

# lab2

January 21, 2025

## 1 Transfer learning - Cats vs. Dogs

## 2 Nuova sezione

Unzip data

```
[8]: # !unzip train.zip -d train/  
     # !unzip test.zip -d test/
```

Code to initialize Tensorflow 2.0 in Colab

```
[9]: from __future__ import absolute_import, division, print_function,  
     ↪ unicode_literals  
     # %tensorflow_version 2.x  
     import tensorflow as tf  
     %load_ext tensorboard  
     import datetime  
     import numpy as np  
     import matplotlib.pyplot as plt
```

[TODO] Create a data loader function that returns a tuple with a `tf.float32` tensor for the image and a label. Images must be resized to 128x128. N.B.: filenames are formatted as `class.number.jpg`

```
[10]: def load_and_preprocess_image(filename): # load images  
       image = tf.io.read_file(filename) # read the raw data from the file as a  
       ↪ string  
       image = tf.image.decode_jpeg(image, channels=3) # decode the jpeg image to  
       ↪ a tensor  
  
       image = tf.image.resize(image, [128, 128]) # resize the image to 128x128  
       image = tf.cast(image, tf.float32) / 255.0 # transform the image to a tf.  
       ↪ float32 type and normalize it to [0, 1]  
       return image  
  
def parse_filename(filename): # load labels  
    label = tf.strings.split(filename, sep='/') # split the filename by '/'  
    ↪ (label[0]: 'train', label[1]: 'class.number.jpg')
```

```

    label = tf.strings.split(label[-1], sep='.')    # split the last element of
    ↪ the filename by '.' (label[0]: 'class', label[1]: 'number', label[2]: 'jpg')
    label = tf.strings.to_number(label[0], out_type=tf.int32)    # convert the
    ↪ label[0] to a tf.int32 type
    return label

```

[TODO] Create a tf.Dataset, map the loader function and prepare a batch object for training

```

[11]: trainDataSet = tf.data.Dataset.list_files('train/*')    # create a dataset from
    ↪ the filenames (filename: train/class.number.jpg)
testDataSet = tf.data.Dataset.list_files('test/*')    # create a dataset from the
    ↪ filenames (filename: test/class.number.jpg)
trainDataSet = trainDataSet.map(lambda x: (load_and_preprocess_image(x),
    ↪ parse_filename(x)))    # for each filename, load the image and the label
testDataSet = testDataSet.map(lambda x: (load_and_preprocess_image(x),
    ↪ parse_filename(x)))    # for each filename, load the image and the label

```

Prepare Keras callback for Tensorboard

```

[12]: logdir = "logs/scalars/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
    %tensorboard --logdir logs
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=logdir,
    ↪ update_freq='batch')

```

<IPython.core.display.HTML object>

[TODO] Import the MobileNetV2 model trained on ImageNet without the final layer

```

[13]: # because build a deep learning model can be too complex, we can consider to
    ↪ use a pre-trained model for performing feature extraction, and then add the
    ↪ final layer to perform the
    # classification as we want

    # import the MobileNetV2 model, input_shape is the shape of the images. They
    ↪ have 3 channels cause they are RGB images
    # include_top=False means that we exclude the last fully connected layer of the
    ↪ model
    # weights='imagenet' means that we initialize the model with pre-trained
    ↪ weights on ImageNet
    base_model = tf.keras.applications.MobileNetV2(input_shape=(128, 128, 3),
    ↪ include_top=False, weights='imagenet')    # load the MobileNetV2 model
    base_model.trainable = False

```

[TODO] Add a final classification layer for 2 classes and create the final Keras model

```

[14]: # We know that the final layer, the one that performs the classification
    ↪ through the softmax function, need that the input is a vector. This is why,
    ↪ as seen in the previous laboratory, we

```

```

# have performed a flattening operation on the input. The problem of flattening
↳ is that than the classification layer depends on the size of the input
↳ (HxW), so if the model receive an image
# with different shapes then the classification will not work. This is why in
↳ this case we use the GlobalAveragePooling2D layer, cause we want that our
↳ model could work with images of
# different shapes. What GlobalAvaragePooling2D does is to evaluate the mean
↳ value of each image feature map and build a F dimensional vector where each
↳ value rappresent the mean value of
# the image on that channel.
x = base_model.output # get the output of the model, on it we will add the
↳ final layers
x = tf.keras.layers.GlobalAveragePooling2D()(x) # evaluate the mean value of
↳ each image feature map. So we receive an input of shape (HxWxF) and we
↳ return an output of shape (F)
y = tf.keras.layers.Dense(2, activation='softmax')(x) # build a pdf with the
↳ two possible classes, dog and cat
model = tf.keras.Model(inputs=base_model.input, outputs=y) # create the
↳ final model

```

[TODO] Compile the Keras model: specify the optimization algorithm, the loss function and the test metric

```

[15]: lr = 0.01 # learning rate
model.compile(optimizer = tf.keras.optimizers.Adam(lr), loss =
↳ 'sparse_categorical_crossentropy', metrics=['accuracy'])

```

[TODO] Train the Keras model

```

[16]: model.fit(trainDataSet.batch(32), epochs=5, callbacks=[tensorboard_callback])
↳ # train the model

```

```

Epoch 1/5
63/63 18s 179ms/step -
accuracy: 0.8613 - loss: 0.4874
Epoch 2/5
63/63 10s 165ms/step -
accuracy: 0.9669 - loss: 0.1241
Epoch 3/5
63/63 11s 170ms/step -
accuracy: 0.9796 - loss: 0.0581
Epoch 4/5
63/63 11s 177ms/step -
accuracy: 0.9921 - loss: 0.0241
Epoch 5/5
63/63 11s 172ms/step -
accuracy: 0.9956 - loss: 0.0192

```

```
[16]: <keras.src.callbacks.history.History at 0x7784246f9f00>
```

[TODO] Print model summary

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

Model: "functional"

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 128, 128, 3)	0	-
Conv1 (Conv2D)	(None, 64, 64, 32)	864	input_layer[0][0]
bn_Conv1 (BatchNormalizatio...	(None, 64, 64, 32)	128	Conv1[0][0]
Conv1_relu (ReLU)	(None, 64, 64, 32)	0	bn_Conv1[0][0]
expanded_conv_dept... (DepthwiseConv2D)	(None, 64, 64, 32)	288	Conv1_relu[0][0]
expanded_conv_dept... (BatchNormalizatio...	(None, 64, 64, 32)	128	expanded_conv_de...
expanded_conv_dept... (ReLU)	(None, 64, 64, 32)	0	expanded_conv_de...
expanded_conv_proj... (Conv2D)	(None, 64, 64, 16)	512	expanded_conv_de...
expanded_conv_proj... (BatchNormalizatio...	(None, 64, 64, 16)	64	expanded_conv_pr...
block_1_expand (Conv2D)	(None, 64, 64, 96)	1,536	expanded_conv_pr...
block_1_expand_BN (BatchNormalizatio...	(None, 64, 64, 96)	384	block_1_expand[0...
block_1_expand_relu (ReLU)	(None, 64, 64, 96)	0	block_1_expand_B...
block_1_pad	(None, 65, 65,	0	block_1_expand_r...

(ZeroPadding2D)	96)		
block_1_depthwise (DepthwiseConv2D)	(None, 32, 32, 96)	864	block_1_pad[0][0]
block_1_depthwise_... (BatchNormalizatio...	(None, 32, 32, 96)	384	block_1_depthwis...
block_1_depthwise_... (ReLU)	(None, 32, 32, 96)	0	block_1_depthwis...
block_1_project (Conv2D)	(None, 32, 32, 24)	2,304	block_1_depthwis...
block_1_project_BN (BatchNormalizatio...	(None, 32, 32, 24)	96	block_1_project[...
block_2_expand (Conv2D)	(None, 32, 32, 144)	3,456	block_1_project_...
block_2_expand_BN (BatchNormalizatio...	(None, 32, 32, 144)	576	block_2_expand[0...
block_2_expand_relu (ReLU)	(None, 32, 32, 144)	0	block_2_expand_B...
block_2_depthwise (DepthwiseConv2D)	(None, 32, 32, 144)	1,296	block_2_expand_r...
block_2_depthwise_... (BatchNormalizatio...	(None, 32, 32, 144)	576	block_2_depthwis...
block_2_depthwise_... (ReLU)	(None, 32, 32, 144)	0	block_2_depthwis...
block_2_project (Conv2D)	(None, 32, 32, 24)	3,456	block_2_depthwis...
block_2_project_BN (BatchNormalizatio...	(None, 32, 32, 24)	96	block_2_project[...
block_2_add (Add)	(None, 32, 32, 24)	0	block_1_project_... block_2_project_...
block_3_expand (Conv2D)	(None, 32, 32, 144)	3,456	block_2_add[0][0]
block_3_expand_BN	(None, 32, 32,	576	block_3_expand[0...

(BatchNormalizatio...	144)		
block_3_expand_relu (ReLU)	(None, 32, 32, 144)	0	block_3_expand_B...
block_3_pad (ZeroPadding2D)	(None, 33, 33, 144)	0	block_3_expand_r...
block_3_depthwise (DepthwiseConv2D)	(None, 16, 16, 144)	1,296	block_3_pad[0][0]
block_3_depthwise_... (BatchNormalizatio...	(None, 16, 16, 144)	576	block_3_depthwis...
block_3_depthwise_... (ReLU)	(None, 16, 16, 144)	0	block_3_depthwis...
block_3_project (Conv2D)	(None, 16, 16, 32)	4,608	block_3_depthwis...
block_3_project_BN (BatchNormalizatio...	(None, 16, 16, 32)	128	block_3_project[...
block_4_expand (Conv2D)	(None, 16, 16, 192)	6,144	block_3_project_...
block_4_expand_BN (BatchNormalizatio...	(None, 16, 16, 192)	768	block_4_expand[0...
block_4_expand_relu (ReLU)	(None, 16, 16, 192)	0	block_4_expand_B...
block_4_depthwise (DepthwiseConv2D)	(None, 16, 16, 192)	1,728	block_4_expand_r...
block_4_depthwise_... (BatchNormalizatio...	(None, 16, 16, 192)	768	block_4_depthwis...
block_4_depthwise_... (ReLU)	(None, 16, 16, 192)	0	block_4_depthwis...
block_4_project (Conv2D)	(None, 16, 16, 32)	6,144	block_4_depthwis...
block_4_project_BN (BatchNormalizatio...	(None, 16, 16, 32)	128	block_4_project[...
block_4_add (Add)	(None, 16, 16,	0	block_3_project_...

	32)		block_4_project_...
block_5_expand (Conv2D)	(None, 16, 16, 192)	6,144	block_4_add[0][0]
block_5_expand_BN (BatchNormalizatio...	(None, 16, 16, 192)	768	block_5_expand[0...
block_5_expand_relu (ReLU)	(None, 16, 16, 192)	0	block_5_expand_B...
block_5_depthwise (DepthwiseConv2D)	(None, 16, 16, 192)	1,728	block_5_expand_r...
block_5_depthwise_... (BatchNormalizatio...	(None, 16, 16, 192)	768	block_5_depthwis...
block_5_depthwise_... (ReLU)	(None, 16, 16, 192)	0	block_5_depthwis...
block_5_project (Conv2D)	(None, 16, 16, 32)	6,144	block_5_depthwis...
block_5_project_BN (BatchNormalizatio...	(None, 16, 16, 32)	128	block_5_project[...
block_5_add (Add)	(None, 16, 16, 32)	0	block_4_add[0][0... block_5_project_...
block_6_expand (Conv2D)	(None, 16, 16, 192)	6,144	block_5_add[0][0]
block_6_expand_BN (BatchNormalizatio...	(None, 16, 16, 192)	768	block_6_expand[0...
block_6_expand_relu (ReLU)	(None, 16, 16, 192)	0	block_6_expand_B...
block_6_pad (ZeroPadding2D)	(None, 17, 17, 192)	0	block_6_expand_r...
block_6_depthwise (DepthwiseConv2D)	(None, 8, 8, 192)	1,728	block_6_pad[0][0]
block_6_depthwise_... (BatchNormalizatio...	(None, 8, 8, 192)	768	block_6_depthwis...
block_6_depthwise_...	(None, 8, 8, 192)	0	block_6_depthwis...

(ReLU)

block_6_project (Conv2D)	(None, 8, 8, 64)	12,288	block_6_depthwis...
block_6_project_BN (BatchNormalizatio...	(None, 8, 8, 64)	256	block_6_project[...
block_7_expand (Conv2D)	(None, 8, 8, 384)	24,576	block_6_project_...
block_7_expand_BN (BatchNormalizatio...	(None, 8, 8, 384)	1,536	block_7_expand[0...
block_7_expand_relu (ReLU)	(None, 8, 8, 384)	0	block_7_expand_B...
block_7_depthwise (DepthwiseConv2D)	(None, 8, 8, 384)	3,456	block_7_expand_r...
block_7_depthwise_... (BatchNormalizatio...	(None, 8, 8, 384)	1,536	block_7_depthwis...
block_7_depthwise_... (ReLU)	(None, 8, 8, 384)	0	block_7_depthwis...
block_7_project (Conv2D)	(None, 8, 8, 64)	24,576	block_7_depthwis...
block_7_project_BN (BatchNormalizatio...	(None, 8, 8, 64)	256	block_7_project[...
block_7_add (Add)	(None, 8, 8, 64)	0	block_6_project_... block_7_project_...
block_8_expand (Conv2D)	(None, 8, 8, 384)	24,576	block_7_add[0][0]
block_8_expand_BN (BatchNormalizatio...	(None, 8, 8, 384)	1,536	block_8_expand[0...
block_8_expand_relu (ReLU)	(None, 8, 8, 384)	0	block_8_expand_B...
block_8_depthwise (DepthwiseConv2D)	(None, 8, 8, 384)	3,456	block_8_expand_r...
block_8_depthwise_...	(None, 8, 8, 384)	1,536	block_8_depthwis...



(BatchNormalizatio...			
block_8_depthwise_... (ReLU)	(None, 8, 8, 384)	0	block_8_depthwis...
block_8_project (Conv2D)	(None, 8, 8, 64)	24,576	block_8_depthwis...
block_8_project_BN (BatchNormalizatio...	(None, 8, 8, 64)	256	block_8_project[...
block_8_add (Add)	(None, 8, 8, 64)	0	block_7_add[0][0... block_8_project_...
block_9_expand (Conv2D)	(None, 8, 8, 384)	24,576	block_8_add[0][0]
block_9_expand_BN (BatchNormalizatio...	(None, 8, 8, 384)	1,536	block_9_expand[0...
block_9_expand_relu (ReLU)	(None, 8, 8, 384)	0	block_9_expand_B...
block_9_depthwise (DepthwiseConv2D)	(None, 8, 8, 384)	3,456	block_9_expand_r...
block_9_depthwise_... (BatchNormalizatio...	(None, 8, 8, 384)	1,536	block_9_depthwis...
block_9_depthwise_... (ReLU)	(None, 8, 8, 384)	0	block_9_depthwis...
block_9_project (Conv2D)	(None, 8, 8, 64)	24,576	block_9_depthwis...
block_9_project_BN (BatchNormalizatio...	(None, 8, 8, 64)	256	block_9_project[...
block_9_add (Add)	(None, 8, 8, 64)	0	block_8_add[0][0... block_9_project_...
block_10_expand (Conv2D)	(None, 8, 8, 384)	24,576	block_9_add[0][0]
block_10_expand_BN (BatchNormalizatio...	(None, 8, 8, 384)	1,536	block_10_expand[...
block_10_expand_re...	(None, 8, 8, 384)	0	block_10_expand_...

(ReLU)			
block_10_depthwise (DepthwiseConv2D)	(None, 8, 8, 384)	3,456	block_10_expand...
block_10_depthwise... (BatchNormalizatio...	(None, 8, 8, 384)	1,536	block_10_depthwi...
block_10_depthwise... (ReLU)	(None, 8, 8, 384)	0	block_10_depthwi...
block_10_project (Conv2D)	(None, 8, 8, 96)	36,864	block_10_depthwi...
block_10_project_BN (BatchNormalizatio...	(None, 8, 8, 96)	384	block_10_project...
block_11_expand (Conv2D)	(None, 8, 8, 576)	55,296	block_10_project...
block_11_expand_BN (BatchNormalizatio...	(None, 8, 8, 576)	2,304	block_11_expand[...
block_11_expand_re... (ReLU)	(None, 8, 8, 576)	0	block_11_expand...
block_11_depthwise (DepthwiseConv2D)	(None, 8, 8, 576)	5,184	block_11_expand...
block_11_depthwise... (BatchNormalizatio...	(None, 8, 8, 576)	2,304	block_11_depthwi...
block_11_depthwise... (ReLU)	(None, 8, 8, 576)	0	block_11_depthwi...
block_11_project (Conv2D)	(None, 8, 8, 96)	55,296	block_11_depthwi...
block_11_project_BN (BatchNormalizatio...	(None, 8, 8, 96)	384	block_11_project...
block_11_add (Add)	(None, 8, 8, 96)	0	block_10_project... block_11_project...
block_12_expand (Conv2D)	(None, 8, 8, 576)	55,296	block_11_add[0][...
block_12_expand_BN	(None, 8, 8, 576)	2,304	block_12_expand[...

(BatchNormalizatio...			
block_12_expand_re... (ReLU)	(None, 8, 8, 576)	0	block_12_expand_...
block_12_depthwise (DepthwiseConv2D)	(None, 8, 8, 576)	5,184	block_12_expand_...
block_12_depthwise... (BatchNormalizatio...	(None, 8, 8, 576)	2,304	block_12_depthwi...
block_12_depthwise... (ReLU)	(None, 8, 8, 576)	0	block_12_depthwi...
block_12_project (Conv2D)	(None, 8, 8, 96)	55,296	block_12_depthwi...
block_12_project_BN (BatchNormalizatio...	(None, 8, 8, 96)	384	block_12_project...
block_12_add (Add)	(None, 8, 8, 96)	0	block_11_add[0][...] block_12_project...
block_13_expand (Conv2D)	(None, 8, 8, 576)	55,296	block_12_add[0][...]
block_13_expand_BN (BatchNormalizatio...	(None, 8, 8, 576)	2,304	block_13_expand[...]
block_13_expand_re... (ReLU)	(None, 8, 8, 576)	0	block_13_expand_...
block_13_pad (ZeroPadding2D)	(None, 9, 9, 576)	0	block_13_expand_...
block_13_depthwise (DepthwiseConv2D)	(None, 4, 4, 576)	5,184	block_13_pad[0][...]
block_13_depthwise... (BatchNormalizatio...	(None, 4, 4, 576)	2,304	block_13_depthwi...
block_13_depthwise... (ReLU)	(None, 4, 4, 576)	0	block_13_depthwi...
block_13_project (Conv2D)	(None, 4, 4, 160)	92,160	block_13_depthwi...
block_13_project_BN	(None, 4, 4, 160)	640	block_13_project...

(BatchNormalizatio...			
block_14_expand (Conv2D)	(None, 4, 4, 960)	153,600	block_13_project...
block_14_expand_BN (BatchNormalizatio...	(None, 4, 4, 960)	3,840	block_14_expand[...
block_14_expand_re... (ReLU)	(None, 4, 4, 960)	0	block_14_expand_...
block_14_depthwise (DepthwiseConv2D)	(None, 4, 4, 960)	8,640	block_14_expand_...
block_14_depthwise... (BatchNormalizatio...	(None, 4, 4, 960)	3,840	block_14_depthwi...
block_14_depthwise... (ReLU)	(None, 4, 4, 960)	0	block_14_depthwi...
block_14_project (Conv2D)	(None, 4, 4, 160)	153,600	block_14_depthwi...
block_14_project_BN (BatchNormalizatio...	(None, 4, 4, 160)	640	block_14_project...
block_14_add (Add)	(None, 4, 4, 160)	0	block_13_project... block_14_project...
block_15_expand (Conv2D)	(None, 4, 4, 960)	153,600	block_14_add[0][...
block_15_expand_BN (BatchNormalizatio...	(None, 4, 4, 960)	3,840	block_15_expand[...
block_15_expand_re... (ReLU)	(None, 4, 4, 960)	0	block_15_expand_...
block_15_depthwise (DepthwiseConv2D)	(None, 4, 4, 960)	8,640	block_15_expand_...
block_15_depthwise... (BatchNormalizatio...	(None, 4, 4, 960)	3,840	block_15_depthwi...
block_15_depthwise... (ReLU)	(None, 4, 4, 960)	0	block_15_depthwi...
block_15_project	(None, 4, 4, 160)	153,600	block_15_depthwi...

(Conv2D)			
block_15_project_BN (BatchNormalizatio...	(None, 4, 4, 160)	640	block_15_project...
block_15_add (Add)	(None, 4, 4, 160)	0	block_14_add[0][... block_15_project...
block_16_expand (Conv2D)	(None, 4, 4, 960)	153,600	block_15_add[0][...
block_16_expand_BN (BatchNormalizatio...	(None, 4, 4, 960)	3,840	block_16_expand[...
block_16_expand_re... (ReLU)	(None, 4, 4, 960)	0	block_16_expand_...
block_16_depthwise (DepthwiseConv2D)	(None, 4, 4, 960)	8,640	block_16_expand_...
block_16_depthwise... (BatchNormalizatio...	(None, 4, 4, 960)	3,840	block_16_depthwi...
block_16_depthwise... (ReLU)	(None, 4, 4, 960)	0	block_16_depthwi...
block_16_project (Conv2D)	(None, 4, 4, 320)	307,200	block_16_depthwi...
block_16_project_BN (BatchNormalizatio...	(None, 4, 4, 320)	1,280	block_16_project...
Conv_1 (Conv2D)	(None, 4, 4, 1280)	409,600	block_16_project...
Conv_1_bn (BatchNormalizatio...	(None, 4, 4, 1280)	5,120	Conv_1[0][0]
out_relu (ReLU)	(None, 4, 4, 1280)	0	Conv_1_bn[0][0]
global_average_poo... (GlobalAveragePool...	(None, 1280)	0	out_relu[0][0]
dense (Dense)	(None, 2)	2,562	global_average_p...

Total params: 2,265,672 (8.64 MB)

Trainable params: 2,562 (10.01 KB)

Non-trainable params: 2,257,984 (8.61 MB)

Optimizer params: 5,126 (20.03 KB)

[TODO] Test the Keras model by computing the accuracy the whole test set

```
[18]: model.evaluate(trainDataSet.batch(32))    # evaluate the model on the training_
      ↪ set
```

```
63/63          13s 173ms/step -
accuracy: 0.9999 - loss: 0.0038
```

```
[18]: [0.0043730284087359905, 0.9994999766349792]
```

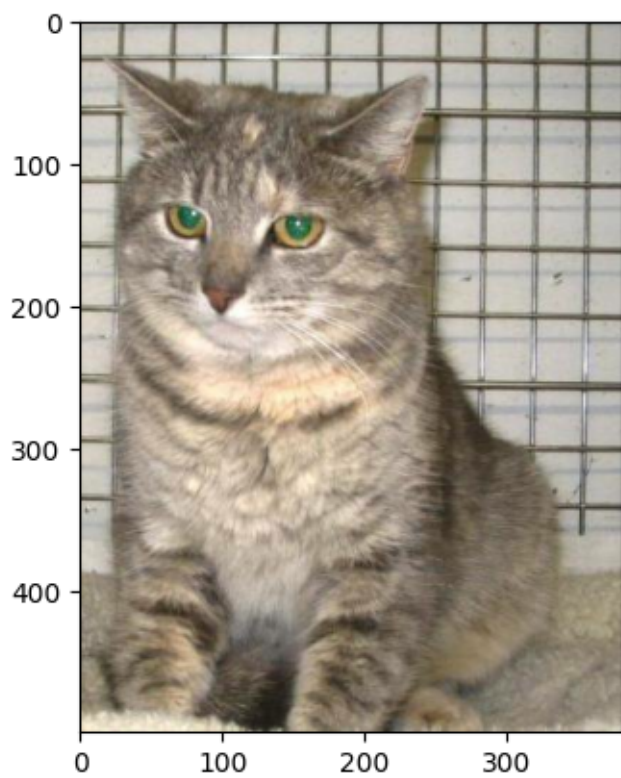
[TODO] Load Test image 'test/0.1047.jpg', visualize it and check the network prediction

```
[19]: # plt.imshow(x_test[47].reshape(28, 28), cmap='gray')
      # plt.title(f'Label: {y_test[47]}')
      # plt.show()

      # y_pred = model.predict(x_test[47][np.newaxis, :, :, :])
      # print(f'Predicted label: {np.argmax(y_pred)}')

      showImage = tf.io.read_file('test/0.1047.jpg')    # read the raw data from the_
      ↪ file as a string
      showImage = tf.image.decode_jpeg(showImage)    # decode the jpeg image to a tensor
      plt.imshow(showImage)    # show the image
      plt.show()

      # prepare the image for the model
      showImage = tf.image.resize(showImage, [128, 128])    # resize the image to_
      ↪ 128x128
      showImage = tf.cast(showImage, tf.float32) / 255.0    # transform the image to a_
      ↪ tf.float32 type and normalize it to [0, 1]
      showImage = showImage[tf.newaxis, :, :, :]    # add a batch dimension
      predict = model.predict(showImage)    # predict the label of the image
      print(f'Predicted label: {np.argmax(predict)}')    # print the predicted label
```



1/1 2s 2s/step  
Predicted label: 0