Stopped at 0.8

x = layers.Dense(1024, activation='selu')(x)

x = layers.Dropout(0.1)(x)

x = layers.Dense(512, activation='selu')(x)

x = layers.Dropout(0.1)(x)

# Add a final sigmoid layer for classification

x = layers.Dense(6, activation='softmax')(x)

model.compile(optimizer = RMSprop(lr=0.0001),

              loss = 'categorical\_crossentropy',

              metrics = ['acc'])

train\_datagen = ImageDataGenerator(rescale = 1./255,

                                  rotation\_range=10,

                                  width\_shift\_range=0.3,

                                  height\_shift\_range=0.3,

                                  # featurewise\_std\_normalization=True,

                                  # samplewise\_std\_normalization=True,

                                  brightness\_range=(0.2,0.5),

                                  # shear\_range=0.2,

                                  # zoom\_range=0.2,

                                  horizontal\_flip=True,

                                  fill\_mode='nearest')

 batch\_size = 20,

history = model.fit(

            train\_generator,

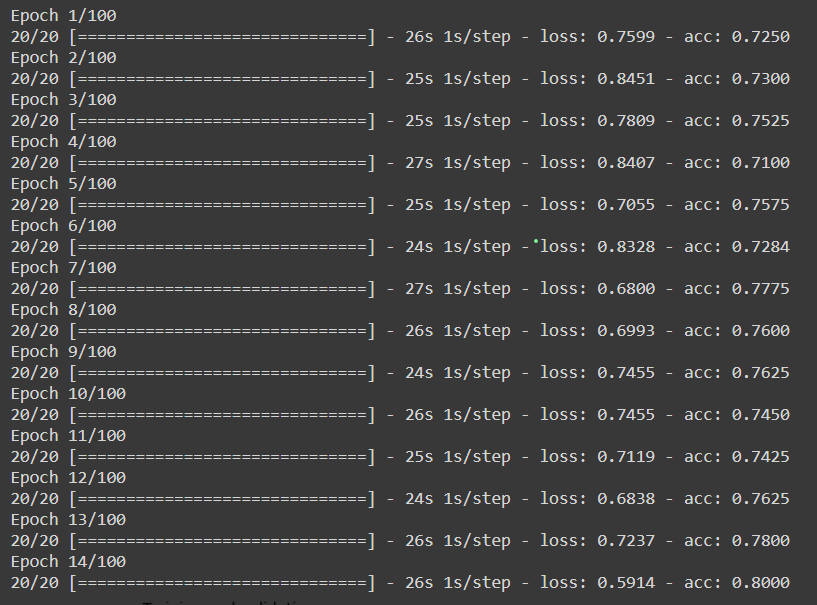
            validation\_data = validation\_generator,

            steps\_per\_epoch = 20,

            epochs = 100,

            validation\_steps = 20,

            verbose = 1,callbacks=[callbacks])



Saved as activity4.h5

Code:

import os

from tensorflow.keras import layers

from tensorflow.keras import Model

!wget --no-check-certificate \

    https://storage.googleapis.com/mledu-datasets/inception\_v3\_weights\_tf\_dim\_ordering\_tf\_kernels\_notop.h5 \

    -O /tmp/inception\_v3\_weights\_tf\_dim\_ordering\_tf\_kernels\_notop.h5

from tensorflow.keras.applications.inception\_v3 import InceptionV3

local\_weights\_file = '/tmp/inception\_v3\_weights\_tf\_dim\_ordering\_tf\_kernels\_notop.h5'

pre\_trained\_model = InceptionV3(input\_shape = (150, 150, 3),

                                include\_top = False,

                                weights = None)

pre\_trained\_model.load\_weights(local\_weights\_file)

for layer in pre\_trained\_model.layers:

  layer.trainable = False

# pre\_trained\_model.summary()

last\_layer = pre\_trained\_model.get\_layer('mixed7')

print('last layer output shape: ', last\_layer.output\_shape)

last\_output = last\_layer.output

from tensorflow.keras.optimizers import RMSprop

# Flatten the output layer to 1 dimension

x = layers.Flatten()(last\_output)

# Add a fully connected layer with 1,024 hidden units and ReLU activation

x = layers.Dense(1024, activation='selu')(x)

x = layers.Dropout(0.1)(x)

x = layers.Dense(512, activation='selu')(x)

x = layers.Dropout(0.1)(x)

# Add a final sigmoid layer for classification

x = layers.Dense(6, activation='softmax')(x)

model = Model( pre\_trained\_model.input, x)

model.compile(optimizer = RMSprop(lr=0.0001),

              loss = 'categorical\_crossentropy',

              metrics = ['acc'])

from tensorflow.keras.preprocessing.image import ImageDataGenerator

train\_dir = r"/content/drive/MyDrive/DatasetNew"

validation\_dir = r"/content/drive/MyDrive/Validation"

# Add our data-augmentation parameters to ImageDataGenerator

train\_datagen = ImageDataGenerator(rescale = 1./255,

                                  rotation\_range=10,

                                  width\_shift\_range=0.3,

                                  height\_shift\_range=0.3,

                                  # featurewise\_std\_normalization=True,

                                  # samplewise\_std\_normalization=True,

                                  brightness\_range=(0.2,0.5),

                                  # shear\_range=0.2,

                                  # zoom\_range=0.2,

                                  horizontal\_flip=True,

                                  fill\_mode='nearest')

# Note that the validation data should not be augmented!

# test\_datagen = ImageDataGenerator( rescale = 1.0/255. )

# Flow training images in batches of 20 using train\_datagen generator

train\_generator = train\_datagen.flow\_from\_directory(train\_dir,

                                                    batch\_size = 20,

                                                    # shuffle="True",

                                                    class\_mode = 'categorical',

                                                    target\_size = (150, 150))

# Flow validation images in batches of 20 using test\_datagen generator

validation\_generator =  train\_datagen.flow\_from\_directory(validation\_dir,

                                                          batch\_size  = 20,

                                                          shuffle="True",

                                                          class\_mode  = 'categorical',

                                                          target\_size = (150, 150))

import tensorflow as tf

from PIL import ImageFile

ImageFile.LOAD\_TRUNCATED\_IMAGES = True

class myCallback(tf.keras.callbacks.Callback):

      def on\_epoch\_end(self, epoch, logs={}):

          if (logs.get('acc')>0.82):

            self.model.stop\_training = True

callbacks=myCallback()

history = model.fit(

            train\_generator,

            validation\_data = validation\_generator,

            steps\_per\_epoch = 20,

            epochs = 100,

            validation\_steps = 20,

            verbose = 1,callbacks=[callbacks])

model.save("activity34.h5")

import matplotlib.pyplot as plt

acc = history.history['acc']

# val\_acc = history.history['val\_acc']

loss = history.history['loss']

# val\_loss = history.history['val\_loss']

epochs = range(len(acc))

model.save("activity4.h5")

plt.plot(epochs, acc, 'r', label='Training accuracy')

# plt.plot(epochs, val\_acc, 'b', label='Validation accuracy')

plt.title('Training and validation accuracy')

plt.legend(loc=0)

plt.figure()

plt.show()

Stopped at 0.82

import os

from tensorflow.keras import layers

from tensorflow.keras import Model

!wget --no-check-certificate \

    https://storage.googleapis.com/mledu-datasets/inception\_v3\_weights\_tf\_dim\_ordering\_tf\_kernels\_notop.h5 \

    -O /tmp/inception\_v3\_weights\_tf\_dim\_ordering\_tf\_kernels\_notop.h5

from tensorflow.keras.applications.inception\_v3 import InceptionV3

local\_weights\_file = '/tmp/inception\_v3\_weights\_tf\_dim\_ordering\_tf\_kernels\_notop.h5'

pre\_trained\_model = InceptionV3(input\_shape = (150, 150, 3),

                                include\_top = False,

                                weights = None)

pre\_trained\_model.load\_weights(local\_weights\_file)

for layer in pre\_trained\_model.layers:

  layer.trainable = False

# pre\_trained\_model.summary()

last\_layer = pre\_trained\_model.get\_layer('mixed7')

print('last layer output shape: ', last\_layer.output\_shape)

last\_output = last\_layer.output

from tensorflow.keras.optimizers import RMSprop

# Flatten the output layer to 1 dimension

x = layers.Flatten()(last\_output)

# Add a fully connected layer with 1,024 hidden units and ReLU activation

x = layers.Dense(1024, activation='selu')(x)

x = layers.Dropout(0.1)(x)

x = layers.Dense(512, activation='selu')(x)

x = layers.Dropout(0.1)(x)

# Add a final sigmoid layer for classification

x = layers.Dense(6, activation='softmax')(x)

model = Model( pre\_trained\_model.input, x)

model.compile(optimizer = RMSprop(lr=0.00001),

              loss = 'categorical\_crossentropy',

              metrics = ['acc'])

from tensorflow.keras.preprocessing.image import ImageDataGenerator

train\_dir = r"/content/drive/MyDrive/DatasetNew"

validation\_dir = r"/content/drive/MyDrive/Validation"

# Add our data-augmentation parameters to ImageDataGenerator

train\_datagen = ImageDataGenerator(rescale = 1./255,

                                  rotation\_range=10,

                                  width\_shift\_range=0.3,

                                  height\_shift\_range=0.3,

                                  # featurewise\_std\_normalization=True,

                                  # samplewise\_std\_normalization=True,

                                  brightness\_range=(0.2,0.5),

                                  # shear\_range=0.2,

                                  # zoom\_range=0.2,

                                  horizontal\_flip=True,

                                  fill\_mode='nearest')

# Note that the validation data should not be augmented!

# test\_datagen = ImageDataGenerator( rescale = 1.0/255. )

# Flow training images in batches of 20 using train\_datagen generator

train\_generator = train\_datagen.flow\_from\_directory(train\_dir,

                                                    batch\_size = 20,

                                                    # shuffle="True",

                                                    class\_mode = 'categorical',

                                                    target\_size = (150, 150))

# Flow validation images in batches of 20 using test\_datagen generator

validation\_generator =  train\_datagen.flow\_from\_directory(validation\_dir,

                                                          batch\_size  = 20,

                                                          shuffle="True",

                                                          class\_mode  = 'categorical',

                                                          target\_size = (150, 150))

import tensorflow as tf

from PIL import ImageFile

ImageFile.LOAD\_TRUNCATED\_IMAGES = True

class myCallback(tf.keras.callbacks.Callback):

      def on\_epoch\_end(self, epoch, logs={}):

          if (logs.get('acc')>0.82):

            self.model.stop\_training = True

callbacks=myCallback()

history = model.fit(

            train\_generator,

            validation\_data = validation\_generator,

            steps\_per\_epoch = 20,

            epochs = 1000,

            validation\_steps = 20,

            verbose = 1,callbacks=[callbacks])

model.save("activity34.h5")

import matplotlib.pyplot as plt

acc = history.history['acc']

# val\_acc = history.history['val\_acc']

loss = history.history['loss']

# val\_loss = history.history['val\_loss']

epochs = range(len(acc))

model.save("activity4.h5")

plt.plot(epochs, acc, 'r', label='Training accuracy')

# plt.plot(epochs, val\_acc, 'b', label='Validation accuracy')

plt.title('Training Accuracy')

plt.legend(loc=0)

plt.figure()

plt.show()

model.save("activity4.h5")

plt.plot(epochs, loss, 'r', label='Training Loss')

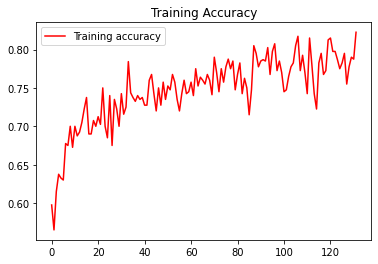
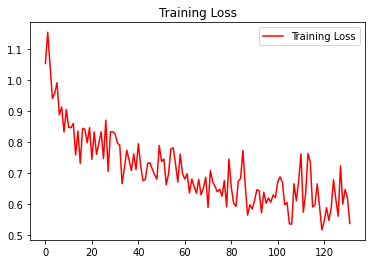
# plt.plot(epochs, val\_acc, 'b', label='Validation accuracy')

plt.title('Training Loss')

plt.legend(loc=0)

plt.figure()

plt.show()

Epoch 1/1000

20/20 [==============================] - 6s 293ms/step - loss: 1.0528 - acc: 0.5975

Epoch 2/1000

20/20 [==============================] - 6s 304ms/step - loss: 1.1537 - acc: 0.5650

Epoch 3/1000

20/20 [==============================] - 8s 390ms/step - loss: 1.0445 - acc: 0.6150

Epoch 4/1000

20/20 [==============================] - 4s 196ms/step - loss: 0.9397 - acc: 0.6375

Epoch 5/1000

20/20 [==============================] - 6s 282ms/step - loss: 0.9567 - acc: 0.6325

Epoch 6/1000

20/20 [==============================] - 5s 231ms/step - loss: 0.9908 - acc: 0.6300

Epoch 7/1000

20/20 [==============================] - 4s 222ms/step - loss: 0.8873 - acc: 0.6775

Epoch 8/1000

20/20 [==============================] - 7s 352ms/step - loss: 0.9123 - acc: 0.6750

Epoch 9/1000

20/20 [==============================] - 5s 255ms/step - loss: 0.8317 - acc: 0.7000

Epoch 10/1000

20/20 [==============================] - 7s 358ms/step - loss: 0.9042 - acc: 0.6726

Epoch 11/1000

20/20 [==============================] - 5s 271ms/step - loss: 0.8471 - acc: 0.7000

Epoch 12/1000

20/20 [==============================] - 7s 369ms/step - loss: 0.8452 - acc: 0.6875

Epoch 13/1000

20/20 [==============================] - 5s 246ms/step - loss: 0.8595 - acc: 0.6925

Epoch 14/1000

20/20 [==============================] - 5s 273ms/step - loss: 0.7574 - acc: 0.7050

Epoch 15/1000

20/20 [==============================] - 7s 336ms/step - loss: 0.8350 - acc: 0.7225

Epoch 16/1000

20/20 [==============================] - 7s 339ms/step - loss: 0.7302 - acc: 0.7375

Epoch 17/1000

20/20 [==============================] - 5s 273ms/step - loss: 0.8436 - acc: 0.6900

Epoch 18/1000

20/20 [==============================] - 6s 292ms/step - loss: 0.8402 - acc: 0.6900

Epoch 19/1000

20/20 [==============================] - 6s 317ms/step - loss: 0.7968 - acc: 0.7075

Epoch 20/1000

20/20 [==============================] - 6s 316ms/step - loss: 0.8455 - acc: 0.7000

Epoch 21/1000

20/20 [==============================] - 6s 323ms/step - loss: 0.7436 - acc: 0.7125

Epoch 22/1000

20/20 [==============================] - 6s 293ms/step - loss: 0.8318 - acc: 0.7025

Epoch 23/1000

20/20 [==============================] - 5s 266ms/step - loss: 0.7593 - acc: 0.7500

Epoch 24/1000

20/20 [==============================] - 6s 286ms/step - loss: 0.7958 - acc: 0.7000

Epoch 25/1000

20/20 [==============================] - 5s 252ms/step - loss: 0.8324 - acc: 0.6850

Epoch 26/1000

20/20 [==============================] - 8s 416ms/step - loss: 0.7458 - acc: 0.7400

Epoch 27/1000

20/20 [==============================] - 6s 305ms/step - loss: 0.8702 - acc: 0.6750

Epoch 28/1000

20/20 [==============================] - 8s 382ms/step - loss: 0.7046 - acc: 0.7350

Epoch 29/1000

20/20 [==============================] - 7s 331ms/step - loss: 0.8324 - acc: 0.7225

Epoch 30/1000

20/20 [==============================] - 9s 449ms/step - loss: 0.8324 - acc: 0.7000

Epoch 31/1000

20/20 [==============================] - 6s 315ms/step - loss: 0.8255 - acc: 0.7425

Epoch 32/1000

20/20 [==============================] - 7s 358ms/step - loss: 0.7959 - acc: 0.7157

Epoch 33/1000

20/20 [==============================] - 7s 371ms/step - loss: 0.7888 - acc: 0.7250

Epoch 34/1000

20/20 [==============================] - 7s 347ms/step - loss: 0.6648 - acc: 0.7843

Epoch 35/1000

20/20 [==============================] - 5s 256ms/step - loss: 0.7151 - acc: 0.7437

Epoch 36/1000

20/20 [==============================] - 6s 292ms/step - loss: 0.7728 - acc: 0.7375

Epoch 37/1000

20/20 [==============================] - 7s 342ms/step - loss: 0.7413 - acc: 0.7325

Epoch 38/1000

20/20 [==============================] - 6s 282ms/step - loss: 0.7070 - acc: 0.7400

Epoch 39/1000

20/20 [==============================] - 6s 316ms/step - loss: 0.7604 - acc: 0.7350

Epoch 40/1000

20/20 [==============================] - 7s 342ms/step - loss: 0.7102 - acc: 0.7375

Epoch 41/1000

20/20 [==============================] - 6s 281ms/step - loss: 0.7938 - acc: 0.7275

Epoch 42/1000

20/20 [==============================] - 6s 286ms/step - loss: 0.7221 - acc: 0.7275

Epoch 43/1000

20/20 [==============================] - 5s 239ms/step - loss: 0.6741 - acc: 0.7600

Epoch 44/1000

20/20 [==============================] - 4s 190ms/step - loss: 0.6791 - acc: 0.7675

Epoch 45/1000

20/20 [==============================] - 6s 287ms/step - loss: 0.7302 - acc: 0.7425

Epoch 46/1000

20/20 [==============================] - 8s 423ms/step - loss: 0.7321 - acc: 0.7200

Epoch 47/1000

20/20 [==============================] - 8s 414ms/step - loss: 0.7120 - acc: 0.7500

Epoch 48/1000

20/20 [==============================] - 6s 302ms/step - loss: 0.6935 - acc: 0.7275

Epoch 49/1000

20/20 [==============================] - 7s 330ms/step - loss: 0.6792 - acc: 0.7575

Epoch 50/1000

20/20 [==============================] - 7s 345ms/step - loss: 0.7882 - acc: 0.7350

Epoch 51/1000

20/20 [==============================] - 5s 260ms/step - loss: 0.7364 - acc: 0.7525

Epoch 52/1000

20/20 [==============================] - 8s 381ms/step - loss: 0.7440 - acc: 0.7475

Epoch 53/1000

20/20 [==============================] - 6s 285ms/step - loss: 0.6615 - acc: 0.7675

Epoch 54/1000

20/20 [==============================] - 7s 359ms/step - loss: 0.6970 - acc: 0.7575

Epoch 55/1000

20/20 [==============================] - 7s 326ms/step - loss: 0.7772 - acc: 0.7350

Epoch 56/1000

20/20 [==============================] - 7s 341ms/step - loss: 0.7805 - acc: 0.7200

Epoch 57/1000

20/20 [==============================] - 6s 305ms/step - loss: 0.7288 - acc: 0.7425

Epoch 58/1000

20/20 [==============================] - 8s 424ms/step - loss: 0.6698 - acc: 0.7600

Epoch 59/1000

20/20 [==============================] - 7s 331ms/step - loss: 0.7602 - acc: 0.7425

Epoch 60/1000

20/20 [==============================] - 7s 342ms/step - loss: 0.6973 - acc: 0.7450

Epoch 61/1000

20/20 [==============================] - 6s 300ms/step - loss: 0.6796 - acc: 0.7575

Epoch 62/1000

20/20 [==============================] - 7s 332ms/step - loss: 0.6967 - acc: 0.7400

Epoch 63/1000

20/20 [==============================] - 7s 339ms/step - loss: 0.6344 - acc: 0.7750

Epoch 64/1000

20/20 [==============================] - 6s 285ms/step - loss: 0.6798 - acc: 0.7525

Epoch 65/1000

20/20 [==============================] - 5s 240ms/step - loss: 0.6553 - acc: 0.7640

Epoch 66/1000

20/20 [==============================] - 7s 358ms/step - loss: 0.6342 - acc: 0.7600

Epoch 67/1000

20/20 [==============================] - 7s 329ms/step - loss: 0.6784 - acc: 0.7550

Epoch 68/1000

20/20 [==============================] - 5s 257ms/step - loss: 0.6291 - acc: 0.7675

Epoch 69/1000

20/20 [==============================] - 7s 326ms/step - loss: 0.6500 - acc: 0.7600

Epoch 70/1000

20/20 [==============================] - 6s 319ms/step - loss: 0.6857 - acc: 0.7411

Epoch 71/1000

20/20 [==============================] - 8s 391ms/step - loss: 0.5883 - acc: 0.7900

Epoch 72/1000

20/20 [==============================] - 7s 364ms/step - loss: 0.7067 - acc: 0.7700

Epoch 73/1000

20/20 [==============================] - 7s 360ms/step - loss: 0.6713 - acc: 0.7450

Epoch 74/1000

20/20 [==============================] - 6s 293ms/step - loss: 0.6559 - acc: 0.7750

Epoch 75/1000

20/20 [==============================] - 8s 400ms/step - loss: 0.6382 - acc: 0.7575

Epoch 76/1000

20/20 [==============================] - 7s 325ms/step - loss: 0.6470 - acc: 0.7775

Epoch 77/1000

20/20 [==============================] - 5s 250ms/step - loss: 0.6243 - acc: 0.7875

Epoch 78/1000

20/20 [==============================] - 7s 363ms/step - loss: 0.6752 - acc: 0.7750

Epoch 79/1000

20/20 [==============================] - 5s 247ms/step - loss: 0.5897 - acc: 0.7850

Epoch 80/1000

20/20 [==============================] - 4s 212ms/step - loss: 0.7444 - acc: 0.7475

Epoch 81/1000

20/20 [==============================] - 6s 323ms/step - loss: 0.6530 - acc: 0.7675

Epoch 82/1000

20/20 [==============================] - 7s 350ms/step - loss: 0.6022 - acc: 0.7825

Epoch 83/1000

20/20 [==============================] - 6s 302ms/step - loss: 0.5915 - acc: 0.7425

Epoch 84/1000

20/20 [==============================] - 6s 313ms/step - loss: 0.6724 - acc: 0.7625

Epoch 85/1000

20/20 [==============================] - 7s 341ms/step - loss: 0.6817 - acc: 0.7500

Epoch 86/1000

20/20 [==============================] - 8s 396ms/step - loss: 0.7718 - acc: 0.7150

Epoch 87/1000

20/20 [==============================] - 7s 374ms/step - loss: 0.6610 - acc: 0.7475

Epoch 88/1000

20/20 [==============================] - 8s 376ms/step - loss: 0.5634 - acc: 0.8050

Epoch 89/1000

20/20 [==============================] - 4s 217ms/step - loss: 0.5965 - acc: 0.7950

Epoch 90/1000

20/20 [==============================] - 5s 267ms/step - loss: 0.5832 - acc: 0.7775

Epoch 91/1000

20/20 [==============================] - 7s 337ms/step - loss: 0.6090 - acc: 0.7850

Epoch 92/1000

20/20 [==============================] - 6s 316ms/step - loss: 0.6453 - acc: 0.7868

Epoch 93/1000

20/20 [==============================] - 6s 325ms/step - loss: 0.6419 - acc: 0.7850

Epoch 94/1000

20/20 [==============================] - 6s 288ms/step - loss: 0.5707 - acc: 0.8025

Epoch 95/1000

20/20 [==============================] - 6s 323ms/step - loss: 0.6369 - acc: 0.7675

Epoch 96/1000

20/20 [==============================] - 7s 330ms/step - loss: 0.6015 - acc: 0.7975

Epoch 97/1000

20/20 [==============================] - 8s 376ms/step - loss: 0.6187 - acc: 0.8075

Epoch 98/1000

20/20 [==============================] - 6s 279ms/step - loss: 0.6048 - acc: 0.7725

Epoch 99/1000

20/20 [==============================] - 9s 429ms/step - loss: 0.6285 - acc: 0.7850

Epoch 100/1000

20/20 [==============================] - 5s 254ms/step - loss: 0.6198 - acc: 0.7700

Epoch 101/1000

20/20 [==============================] - 7s 340ms/step - loss: 0.6712 - acc: 0.7450

Epoch 102/1000

20/20 [==============================] - 6s 280ms/step - loss: 0.6869 - acc: 0.7475

Epoch 103/1000

20/20 [==============================] - 6s 302ms/step - loss: 0.6699 - acc: 0.7650

Epoch 104/1000

20/20 [==============================] - 7s 345ms/step - loss: 0.5966 - acc: 0.7775

Epoch 105/1000

20/20 [==============================] - 5s 274ms/step - loss: 0.6042 - acc: 0.7825

Epoch 106/1000

20/20 [==============================] - 6s 299ms/step - loss: 0.5354 - acc: 0.8050

Epoch 107/1000

20/20 [==============================] - 7s 358ms/step - loss: 0.5340 - acc: 0.8173

Epoch 108/1000

20/20 [==============================] - 6s 289ms/step - loss: 0.6644 - acc: 0.7725

Epoch 109/1000

20/20 [==============================] - 7s 348ms/step - loss: 0.6090 - acc: 0.7925

Epoch 110/1000

20/20 [==============================] - 8s 376ms/step - loss: 0.6747 - acc: 0.7700

Epoch 111/1000

20/20 [==============================] - 6s 311ms/step - loss: 0.7611 - acc: 0.7425

Epoch 112/1000

20/20 [==============================] - 6s 303ms/step - loss: 0.5724 - acc: 0.8150

Epoch 113/1000

20/20 [==============================] - 8s 400ms/step - loss: 0.6328 - acc: 0.7800

Epoch 114/1000

20/20 [==============================] - 5s 235ms/step - loss: 0.7620 - acc: 0.7425

Epoch 115/1000

20/20 [==============================] - 5s 264ms/step - loss: 0.7345 - acc: 0.7225

Epoch 116/1000

20/20 [==============================] - 8s 378ms/step - loss: 0.5895 - acc: 0.7825

Epoch 117/1000

20/20 [==============================] - 6s 288ms/step - loss: 0.5960 - acc: 0.7950

Epoch 118/1000

20/20 [==============================] - 5s 265ms/step - loss: 0.6642 - acc: 0.7675

Epoch 119/1000

20/20 [==============================] - 7s 359ms/step - loss: 0.5898 - acc: 0.7725

Epoch 120/1000

20/20 [==============================] - 5s 243ms/step - loss: 0.5155 - acc: 0.8125

Epoch 121/1000

20/20 [==============================] - 8s 424ms/step - loss: 0.5441 - acc: 0.8150

Epoch 122/1000

20/20 [==============================] - 8s 406ms/step - loss: 0.5873 - acc: 0.7975

Epoch 123/1000

20/20 [==============================] - 4s 221ms/step - loss: 0.5462 - acc: 0.7975

Epoch 124/1000

20/20 [==============================] - 7s 375ms/step - loss: 0.5869 - acc: 0.7868

Epoch 125/1000

20/20 [==============================] - 6s 285ms/step - loss: 0.6775 - acc: 0.7750

Epoch 126/1000

20/20 [==============================] - 7s 370ms/step - loss: 0.6143 - acc: 0.7825

Epoch 127/1000

20/20 [==============================] - 7s 357ms/step - loss: 0.5597 - acc: 0.7950

Epoch 128/1000

20/20 [==============================] - 5s 268ms/step - loss: 0.7223 - acc: 0.7550

Epoch 129/1000

20/20 [==============================] - 7s 340ms/step - loss: 0.5987 - acc: 0.7775

Epoch 130/1000

20/20 [==============================] - 8s 397ms/step - loss: 0.6457 - acc: 0.7900

Epoch 131/1000

20/20 [==============================] - 5s 273ms/step - loss: 0.6190 - acc: 0.7875

Epoch 132/1000

20/20 [==============================] - 6s 291ms/step - loss: 0.5363 - acc: 0.8225