

```
import kagglehub

# Download latest version
path = kagglehub.dataset_download("vipooooool/new-plant-diseases-dataset")

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

➡ Path to dataset files: /root/.cache/kagglehub/datasets/vipooooool/new-plant-diseases-dataset/ve
```

```
import tensorflow as tf
from tensorflow.keras.applications import VGG16
from tensorflow.keras import layers, models, optimizers
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```

```
os.listdir(path)
```

```
➡ ['test',
   'new plant diseases dataset(augmented)',
   'New Plant Diseases Dataset(Augmented)']
```

```
train_path = os.path.join(path, "new plant diseases dataset(augmented)", "New Plant Diseases Dataset(,
valid_path = os.path.join(path, "new plant diseases dataset(augmented)", "New Plant Diseases Dataset(,
test_path = os.path.join(path, 'test', "test")
```

```
names = os.listdir(train_path)
```

```
values = [i for i in range(len(names))]
classes_index = {value:key for (key,value) in zip(names,values)}
classes_names= {key:value for (key,value) in zip(names,values)}
```

## ▼ Data Preparation

```
# Data augmentation & preprocessing
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
```

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

```
# Load data
```

```
train_generator = train_datagen.flow_from_directory(
```

```

train_generator = train_datagen.flow_from_directory(
    train_path,
    target_size=(224, 224), # VGG input size
    batch_size=32,
    class_mode='categorical'
)

val_generator = val_datagen.flow_from_directory(
    valid_path,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical'
)

```

## ✓ Model Building (Using transfer Learning)

```

# Load pre-trained VGG16 (without top layer)
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

# Freeze convolutional layers (optional)
for layer in base_model.layers:
    layer.trainable = False # Freeze to avoid overfitting

# Add custom classification head
model = models.Sequential([
    base_model,
    layers.Flatten(),
    layers.Dense(256, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(train_generator.num_classes, activation='softmax') # Output layer
])

# Compile
model.compile(
    optimizer=optimizers.Adam(learning_rate=1e-4),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)

# Train
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // 32,
    epochs=10,
    validation_data=val_generator,
    validation_steps=val_generator.samples // 32
)

# Save model
model.save('my_vgg_model.h5')

```

```

➡ Found 70295 images belonging to 38 classes.
Found 17572 images belonging to 38 classes.
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16\_58889256/58889256 0s 0us/step
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py
self._warn_if_super_not_called()
Epoch 1/10
2196/2196 ————— 1086s 488ms/step - accuracy: 0.3535 - loss: 2.3670 - val_acc
Epoch 2/10
1/2196 ————— 6:05 166ms/step - accuracy: 0.7812 - loss: 0.8308/usr/local/
self._interrupted_warning()
2196/2196 ————— 96s 44ms/step - accuracy: 0.7812 - loss: 0.8308 - val_accu
Epoch 3/10
2196/2196 ————— 1103s 502ms/step - accuracy: 0.6666 - loss: 1.1047 - val_acc
Epoch 4/10
2196/2196 ————— 142s 65ms/step - accuracy: 0.8125 - loss: 0.6109 - val_accu
Epoch 5/10
2196/2196 ————— 1102s 485ms/step - accuracy: 0.7246 - loss: 0.8863 - val_acc
Epoch 6/10
2196/2196 ————— 96s 44ms/step - accuracy: 0.7812 - loss: 0.8392 - val_accu
Epoch 7/10
2196/2196 ————— 1030s 468ms/step - accuracy: 0.7604 - loss: 0.7615 - val_acc
Epoch 8/10
2196/2196 ————— 96s 44ms/step - accuracy: 0.8438 - loss: 0.4734 - val_accu
Epoch 9/10
2196/2196 ————— 1043s 475ms/step - accuracy: 0.7814 - loss: 0.6926 - val_acc
Epoch 10/10
2196/2196 ————— 142s 65ms/step - accuracy: 0.7812 - loss: 0.5083 - val_accu
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.savi

```

```

test_imgs = []
root = os.path.dirname(test_path)
for file in os.listdir(test_path):
    file_source = os.path.join(root, file)
    test_imgs.append(file_source)
test_imgs[0]

```

```

def predict_image(img_path):
    # Load and preprocess image
    img = image.load_img(img_path, target_size=(224, 224))
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0)
    img_array = img_array / 255.0 # Normalize (same as training)

    # Predict
    predictions = model.predict(img_array)
    predicted_class = np.argmax(predictions)
    confidence = np.max(predictions)

    return predicted_class, confidence

```

```

# Example usage
img_path = train_imgs[0]
predicted_class, confidence = predict_image(img_path)
print(f"Predicted: {predicted_class} (Confidence: {confidence:.2f})")

```

```

➡ 1/1 ————— 0s 64ms/step
Predicted: 22 (Confidence: 1.00)

```

classes\_names

```
➡ {'Potato__healthy': 0,
  'Raspberry__healthy': 1,
  'Soybean__healthy': 2,
  'Potato__Late_blight': 3,
  'Strawberry__Leaf_scorch': 4,
  'Apple__Cedar_apple_rust': 5,
  'Potato__Early_blight': 6,
  'Tomato__Leaf_Mold': 7,
  'Cherry_(including_sour)__Powdery_mildew': 8,
  'Peach__Bacterial_spot': 9,
  'Tomato__Tomato_mosaic_virus': 10,
  'Cherry_(including_sour)__healthy': 11,
  'Peach__healthy': 12,
  'Tomato__Spider_mites Two-spotted_spider_mite': 13,
  'Apple__Black_rot': 14,
  'Corn_(maize)__Common_rust_': 15,
  'Apple__Apple_scab': 16,
  'Corn_(maize)__healthy': 17,
  'Squash__Powdery_mildew': 18,
  'Grape__Leaf_blight_(Isariopsis_Leaf_Spot)': 19,
  'Corn_(maize)__Northern_Leaf_Blight': 20,
  'Tomato__Septoria_leaf_spot': 21,
  'Grape__healthy': 22,
  'Pepper,_bell__Bacterial_spot': 23,
  'Corn_(maize)__Cercospora_leaf_spot Gray_leaf_spot': 24,
  'Grape__Black_rot': 25,
  'Blueberry__healthy': 26,
  'Tomato__Tomato_Yellow_Leaf_Curl_Virus': 27,
  'Tomato__Bacterial_spot': 28,
  'Tomato__healthy': 29,
  'Tomato__Early_blight': 30,
  'Apple__healthy': 31,
  'Tomato__Target_Spot': 32,
  'Grape__Esca_(Black_Measles)': 33,
  'Strawberry__healthy': 34,
  'Orange__Haunglongbing_(Citrus_greening)': 35,
  'Tomato__Late_blight': 36,
  'Pepper,_bell__healthy': 37}
```

train\_imgs[0]

```
➡ '/root/.cache/kagglehub/datasets/vip000ool/new-plant-diseases-dataset/versions/2/new plant diseases dataset(augmented)/New Plant Diseases Dataset(Augmented)/train/Potato__healthy/20ac28d6-5708-4e31-8676-1b1979de094f__RS_HL_1735_new30degFlipTB.JPG'
```

classes\_index

```
➡ {0: 'Potato__healthy',
  1: 'Raspberry__healthy',
  2: 'Soybean__healthy',
  3: 'Potato__Late_blight',
  4: 'Strawberry__Leaf_scorch',
  5: 'Apple__Cedar_apple_rust',
  6: 'Potato__Early_blight',
  7: 'Tomato__Leaf_Mold',
  8: 'Cherry_(including_sour)__Powdery_mildew',
  9: 'Peach__Bacterial_spot',
  10: 'Tomato__Tomato_mosaic_virus',
  11: 'Cherry_(including_sour)__healthy',
  12: 'Peach__healthy',
  13: 'Tomato__Spider_mites Two-spotted_spider_mite',
```

```
14: 'Apple__Black_rot',
15: 'Corn_(maize)__Common_rust_',
16: 'Apple__Apple_scab',
17: 'Corn_(maize)__healthy',
18: 'Squash__Powdery_mildew',
19: 'Grape__Leaf_blight_(Isariopsis_Leaf_Spot)',
20: 'Corn_(maize)__Northern_Leaf_Blight',
21: 'Tomato__Septoria_leaf_spot',
22: 'Grape__healthy',
23: 'Pepper,_bell__Bacterial_spot',
24: 'Corn_(maize)__Cercospora_leaf_spot Gray_leaf_spot',
25: 'Grape__Black_rot',
26: 'Blueberry__healthy',
27: 'Tomato__Tomato_Yellow_Leaf_Curl_Virus',
28: 'Tomato__Bacterial_spot',
29: 'Tomato__healthy',
30: 'Tomato__Early_blight',
31: 'Apple__healthy',
32: 'Tomato__Target_Spot',
33: 'Grape__Esca_(Black_Measles)',
34: 'Strawberry__healthy',
35: 'Orange__Haunglongbing_(Citrus_greening)',
36: 'Tomato__Late_blight',
37: 'Pepper,_bell__healthy'}
```