In [1]:
```python
# importing the required libraries
import tensorflow as tf
from tensorflow import keras
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras import models,layers
```

```
2023-10-05 15:08:03.491300: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations:  AVX2 AVX512F FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
VOC-NOTICE: GPU memory for this assignment is capped at 1024MiB
2023-10-05 15:08:06.591857: E tensorflow/compiler/xla/stream_executor/cuda/cuda_driver.cc:267] failed call to cuInit: CUDA_ERROR_NO_DEVICE: no CUDA-capable device is detected
```

In [2]:
```python
dataflow = ImageDataGenerator( rescale = 1.0 / 255.0 )
```

In [3]:
```python
# creating a train variable to store all the training data into a variable
train = dataflow.flow_from_directory( './data/train/', class_mode = 'binary')
```

```
Found 40 images belonging to 2 classes.
```

In [4]:
```python
# creating a test variable to store all the testing data into a variable
test = dataflow.flow_from_directory( './data/test', class_mode = 'binary')
```

```
Found 20 images belonging to 2 classes.
```

In [5]:
```python
# intializing the model with a variable name model
model = models.Sequential()

#Convolutional layer 1 with 32 filters of kernel size[5,5]
model.add( layers.Conv2D( 32, (5,5), activation = 'relu', padding = 'same', input_shape = (256, 256, 3)))

# Pooling layer 1 with pool size[2,2] and stride 2
model.add( layers.MaxPooling2D(2,2))

# Convolutional layer 2 with 64 filters of kernel size[5,5]
model.add( layers.Conv2D(64, (5,5), activation = 'relu' ))

# Pooling layer 2 with pool size[2,2] and stride 2
model.add( layers.MaxPooling2D(2,2))

# Dense layer whose output size is fixed in the hyper parameter: fc_size=32
model.add( layers.Flatten())
model.add( layers.Dense( 32, activation='relu'))

#Dropout layer with dropout probability 0.4
model.add(layers.Dropout(0.4))
model.add(layers.Dense(1, activation = 'sigmoid'))
```

In [6]:
```python
# displaying the summary of the model
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 256, 256, 32)      2432

 max_pooling2d (MaxPooling2D  (None, 128, 128, 32)     0
 )

 conv2d_1 (Conv2D)           (None, 124, 124, 64)      51264

 max_pooling2d_1 (MaxPooling  (None, 62, 62, 64)       0
 2D)

 flatten (Flatten)           (None, 246016)            0

 dense (Dense)               (None, 32)                7872544

 dropout (Dropout)           (None, 32)                0

 dense_1 (Dense)             (None, 1)                 33

=================================================================
Total params: 7,926,273
Trainable params: 7,926,273
Non-trainable params: 0
_________________________________________________________________
```

In [7]:
```python
sgd_opt = tf.keras.optimizers.SGD( learning_rate = 0.01)
```

In [8]:
```python
model.compile( optimizer = sgd_opt, loss = 'binary_crossentropy', metrics = ['accuracy'])
```

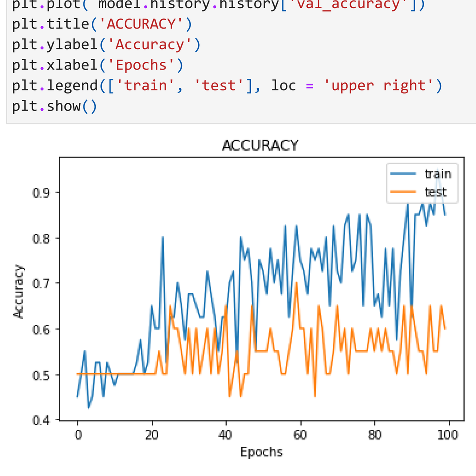
In [9]:
```python
model.fit( train, validation_data= test, epochs = 100 )
```

```
Epoch 1/100
2/2 [==============================] - 10s 2s/step - loss: 0.6948 - accuracy: 0.4500 - val_loss: 0.7075 - val_accuracy: 0.5000
Epoch 2/100
2/2 [==============================] - 7s 2s/step - loss: 0.7081 - accuracy: 0.5000 - val_loss: 0.7320 - val_accuracy: 0.5000
Epoch 3/100
2/2 [==============================] - 7s 2s/step - loss: 0.6864 - accuracy: 0.5500 - val_loss: 0.6951 - val_accuracy: 0.5000
Epoch 4/100
2/2 [==============================] - 7s 2s/step - loss: 0.7174 - accuracy: 0.4250 - val_loss: 0.6905 - val_accuracy: 0.5000
Epoch 5/100
2/2 [==============================] - 7s 5s/step - loss: 0.6868 - accuracy: 0.4500 - val_loss: 0.6906 - val_accuracy: 0.5000
Epoch 6/100
2/2 [==============================] - 7s 2s/step - loss: 0.6821 - accuracy: 0.5250 - val_loss: 0.6940 - val_accuracy: 0.5000
Epoch 7/100
2/2 [==============================] - 7s 2s/step - loss: 0.6992 - accuracy: 0.5250 - val_loss: 0.6928 - val_accuracy: 0.5000
Epoch 8/100
2/2 [==============================] - 7s 5s/step - loss: 0.6897 - accuracy: 0.4500 - val_loss: 0.6917 - val_accuracy: 0.5000
Epoch 9/100
2/2 [==============================] - 7s 2s/step - loss: 0.6827 - accuracy: 0.5250 - val_loss: 0.6897 - val_accuracy: 0.5000
Epoch 10/100
2/2 [==============================] - 7s 5s/step - loss: 0.6981 - accuracy: 0.5000 - val_loss: 0.6892 - val_accuracy: 0.5000
Epoch 11/100
2/2 [==============================] - 7s 2s/step - loss: 0.6774 - accuracy: 0.4750 - val_loss: 0.6891 - val_accuracy: 0.5000
Epoch 12/100
2/2 [==============================] - 7s 2s/step - loss: 0.6691 - accuracy: 0.5000 - val_loss: 0.6943 - val_accuracy: 0.5000
Epoch 13/100
2/2 [==============================] - 7s 2s/step - loss: 0.7009 - accuracy: 0.5000 - val_loss: 0.7016 - val_accuracy: 0.5000
Epoch 14/100
2/2 [==============================] - 7s 2s/step - loss: 0.7164 - accuracy: 0.5000 - val_loss: 0.6919 - val_accuracy: 0.5000
Epoch 15/100
2/2 [==============================] - 7s 5s/step - loss: 0.6953 - accuracy: 0.5000 - val_loss: 0.6926 - val_accuracy: 0.5000
Epoch 16/100
2/2 [==============================] - 7s 2s/step - loss: 0.6786 - accuracy: 0.5000 - val_loss: 0.6913 - val_accuracy: 0.5000
Epoch 17/100
2/2 [==============================] - 7s 2s/step - loss: 0.6606 - accuracy: 0.5250 - val_loss: 0.6940 - val_accuracy: 0.5000
Epoch 18/100
2/2 [==============================] - 7s 5s/step - loss: 0.6671 - accuracy: 0.5750 - val_loss: 0.6929 - val_accuracy: 0.5000
Epoch 19/100
2/2 [==============================] - 7s 5s/step - loss: 0.6643 - accuracy: 0.5000 - val_loss: 0.6931 - val_accuracy: 0.5000
Epoch 20/100
2/2 [==============================] - 7s 5s/step - loss: 0.6624 - accuracy: 0.5250 - val_loss: 0.6925 - val_accuracy: 0.5000
Epoch 21/100
2/2 [==============================] - 7s 5s/step - loss: 0.6402 - accuracy: 0.6500 - val_loss: 0.6978 - val_accuracy: 0.5000
Epoch 22/100
2/2 [==============================] - 7s 5s/step - loss: 0.6526 - accuracy: 0.6000 - val_loss: 0.6908 - val_accuracy: 0.5000
Epoch 23/100
2/2 [==============================] - 7s 5s/step - loss: 0.6824 - accuracy: 0.6000 - val_loss: 0.6906 - val_accuracy: 0.5500
Epoch 24/100
2/2 [==============================] - 7s 2s/step - loss: 0.6528 - accuracy: 0.8000 - val_loss: 0.6905 - val_accuracy: 0.5000
Epoch 25/100
2/2 [==============================] - 7s 2s/step - loss: 0.6568 - accuracy: 0.5250 - val_loss: 0.6911 - val_accuracy: 0.5000
Epoch 26/100
2/2 [==============================] - 7s 5s/step - loss: 0.6902 - accuracy: 0.6250 - val_loss: 0.6892 - val_accuracy: 0.6500
Epoch 27/100
2/2 [==============================] - 7s 2s/step - loss: 0.6706 - accuracy: 0.6250 - val_loss: 0.6885 - val_accuracy: 0.6000
Epoch 28/100
2/2 [==============================] - 7s 5s/step - loss: 0.6685 - accuracy: 0.7000 - val_loss: 0.6862 - val_accuracy: 0.6000
Epoch 29/100
2/2 [==============================] - 7s 5s/step - loss: 0.6617 - accuracy: 0.6500 - val_loss: 0.6855 - val_accuracy: 0.5500
Epoch 30/100
2/2 [==============================] - 7s 2s/step - loss: 0.6457 - accuracy: 0.5750 - val_loss: 0.6919 - val_accuracy: 0.5000
Epoch 31/100
2/2 [==============================] - 7s 5s/step - loss: 0.6492 - accuracy: 0.6750 - val_loss: 0.6867 - val_accuracy: 0.6000
Epoch 32/100
2/2 [==============================] - 7s 2s/step - loss: 0.6183 - accuracy: 0.6750 - val_loss: 0.6928 - val_accuracy: 0.5000
Epoch 33/100
2/2 [==============================] - 7s 5s/step - loss: 0.6490 - accuracy: 0.6500 - val_loss: 0.6904 - val_accuracy: 0.6000
Epoch 34/100
2/2 [==============================] - 7s 2s/step - loss: 0.6523 - accuracy: 0.6250 - val_loss: 0.6935 - val_accuracy: 0.5000
Epoch 35/100
2/2 [==============================] - 7s 5s/step - loss: 0.6122 - accuracy: 0.6250 - val_loss: 0.6939 - val_accuracy: 0.5500
Epoch 36/100
2/2 [==============================] - 7s 5s/step - loss: 0.6178 - accuracy: 0.7250 - val_loss: 0.6934 - val_accuracy: 0.6000
Epoch 37/100
2/2 [==============================] - 7s 2s/step - loss: 0.6327 - accuracy: 0.6750 - val_loss: 0.7024 - val_accuracy: 0.5000
Epoch 38/100
2/2 [==============================] - 7s 2s/step - loss: 0.6088 - accuracy: 0.6250 - val_loss: 0.6915 - val_accuracy: 0.6000
Epoch 39/100
2/2 [==============================] - 7s 5s/step - loss: 0.6154 - accuracy: 0.5500 - val_loss: 0.6928 - val_accuracy: 0.5000
Epoch 40/100
2/2 [==============================] - 7s 2s/step - loss: 0.6364 - accuracy: 0.6250 - val_loss: 0.6809 - val_accuracy: 0.5500
Epoch 41/100
2/2 [==============================] - 7s 5s/step - loss: 0.6174 - accuracy: 0.6250 - val_loss: 0.6890 - val_accuracy: 0.6500
Epoch 42/100
2/2 [==============================] - 7s 5s/step - loss: 0.5969 - accuracy: 0.7000 - val_loss: 0.6865 - val_accuracy: 0.4500
Epoch 43/100
2/2 [==============================] - 7s 2s/step - loss: 0.5701 - accuracy: 0.7250 - val_loss: 0.7476 - val_accuracy: 0.5000
Epoch 44/100
2/2 [==============================] - 7s 2s/step - loss: 0.6045 - accuracy: 0.5250 - val_loss: 0.6967 - val_accuracy: 0.5500
Epoch 45/100
2/2 [==============================] - 7s 2s/step - loss: 0.5873 - accuracy: 0.8000 - val_loss: 0.6930 - val_accuracy: 0.4500
Epoch 46/100
2/2 [==============================] - 7s 5s/step - loss: 0.5892 - accuracy: 0.7500 - val_loss: 0.6953 - val_accuracy: 0.5000
Epoch 47/100
2/2 [==============================] - 7s 5s/step - loss: 0.5276 - accuracy: 0.7750 - val_loss: 0.7081 - val_accuracy: 0.5000
Epoch 48/100
2/2 [==============================] - 7s 2s/step - loss: 0.5357 - accuracy: 0.7000 - val_loss: 0.6965 - val_accuracy: 0.6500
Epoch 49/100
2/2 [==============================] - 7s 5s/step - loss: 0.6455 - accuracy: 0.5500 - val_loss: 0.7012 - val_accuracy: 0.5500
Epoch 50/100
2/2 [==============================] - 6s 5s/step - loss: 0.5724 - accuracy: 0.7500 - val_loss: 0.6966 - val_accuracy: 0.5500
Epoch 51/100
2/2 [==============================] - 7s 2s/step - loss: 0.5566 - accuracy: 0.7250 - val_loss: 0.6994 - val_accuracy: 0.5500
Epoch 52/100
2/2 [==============================] - 7s 5s/step - loss: 0.5667 - accuracy: 0.6750 - val_loss: 0.7001 - val_accuracy: 0.5500
Epoch 53/100
2/2 [==============================] - 7s 5s/step - loss: 0.6067 - accuracy: 0.7750 - val_loss: 0.7536 - val_accuracy: 0.6000
Epoch 54/100
2/2 [==============================] - 7s 5s/step - loss: 0.5202 - accuracy: 0.7000 - val_loss: 0.7057 - val_accuracy: 0.5500
Epoch 55/100
2/2 [==============================] - 7s 2s/step - loss: 0.5433 - accuracy: 0.7500 - val_loss: 0.6916 - val_accuracy: 0.5500
Epoch 56/100
2/2 [==============================] - 7s 5s/step - loss: 0.5783 - accuracy: 0.6750 - val_loss: 0.7092 - val_accuracy: 0.5000
Epoch 57/100
2/2 [==============================] - 7s 2s/step - loss: 0.5045 - accuracy: 0.8250 - val_loss: 0.8290 - val_accuracy: 0.5000
Epoch 58/100
2/2 [==============================] - 7s 5s/step - loss: 0.6219 - accuracy: 0.6250 - val_loss: 0.6779 - val_accuracy: 0.5500
Epoch 59/100
2/2 [==============================] - 7s 2s/step - loss: 0.5834 - accuracy: 0.7250 - val_loss: 0.6652 - val_accuracy: 0.6000
Epoch 60/100
2/2 [==============================] - 7s 2s/step - loss: 0.5509 - accuracy: 0.8250 - val_loss: 0.6775 - val_accuracy: 0.7000
Epoch 61/100
2/2 [==============================] - 7s 2s/step - loss: 0.5065 - accuracy: 0.7500 - val_loss: 0.7205 - val_accuracy: 0.6000
Epoch 62/100
2/2 [==============================] - 7s 2s/step - loss: 0.5731 - accuracy: 0.7250 - val_loss: 0.7746 - val_accuracy: 0.6000
Epoch 63/100
2/2 [==============================] - 7s 5s/step - loss: 0.5573 - accuracy: 0.6750 - val_loss: 0.7147 - val_accuracy: 0.5000
Epoch 64/100
2/2 [==============================] - 7s 5s/step - loss: 0.4964 - accuracy: 0.7750 - val_loss: 0.7415 - val_accuracy: 0.6000
Epoch 65/100
2/2 [==============================] - 7s 5s/step - loss: 0.4414 - accuracy: 0.7500 - val_loss: 0.7413 - val_accuracy: 0.4500
Epoch 66/100
2/2 [==============================] - 7s 2s/step - loss: 0.4226 - accuracy: 0.7750 - val_loss: 0.7846 - val_accuracy: 0.6500
Epoch 67/100
2/2 [==============================] - 7s 5s/step - loss: 0.4878 - accuracy: 0.7250 - val_loss: 0.7701 - val_accuracy: 0.6000
Epoch 68/100
2/2 [==============================] - 7s 2s/step - loss: 0.4179 - accuracy: 0.8000 - val_loss: 0.7392 - val_accuracy: 0.5000
Epoch 69/100
2/2 [==============================] - 7s 2s/step - loss: 0.6084 - accuracy: 0.6500 - val_loss: 0.7178 - val_accuracy: 0.5000
Epoch 70/100
2/2 [==============================] - 7s 5s/step - loss: 0.5126 - accuracy: 0.8250 - val_loss: 0.7503 - val_accuracy: 0.5500
Epoch 71/100
2/2 [==============================] - 7s 5s/step - loss: 0.4944 - accuracy: 0.7250 - val_loss: 0.8295 - val_accuracy: 0.6500
Epoch 72/100
2/2 [==============================] - 7s 5s/step - loss: 0.4426 - accuracy: 0.7000 - val_loss: 0.7419 - val_accuracy: 0.5500
Epoch 73/100
2/2 [==============================] - 7s 2s/step - loss: 0.4042 - accuracy: 0.8250 - val_loss: 0.7639 - val_accuracy: 0.6000
Epoch 74/100
2/2 [==============================] - 7s 2s/step - loss: 0.3894 - accuracy: 0.8500 - val_loss: 0.8061 - val_accuracy: 0.5000
Epoch 75/100
2/2 [==============================] - 7s 5s/step - loss: 0.4597 - accuracy: 0.7250 - val_loss: 0.8236 - val_accuracy: 0.6000
Epoch 76/100
2/2 [==============================] - 7s 2s/step - loss: 0.3831 - accuracy: 0.7500 - val_loss: 0.7381 - val_accuracy: 0.5500
Epoch 77/100
2/2 [==============================] - 7s 5s/step - loss: 0.3467 - accuracy: 0.8500 - val_loss: 0.7566 - val_accuracy: 0.5500
Epoch 78/100
2/2 [==============================] - 7s 2s/step - loss: 0.4976 - accuracy: 0.6500 - val_loss: 0.7355 - val_accuracy: 0.5500
Epoch 79/100
2/2 [==============================] - 7s 5s/step - loss: 0.4212 - accuracy: 0.8500 - val_loss: 0.6700 - val_accuracy: 0.5500
Epoch 80/100
2/2 [==============================] - 7s 2s/step - loss: 0.4957 - accuracy: 0.8250 - val_loss: 0.7851 - val_accuracy: 0.6000
Epoch 81/100
2/2 [==============================] - 7s 5s/step - loss: 0.7599 - accuracy: 0.6500 - val_loss: 0.6809 - val_accuracy: 0.5500
Epoch 82/100
2/2 [==============================] - 7s 2s/step - loss: 0.5788 - accuracy: 0.6750 - val_loss: 0.6885 - val_accuracy: 0.6000
Epoch 83/100
2/2 [==============================] - 7s 2s/step - loss: 0.5826 - accuracy: 0.6250 - val_loss: 0.6778 - val_accuracy: 0.5500
Epoch 84/100
2/2 [==============================] - 7s 2s/step - loss: 0.4822 - accuracy: 0.7750 - val_loss: 0.6835 - val_accuracy: 0.6000
Epoch 85/100
2/2 [==============================] - 7s 2s/step - loss: 0.5559 - accuracy: 0.6500 - val_loss: 0.6994 - val_accuracy: 0.5500
Epoch 86/100
2/2 [==============================] - 7s 2s/step - loss: 0.4519 - accuracy: 0.7750 - val_loss: 0.7011 - val_accuracy: 0.5500
Epoch 87/100
2/2 [==============================] - 7s 5s/step - loss: 0.7064 - accuracy: 0.5750 - val_loss: 0.7343 - val_accuracy: 0.5000
Epoch 88/100
2/2 [==============================] - 7s 2s/step - loss: 0.5454 - accuracy: 0.7250 - val_loss: 0.7098 - val_accuracy: 0.5500
Epoch 89/100
2/2 [==============================] - 7s 2s/step - loss: 0.4760 - accuracy: 0.8000 - val_loss: 0.7282 - val_accuracy: 0.6500
Epoch 90/100
2/2 [==============================] - 7s 5s/step - loss: 0.4169 - accuracy: 0.8750 - val_loss: 0.7698 - val_accuracy: 0.5000
Epoch 91/100
2/2 [==============================] - 7s 5s/step - loss: 0.6617 - accuracy: 0.6250 - val_loss: 0.6846 - val_accuracy: 0.6500
Epoch 92/100
2/2 [==============================] - 7s 5s/step - loss: 0.5094 - accuracy: 0.8500 - val_loss: 0.6946 - val_accuracy: 0.6000
Epoch 93/100
2/2 [==============================] - 7s 5s/step - loss: 0.4436 - accuracy: 0.8500 - val_loss: 0.7174 - val_accuracy: 0.5500
Epoch 94/100
2/2 [==============================] - 7s 5s/step - loss: 0.3960 - accuracy: 0.8750 - val_loss: 0.7154 - val_accuracy: 0.5500
Epoch 95/100
2/2 [==============================] - 7s 2s/step - loss: 0.3796 - accuracy: 0.8250 - val_loss: 0.7453 - val_accuracy: 0.5000
Epoch 96/100
2/2 [==============================] - 7s 2s/step - loss: 0.4107 - accuracy: 0.8750 - val_loss: 0.7345 - val_accuracy: 0.6500
Epoch 97/100
2/2 [==============================] - 7s 5s/step - loss: 0.4424 - accuracy: 0.8500 - val_loss: 0.6866 - val_accuracy: 0.5500
Epoch 98/100
2/2 [==============================] - 7s 5s/step - loss: 0.3612 - accuracy: 0.9500 - val_loss: 0.6958 - val_accuracy: 0.5500
Epoch 99/100
2/2 [==============================] - 7s 2s/step - loss: 0.3259 - accuracy: 0.9000 - val_loss: 0.7405 - val_accuracy: 0.6500
Epoch 100/100
2/2 [==============================] - 7s 2s/step - loss: 0.4126 - accuracy: 0.8500 - val_loss: 0.7135 - val_accuracy: 0.6000
```

Out[9]: <keras.callbacks.History at 0x7fc428bf38e0>

In [10]:
```python
# comparing the training data and testing data reults through a line chart
from matplotlib import pyplot as plt

plt.plot( model.history.history['accuracy'])
plt.plot( model.history.history['val_accuracy'])
plt.title('ACCURACY')
plt.ylabel('Accuracy')
plt.xlabel('Epochs')
plt.legend(['train', 'test'], loc = 'upper right')
plt.show()
```



In [ ]: