```
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).

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator
IMG_SIZE = 224
BATCH_SIZE=32

train_datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2)
train_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/archive/Vegetable Images/train',
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    subset='training'
)
val_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/archive/Vegetable Images/train',
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    subset='validation'
)

Found 197 images belonging to 4 classes.
Found 49 images belonging to 4 classes.

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')
])

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

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

Epoch 1/5
7/7 [==============================] - 55s 7s/step - loss: -26.0336 -
```

```
accuracy: 0.3046 - val_loss: -143.4800 - val_accuracy: 0.3061
Epoch 2/5
7/7 [==============================] - 24s 3s/step - loss: -435.8766 -
accuracy: 0.3046 - val_loss: -1498.4027 - val_accuracy: 0.3061
Epoch 3/5
7/7 [==============================] - 26s 3s/step - loss: -2801.8159 -
accuracy: 0.3046 - val_loss: -8108.7212 - val_accuracy: 0.3061
Epoch 4/5
7/7 [==============================] - 25s 4s/step - loss: -11872.0430 -
accuracy: 0.3046 - val_loss: -30162.5332 - val_accuracy: 0.3061
Epoch 5/5
7/7 [==============================] - 25s 3s/step - loss: -52670.7969 -
accuracy: 0.3046 - val_loss: -93746.5078 - val_accuracy: 0.3061

<keras.src.callbacks.History at 0x790d60ff3c40>
```

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

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103:
UserWarning: You are saving your model as an HDF5 file via `model.save()`.
This file format is considered legacy. We recommend using instead the native
Keras format, e.g. `model.save('my_model.keras')`.
  saving_api.save_model(
```

```python
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/archive/Vegetable
Images/train/Carrot/0914.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)
```

```
1/1 [==============================] - 0s 145ms/step
[[1.]]
```

```python
if predictions < 0.25:
 print('It is a tomato')
elif predictions < 0.50:
 Print('it is a papaya')
elif predictions <0.75:
 print('It is a potato')
else:
 print('It is a carrot')
```

It is a carrot

DAY1_026.ipynb

File   Edit   View   Insert   Runtime   Tools   Help   Last saved at 9:24 PM

Comment   Share

Files

+ Code   + Text

RAM
Disk

Gemini

0914.jpg

..

Queues

archive

Vegetable Images

test

train

Carrot

0914.jpg

0915.jpg

0916.jpg

0917.jpg

0918.jpg

Disk   80.15 GB available

```python
img_array = img_array / 255.0

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

```
1/1 [==============================] - 0s 145ms/step
[[1.]]
```

```python
if predictions < 0.25:
    print('It is a tomato')
elif predictions < 0.50:
    Print('it is a papaya')
elif predictions <0.75:
    print('It is a potato')
else:
    print('It is a carrot')
```

```
It is a carrot
```


✓ Connected to Python 3 Google Compute Engine backend