In [1]:
```
import tensorflow as tf
import tensorflow.keras as keras
import tensorflow_probability as tf
import pandas as pd
import numpy as np
import io
from PIL import Image
import os
import matplotlib.pyplot as plt
import shutil
from glob import glob
from sklearn.model_selection import train_test_split
import cv2
%matplotlib inline
import matplotlib.pyplot as plt

import pandas as pd
import pickle
import csv

from sklearn.metrics import confusion_matrix, classification_report
import tensorflow as tf
from PIL import Image

import os
```

```
2023-12-04 07:35:11.931545: I tensorflow/core/platform/cpu_feature_guard.cc:182] This Te
nsorFlow binary is optimized to use available CPU instructions in performance-critical o
perations.
To enable the following instructions: SSE4.1 SSE4.2, in other operations, rebuild Tensor
Flow with the appropriate compiler flags.
```

In [3]:
```
from keras.preprocessing.image import ImageDataGenerator
from keras.applications import densenet
from keras.models import Sequential,Model
from keras.layers import Conv2D, MaxPooling2D, Dense, Flatten, Dropout,Activation
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam
```

In [4]:
```
df=pd.read_csv('archivo.csv')
```

In [5]:
```
magen = sorted(glob(os.path.join("YO", "*")))
```

Intenté seguir la metodología del caso anterior

In [6]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Lee el archivo CSV que contiene los nombres de las imágenes y las etiquetas
df_train = pd.read_csv('archivo.csv')
df_train['Values'] = df_train['Values'].astype('str')

# Crea un generador de datos de imágenes desde el DataFrame
datagen = ImageDataGenerator(rescale=1.0/255) # Normaliza los valores de píxeles entre 0
train_generator_df = datagen.flow_from_dataframe(
    dataframe=df_train,
    directory='Yo',
    x_col="File Names",  # Columna que contiene los nombres de las imágenes
    y_col="Values",  # Columna que contiene las etiquetas (en formato binario)
    class_mode="binary",  # Modo de clasificación (en este caso binario)
```

```
    target_size=(200, 200),  # Tamaño al que se redimensionarán las imágenes
    batch_size=1,

)
for i in range(4):#Para las demás imagenes
    image, label = next(train_generator_df)

    image = image[0].astype('uint8')#porque hace rato me arrojaba un error de incompatib

    # Graficar la imagen
    ax[i].imshow(image)
    ax[i].axis('off')
```

```
Found 73 validated image filenames belonging to 2 classes.
/opt/anaconda3/envs/testenv/lib/python3.11/site-packages/keras/preprocessing/image.py:11
37: UserWarning: Found 4 invalid image filename(s) in x_col="File Names". These filename
(s) will be ignored.
  warnings.warn(
---------------------------------------------------------------------------
UnidentifiedImageError                    Traceback (most recent call last)
Cell In[6], line 23
     12 train_generator_df = datagen.flow_from_dataframe(
     13     dataframe=df_train,
     14     directory='Yo',
   (...)
     20
     21 )
     22 for i in range(4):#Para las demás imagenes
---> 23     image, label = next(train_generator_df)
     25     image = image[0].astype('uint8')#porque hace rato me arrojaba un error de in
compatibilidad
     27     # Graficar la imagen

File /opt/anaconda3/envs/testenv/lib/python3.11/site-packages/keras/preprocessing/image.
py:156, in Iterator.__next__(self, *args, **kwargs)
    155 def __next__(self, *args, **kwargs):
--> 156     return self.next(*args, **kwargs)

File /opt/anaconda3/envs/testenv/lib/python3.11/site-packages/keras/preprocessing/image.
py:168, in Iterator.next(self)
    165     index_array = next(self.index_generator)
    166 # The transformation of images is not under thread lock
    167 # so it can be done in parallel
--> 168 return self._get_batches_of_transformed_samples(index_array)

File /opt/anaconda3/envs/testenv/lib/python3.11/site-packages/keras/preprocessing/image.
py:370, in BatchFromFilesMixin._get_batches_of_transformed_samples(self, index_array)
    368 filepaths = self.filepaths
    369 for i, j in enumerate(index_array):
--> 370     img = image_utils.load_img(
    371         filepaths[j],
    372         color_mode=self.color_mode,
    373         target_size=self.target_size,
    374         interpolation=self.interpolation,
    375         keep_aspect_ratio=self.keep_aspect_ratio,
    376     )
    377     x = image_utils.img_to_array(img, data_format=self.data_format)
    378     # Pillow images should be closed after `load_img`,
    379     # but not PIL images.

File /opt/anaconda3/envs/testenv/lib/python3.11/site-packages/keras/utils/image_utils.p
y:423, in load_img(path, grayscale, color_mode, target_size, interpolation, keep_aspect_
ratio)
    421         path = str(path.resolve())
    422     with open(path, "rb") as f:
```

```
--> 423         img = pil_image.open(io.BytesIO(f.read()))
    424 else:
    425     raise TypeError(
    426         f"path should be path-like or io.BytesIO, not {type(path)}"
    427     )

File /opt/anaconda3/envs/testenv/lib/python3.11/site-packages/PIL/Image.py:3280, in open
(fp, mode, formats)
   3278     warnings.warn(message)
   3279 msg = "cannot identify image file %r" % (filename if filename else fp)
-> 3280 raise UnidentifiedImageError(msg)

UnidentifiedImageError: cannot identify image file <_io.BytesIO object at 0x21eceb2e0>
```

In [7]:

```
model = tf.keras.models.load_model("my_model.h5")
modeloo=keras.models.Sequential()
model.add(Conv2D(64,kernel_size=3,input_shape=(224,224,3)))
model.add("my_model.h5".layers[-2].output)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[7], line 3
      1 model = tf.keras.models.load_model("my_model.h5")
      2 modeloo=keras.models.Sequential()
----> 3 model.add(Conv2D(64,kernel_size=3,input_shape=(224,224,3)))
      4 model.add("my_model.h5".layers[-2].output)

File /opt/anaconda3/envs/testenv/lib/python3.11/site-packages/tensorflow/python/trackabl
e/base.py:205, in no_automatic_dependency_tracking.<locals>._method_wrapper(self, *args,
 **kwargs)
    203 self._self_setattr_tracking = False  # pylint: disable=protected-access
    204 try:
--> 205   result = method(self, *args, **kwargs)
    206 finally:
    207   self._self_setattr_tracking = previous_value  # pylint: disable=protected-acce
ss

File /opt/anaconda3/envs/testenv/lib/python3.11/site-packages/keras/utils/traceback_util
s.py:70, in filter_traceback.<locals>.error_handler(*args, **kwargs)
     67     filtered_tb = _process_traceback_frames(e.__traceback__)
     68     # To get the full stack trace, call:
     69     # `tf.debugging.disable_traceback_filtering()`
---> 70     raise e.with_traceback(filtered_tb) from None
     71 finally:
     72     del filtered_tb

File /opt/anaconda3/envs/testenv/lib/python3.11/site-packages/keras/engine/input_spec.p
y:253, in assert_input_compatibility(input_spec, inputs, layer_name)
    251     ndim = x.shape.rank
    252     if ndim is not None and ndim < spec.min_ndim:
--> 253         raise ValueError(
    254             f'Input {input_index} of layer "{layer_name}" '
    255             "is incompatible with the layer: "
    256             f"expected min_ndim={spec.min_ndim}, "
    257             f"found ndim={ndim}. "
    258             f"Full shape received: {tuple(shape)}"
    259         )
    260 # Check dtype.
    261 if spec.dtype is not None:

ValueError: Input 0 of layer "conv2d" is incompatible with the layer: expected min_ndim=
4, found ndim=2. Full shape received: (None, 40)
```

En el paso anterior intenté quitar las dos capas pero o lo logré y luego iría entrenar de nuevo, pero también me enredé para definir

In [ ]:
```
modeloo = Sequential(
    new model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy']
```