

```
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 [=====] - 76s 10s/step - loss: -26.9391 -
accuracy: 0.3147 - val_loss: -121.8270 - val_accuracy: 0.3061
Epoch 2/5
7/7 [=====] - 29s 4s/step - loss: -389.3503 -
accuracy: 0.3046 - val_loss: -1541.7667 - val_accuracy: 0.3061
Epoch 3/5
7/7 [=====] - 28s 4s/step - loss: -3138.5518
- accuracy: 0.3046 - val_loss: -8939.4834 - val_accuracy: 0.3061
Epoch 4/5
7/7 [=====] - 28s 4s/step - loss: -12494.0254
- accuracy: 0.3046 - val_loss: -31428.4805 - val_accuracy: 0.3061
Epoch 5/5
7/7 [=====] - 29s 4s/step - loss: -36952.1016
- accuracy: 0.3046 - val_loss: -63881.8867 - val_accuracy: 0.3061

<keras.src.callbacks.History at 0x79467800e140>

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(

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 182ms/step
[[1.]]

if predictions < 0.5:
    print('It is papaya')
else:
    print('It is carrot')

```

It is carrot