

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

Mounted at /content/drive

```
import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.optimizers import Adam
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint, ReduceLROnPlateau
```

```
train_dir = '/content/drive/MyDrive/split_minip/train'
val_dir = '/content/drive/MyDrive/split_minip/val'
test_dir = '/content/drive/MyDrive/split_minip/test'
```

```
train_gen = ImageDataGenerator(rescale=1./255)
val_gen = ImageDataGenerator(rescale=1./255)
test_gen = ImageDataGenerator(rescale=1./255)
```

```
train_data = train_gen.flow_from_directory(train_dir, target_size=(224, 224), batch_size=32, class_mode='categorical')
val_data = val_gen.flow_from_directory(val_dir, target_size=(224, 224), batch_size=32, class_mode='categorical')
test_data = test_gen.flow_from_directory(test_dir, target_size=(224, 224), batch_size=32, class_mode='categorical')
```

Found 5600 images belonging to 8 classes.  
Found 1200 images belonging to 8 classes.

```
# Base model: VGG16 (without top/fully connected layers)
base_model = tf.keras.applications.VGG16(weights="imagenet", include_top=False, input_shape=(224, 224, 3))
```

Downloading data from [https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16\\_weights\\_tf\\_dim\\_ordering\\_tf\\_kernels\\_not\\_top.h5](https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_not_top.h5)  
58889256/58889256 4s 0us/step

```
for layer in base_model.layers:
    layer.trainable = False
x = base_model.output
x = Flatten()(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.5)(x)
output = Dense(8, activation='softmax')(x)

model = tf.keras.Model(inputs=base_model.input, outputs=output)
```

```
model.compile(optimizer=Adam(learning_rate=0.0001), loss='categorical_crossentropy', metrics=['accuracy'])
```

```
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
checkpoint_path = '/content/drive/MyDrive/vggbest_model.keras'
final_model_path = '/content/drive/MyDrive/vggfinal_model.keras'
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
```

```
model_checkpoint = ModelCheckpoint(
    checkpoint_path,
    monitor='val_loss',
    save_best_only=True,
)
```

```
history = model.fit(
    train_data,
    validation_data=val_data,
    epochs=30,
```

```
callbacks=[early_stopping, model_checkpoint]
)
model.save(final_model_path)
```

Epoch 1/30  
/usr/local/lib/python3.10/dist-packages/keras/src/trainers/data\_adapters/py\_dataset\_adapter.py:122: UserWarning: Your `PyDatasetAdapter` class does not implement the `warn\_if\_super\_not\_called` method.  
self.\_warn\_if\_super\_not\_called()

175/175	4759s	26s/step	- accuracy: 0.5079	- loss: 1.3424	- val_accuracy: 0.8077	- val_loss: 0.5308
Epoch 2/30						
175/175	39s	194ms/step	- accuracy: 0.7804	- loss: 0.5804	- val_accuracy: 0.8718	- val_loss: 0.4030
Epoch 3/30						
175/175	40s	188ms/step	- accuracy: 0.8314	- loss: 0.4609	- val_accuracy: 0.8676	- val_loss: 0.3817
Epoch 4/30						
175/175	39s	216ms/step	- accuracy: 0.8455	- loss: 0.4072	- val_accuracy: 0.8776	- val_loss: 0.3495
Epoch 5/30						
175/175	35s	191ms/step	- accuracy: 0.8670	- loss: 0.3521	- val_accuracy: 0.8709	- val_loss: 0.3326
Epoch 6/30						
175/175	39s	181ms/step	- accuracy: 0.8921	- loss: 0.2892	- val_accuracy: 0.8743	- val_loss: 0.3412
Epoch 7/30						
175/175	41s	185ms/step	- accuracy: 0.8996	- loss: 0.2766	- val_accuracy: 0.8876	- val_loss: 0.2949
Epoch 8/30						
175/175	35s	192ms/step	- accuracy: 0.9154	- loss: 0.2387	- val_accuracy: 0.8934	- val_loss: 0.2915
Epoch 9/30						
175/175	34s	186ms/step	- accuracy: 0.9215	- loss: 0.2139	- val_accuracy: 0.8809	- val_loss: 0.3123
Epoch 10/30						
175/175	46s	219ms/step	- accuracy: 0.9338	- loss: 0.1930	- val_accuracy: 0.8951	- val_loss: 0.2857
Epoch 11/30						
175/175	33s	181ms/step	- accuracy: 0.9380	- loss: 0.1777	- val_accuracy: 0.8851	- val_loss: 0.2908
Epoch 12/30						
175/175	32s	179ms/step	- accuracy: 0.9424	- loss: 0.1662	- val_accuracy: 0.8818	- val_loss: 0.3258
Epoch 13/30						
175/175	41s	178ms/step	- accuracy: 0.9486	- loss: 0.1462	- val_accuracy: 0.8834	- val_loss: 0.2879
Epoch 14/30						
175/175	32s	180ms/step	- accuracy: 0.9503	- loss: 0.1409	- val_accuracy: 0.8884	- val_loss: 0.2893
Epoch 15/30						
175/175	41s	178ms/step	- accuracy: 0.9632	- loss: 0.1198	- val_accuracy: 0.8893	- val_loss: 0.2987

```
val_loss, val_accuracy = model.evaluate(val_data) #model evaluation on val data
```

```
print(f'Validation Loss: {val_loss}')
print(f'Validation Accuracy: {val_accuracy}')
```

38/38 5s 131ms/step - accuracy: 0.9079 - loss: 0.2610  
Validation Loss: 0.2857494056224823  
Validation Accuracy: 0.8950874209403992

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import warnings
warnings.filterwarnings("ignore", category=UserWarning, module="keras.src.trainers.data_adapters.py_dataset_adapter")
```

```
test_dir = '/content/drive/MyDrive/split_minip/test'
test_gen = ImageDataGenerator(rescale=1./255)
test_data = test_gen.flow_from_directory(test_dir, target_size=(224, 224), batch_size=32, class_mode='categorical')
```

```
test_loss, test_accuracy = model.evaluate(test_data)
print(f"Test Loss: {test_loss:.4f}")
print(f"Test Accuracy: {test_accuracy * 100:.4f}")
```

Found 1160 images belonging to 8 classes.  
37/37 326s 9s/step - accuracy: 0.8905 - loss: 0.2944  
Test Loss: 0.2889  
Test Accuracy: 88.7931

