

```
from google.colab import drive
```

```
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).

```
!wget https://bitbucket.org/ishaanjav/code-and-deploy-custom-tensorflow-lite-model/raw/a4febbfee178324b2083e322cdead7465d6dfd95/fruits.;
```

↗ --2025-08-13 09:46:10-- [https://bitbucket.org/ishaanjav/code-and-deploy-custom-tensorflow-lite-model/raw/a4febbfee178324b2083e322cdead7465d6dfd95/fruits.](https://bitbucket.org/ishaanjav/code-and-deploy-custom-tensorflow-lite-model/raw/a4febbfee178324b2083e322cdead7465d6dfd95/fruits.;)
Resolving bitbucket.org (bitbucket.org)... 104.192.142.24, 104.192.142.25, 104.192.142.26, ...
Connecting to bitbucket.org (bitbucket.org)|104.192.142.24|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 105946856 (101M) [application/zip]
Saving to: 'fruits.zip'

```
fruits.zip          100%[=====>] 101.04M  17.1MB/s   in 6.2s
```

```
2025-08-13 09:46:18 (16.2 MB/s) - 'fruits.zip' saved [105946856/105946856]
```

```
!unzip fruits.zip
```

↗ inflating: fruits/validation/banana/Screen Shot 2018-06-12 at 9.38.04 PM.png
inflating: __MACOSX/fruits/validation/banana/._Screen Shot 2018-06-12 at 9.38.04 PM.png
inflating: fruits/validation/banana/Screen Shot 2018-06-12 at 9.38.51 PM.png
inflating: __MACOSX/fruits/validation/banana/._Screen Shot 2018-06-12 at 9.38.51 PM.png
inflating: fruits/validation/banana/Screen Shot 2018-06-12 at 9.39.53 PM.png
inflating: __MACOSX/fruits/validation/banana/._Screen Shot 2018-06-12 at 9.39.53 PM.png
inflating: fruits/validation/banana/Screen Shot 2018-06-12 at 9.39.22 PM.png
inflating: __MACOSX/fruits/validation/banana/._Screen Shot 2018-06-12 at 9.39.22 PM.png
inflating: fruits/validation/banana/Screen Shot 2018-06-12 at 9.38.38 PM.png
inflating: __MACOSX/fruits/validation/banana/._Screen Shot 2018-06-12 at 9.38.38 PM.png
inflating: fruits/validation/banana/Screen Shot 2018-06-12 at 9.38.10 PM.png
inflating: __MACOSX/fruits/validation/banana/._Screen Shot 2018-06-12 at 9.38.10 PM.png
inflating: fruits/validation/banana/Screen Shot 2018-06-12 at 9.39.08 PM.png
inflating: __MACOSX/fruits/validation/banana/._Screen Shot 2018-06-12 at 9.39.08 PM.png
inflating: fruits/validation/orange/Screen Shot 2018-06-12 at 11.55.58 PM.png
inflating: __MACOSX/fruits/validation/orange/._Screen Shot 2018-06-12 at 11.55.58 PM.png
inflating: fruits/validation/orange/Screen Shot 2018-06-12 at 11.55.48 PM.png
inflating: __MACOSX/fruits/validation/orange/._Screen Shot 2018-06-12 at 11.55.48 PM.png
inflating: fruits/validation/orange/.DS_Store
inflating: __MACOSX/fruits/validation/orange/._DS_Store
inflating: fruits/validation/orange/Screen Shot 2018-06-12 at 11.56.16 PM.png
inflating: __MACOSX/fruits/validation/orange/._Screen Shot 2018-06-12 at 11.56.16 PM.png
inflating: fruits/validation/orange/Screen Shot 2018-06-12 at 11.56.43 PM.png
inflating: __MACOSX/fruits/validation/orange/._Screen Shot 2018-06-12 at 11.56.43 PM.png
inflating: fruits/validation/orange/Screen Shot 2018-06-12 at 11.56.20 PM.png
inflating: __MACOSX/fruits/validation/orange/._Screen Shot 2018-06-12 at 11.56.20 PM.png
inflating: fruits/validation/orange/Screen Shot 2018-06-12 at 11.58.56 PM.png
inflating: __MACOSX/fruits/validation/orange/._Screen Shot 2018-06-12 at 11.58.56 PM.png
inflating: fruits/validation/orange/Screen Shot 2018-06-12 at 11.57.45 PM.png
inflating: __MACOSX/fruits/validation/orange/._Screen Shot 2018-06-12 at 11.57.45 PM.png
inflating: fruits/validation/orange/Screen Shot 2018-06-12 at 11.58.11 PM.png
inflating: __MACOSX/fruits/validation/orange/._Screen Shot 2018-06-12 at 11.58.11 PM.png
inflating: fruits/validation/orange/Screen Shot 2018-06-12 at 11.56.55 PM.png
inflating: __MACOSX/fruits/validation/orange/._Screen Shot 2018-06-12 at 11.56.55 PM.png
inflating: fruits/validation/orange/Screen Shot 2018-06-12 at 11.58.50 PM.png
inflating: __MACOSX/fruits/validation/orange/._Screen Shot 2018-06-12 at 11.58.50 PM.png
inflating: fruits/validation/orange/Screen Shot 2018-06-12 at 11.56.02 PM.png
inflating: __MACOSX/fruits/validation/orange/._Screen Shot 2018-06-12 at 11.56.02 PM.png
inflating: fruits/validation/orange/Screen Shot 2018-06-12 at 11.58.24 PM.png
inflating: __MACOSX/fruits/validation/orange/._Screen Shot 2018-06-12 at 11.58.24 PM.png
inflating: fruits/validation/orange/Screen Shot 2018-06-12 at 11.58.28 PM.png
inflating: __MACOSX/fruits/validation/orange/._Screen Shot 2018-06-12 at 11.58.28 PM.png
inflating: fruits/validation/orange/Screen Shot 2018-06-12 at 11.57.37 PM.png
inflating: __MACOSX/fruits/validation/orange/._Screen Shot 2018-06-12 at 11.57.37 PM.png
inflating: fruits/validation/orange/Screen Shot 2018-06-12 at 11.58.02 PM.png
inflating: __MACOSX/fruits/validation/orange/._Screen Shot 2018-06-12 at 11.58.02 PM.png
inflating: fruits/validation/orange/Screen Shot 2018-06-12 at 11.57.52 PM.png
inflating: __MACOSX/fruits/validation/orange/._Screen Shot 2018-06-12 at 11.57.52 PM.png
inflating: fruits/validation/orange/Screen Shot 2018-06-12 at 11.57.07 PM.png
inflating: __MACOSX/fruits/validation/orange/._Screen Shot 2018-06-12 at 11.57.07 PM.png
inflating: fruits/validation/orange/Screen Shot 2018-06-12 at 11.58.18 PM.png
inflating: __MACOSX/fruits/validation/orange/._Screen Shot 2018-06-12 at 11.58.18 PM.png
inflating: fruits/validation/orange/Screen Shot 2018-06-12 at 11.56.48 PM.png
inflating: __MACOSX/fruits/validation/orange/._Screen Shot 2018-06-12 at 11.56.48 PM.png
inflating: fruits/validation/orange/Screen Shot 2018-06-12 at 11.56.35 PM.png
inflating: __MACOSX/fruits/validation/orange/._Screen Shot 2018-06-12 at 11.56.35 PM.png
inflating: fruits/validation/orange/Screen Shot 2018-06-12 at 11.58.43 PM.png
inflating: __MACOSX/fruits/validation/orange/._Screen Shot 2018-06-12 at 11.58.43 PM.png

```
import tensorflow as tf
import matplotlib.pyplot as plt
```

```
tf.__version__
```

```
↗ '2.19.0'
```

```
img_height, img_width = 32, 32
batch_size = 20
```

```
train_ds = tf.keras.utils.image_dataset_from_directory(
    "fruits/train",
    image_size = (img_height, img_width),
    batch_size = batch_size
)
val_ds = tf.keras.utils.image_dataset_from_directory(
    "fruits/validation",
    image_size = (img_height, img_width),
    batch_size = batch_size
)
test_ds = tf.keras.utils.image_dataset_from_directory(
    "fruits/test",
    image_size = (img_height, img_width),
    batch_size = batch_size
)
```

```
↗ Found 460 files belonging to 3 classes.
Found 66 files belonging to 3 classes.
Found 130 files belonging to 3 classes.
```

```
class_names = ["apple", "banana", "orange"]
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(class_names[labels[i]])
        plt.axis("off")
```



banana



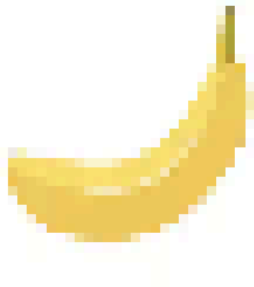
apple



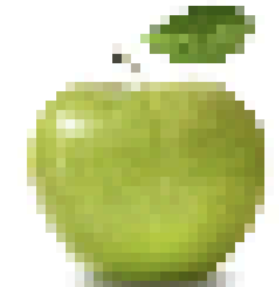
apple



banana



apple



banana



apple



banana



orange



```

model = tf.keras.Sequential(
    [
        tf.keras.layers.Rescaling(1./255),
        tf.keras.layers.Conv2D(32, 3, activation="relu"),
        tf.keras.layers.MaxPooling2D(),
        tf.keras.layers.Conv2D(64, 3, activation="relu"),
        tf.keras.layers.MaxPooling2D(),
        tf.keras.layers.Conv2D(128, 3, activation="relu"),
        tf.keras.layers.MaxPooling2D(),
        tf.keras.layers.Flatten(),
        tf.keras.layers.Dense(128, activation="softmax"),
        tf.keras.layers.Dense(3)
    ]
)

model.compile(
    optimizer="rmsprop",
    loss=tf.losses.SparseCategoricalCrossentropy(from_logits = True),
    metrics=['accuracy']
)

model.fit(
    train_ds,
    validation_data = val_ds,
    epochs = 20
)

```



```

Epoch 1/20
23/23 — 7s 112ms/step - accuracy: 0.3369 - loss: 1.0984 - val_accuracy: 0.6061 - val_loss: 1.0961
Epoch 2/20
23/23 — 2s 101ms/step - accuracy: 0.4700 - loss: 1.0917 - val_accuracy: 0.4697 - val_loss: 1.0570
Epoch 3/20
23/23 — 2s 85ms/step - accuracy: 0.4855 - loss: 1.0536 - val_accuracy: 0.5000 - val_loss: 1.0289
Epoch 4/20
23/23 — 2s 69ms/step - accuracy: 0.5902 - loss: 1.0037 - val_accuracy: 0.6061 - val_loss: 0.9700
Epoch 5/20
23/23 — 1s 64ms/step - accuracy: 0.5666 - loss: 0.9860 - val_accuracy: 0.6515 - val_loss: 0.9557

```

```

Epoch 6/20
23/23 ————— 3s 63ms/step - accuracy: 0.6278 - loss: 0.9529 - val_accuracy: 0.5758 - val_loss: 0.9670
Epoch 7/20
23/23 ————— 3s 66ms/step - accuracy: 0.6045 - loss: 0.9456 - val_accuracy: 0.6364 - val_loss: 0.9214
Epoch 8/20
23/23 ————— 4s 125ms/step - accuracy: 0.6161 - loss: 0.9229 - val_accuracy: 0.4848 - val_loss: 0.9825
Epoch 9/20
23/23 ————— 2s 76ms/step - accuracy: 0.6683 - loss: 0.9053 - val_accuracy: 0.6818 - val_loss: 0.8996
Epoch 10/20
23/23 ————— 2s 63ms/step - accuracy: 0.6846 - loss: 0.8801 - val_accuracy: 0.7121 - val_loss: 0.8731
Epoch 11/20
23/23 ————— 1s 62ms/step - accuracy: 0.7465 - loss: 0.8714 - val_accuracy: 0.8485 - val_loss: 0.8270
Epoch 12/20
23/23 ————— 3s 70ms/step - accuracy: 0.7927 - loss: 0.8426 - val_accuracy: 0.8485 - val_loss: 0.8134
Epoch 13/20
23/23 ————— 2s 68ms/step - accuracy: 0.8838 - loss: 0.7954 - val_accuracy: 0.8788 - val_loss: 0.7855
Epoch 14/20
23/23 ————— 2s 75ms/step - accuracy: 0.8761 - loss: 0.7696 - val_accuracy: 0.9091 - val_loss: 0.7447
Epoch 15/20
23/23 ————— 3s 110ms/step - accuracy: 0.9151 - loss: 0.7350 - val_accuracy: 0.8636 - val_loss: 0.7584
Epoch 16/20
23/23 ————— 4s 63ms/step - accuracy: 0.9427 - loss: 0.6935 - val_accuracy: 0.9394 - val_loss: 0.6955
Epoch 17/20
23/23 ————— 1s 63ms/step - accuracy: 0.9458 - loss: 0.6757 - val_accuracy: 0.9091 - val_loss: 0.6924
Epoch 18/20
23/23 ————— 3s 62ms/step - accuracy: 0.9599 - loss: 0.6459 - val_accuracy: 0.9242 - val_loss: 0.6538
Epoch 19/20
23/23 ————— 3s 68ms/step - accuracy: 0.9350 - loss: 0.6479 - val_accuracy: 0.9545 - val_loss: 0.6213
Epoch 20/20
23/23 ————— 3s 125ms/step - accuracy: 0.9670 - loss: 0.6045 - val_accuracy: 0.9091 - val_loss: 0.6280
<keras.src.callbacks.history.History at 0x7a5188b96950>

```

```
model.evaluate(test_ds)
```

```

7/7 ————— 1s 146ms/step - accuracy: 0.9471 - loss: 0.6202
[0.6357595324516296, 0.9384615421295166]

```

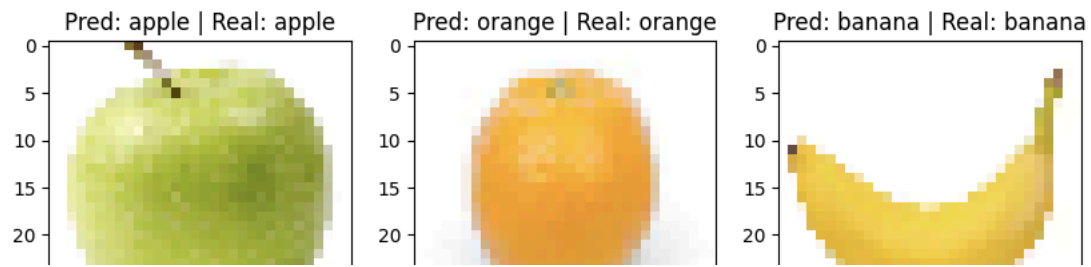
```
import numpy
```

```

plt.figure(figsize=(10,10))
for images, labels in test_ds.take(1):
    classifications = model(images)
    # print(classifications)

for i in range(9):
    ax = plt.subplot(3, 3, i + 1)
    plt.imshow(images[i].numpy().astype("uint8"))
    index = numpy.argmax(classifications[i])
    plt.title("Pred: " + class_names[index] + " | Real: " + class_names[labels[i]])

```



```
converter = tf.lite.TFLiteConverter.from_keras_model(model)
```