

```

import os, numpy as np, matplotlib.pyplot as plt, seaborn as sns
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.utils.class_weight import compute_class_weight
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import VGG16
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, Dropout,
GlobalAveragePooling2D
from tensorflow.keras.callbacks import EarlyStopping,
ReduceLRonPlateau
from tensorflow.keras.optimizers import Adam

# === PARAMETERS ===
img_size = 224
batch_size = 32
epochs = 10
path = "/kaggle/input/plantdisease/PlantVillage"

# === DATA GENERATORS ===
datagen = ImageDataGenerator(
    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,
    validation_split=0.2
)

train_gen = datagen.flow_from_directory(
    path, target_size=(img_size, img_size), batch_size=batch_size,
    class_mode='categorical', subset='training', shuffle=True
)
val_gen = datagen.flow_from_directory(
    path, target_size=(img_size, img_size), batch_size=batch_size,
    class_mode='categorical', subset='validation', shuffle=False
)
class_names = list(train_gen.class_indices.keys())
num_classes = len(class_names)

# === CLASS WEIGHTS ===
class_weights = compute_class_weight(class_weight='balanced',

classes=np.unique(train_gen.classes),
                                y=train_gen.classes)
class_weights_dict = dict(enumerate(class_weights))

# === MODEL ===
base = VGG16(weights='imagenet', include_top=False,

```

```

input_shape=(img_size, img_size, 3))
base.trainable = True # Fine-tune entire model (can freeze first 10
layers if needed)

x = GlobalAveragePooling2D()(base.output)
x = Dropout(0.4)(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.3)(x)
out = Dense(num_classes, activation='softmax')(x)
model = Model(inputs=base.input, outputs=out)
model.compile(optimizer=Adam(1e-4), loss='categorical_crossentropy',
metrics=['accuracy'])

# === TRAIN ===
callbacks = [
    EarlyStopping(monitor='val_loss', patience=5,
restore_best_weights=True),
    ReduceLROnPlateau(monitor='val_loss', patience=2, factor=0.2,
verbose=1)
]

history = model.fit(train_gen, validation_data=val_gen, epochs=epochs,
                    callbacks=callbacks,
                    class_weight=class_weights_dict)

# === EVALUATE ===
val_gen.reset()
y_true = val_gen.classes
y_pred = np.argmax(model.predict(val_gen), axis=1)
print("Classification Report:\n")
print(classification_report(y_true, y_pred, target_names=class_names))

cm = confusion_matrix(y_true, y_pred)
plt.figure(figsize=(12, 10))
sns.heatmap(cm, annot=True, fmt='d', xticklabels=class_names,
            yticklabels=class_names, cmap='Blues')
plt.xlabel("Predicted"); plt.ylabel("True"); plt.title("Confusion
Matrix")
plt.show()

# === VISUALIZE PREDICTIONS ===
x_batch, y_batch = next(val_gen)
y_true_batch = np.argmax(y_batch, axis=1)
y_pred_batch = np.argmax(model.predict(x_batch), axis=1)
plt.figure(figsize=(16, 6))
for i in range(8):
    plt.subplot(2, 4, i + 1)
    plt.imshow(x_batch[i])
    true = class_names[y_true_batch[i]]
    pred = class_names[y_pred_batch[i]]

```

```
plt.title(f"True: {true}\nPred: {pred}", color='green' if
true==pred else 'red')
plt.axis('off')
plt.tight_layout(); plt.show()
```

2025-08-05 09:38:24.098462: E

external/local\_xla/xla/stream\_executor/cuda/cuda\_fft.cc:477] Unable to register cuFFT factory: Attempting to register factory for plugin cuFFT when one has already been registered

WARNING: All log messages before absl::InitializeLog() is called are written to STDERR

E0000 00:00:1754386704.311293 36 cuda\_dnn.cc:8310] Unable to register cuDNN factory: Attempting to register factory for plugin cuDNN when one has already been registered

E0000 00:00:1754386704.380645 36 cuda\_blas.cc:1418] Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered

Found 16516 images belonging to 15 classes.

Found 4122 images belonging to 15 classes.

I0000 00:00:1754386725.266864 36 gpu\_device.cc:2022] Created device /job:localhost/replica:0/task:0/device:GPU:0 with 13942 MB memory: -> device: 0, name: Tesla T4, pci bus id: 0000:00:04.0, compute capability: 7.5

I0000 00:00:1754386725.267660 36 gpu\_device.cc:2022] Created device /job:localhost/replica:0/task:0/device:GPU:1 with 13942 MB memory: -> device: 1, name: Tesla T4, pci bus id: 0000:00:05.0, compute capability: 7.5

Downloading data from [https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16\\_weights\\_tf\\_dim\\_ordering\\_tf\\_kernels\\_notop.h5](https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5)  
58889256/58889256 0s 0us/step

/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data\_adapters/py\_dataset\_adapter.py:121: UserWarning: Your `PyDataset` class should call `super().\_\_init\_\_(\*\*kwargs)` in its constructor. `\*\*kwargs` can include `workers`, `use\_multiprocessing`, `max\_queue\_size`. Do not pass these arguments to `fit()`, as they will be ignored.

```
self._warn_if_super_not_called()
```

Epoch 1/10

WARNING: All log messages before absl::InitializeLog() is called are written to STDERR

I0000 00:00:1754386734.801499 112 service.cc:148] XLA service 0x7fad7c002370 initialized for platform CUDA (this does not guarantee that XLA will be used). Devices:

I0000 00:00:1754386734.802631 112 service.cc:156] StreamExecutor device (0): Tesla T4, Compute Capability 7.5

```
I0000 00:00:1754386734.802652    112 service.cc:156] StreamExecutor
device (1): Tesla T4, Compute Capability 7.5
I0000 00:00:1754386735.652850    112 cuda_dnn.cc:529] Loaded cuDNN
version 90300
I0000 00:00:1754386767.798869    112 device_compiler.h:188] Compiled
cluster using XLA! This line is logged at most once for the lifetime
of the process.
```

```
517/517 _____ 416s 729ms/step - accuracy: 0.0650 -
loss: 2.7284 - val_accuracy: 0.0473 - val_loss: 2.6922 -
learning_rate: 1.0000e-04
```

Epoch 2/10

```
517/517 _____ 299s 577ms/step - accuracy: 0.1418 -
loss: 2.5406 - val_accuracy: 0.2623 - val_loss: 2.0998 -
learning_rate: 1.0000e-04
```

Epoch 3/10

```
517/517 _____ 299s 578ms/step - accuracy: 0.3385 -
loss: 1.8892 - val_accuracy: 0.5835 - val_loss: 1.1939 -
learning_rate: 1.0000e-04
```

Epoch 4/10

```
517/517 _____ 297s 575ms/step - accuracy: 0.5785 -
loss: 1.1484 - val_accuracy: 0.7555 - val_loss: 0.7263 -
learning_rate: 1.0000e-04
```

Epoch 5/10

```
517/517 _____ 299s 578ms/step - accuracy: 0.7871 -
loss: 0.6165 - val_accuracy: 0.8857 - val_loss: 0.3451 -
learning_rate: 1.0000e-04
```

Epoch 6/10

```
517/517 _____ 297s 574ms/step - accuracy: 0.8637 -
loss: 0.4006 - val_accuracy: 0.9224 - val_loss: 0.2437 -
learning_rate: 1.0000e-04
```

Epoch 7/10

```
517/517 _____ 298s 576ms/step - accuracy: 0.9010 -
loss: 0.2960 - val_accuracy: 0.9476 - val_loss: 0.1703 -
learning_rate: 1.0000e-04
```

Epoch 8/10

```
517/517 _____ 298s 575ms/step - accuracy: 0.9220 -
loss: 0.2441 - val_accuracy: 0.9561 - val_loss: 0.1418 -
learning_rate: 1.0000e-04
```

Epoch 9/10

```
517/517 _____ 298s 576ms/step - accuracy: 0.9354 -
loss: 0.1926 - val_accuracy: 0.9491 - val_loss: 0.1548 -
learning_rate: 1.0000e-04
```

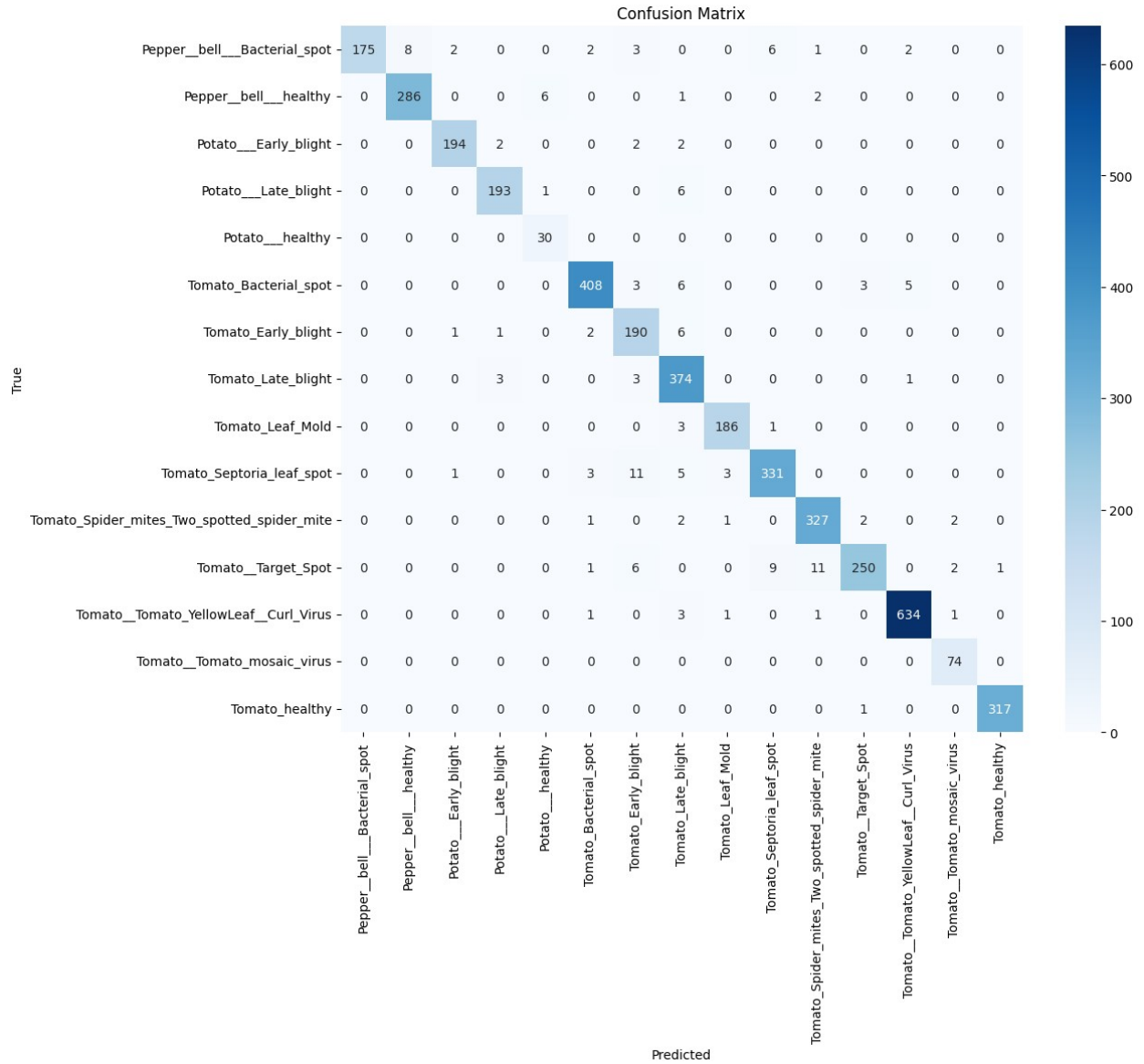
Epoch 10/10

```
517/517 _____ 297s 574ms/step - accuracy: 0.9479 -
loss: 0.1609 - val_accuracy: 0.9646 - val_loss: 0.1135 -
learning_rate: 1.0000e-04
```

```
129/129 _____ 47s 358ms/step
```

Classification Report:

score	support		precision	recall	f1-
0.94	199	Pepper__bell__Bacterial_spot	1.00	0.88	
0.97	295	Pepper__bell__healthy	0.97	0.97	
0.97	200	Potato__Early_blight	0.98	0.97	
0.97	200	Potato__Late_blight	0.97	0.96	
0.90	30	Potato__healthy	0.81	1.00	
0.97	425	Tomato_Bacterial_spot	0.98	0.96	
0.91	200	Tomato_Early_blight	0.87	0.95	
0.95	381	Tomato_Late_blight	0.92	0.98	
0.98	190	Tomato_Leaf_Mold	0.97	0.98	
0.94	354	Tomato_Septoria_leaf_spot	0.95	0.94	
0.97	335	Tomato_Spider_mites_Two_spotted_spider_mite	0.96	0.98	
0.93	280	Tomato__Target_Spot	0.98	0.89	
0.99	641	Tomato__Tomato_YellowLeaf__Curl_Virus	0.99	0.99	
0.97	74	Tomato__Tomato_mosaic_virus	0.94	1.00	
1.00	318	Tomato_healthy	1.00	1.00	
0.96	4122	accuracy			
0.96	4122	macro avg	0.95	0.96	
0.96	4122	weighted avg	0.96	0.96	



True: Pepper\_bell\_\_Bacterial\_spot  
Pred: Pepper\_bell\_\_Bacterial\_spot



True: Pepper\_bell\_\_Bacterial\_spot  
Pred: Pepper\_bell\_\_Bacterial\_spot



True: Pepper\_bell\_\_Bacterial\_spot  
Pred: Pepper\_bell\_\_Bacterial\_spot



True: Pepper\_bell\_\_Bacterial\_spot  
Pred: Pepper\_bell\_\_Bacterial\_spot



True: Pepper\_bell\_\_Bacterial\_spot  
Pred: Pepper\_bell\_\_Bacterial\_spot



True: Pepper\_bell\_\_Bacterial\_spot  
Pred: Pepper\_bell\_\_Bacterial\_spot



True: Pepper\_bell\_\_Bacterial\_spot  
Pred: Pepper\_bell\_\_Bacterial\_spot



True: Pepper\_bell\_\_Bacterial\_spot  
Pred: Pepper\_bell\_\_Bacterial\_spot

