TECNOLÓGICO DE MONTERREY®

Maestría en Inteligencia Artificial Aplicada

Curso: Navegación autónoma

Tecnológico de Monterrey

Prof Titular y Tutor: Dr. David Antonio Torres

Prof Asistente: Maricarmen Vázquez Rojí

ALUMNO: Luis Alfonso Sabanero Esquivel

MATRICULA: A01273286

ALUMNO: Jose Mtanous

MATRICULA: A00169781

ALUMNO: Guillermo Alfonso Muñiz Hermosillo

MATRICULA: A01793101

ALUMNO: Jorge Mariles Estrada

MATRICULA: A01335663

Actividad de la Semana 07

Actividad 4.1 - Ejercicio de clasificación con Keras

Junio 2023

- Enlace a youtube: https://www.youtube.com/watch?v=_hCi91eoft0

In [ ]:
```
#Cargamos las librerias necesarias para trabajar con redes neuronales
import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt
import random
from tensorflow.keras import layers
import os
```

```
2023-06-01 18:55:17.378460: I tensorflow/core/util/port.cc:110] oneDNN custom operati
ons are on. You may see slightly different numerical results due to floating-point ro
und-off errors from different computation orders. To turn them off, set the environme
nt variable `TF_ENABLE_ONEDNN_OPTS=0`.
2023-06-01 18:55:17.584482: I tensorflow/core/platform/cpu_feature_guard.cc:182] This
TensorFlow binary is optimized to use available CPU instructions in performance-criti
cal operations.
To enable the following instructions: AVX2 AVX512F AVX512_VNNI FMA, in other operatio
ns, rebuild TensorFlow with the appropriate compiler flags.
2023-06-01 18:55:18.857294: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] T
F-TRT Warning: Could not find TensorRT
```

Enseguida comenzamos con la importacion de datos. Para el caso de google drive tenemos que montar nuestro drive para poder ver los datos de nuestro dataset.

En caso de contar con una implementacion local, necesitamos navegar al folder donde se encuentren nuestro dataset para comenzar a trabajar.

In [ ]:
```
#Codigo necesario para google collab
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [ ]:
```
# datadir = "/home/gmuniz/JupyterFolder/Tareas/navegacionAutonoma/data"
%cd "/home/gmuniz/JupyterFolder/Tareas/navegacionAutonoma/data"
homedir = %pwd

print(homedir)
import os
```

```
/home/gmuniz/JupyterFolder/Tareas/navegacionAutonoma/data
/home/gmuniz/JupyterFolder/Tareas/navegacionAutonoma/data
```

Una vez instalados en el directorio donde se encuentran nuestras imagenes de peatones y no peatones. Convertimos dichas imagenes en un formato unificado, es decir jpg.

Repetimos el proceso para ambos tipos de imagenes

In [ ]:
```
#La funcion de tensorflow requiere que el label de la clase venga como parte del nombr
directorio = 'NoPedestrians'  # Ruta del directorio donde se encuentran los archivos

for nombre_archivo in os.listdir(directorio):
    if nombre_archivo.endswith('.png'):  # Renombrar solo los archivos que terminan er
        antiguo_nombre = os.path.join(directorio, nombre_archivo)
```

```
        nuevo_nombre = os.path.join(directorio, 'NoPedestrians_' + nombre_archivo)
        os.rename(antiguo_nombre, nuevo_nombre)
```

In [ ]:
```
#La funcion de tensorflow requiere que el label de la clase venga como parte del nombr

directorio = 'Pedestrians'  # Ruta del directorio donde se encuentran los archivos

for nombre_archivo in os.listdir(directorio):
    if nombre_archivo.endswith('.png'):  # Renombrar solo los archivos que terminan er
        antiguo_nombre = os.path.join(directorio, nombre_archivo)
        nuevo_nombre = os.path.join(directorio, 'Pedestrians_' + nombre_archivo)
        os.rename(antiguo_nombre, nuevo_nombre)
```

Una vez convertidas a formato jpg, importamos nuestras imagenes a un dataset utilizando la libreria de keras la cual nos permite establecer nuestros conjuntos de entrenamiento y validacion, asi como convertir nuestras imagenes a escala de grises y definir un tamaño de imagen y de batch, todo en un mismo comando.

In [ ]:
```
image_size = (64, 32) #Variable necesaria para hacer el resize a la imagen
batch_size = 128  #Tamaño del lote

#La funcion opera de los siguientes parametros:
#Ruta de las imagenes, tamaño del set de entrenamiento, requiere o no validatio set, s
#las dos variables de tamaño de imagen y tamaño de lote
#Como solo tenemos dos tipo de imagenes, se considera clasificación binaria
train_ds, val_ds = tf.keras.utils.image_dataset_from_directory(
    "/home/gmuniz/JupyterFolder/Tareas/navegacionAutonoma/data",
    validation_split=0.2,
    subset="both",
    seed=1337,
    labels='inferred',
    label_mode='binary',
    color_mode='grayscale',
    image_size=image_size,
    batch_size=batch_size,
)
```

```
Found 2222 files belonging to 2 classes.
Using 1778 files for training.
Using 444 files for validation.
```

In [ ]:
```
#Probamos que el codigo funcione, vemos las caracteristicas del objeto
print(train_ds.take(1))
```

```
<_TakeDataset element_spec=(TensorSpec(shape=(None, 64, 32, 1), dtype=tf.float32, nam
e=None), TensorSpec(shape=(None, 1), dtype=tf.float32, name=None))>
```

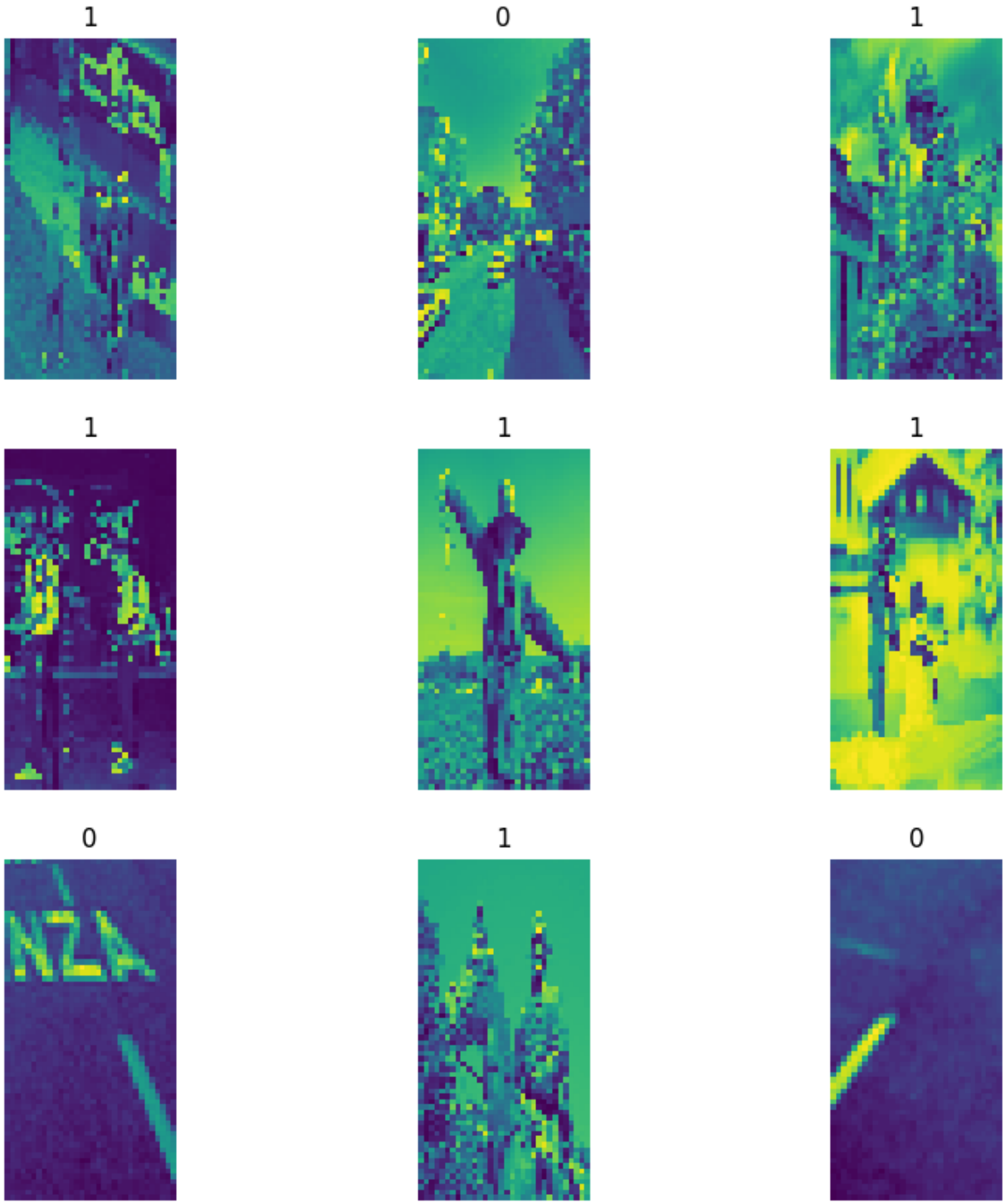
Una vez definido nuestro conjunto de datos podemos imprimir un subconjunto de nuestras imagenes y sus etiquetas para ver la data con la que estaremos trabajando.

In [ ]:
```
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 10))
for images, labels in train_ds.take(1):
    for i in range(9):
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))
```

```
        plt.title(int(labels[i]))
        plt.axis("off")
```

```
2023-06-01 19:02:30.267437: I tensorflow/core/common_runtime/executor.cc:1197] [/devi
ce:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and y
ou can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder
tensor 'Placeholder/_4' with dtype int32 and shape [1778]
         [[{{node Placeholder/_4}}]]
2023-06-01 19:02:30.267828: I tensorflow/core/common_runtime/executor.cc:1197] [/devi
ce:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and y
ou can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder
tensor 'Placeholder/_4' with dtype int32 and shape [1778]
         [[{{node Placeholder/_4}}]]
```



El siguiente paso sera ahora si construir nuestra red neuronal. En este paso es necesario jugar con los parametros por lo que al final mostraremos solo el modelo que consideramos entrega los mejores parametros.

In [ ]:
```
target_shape=(64,32)

#Dado que usamos imagenes de 64x32, el numero de neuronas necesarias para la primera c
#Todas las capas usaran el metodo de activación Relu
#La capa de salida, usara el metodo de activación sigmoide al ser binaria la clasifica
#Usaremos dos capas de dropout para prevenir el sobreentrenamiento del modelo de 0.5
#Se aplicara un rescaling para normalizar todas las entradas y un flatten para hacer q

#Importante: No usar un numero grande neuronas en local, problemas de memoria
model = tf.keras.Sequential([
     tf.keras.layers.Rescaling(1./255, input_shape=(image_size[0],image_size[1], 1)),
    tf.keras.layers.Flatten(input_shape=(image_size[0],image_size[1], 1)),
    tf.keras.layers.Dense(2048, activation='relu'),
    tf.keras.layers.Dense(1024, activation='relu'),
     tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(256, activation='relu'),
    tf.keras.layers.Dense(128, activation='relu'),
     tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(32, activation='relu'),
    tf.keras.layers.Dense(16, activation='relu'),
    tf.keras.layers.Dense(1,activation='sigmoid')
])
```

In [ ]:
```
#Usaremos el metodo de optimización adam, junto con la funcion de perdida binarycrosse
#La metrica de este modelo sera el accurancy
model.compile(optimizer='adam',
              loss=tf.keras.losses.BinaryCrossentropy(from_logits=True),
              metrics=['accuracy'])
```

In [ ]:
```
#Empezamos el entrenamiento, por mejores resultados se usaron 35 epocas
history=model.fit(train_ds, validation_data = val_ds, epochs = 35,)
```

```
Epoch 1/35
2023-06-01 19:04:54.721418: I tensorflow/core/common_runtime/executor.cc:1197] [/devi
ce:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and y
ou can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder
tensor 'Placeholder/_0' with dtype string and shape [1778]
         [[{{node Placeholder/_0}}]]
2023-06-01 19:04:54.721700: I tensorflow/core/common_runtime/executor.cc:1197] [/devi
ce:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and y
ou can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder
tensor 'Placeholder/_0' with dtype string and shape [1778]
         [[{{node Placeholder/_0}}]]
/home/gmuniz/.local/lib/python3.10/site-packages/keras/backend.py:5703: UserWarning:
"`binary_crossentropy` received `from_logits=True`, but the `output` argument was pro
duced by a Sigmoid activation and thus does not represent logits. Was this intended?
  output, from_logits = _get_logits(
14/14 [==============================] - ETA: 0s - loss: 0.7202 - accuracy: 0.4994
```

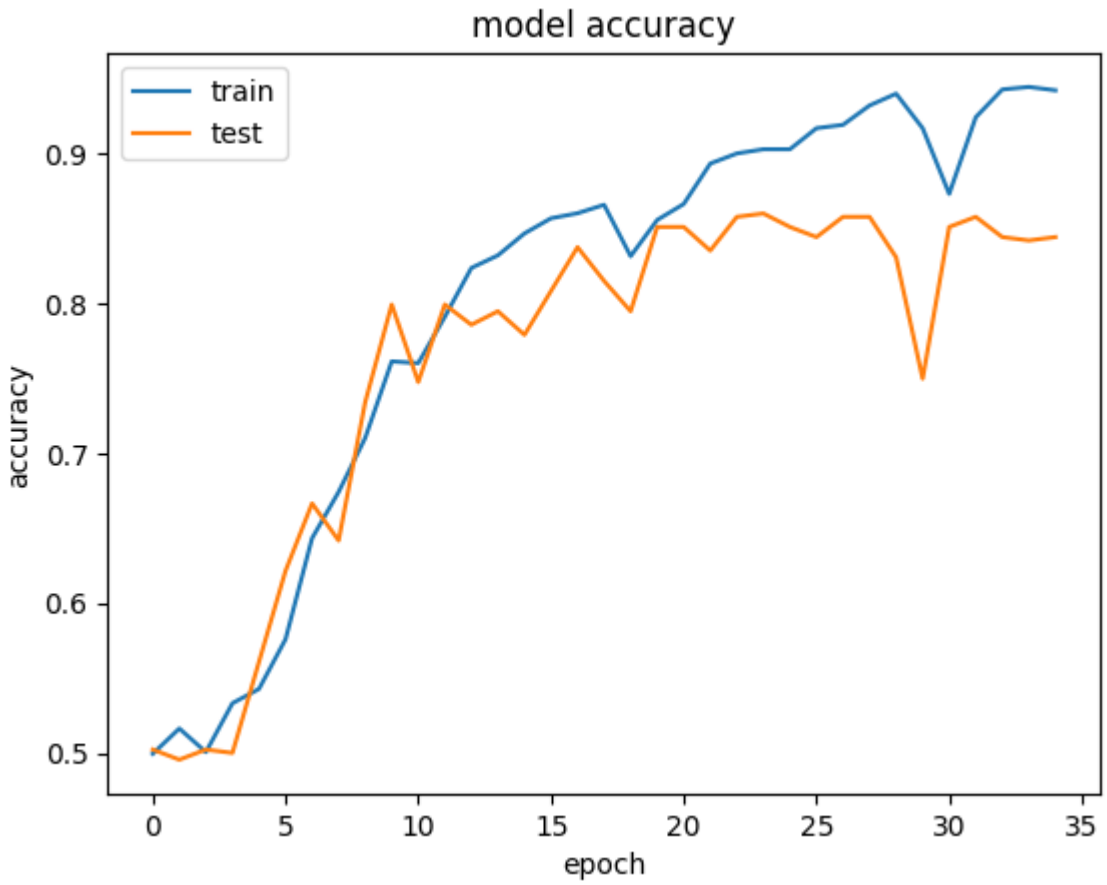
```
2023-06-01 19:04:57.234009: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'Placeholder/_4' with dtype int32 and shape [444]
         [[{{node Placeholder/_4}}]]
2023-06-01 19:04:57.234282: I tensorflow/core/common_runtime/executor.cc:1197] [/device:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and you can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder tensor 'Placeholder/_4' with dtype int32 and shape [444]
         [[{{node Placeholder/_4}}]]
```

```
14/14 [==============================] - 3s 94ms/step - loss: 0.7202 - accuracy: 0.49
94 - val_loss: 0.7435 - val_accuracy: 0.5023
Epoch 2/35
14/14 [==============================] - 2s 90ms/step - loss: 0.7089 - accuracy: 0.51
63 - val_loss: 0.6853 - val_accuracy: 0.4955
Epoch 3/35
14/14 [==============================] - 2s 91ms/step - loss: 0.6932 - accuracy: 0.50
06 - val_loss: 0.6861 - val_accuracy: 0.5023
Epoch 4/35
14/14 [==============================] - 2s 89ms/step - loss: 0.6836 - accuracy: 0.53
32 - val_loss: 0.6808 - val_accuracy: 0.5000
Epoch 5/35
14/14 [==============================] - 2s 88ms/step - loss: 0.6771 - accuracy: 0.54
27 - val_loss: 0.6723 - val_accuracy: 0.5608
Epoch 6/35
14/14 [==============================] - 2s 91ms/step - loss: 0.6516 - accuracy: 0.57
59 - val_loss: 0.6508 - val_accuracy: 0.6216
Epoch 7/35
14/14 [==============================] - 2s 90ms/step - loss: 0.6511 - accuracy: 0.64
34 - val_loss: 0.6515 - val_accuracy: 0.6667
Epoch 8/35
14/14 [==============================] - 2s 91ms/step - loss: 0.6250 - accuracy: 0.67
44 - val_loss: 0.6221 - val_accuracy: 0.6419
Epoch 9/35
14/14 [==============================] - 2s 91ms/step - loss: 0.5904 - accuracy: 0.71
03 - val_loss: 0.5660 - val_accuracy: 0.7342
Epoch 10/35
14/14 [==============================] - 2s 91ms/step - loss: 0.5267 - accuracy: 0.76
15 - val_loss: 0.4847 - val_accuracy: 0.7995
Epoch 11/35
14/14 [==============================] - 2s 91ms/step - loss: 0.5003 - accuracy: 0.76
04 - val_loss: 0.5238 - val_accuracy: 0.7477
Epoch 12/35
14/14 [==============================] - 2s 95ms/step - loss: 0.4618 - accuracy: 0.79
13 - val_loss: 0.4460 - val_accuracy: 0.7995
Epoch 13/35
14/14 [==============================] - 2s 99ms/step - loss: 0.3947 - accuracy: 0.82
40 - val_loss: 0.5728 - val_accuracy: 0.7860
Epoch 14/35
14/14 [==============================] - 2s 101ms/step - loss: 0.3864 - accuracy: 0.8
324 - val_loss: 0.5123 - val_accuracy: 0.7950
Epoch 15/35
14/14 [==============================] - 2s 96ms/step - loss: 0.3732 - accuracy: 0.84
70 - val_loss: 0.4594 - val_accuracy: 0.7793
Epoch 16/35
14/14 [==============================] - 2s 93ms/step - loss: 0.3390 - accuracy: 0.85
71 - val_loss: 0.4178 - val_accuracy: 0.8086
Epoch 17/35
14/14 [==============================] - 2s 92ms/step - loss: 0.3229 - accuracy: 0.86
05 - val_loss: 0.3942 - val_accuracy: 0.8378
Epoch 18/35
14/14 [==============================] - 2s 92ms/step - loss: 0.3181 - accuracy: 0.86
61 - val_loss: 0.3943 - val_accuracy: 0.8153
Epoch 19/35
14/14 [==============================] - 2s 93ms/step - loss: 0.3772 - accuracy: 0.83
18 - val_loss: 0.4757 - val_accuracy: 0.7950
Epoch 20/35
14/14 [==============================] - 2s 89ms/step - loss: 0.3147 - accuracy: 0.85
60 - val_loss: 0.4834 - val_accuracy: 0.8514
Epoch 21/35
```

```
14/14 [==============================] - 2s 93ms/step - loss: 0.3076 - accuracy: 0.86
67 - val_loss: 0.3865 - val_accuracy: 0.8514
Epoch 22/35
14/14 [==============================] - 2s 90ms/step - loss: 0.2597 - accuracy: 0.89
37 - val_loss: 0.4458 - val_accuracy: 0.8356
Epoch 23/35
14/14 [==============================] - 2s 91ms/step - loss: 0.2457 - accuracy: 0.90
04 - val_loss: 0.3507 - val_accuracy: 0.8581
Epoch 24/35
14/14 [==============================] - 2s 96ms/step - loss: 0.2330 - accuracy: 0.90
33 - val_loss: 0.3765 - val_accuracy: 0.8604
Epoch 25/35
14/14 [==============================] - 2s 92ms/step - loss: 0.2288 - accuracy: 0.90
33 - val_loss: 0.3953 - val_accuracy: 0.8514
Epoch 26/35
14/14 [==============================] - 3s 119ms/step - loss: 0.2174 - accuracy: 0.9
173 - val_loss: 0.3767 - val_accuracy: 0.8446
Epoch 27/35
14/14 [==============================] - 3s 115ms/step - loss: 0.1944 - accuracy: 0.9
196 - val_loss: 0.4596 - val_accuracy: 0.8581
Epoch 28/35
14/14 [==============================] - 3s 120ms/step - loss: 0.1697 - accuracy: 0.9
325 - val_loss: 0.3496 - val_accuracy: 0.8581
Epoch 29/35
14/14 [==============================] - 2s 103ms/step - loss: 0.1688 - accuracy: 0.9
404 - val_loss: 0.5542 - val_accuracy: 0.8311
Epoch 30/35
14/14 [==============================] - 2s 111ms/step - loss: 0.2195 - accuracy: 0.9
173 - val_loss: 0.4848 - val_accuracy: 0.7500
Epoch 31/35
14/14 [==============================] - 2s 104ms/step - loss: 0.2846 - accuracy: 0.8
735 - val_loss: 0.3297 - val_accuracy: 0.8514
Epoch 32/35
14/14 [==============================] - 2s 94ms/step - loss: 0.2009 - accuracy: 0.92
46 - val_loss: 0.3520 - val_accuracy: 0.8581
Epoch 33/35
14/14 [==============================] - 2s 97ms/step - loss: 0.1559 - accuracy: 0.94
32 - val_loss: 0.5835 - val_accuracy: 0.8446
Epoch 34/35
14/14 [==============================] - 2s 104ms/step - loss: 0.1417 - accuracy: 0.9
449 - val_loss: 0.5381 - val_accuracy: 0.8423
Epoch 35/35
14/14 [==============================] - 2s 97ms/step - loss: 0.1478 - accuracy: 0.94
26 - val_loss: 0.3972 - val_accuracy: 0.8446
```

In [ ]:

```
#Graficamos la evolución del entrenamiento
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```



In [ ]:
```
#Obtenemos imagenes de muestra para verificar las predicciones obtenida
images_full=[images for images, labels in val_ds.take(4)]
labels_full=[labels for images, labels in val_ds.take(4)]
```

```
2023-06-01 19:35:20.210873: I tensorflow/core/common_runtime/executor.cc:1197] [/devi
ce:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and y
ou can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder
tensor 'Placeholder/_4' with dtype int32 and shape [444]
         [[{{node Placeholder/_4}}]]
2023-06-01 19:35:20.211138: I tensorflow/core/common_runtime/executor.cc:1197] [/devi
ce:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and y
ou can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder
tensor 'Placeholder/_4' with dtype int32 and shape [444]
         [[{{node Placeholder/_4}}]]
2023-06-01 19:35:20.524933: I tensorflow/core/common_runtime/executor.cc:1197] [/devi
ce:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and y
ou can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder
tensor 'Placeholder/_4' with dtype int32 and shape [444]
         [[{{node Placeholder/_4}}]]
2023-06-01 19:35:20.525159: I tensorflow/core/common_runtime/executor.cc:1197] [/devi
ce:CPU:0] (DEBUG INFO) Executor start aborting (this does not indicate an error and y
ou can ignore this message): INVALID_ARGUMENT: You must feed a value for placeholder
tensor 'Placeholder/_4' with dtype int32 and shape [444]
         [[{{node Placeholder/_4}}]]
```

In [ ]:
```
#Imprimos algunas predicciones obtenidas por el modelo entrenado
cnt_samples=9
plt.figure(figsize=(10, 10))
batch= random.randint(1,val_ds.cardinality().numpy()-1)
images=images_full[batch]
labels=labels_full[batch]
```

```
predictions = model.predict(images).round() #Pasamos el grupo de imagenes obtenidas pa
l_items=[x for x in range(predictions.shape[0])]
sample_=random.sample(l_items,cnt_samples)


for tile,i in enumerate(sample_):
    ax = plt.subplot(3, 3, tile + 1)

    plt.imshow(images[i].numpy().astype("uint8"))
    plt.title(f'y={int(labels[i])}, yhat={int(predictions[i])}')
    plt.axis("off")
```

```
4/4 [==============================] - 0s 6ms/step
```