```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator

IMG_SIZE = 244
BATCH_SIZE = 32

train_datagen =
ImageDataGenerator(rescale=1./255,validation_split=0.2)
train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/1sv21cs040/images',
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='training'
)
val_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/1sv21cs040/images',
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='validation'
)
```

```
Found 288 images belonging to 4 classes.
Found 71 images belonging to 4 classes.
```

```
# Define the model
model = keras.Sequential([
    layers.Conv2D(32,
(3,3),activation='relu',input_shape=(IMG_SIZE,IMG_SIZE,3)),
    layers.MaxPooling2D(2,2),
    layers.Conv2D(64,(3,3),activation='relu'),
    layers.MaxPooling2D(2,2),
    layers.Conv2D(128,(3,3),activation='relu'),
    layers.MaxPooling2D(2,2),
    layers.Flatten(),
    layers.Dense(128,activation='relu'),
    layers.Dense(1,activation='sigmoid') #output layer
])

model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])

model.fit(train_generator,validation_data=val_generator,epochs=5)
```

```
Epoch 1/5
9/9 [==============================] - 143s 16s/step - loss: 1.2605 -
accuracy: 0.6858 - val_loss: 0.5872 - val_accuracy: 0.7500
Epoch 2/5
```

```
9/9 [==============================] - 49s 5s/step - loss: 0.5713 -
accuracy: 0.7500 - val_loss: 0.5662 - val_accuracy: 0.7500
Epoch 3/5
9/9 [==============================] - 47s 5s/step - loss: 0.5662 -
accuracy: 0.7500 - val_loss: 0.5648 - val_accuracy: 0.7500
Epoch 4/5
9/9 [==============================] - 47s 5s/step - loss: 0.5651 -
accuracy: 0.7500 - val_loss: 0.5636 - val_accuracy: 0.7500
Epoch 5/5
9/9 [==============================] - 48s 5s/step - loss: 0.5642 -
accuracy: 0.7500 - val_loss: 0.5631 - val_accuracy: 0.7500

<keras.src.callbacks.History at 0x7b435d03e0e0>

model.save("Model.h5","label.txt")

from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np

model = load_model('/content/Model.h5')
test_image_path = '/content/drive/MyDrive/1sv21cs040/images/apple
fruit/Image_1.jpg'
img = image.load_img(test_image_path, target_size=(224, 224))
img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array, axis=0)

img_array = img_array / 255.0

predictions = model.predict(img_array)
print(predictions)

---------------------------------------------------------------------
-----
ValueError                                Traceback (most recent call
last)
<ipython-input-24-065adbfd71df> in <cell line: 13>()
     11 img_array = img_array / 255.0
     12
---> 13 predictions = model.predict(img_array)
     14 print(predictions)

/usr/local/lib/python3.10/dist-packages/keras/src/utils/traceback_util
s.py in error_handler(*args, **kwargs)
     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

/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py
```

```
in tf__predict_function(iterator)
     13                 try:
     14                     do_return = True
---> 15                     retval_ =
ag__.converted_call(ag__.ld(step_function), (ag__.ld(self),
ag__.ld(iterator)), None, fscope)
     16                 except:
     17                     do_return = False

ValueError: in user code:

    File
"/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py"
, line 2440, in predict_function  *
        return step_function(self, iterator)
    File
"/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py"
, line 2425, in step_function  **
        outputs = model.distribute_strategy.run(run_step,
args=(data,))
    File
"/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py"
, line 2413, in run_step  **
        outputs = model.predict_step(data)
    File
"/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py"
, line 2381, in predict_step
        return self(x, training=False)
    File
"/usr/local/lib/python3.10/dist-packages/keras/src/utils/traceback_uti
ls.py", line 70, in error_handler
        raise e.with_traceback(filtered_tb) from None
    File
"/usr/local/lib/python3.10/dist-packages/keras/src/engine/input_spec.p
y", line 298, in assert_input_compatibility
        raise ValueError(

    ValueError: Input 0 of layer "sequential" is incompatible with the
layer: expected shape=(None, 244, 244, 3), found shape=(None, 224,
224, 3)


from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly
remount, call drive.mount("/content/drive", force_remount=True).

model.save("Model.h5","label.txt")
```

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np

model = load_model('Model.h5')
test_image_path = '/content/drive/MyDrive/1sv21cs040/images/apple
fruit/Image_1.jpg'
img = image.load_img(test_image_path, target_size=(244, 244)) # Change
target_size to match model input
img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array, axis=0)

img_array = img_array / 255.0

predictions = model.predict(img_array)
print(predictions)
```

```
1/1 [==============================] - 0s 111ms/step
[[0.2679664]]
```

```
if predictions < 0.5:
    print('It is a apple')
else:
    print('It is a banana')
```

```
It is a apple
```