modelo sin entrenar
loss_inicial, accuracy_inicial = model.evaluate(validation_dataset)
```

```
2/2 [==============================] - 8s 1s/step - loss: 1.1608 - accuracy:
0.3667
```

[ ]:
```
#Entrenar el modelo
```

```
history = model.fit(train_dataset, epochs=10, validation_data =␣
  ↪validation_dataset)
```

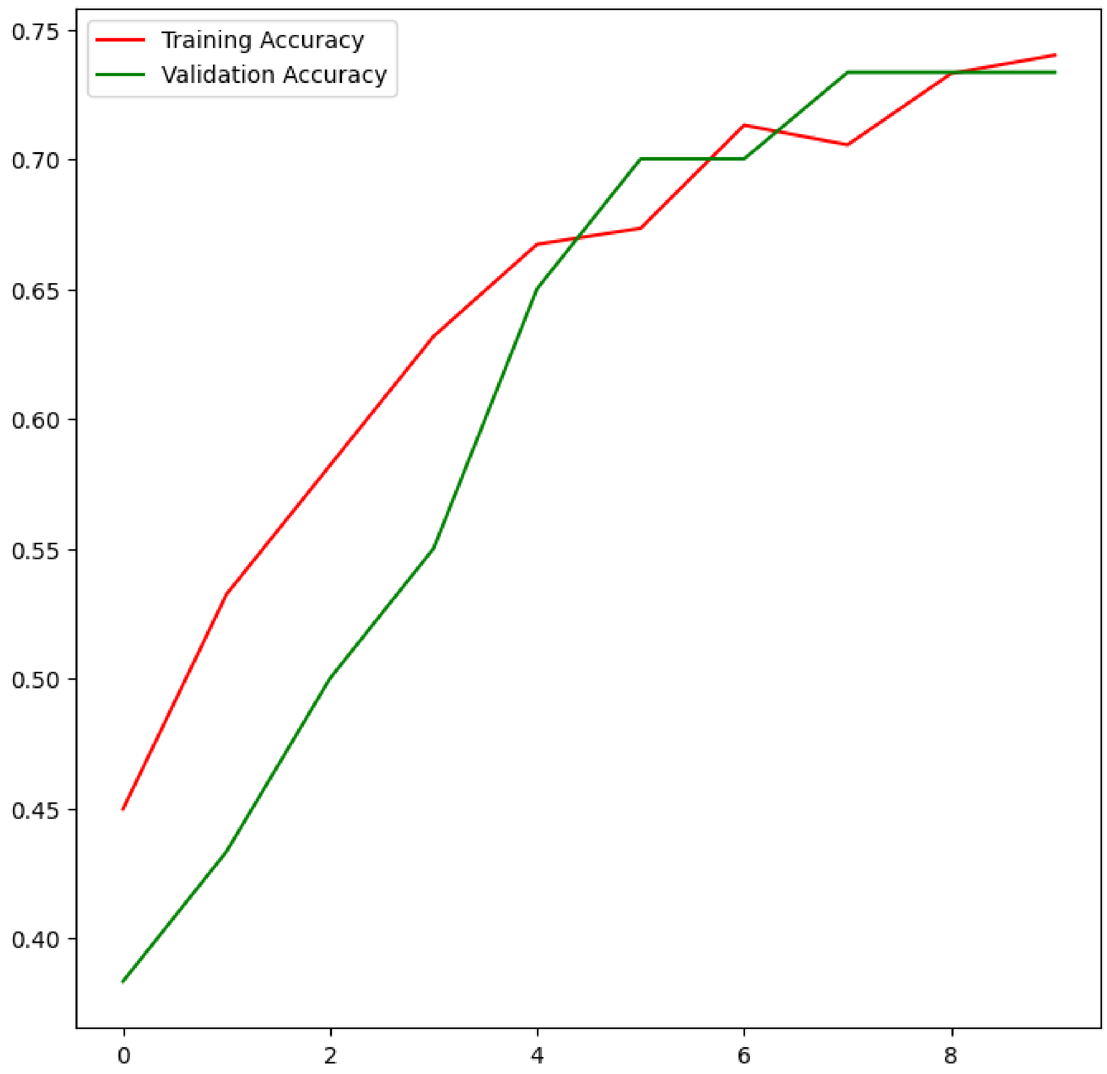
```
Epoch 1/10
67/67 [==============================] - 15s 166ms/step - loss: 1.0613 -
accuracy: 0.4496 - val_loss: 1.0561 - val_accuracy: 0.3833
Epoch 2/10
67/67 [==============================] - 9s 125ms/step - loss: 0.9609 -
accuracy: 0.5324 - val_loss: 0.9756 - val_accuracy: 0.4333
Epoch 3/10
67/67 [==============================] - 8s 125ms/step - loss: 0.8946 -
accuracy: 0.5820 - val_loss: 0.9053 - val_accuracy: 0.5000
Epoch 4/10
67/67 [==============================] - 9s 134ms/step - loss: 0.8418 -
accuracy: 0.6317 - val_loss: 0.8378 - val_accuracy: 0.5500
Epoch 5/10
67/67 [==============================] - 9s 127ms/step - loss: 0.7988 -
accuracy: 0.6671 - val_loss: 0.7932 - val_accuracy: 0.6500
Epoch 6/10
67/67 [==============================] - 8s 119ms/step - loss: 0.7596 -
accuracy: 0.6733 - val_loss: 0.7729 - val_accuracy: 0.7000
Epoch 7/10
67/67 [==============================] - 8s 123ms/step - loss: 0.7297 -
accuracy: 0.7130 - val_loss: 0.7378 - val_accuracy: 0.7000
Epoch 8/10
67/67 [==============================] - 8s 120ms/step - loss: 0.7172 -
accuracy: 0.7054 - val_loss: 0.7021 - val_accuracy: 0.7333
Epoch 9/10
67/67 [==============================] - 8s 123ms/step - loss: 0.6817 -
accuracy: 0.7329 - val_loss: 0.6721 - val_accuracy: 0.7333
Epoch 10/10
67/67 [==============================] - 10s 144ms/step - loss: 0.6614 -
accuracy: 0.7400 - val_loss: 0.6527 - val_accuracy: 0.7333
```

[ ]:

```
#Evaluar accuracy vs val accuracy a través de las épocas
acc = history.history["accuracy"]
val_acc = history.history["val_accuracy"]
plt.figure(figsize=(8,8))
plt.plot(acc,label="Training Accuracy",color="red")
plt.plot(val_acc,label="Validation Accuracy",color="green")
plt.legend()
```

[ ]:

```
<matplotlib.legend.Legend at 0x790a821465c0>
```



0.5 ## Fine-Tuning del modelo

[ ]:
```
base_model.trainable = True
print("Número de capas del modelo:", len(base_model.layers))
```

```
Número de capas del modelo: 175
```

[ ]:
```
fine_tune_at = 100
for layer in base_model.layers[:fine_tune_at]:
  layer.trainable=False
```

[ ]:
```
model.compile(tf.keras.optimizers.RMSprop(learning_rate=0.0001), loss=tf.keras.
 ↪losses.SparseCategoricalCrossentropy(from_logits=True), metrics=["accuracy"])
```

```
model.summary()
```

```
Model: "model"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 224, 224, 3)]     0

 sequential (Sequential)     (None, 224, 224, 3)       0

 tf.math.truediv (TFOpLambd  (None, 224, 224, 3)       0
 a)

 tf.math.subtract (TFOpLamb  (None, 224, 224, 3)       0
 da)

 resnet50 (Functional)       (None, 7, 7, 2048)        23587712

 global_average_pooling2d (  (None, 2048)              0
 GlobalAveragePooling2D)

 dropout (Dropout)           (None, 2048)              0

 dense (Dense)               (None, 3)                 6147

=================================================================
Total params: 23593859 (90.00 MB)
Trainable params: 19459075 (74.23 MB)
Non-trainable params: 4134784 (15.77 MB)
_________________________________________________________________
```

[ ]:

```
model.save("model.keras")
```

[ ]:

```
#Definir un callback para impedir overfitting y elegir el mejor modelo
checkpoint = tf.keras.callbacks.ModelCheckpoint(filepath="model.keras",
    save_best_only=True,
    monitor='val_loss',
    mode='min',
    save_weights_only=True,
    verbose=1  )
```

[ ]:

```
#Entrenar el modelo con fine tuning
epochs = 30 #10 iniciales + 20 fine tuning
history_ft = model.fit(train_dataset,epochs = epochs, initial_epoch=10,␣
 ↪validation_data=validation_dataset, callbacks=[checkpoint])
```

```
Epoch 11/30
```

```
67/67 [==============================] - ETA: 0s - loss: 1.3046 - accuracy:
0.5352
Epoch 11: val_loss improved from inf to 2.02484, saving model to model.keras
67/67 [==============================] - 31s 219ms/step - loss: 1.3046 -
accuracy: 0.5352 - val_loss: 2.0248 - val_accuracy: 0.1833
Epoch 12/30
67/67 [==============================] - ETA: 0s - loss: 0.7332 - accuracy:
0.7310
Epoch 12: val_loss improved from 2.02484 to 0.57773, saving model to model.keras
67/67 [==============================] - 15s 217ms/step - loss: 0.7332 -
accuracy: 0.7310 - val_loss: 0.5777 - val_accuracy: 0.8000
Epoch 13/30
66/67 [============================>.] - ETA: 0s - loss: 0.5057 - accuracy:
0.7992
Epoch 13: val_loss did not improve from 0.57773
67/67 [==============================] - 13s 195ms/step - loss: 0.5066 -
accuracy: 0.7986 - val_loss: 0.6362 - val_accuracy: 0.5667
Epoch 14/30
66/67 [============================>.] - ETA: 0s - loss: 0.4404 - accuracy:
0.8452
Epoch 14: val_loss improved from 0.57773 to 0.34269, saving model to model.keras
67/67 [==============================] - 14s 200ms/step - loss: 0.4398 -
accuracy: 0.8454 - val_loss: 0.3427 - val_accuracy: 0.8500
Epoch 15/30
66/67 [============================>.] - ETA: 0s - loss: 0.3742 - accuracy:
0.8670
Epoch 15: val_loss did not improve from 0.34269
67/67 [==============================] - 13s 193ms/step - loss: 0.3744 -
accuracy: 0.8667 - val_loss: 0.6719 - val_accuracy: 0.7167
Epoch 16/30
66/67 [============================>.] - ETA: 0s - loss: 0.3434 - accuracy:
0.8722
Epoch 16: val_loss did not improve from 0.34269
67/67 [==============================] - 13s 197ms/step - loss: 0.3430 -
accuracy: 0.8723 - val_loss: 0.9536 - val_accuracy: 0.6833
Epoch 17/30
66/67 [============================>.] - ETA: 0s - loss: 0.3436 - accuracy:
0.8736
Epoch 17: val_loss improved from 0.34269 to 0.30858, saving model to model.keras
67/67 [==============================] - 15s 220ms/step - loss: 0.3431 -
accuracy: 0.8738 - val_loss: 0.3086 - val_accuracy: 0.9000
Epoch 18/30
66/67 [============================>.] - ETA: 0s - loss: 0.3196 - accuracy:
0.8812
Epoch 18: val_loss did not improve from 0.30858
67/67 [==============================] - 13s 194ms/step - loss: 0.3204 -
accuracy: 0.8809 - val_loss: 0.4162 - val_accuracy: 0.8000
Epoch 19/30
```

```
66/67 [============================>.] - ETA: 0s - loss: 0.2510 - accuracy:
0.9091
Epoch 19: val_loss improved from 0.30858 to 0.27282, saving model to model.keras
67/67 [==============================] - 13s 199ms/step - loss: 0.2506 -
accuracy: 0.9092 - val_loss: 0.2728 - val_accuracy: 0.9167
Epoch 20/30
67/67 [==============================] - ETA: 0s - loss: 0.2543 - accuracy:
0.9130
Epoch 20: val_loss improved from 0.27282 to 0.11010, saving model to model.keras
67/67 [==============================] - 14s 211ms/step - loss: 0.2543 -
accuracy: 0.9130 - val_loss: 0.1101 - val_accuracy: 0.9500
Epoch 21/30
66/67 [============================>.] - ETA: 0s - loss: 0.2528 - accuracy:
0.9138
Epoch 21: val_loss did not improve from 0.11010
67/67 [==============================] - 15s 217ms/step - loss: 0.2524 -
accuracy: 0.9139 - val_loss: 0.3350 - val_accuracy: 0.9167
Epoch 22/30
66/67 [============================>.] - ETA: 0s - loss: 0.2428 - accuracy:
0.9190
Epoch 22: val_loss did not improve from 0.11010
67/67 [==============================] - 13s 193ms/step - loss: 0.2424 -
accuracy: 0.9191 - val_loss: 0.2514 - val_accuracy: 0.9333
Epoch 23/30
66/67 [============================>.] - ETA: 0s - loss: 0.2375 - accuracy:
0.9186
Epoch 23: val_loss did not improve from 0.11010
67/67 [==============================] - 13s 193ms/step - loss: 0.2373 -
accuracy: 0.9187 - val_loss: 0.2363 - val_accuracy: 0.9333
Epoch 24/30
66/67 [============================>.] - ETA: 0s - loss: 0.2461 - accuracy:
0.9162
Epoch 24: val_loss did not improve from 0.11010
67/67 [==============================] - 13s 196ms/step - loss: 0.2459 -
accuracy: 0.9163 - val_loss: 0.2955 - val_accuracy: 0.9167
Epoch 25/30
66/67 [============================>.] - ETA: 0s - loss: 0.2339 - accuracy:
0.9157
Epoch 25: val_loss did not improve from 0.11010
67/67 [==============================] - 13s 199ms/step - loss: 0.2344 -
accuracy: 0.9154 - val_loss: 0.2673 - val_accuracy: 0.9333
Epoch 26/30
66/67 [============================>.] - ETA: 0s - loss: 0.1915 - accuracy:
0.9276
Epoch 26: val_loss did not improve from 0.11010
67/67 [==============================] - 13s 196ms/step - loss: 0.1918 -
accuracy: 0.9272 - val_loss: 4.1172 - val_accuracy: 0.5833
Epoch 27/30
```

```
66/67 [============================>.] - ETA: 0s - loss: 0.2359 - accuracy:
0.9238
Epoch 27: val_loss did not improve from 0.11010
67/67 [==============================] - 13s 194ms/step - loss: 0.2362 -
accuracy: 0.9234 - val_loss: 0.2158 - val_accuracy: 0.9167
Epoch 28/30
66/67 [============================>.] - ETA: 0s - loss: 0.2020 - accuracy:
0.9295
Epoch 28: val_loss did not improve from 0.11010
67/67 [==============================] - 13s 195ms/step - loss: 0.2018 -
accuracy: 0.9296 - val_loss: 1.3978 - val_accuracy: 0.7667
Epoch 29/30
66/67 [============================>.] - ETA: 0s - loss: 0.1897 - accuracy:
0.9356
Epoch 29: val_loss did not improve from 0.11010
67/67 [==============================] - 13s 194ms/step - loss: 0.1895 -
accuracy: 0.9357 - val_loss: 0.1893 - val_accuracy: 0.9167
Epoch 30/30
67/67 [==============================] - ETA: 0s - loss: 0.1883 - accuracy:
0.9352
Epoch 30: val_loss did not improve from 0.11010
67/67 [==============================] - 15s 229ms/step - loss: 0.1883 -
accuracy: 0.9352 - val_loss: 0.1228 - val_accuracy: 0.9500
```

[ ]:

```
model.save("model.keras")
```

[ ]:

```
model.get_metrics_result()
```

[ ]:

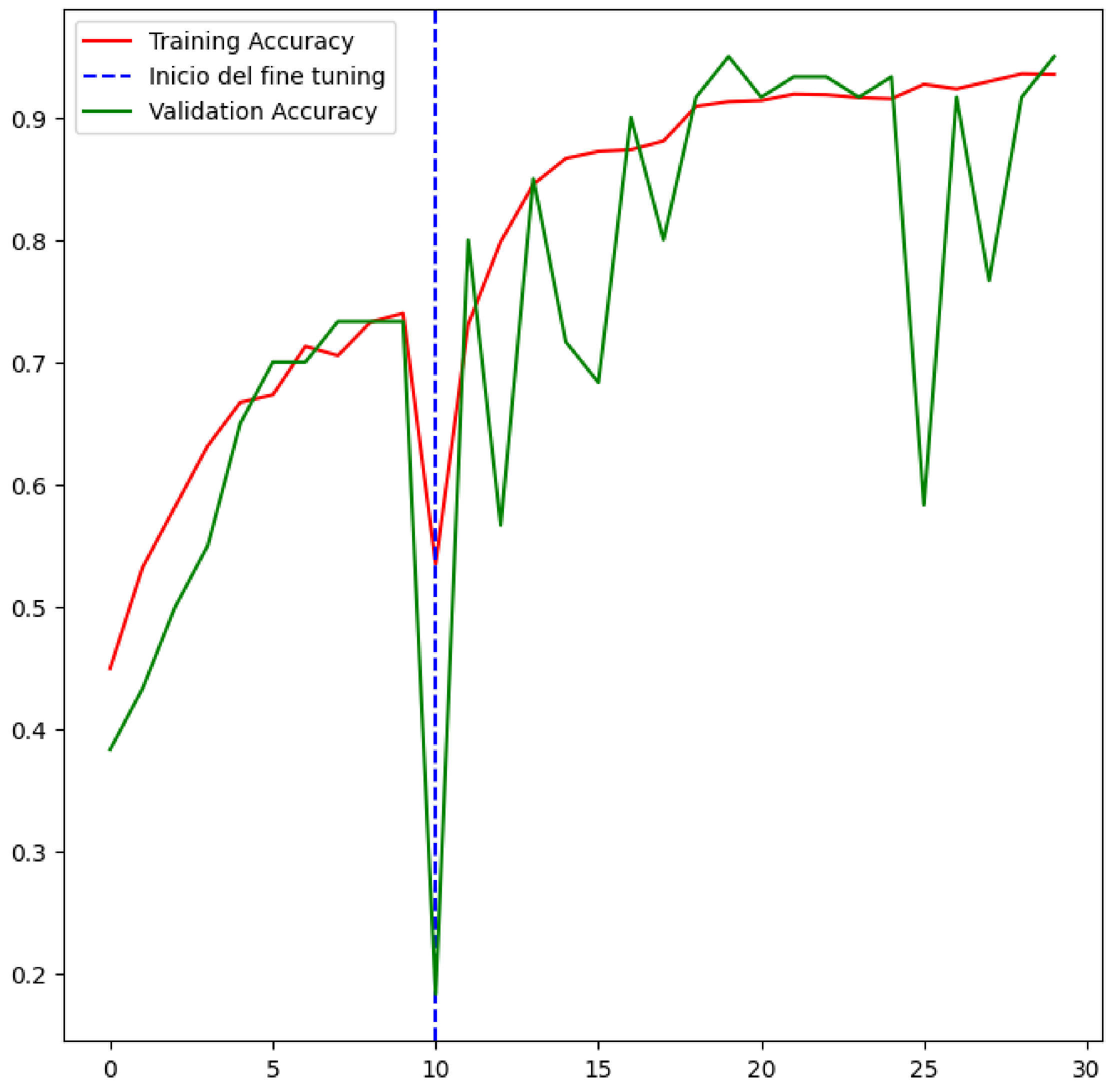
```
{'loss': <tf.Tensor: shape=(), dtype=float32, numpy=0.12280381>,
 'accuracy': <tf.Tensor: shape=(), dtype=float32, numpy=0.95>}
```

[ ]:

```
#Grafica accuracy vs val accuracy con fine tuning
acc += history_ft.history["accuracy"]
val_acc += history_ft.history["val_accuracy"]
plt.figure(figsize=(8,8))
plt.plot(acc,label="Training Accuracy",color="red")
plt.axvline(x=10, color='blue', linestyle='--', label='Inicio del fine tuning')
plt.plot(val_acc,label="Validation Accuracy",color="green")
plt.legend()
```

[ ]:

```
<matplotlib.legend.Legend at 0x790aa028fa90>
```



0.6 ## Predicciones

[ ]:
```
#Generar predicciones con el modelo final
predicciones = model.predict(test_dataset)
```

```
1/1 [==============================] - 2s 2s/step
```

[ ]:
```
predicciones[0] #probabilidades de cada una de las neruonas
```

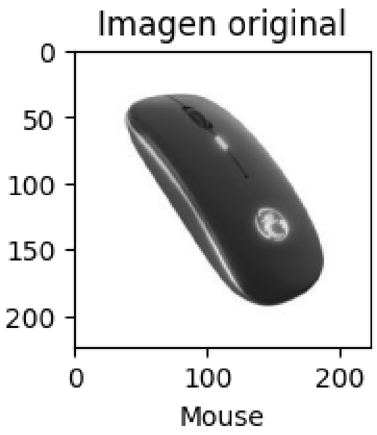
[ ]:
```
array([-4.974368 , -3.1757433,  5.122171 ], dtype=float32)
```

[ ]:
```
print("La imagen pertence al grupo {} con una probabilidad de {:.2f} %"
.format(class_names[np.argmax(predicciones[0])], 100 + np.max(predicciones[0])))
```

```
La imagen pertence al grupo Mouse con una probabilidad de 105.12 %
```

[40]:

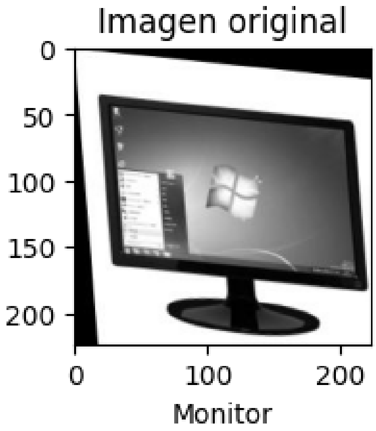
```
n= 10 #Número de imagen

for images, labels in test_dataset.take(1):
    numpy_images = images.numpy().astype("uint8")
    numpy_labels = labels.numpy()

plt.figure(figsize= (2,2))
plt.imshow(numpy_images[n])
plt.xlabel(class_names[numpy_labels[n]])
plt.title('Imagen original')
plt.show()

#Generar predicciones con el modelo final
predicciones = model.predict(test_dataset)

predicciones[n] #probabilidades de cada una de las neruonas

print("La imagen pertence al grupo {} con una probabilidad de {:.2f} %"
.format(class_names[np.argmax(predicciones[n])], 100 + np.max(predicciones[n])))
```



```
1/1 [==============================] - 0s 38ms/step
La imagen pertence al grupo Mouse con una probabilidad de 107.70 %
```

[37]:

```
n= 25 #Número de imagen

for images, labels in test_dataset.take(1):
    numpy_images = images.numpy().astype("uint8")
    numpy_labels = labels.numpy()

plt.figure(figsize= (2,2))
plt.imshow(numpy_images[n])
plt.xlabel(class_names[numpy_labels[n]])
plt.title('Imagen original')
plt.show()

#Generar predicciones con el modelo final
predicciones = model.predict(test_dataset)

predicciones[n] #probabilidades de cada una de las neruonas

print("La imagen pertence al grupo {} con una probabilidad de {:.2f} %"
.format(class_names[np.argmax(predicciones[n])], 100 + np.max(predicciones[n])))
```



```
1/1 [==============================] - 0s 36ms/step
La imagen pertence al grupo Monitor con una probabilidad de 105.53 %
```