

PROGRAM:

```
import TensorFlow as tf  
from tensorflow.keras.preprocessing.image import  
ImageGenerator  
from tensorflow.keras.applications import VGG16  
from tensorflow.keras.layers import Dense,  
GlobalAveragePooling2D  
from tensorflow.keras.models import Model  
  
# Define paths to your dataset  
train Dir = 'path/to/train'  
testier = 'path/to/test'  
  
# Data preprocessing and augmentation  
train_datagen = ImageGenerator(  
    rescale=1. /255,  
    rotation range=40,  
    width_shift_range=0.2,  
    height_shift_range=0.2,  
    shear range=0.2,  
    zoom range=0.2,  
    horizontal flip=True,  
    fill mode='nearest'  
)
```



Edit with WPS Office

```
testudinate = ImageDataGenerator(rescale=1./255)

# Batch size
batch size = 32

# Load and augment data
train generator = train_datagen.flow_from_directory(
    train Dir,
    target size=(224, 224), # Adjust size as per your model
requirements
    batch size=batch size,
    class mode='categorical'
)

test generator = test_datagen.flow_from_directory(
    testier,
    target size=(224, 224),
    batch size=batch size,
    class mode='categorical'
)

# Load pre-trained model (VGG16 in this example)
base model = VGG16(weights='ImageNet', include
top=False)
```



Edit with WPS Office

```
# Add custom classification layers  
x = base_model.output  
x = GlobalAveragePooling2D()(x)  
x = Dense(1024, activation='relu')(x)  
predictions = Dense(Num classes, activation='SoftMax')(x)
```

```
# Combine base model and custom layers  
model = Model(inputs=base_model.input,  
outputs=predictions)
```

```
# Freeze pre-trained layers  
for layer in base_model.layers:  
    layer.Trainable = False
```

```
# Compile the model  
model.compile(optimizer='Adam',  
loss='categorical_crossentropy', metrics=['accuracy'])
```

```
# Train the model  
model.fit(train_generator, epochs=10, validation  
data=test_generator)
```

```
# Evaluate the model  
loss, accuracy = model.evaluate(test_generator)  
print("Test Accuracy:", accuracy)
```



Edit with WPS Office

```
# Make predictions
```

```
# Use the trained model to predict disease classes of new  
images
```



Edit with WPS Office