```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Dropout
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
import os

# Paths to dataset folders
train_dir = "dataset/train"
val_dir = "dataset/val"
test_dir = "dataset/test"
IMG_SIZE = 224
BATCH_SIZE = 32
EPOCHS = 10

# Data augmentation
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    zoom_range=0.2,
    horizontal_flip=True
)

val_datagen = ImageDataGenerator(rescale=1./255)

train_data = train_datagen.flow_from_directory(
    train_dir, target_size=(IMG_SIZE, IMG_SIZE), batch_size=BATCH_SIZE,
class_mode='categorical'
)

val_data = val_datagen.flow_from_directory(
    val_dir, target_size=(IMG_SIZE, IMG_SIZE), batch_size=BATCH_SIZE,
class_mode='categorical'
)

test_data = val_datagen.flow_from_directory(
    test_dir, target_size=(IMG_SIZE, IMG_SIZE), batch_size=1, class_mode='categorical',
shuffle=False
)

# Load MobileNetV2 base model
```

```
base_model = MobileNetV2(weights='imagenet', include_top=False, input_shape=(IMG_SIZE,
IMG_SIZE, 3))
base_model.trainable = False  # Freeze base layers

# Add custom classification head
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dropout(0.3)(x)
predictions = Dense(train_data.num_classes, activation='softmax')(x)
model = Model(inputs=base_model.input, outputs=predictions)

# Compile model
```