

```
import tensorflow as tf
from tensorflow.keras.preprocessing.image
import ImageDataGenerator
from tensorflow.keras.applications import
MobileNetV2
from tensorflow.keras.layers import Dense,
GlobalAveragePooling2D, Dropout
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import
Adam
import os
```

```
# === CONFIGURATION ===
```

```
IMAGE_SIZE = 224
```

```
BATCH_SIZE = 32
```

```
EPOCHS = 10
```

```
DATA_DIR = 'dataset' # Change to your
dataset path
```

```
# === DATA AUGMENTATION & LOADING
===
```

```
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=30,
    zoom_range=0.2,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    horizontal_flip=True,
    validation_split=0.2
)
```

```
train_generator =
train_datagen.flow_from_directory(
    DATA_DIR,
    target_size=(IMAGE_SIZE, IMAGE_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
```

```

        subset='training'
    )

    val_generator =
train_datagen.flow_from_directory(
    DATA_DIR,
    target_size=(IMAGE_SIZE, IMAGE_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='validation'
)

# === TRANSFER LEARNING MODEL ===
base_model =
MobileNetV2(weights='imagenet',
include_top=False,
input_shape=(IMAGE_SIZE, IMAGE_SIZE, 3))
base_model.trainable = False # Freeze the
base model

# Add custom layers
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dropout(0.3)(x)
predictions =
Dense(train_generator.num_classes,
activation='softmax')(x)

model = Model(inputs=base_model.input,
outputs=predictions)

# === COMPILE THE MODEL ===
model.compile(optimizer=Adam(learning_rate=0.0001), loss='categorical_crossentropy',
metrics=['accuracy'])

# === TRAINING ===
history = model.fit(
    train_generator,

```

```
train_generator,  
validation_data=val_generator,  
epochs=EPOCHS,  
verbose=1
```

```
)  
  
# === SAVE MODEL ===  
model.save('fruit_veg_classifier.h5')  
print("Model saved as  
fruit_veg_classifier.h5")
```
