```
from tensorflow.keras.applications import VGG19
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, Flatten, Dropout
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.optimizers import Adam


import kagglehub

# Download latest version
path = kagglehub.dataset_download("nirmalsankalana/rice-leaf-disease-image")

print("Path to dataset files:", path)
```

```
Downloading from https://www.kaggle.com/api/v1/datasets/download/nirmalsankalana/rice-leaf-disease-image?dataset_version_number=1..
100%|██████████| 195M/195M [00:10<00:00, 20.4MB/s]Extracting files...

Path to dataset files: /root/.cache/kagglehub/datasets/nirmalsankalana/rice-leaf-disease-image/versions/1
```

```
# Veri artırma
datagen = ImageDataGenerator(
    rescale=1./255,
    validation_split=0.2,
    rotation_range=20,
    zoom_range=0.2,
    horizontal_flip=True
)



train_generator = datagen.flow_from_directory(
    path,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    subset='training'
)
```

```
Found 4747 images belonging to 4 classes.
```

```
val_generator = datagen.flow_from_directory(
    path,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    subset='validation'
)
```

```
Found 1185 images belonging to 4 classes.
```

```
base_model = VGG19(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg19/vgg19_weights_tf_dim_ordering_tf_kernels_no
80134624/80134624 ━━━━━━━━━━━━━━━━━━━━ 4s 0us/step
```

```
for layer in base_model.layers:
    layer.trainable = False



# Yeni sınıflandırıcı katmanlar
x = Flatten()(base_model.output)
x = Dense(256, activation='relu')(x)
x = Dropout(0.5)(x)
output = Dense(4, activation='softmax')(x)


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


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


model.fit(train_generator, validation_data=val_generator, epochs=28)
```

```
Epoch 1/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 109s 636ms/step - accuracy: 0.5794 - loss: 0.9935 - val_accuracy: 0.8059 - val_loss: 0.5094
Epoch 2/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 88s 591ms/step - accuracy: 0.8092 - loss: 0.4869 - val_accuracy: 0.8599 - val_loss: 0.3745
Epoch 3/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 84s 565ms/step - accuracy: 0.8638 - loss: 0.3643 - val_accuracy: 0.9063 - val_loss: 0.2953
Epoch 4/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 141s 560ms/step - accuracy: 0.8939 - loss: 0.2983 - val_accuracy: 0.9325 - val_loss: 0.2217
Epoch 5/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 84s 565ms/step - accuracy: 0.9182 - loss: 0.2378 - val_accuracy: 0.9342 - val_loss: 0.2007
Epoch 6/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 84s 564ms/step - accuracy: 0.9203 - loss: 0.2234 - val_accuracy: 0.9376 - val_loss: 0.1819
Epoch 7/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 84s 565ms/step - accuracy: 0.9327 - loss: 0.2006 - val_accuracy: 0.9359 - val_loss: 0.1838
Epoch 8/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 85s 571ms/step - accuracy: 0.9456 - loss: 0.1670 - val_accuracy: 0.9536 - val_loss: 0.1497
Epoch 9/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 88s 591ms/step - accuracy: 0.9468 - loss: 0.1546 - val_accuracy: 0.9671 - val_loss: 0.1114
Epoch 10/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 84s 564ms/step - accuracy: 0.9557 - loss: 0.1346 - val_accuracy: 0.9696 - val_loss: 0.1128
Epoch 11/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 142s 566ms/step - accuracy: 0.9560 - loss: 0.1333 - val_accuracy: 0.9679 - val_loss: 0.1056
Epoch 12/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 142s 565ms/step - accuracy: 0.9514 - loss: 0.1355 - val_accuracy: 0.9738 - val_loss: 0.0903
Epoch 13/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 85s 567ms/step - accuracy: 0.9633 - loss: 0.1171 - val_accuracy: 0.9679 - val_loss: 0.1034
Epoch 14/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 85s 569ms/step - accuracy: 0.9643 - loss: 0.1088 - val_accuracy: 0.9671 - val_loss: 0.0995
Epoch 15/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 87s 584ms/step - accuracy: 0.9636 - loss: 0.1076 - val_accuracy: 0.9831 - val_loss: 0.0635
Epoch 16/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 88s 588ms/step - accuracy: 0.9696 - loss: 0.1000 - val_accuracy: 0.9814 - val_loss: 0.0688
Epoch 17/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 88s 590ms/step - accuracy: 0.9752 - loss: 0.0871 - val_accuracy: 0.9797 - val_loss: 0.0675
Epoch 18/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 88s 593ms/step - accuracy: 0.9790 - loss: 0.0768 - val_accuracy: 0.9831 - val_loss: 0.0513
Epoch 19/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 88s 593ms/step - accuracy: 0.9763 - loss: 0.0777 - val_accuracy: 0.9865 - val_loss: 0.0538
Epoch 20/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 85s 572ms/step - accuracy: 0.9730 - loss: 0.0802 - val_accuracy: 0.9797 - val_loss: 0.0666
Epoch 21/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 84s 564ms/step - accuracy: 0.9727 - loss: 0.0781 - val_accuracy: 0.9705 - val_loss: 0.0787
Epoch 22/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 84s 564ms/step - accuracy: 0.9703 - loss: 0.0836 - val_accuracy: 0.9848 - val_loss: 0.0557
Epoch 23/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 87s 581ms/step - accuracy: 0.9800 - loss: 0.0652 - val_accuracy: 0.9907 - val_loss: 0.0441
Epoch 24/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 87s 583ms/step - accuracy: 0.9822 - loss: 0.0625 - val_accuracy: 0.9873 - val_loss: 0.0457
Epoch 25/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 87s 584ms/step - accuracy: 0.9809 - loss: 0.0645 - val_accuracy: 0.9890 - val_loss: 0.0459
Epoch 26/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 88s 591ms/step - accuracy: 0.9757 - loss: 0.0719 - val_accuracy: 0.9882 - val_loss: 0.0374
Epoch 27/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 88s 590ms/step - accuracy: 0.9822 - loss: 0.0577 - val_accuracy: 0.9730 - val_loss: 0.0705
Epoch 28/28
149/149 ━━━━━━━━━━━━━━━━━━━━ 88s 592ms/step - accuracy: 0.9834 - loss: 0.0539 - val_accuracy: 0.9840 - val_loss: 0.0520
<keras.src.callbacks.history.History at 0x7adcbbd0d990>
```

```
loss, accuracy = model.evaluate(val_generator)
print("Validation Accuracy:", accuracy)
```

```
38/38 ━━━━━━━━━━━━━━━━━━━━ 18s 459ms/step - accuracy: 0.9762 - loss: 0.0673
Validation Accuracy: 0.9789029359817505
```

```
model.save('vgg19_rice_model.keras')
```

```
from tensorflow.keras.models import load_model
model = load_model('vgg19_rice_model.keras')
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/saving/saving_lib.py:757: UserWarning: Skipping variable loading for optimizer 'rm
  saveable.load_own_variables(weights_store.get(inner_path))
```

```
from google.colab import files
files.download('vgg19_rice_model.keras')
```

```
Start coding or generate with AI.
```