

```
from google.colab import drive
drive.mount('/content/gdrive')
```

Mounted at /content/gdrive

```
!ls gdrive/MyDrive
```

```
BECOB265_LP1.pdf  'Data Science Question Bank .gdoc'
Classroom        PlantVillage
'Colab Notebooks' 'SAKSHI SURYAWANSHI - RESUME.pdf'
```

```
!unzip gdrive/MyDrive/PlantVillage.zip
```

```
import tensorflow as tf
from tensorflow.keras import models, layers
import matplotlib.pyplot as plt
from IPython.display import HTML
```

```
BATCH_SIZE = 32
IMAGE_SIZE = 256
CHANNELS=3
EPOCHS=50
```

```
!unzip PlantVillage.zip
```

```
Archive:  PlantVillage.zip
  creating:  PlantVillage/Potato__Early_blight/
  inflating:  PlantVillage/Potato__Early_blight/001187a0-57ab-4329-baff-e7246a9ede
  inflating:  PlantVillage/Potato__Early_blight/002a55fb-7a3d-4a3a-aca8-ce2d5ebc69
  inflating:  PlantVillage/Potato__Early_blight/009c8c31-f22d-4ffd-8f16-189c6f06c5
  inflating:  PlantVillage/Potato__Early_blight/00d8f10f-5038-4e0f-bb58-0b885ddc0c
  inflating:  PlantVillage/Potato__Early_blight/0182e991-97f0-4805-a1f7-6e1b4306d5
  inflating:  PlantVillage/Potato__Early_blight/02578b86-b234-4ac0-9bc3-691b5610e2
  inflating:  PlantVillage/Potato__Early_blight/0267d4ca-522e-4ca0-b1a2-ce925e5b54
  inflating:  PlantVillage/Potato__Early_blight/028f9b73-142f-499a-9c7b-d7c1ed5e55
  inflating:  PlantVillage/Potato__Early_blight/034959c1-f1e8-4a79-a6d5-3c1d14efa2
  inflating:  PlantVillage/Potato__Early_blight/03b0d3c1-b5b0-48f4-98aa-f890467029
  inflating:  PlantVillage/Potato__Early_blight/042135e2-e126-4900-9212-d42d900b81
```

```

inflating: PlantVillage/Potato___Early_blight/044c3abc-0bc9-45fb-8fd5-094aeb605f
inflating: PlantVillage/Potato___Early_blight/048d18ae-98b1-484d-97da-5a0e69b9eb
inflating: PlantVillage/Potato___Early_blight/04c8e6b9-7710-4cdd-b259-2d78b15d10
inflating: PlantVillage/Potato___Early_blight/04ee51b6-07e2-4182-84f8-46b22c8938
inflating: PlantVillage/Potato___Early_blight/04fd2a46-ddd4-4b0b-8f19-5ecca482a7
inflating: PlantVillage/Potato___Early_blight/05c35093-11b8-4cd0-b67a-1488597544
inflating: PlantVillage/Potato___Early_blight/0604174e-3018-4faa-9975-0be32d2c07
inflating: PlantVillage/Potato___Early_blight/060fd5a7-1606-4a59-895b-604c90d6b4
inflating: PlantVillage/Potato___Early_blight/065fc68f-88c9-4fc3-b0a6-a6f5e1072e
inflating: PlantVillage/Potato___Early_blight/06ac6596-8d65-46dd-a343-a2209f3480
inflating: PlantVillage/Potato___Early_blight/06d9fcc9-4eea-4736-8392-5e483e2e94
inflating: PlantVillage/Potato___Early_blight/07953ca1-8935-449f-b338-4357ed683b
inflating: PlantVillage/Potato___Early_blight/07baabd8-e118-47dd-9d8d-132d434783
inflating: PlantVillage/Potato___Early_blight/07d777f8-2c5a-48da-935e-f17c572b1e
inflating: PlantVillage/Potato___Early_blight/08029ccc-387e-4be6-9389-04f7b82fdb
inflating: PlantVillage/Potato___Early_blight/08194ca3-f0b2-4aaa-8df8-5ec5ddc669
inflating: PlantVillage/Potato___Early_blight/08392b44-ecc6-4f38-8566-361b552cfe
inflating: PlantVillage/Potato___Early_blight/0898bffc-57aa-4fdb-92d1-fd6a03d2a0
inflating: PlantVillage/Potato___Early_blight/089fb289-4bea-48b6-a29a-3a1d268d9d
inflating: PlantVillage/Potato___Early_blight/08a892eb-19cc-45ea-babc-181b98f892
inflating: PlantVillage/Potato___Early_blight/08bbd7f1-169d-4e38-946f-8d91589b26
inflating: PlantVillage/Potato___Early_blight/08cf1add-91f8-41ea-bc13-54e7d1cb6d
inflating: PlantVillage/Potato___Early_blight/094fbf4c-da00-4037-82af-03e712d8db
inflating: PlantVillage/Potato___Early_blight/096a2c48-104c-4570-9e34-8dd60aa5eb
inflating: PlantVillage/Potato___Early_blight/096f6928-24dd-4efd-acc6-b824631867
inflating: PlantVillage/Potato___Early_blight/09ab9573-90d5-4001-9b74-14551935de
inflating: PlantVillage/Potato___Early_blight/0a0744dc-8486-4fbb-a44b-4d63e6db61
inflating: PlantVillage/Potato___Early_blight/0a47f32c-1724-4c8d-bfe4-986cedd358
inflating: PlantVillage/Potato___Early_blight/0a6983a5-895e-4e68-9edb-88adf79211
inflating: PlantVillage/Potato___Early_blight/0a79700b-f834-41f5-ae51-6ceda6f67a
inflating: PlantVillage/Potato___Early_blight/0a8a68ee-f587-4dea-beec-79d02e7d3f
inflating: PlantVillage/Potato___Early_blight/0ad3ba53-f01b-403b-a99d-5991eed850
inflating: PlantVillage/Potato___Early_blight/0bbb8bce-2020-416b-8bd6-c160c2db99
inflating: PlantVillage/Potato___Early_blight/0c4f6f72-c7a2-42e1-9671-41ab3bf37f
inflating: PlantVillage/Potato___Early_blight/0c5b14d9-8b1c-4c39-bb23-1835b5760c
inflating: PlantVillage/Potato___Early_blight/0caf6a39-3f5f-4201-a4d7-3ea35fdf13
inflating: PlantVillage/Potato___Early_blight/0d2325ff-4e3e-44bf-9614-e5ad6c23fc
inflating: PlantVillage/Potato___Early_blight/0d2e2971-f1c9-4278-b35c-91dd8a22a6
inflating: PlantVillage/Potato___Early_blight/0d987d4a-26bc-4f74-8a16-12f8969dfe
inflating: PlantVillage/Potato___Early_blight/0d9dbf50-53a9-42b2-8b29-0360fb7dbd
inflating: PlantVillage/Potato___Early_blight/0ddd62cd-a999-4d58-a8f1-506e1004a5
inflating: PlantVillage/Potato___Early_blight/0e0a1b51-f61c-4934-bc57-a820af1faa
inflating: PlantVillage/Potato___Early_blight/0e6b9e09-2bcd-41e0-b001-b80a33a8a7
inflating: PlantVillage/Potato___Early_blight/0ed45bc2-c8cc-4d65-8f53-bdc10d78a7

```

```

dataset = tf.keras.preprocessing.image_dataset_from_directory(
    "PlantVillage",
    seed=123,
    shuffle=True,
    image_size=(IMAGE_SIZE, IMAGE_SIZE),
    batch_size=BATCH_SIZE
)

```

Found 2152 files belonging to 3 classes.

```

class_names = dataset.class_names
class_names

```

```
['Potato__Early_blight', 'Potato__Late_blight', 'Potato__healthy']
```

```
for image_batch, labels_batch in dataset.take(1):
    print(image_batch.shape)
    print(labels_batch.numpy())
```

```
(32, 256, 256, 3)
```

```
[1 1 1 0 0 0 0 0 1 1 1 1 0 1 0 1 1 1 0 1 0 1 0 0 1 0 0 1 1 2 0 0]
```

```
plt.figure(figsize=(10, 10))
for image_batch, labels_batch in dataset.take(1):
    for i in range(12):
        ax = plt.subplot(3, 4, i + 1)
        plt.imshow(image_batch[i].numpy().astype("uint8"))
        plt.title(class_names[labels_batch[i]])
        plt.axis("off")
```

Potato__Late_blight



Potato__Late_blight



Potato__Early_blight



Potato__Late_blight



Potato__Early_blight



Potato__Late_blight



Potato__Late_blight



Potato__Late_blight



Potato__Late_blight



Potato__Early_blight



Potato__healthy



Potato__Late_blight



```
len(dataset)
```

```
68
```

```
train_size = 0.8
```

```
len(dataset)*train_size
```

```
54.400000000000006
```

```
train_ds = dataset.take(54)
len(train_ds)
```

```
54
```

```
test_ds = dataset.skip(54)
len(test_ds)
```

```
14
```

```
val_size=0.1
len(dataset)*val_size
```

```
6.800000000000001
```

```
val_ds = test_ds.take(6)
len(val_ds)
```

```
6
```

```
test_ds = test_ds.skip(6)
len(test_ds)
```

```
8
```

```
def get_dataset_partitions_tf(ds, train_split=0.8, val_split=0.1, test_split=0.1, shuffle=
    assert (train_split + test_split + val_split) == 1

    ds_size = len(ds)

    if shuffle:
        ds = ds.shuffle(shuffle_size, seed=12)

    train_size = int(train_split * ds_size)
    val_size = int(val_split * ds_size)

    train_ds = ds.take(train_size)
    val_ds = ds.skip(train_size).take(val_size)
    test_ds = ds.skip(train_size).skip(val_size)

    return train_ds, val_ds, test_ds
```

```
train_ds, val_ds, test_ds = get_dataset_partitions_tf(dataset)
```

```
len(train_ds)
```

```
54
```

```
len(val_ds)
```

```
6
```

```
len(test_ds)
```

```
8
```

```
train_ds = train_ds.cache().shuffle(1000).prefetch(buffer_size=tf.data.AUTOTUNE)
val_ds = val_ds.cache().shuffle(1000).prefetch(buffer_size=tf.data.AUTOTUNE)
test_ds = test_ds.cache().shuffle(1000).prefetch(buffer_size=tf.data.AUTOTUNE)
```

```
resize_and_rescale = tf.keras.Sequential([
    layers.experimental.preprocessing.Resizing(IMAGE_SIZE, IMAGE_SIZE),
    layers.experimental.preprocessing.Rescaling(1./255),
])
```

```
data_augmentation = tf.keras.Sequential([
    layers.experimental.preprocessing.RandomFlip("horizontal_and_vertical"),
    layers.experimental.preprocessing.RandomRotation(0.2),
])
```

```
train_ds = train_ds.map(
    lambda x, y: (data_augmentation(x, training=True), y)
).prefetch(buffer_size=tf.data.AUTOTUNE)
```

```
input_shape = (BATCH_SIZE, IMAGE_SIZE, IMAGE_SIZE, CHANNELS)
n_classes = 3
```

```
model = models.Sequential([
    resize_and_rescale,
    layers.Conv2D(32, kernel_size = (3,3), activation='relu', input_shape=input_shape),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(n_classes, activation='softmax'),
])
```

```
model.summary()
```

Model: "sequential_2"

Layer (type)	Output Shape	Param #
=====		
sequential (Sequential)	(32, 256, 256, 3)	0
conv2d (Conv2D)	(32, 254, 254, 32)	896
max_pooling2d (MaxPooling2D)	(32, 127, 127, 32)	0
conv2d_1 (Conv2D)	(32, 125, 125, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(32, 62, 62, 64)	0
conv2d_2 (Conv2D)	(32, 60, 60, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(32, 30, 30, 64)	0
conv2d_3 (Conv2D)	(32, 28, 28, 64)	36928
max_pooling2d_3 (MaxPooling2D)	(32, 14, 14, 64)	0
conv2d_4 (Conv2D)	(32, 12, 12, 64)	36928
max_pooling2d_4 (MaxPooling2D)	(32, 6, 6, 64)	0
conv2d_5 (Conv2D)	(32, 4, 4, 64)	36928
max_pooling2d_5 (MaxPooling2D)	(32, 2, 2, 64)	0
flatten (Flatten)	(32, 256)	0
dense (Dense)	(32, 64)	16448
dense_1 (Dense)	(32, 3)	195
=====		
Total params: 183,747		
Trainable params: 183,747		
Non-trainable params: 0		

```
model.compile(
    optimizer='adam',
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),
    metrics=['accuracy']
)
```

```

history = model.fit(
    train_ds,
    batch_size=BATCH_SIZE,
    validation_data=val_ds,
    verbose=1,
    epochs=50,
)

```

```

Epoch 1/50
54/54 [=====] - 212s 4s/step - loss: 0.9076 - accuracy: 0
Epoch 2/50
54/54 [=====] - 204s 4s/step - loss: 0.7274 - accuracy: 0
Epoch 3/50
54/54 [=====] - 202s 4s/step - loss: 0.4707 - accuracy: 0
Epoch 4/50
54/54 [=====] - 203s 4s/step - loss: 0.3812 - accuracy: 0
Epoch 5/50
54/54 [=====] - 203s 4s/step - loss: 0.2827 - accuracy: 0
Epoch 6/50
54/54 [=====] - 203s 4s/step - loss: 0.2331 - accuracy: 0
Epoch 7/50
54/54 [=====] - 203s 4s/step - loss: 0.1965 - accuracy: 0
Epoch 8/50
54/54 [=====] - 207s 4s/step - loss: 0.2065 - accuracy: 0
Epoch 9/50
54/54 [=====] - 218s 4s/step - loss: 0.1633 - accuracy: 0
Epoch 10/50
54/54 [=====] - 202s 4s/step - loss: 0.1531 - accuracy: 0
Epoch 11/50
54/54 [=====] - 202s 4s/step - loss: 0.1371 - accuracy: 0
Epoch 12/50
54/54 [=====] - 202s 4s/step - loss: 0.2872 - accuracy: 0
Epoch 13/50
54/54 [=====] - 202s 4s/step - loss: 0.1657 - accuracy: 0
Epoch 14/50
54/54 [=====] - 201s 4s/step - loss: 0.1579 - accuracy: 0
Epoch 15/50
54/54 [=====] - 201s 4s/step - loss: 0.1047 - accuracy: 0
Epoch 16/50
54/54 [=====] - 201s 4s/step - loss: 0.1112 - accuracy: 0
Epoch 17/50
54/54 [=====] - 200s 4s/step - loss: 0.0955 - accuracy: 0
Epoch 18/50
54/54 [=====] - 201s 4s/step - loss: 0.0980 - accuracy: 0
Epoch 19/50
54/54 [=====] - 201s 4s/step - loss: 0.1053 - accuracy: 0
Epoch 20/50
54/54 [=====] - 201s 4s/step - loss: 0.1188 - accuracy: 0
Epoch 21/50
54/54 [=====] - 201s 4s/step - loss: 0.0742 - accuracy: 0
Epoch 22/50
54/54 [=====] - 201s 4s/step - loss: 0.0628 - accuracy: 0
Epoch 23/50
54/54 [=====] - 201s 4s/step - loss: 0.0700 - accuracy: 0
Epoch 24/50
54/54 [=====] - 202s 4s/step - loss: 0.0541 - accuracy: 0
Epoch 25/50
54/54 [=====] - 202s 4s/step - loss: 0.0920 - accuracy: 0
Epoch 26/50

```

```
54/54 [=====] - 202s 4s/step - loss: 0.0857 - accuracy: 0
Epoch 27/50
54/54 [=====] - 203s 4s/step - loss: 0.0550 - accuracy: 0
Epoch 28/50
54/54 [=====] - 203s 4s/step - loss: 0.0699 - accuracy: 0
Epoch 29/50
```

```
scores = model.evaluate(test_ds)
```

```
8/8 [=====] - 10s 884ms/step - loss: 0.2651 - accuracy: 0.93
```

```
scores
```

```
[0.26508617401123047, 0.93359375]
```

```
history
```

```
<keras.callbacks.History at 0x7efeba7d9550>
```

```
history.params
```

```
{'epochs': 50, 'steps': 54, 'verbose': 1}
```

```
history.history.keys()
```

```
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```

```
type(history.history['loss'])
```

```
list
```

```
len(history.history['loss'])
```

```
50
```

```
history.history['loss'][:5] # show loss for first 5 epochs
```

```
[0.9076286554336548,
 0.7273920774459839,
 0.47073042392730713,
 0.3811952471733093,
 0.2826952636241913]
```

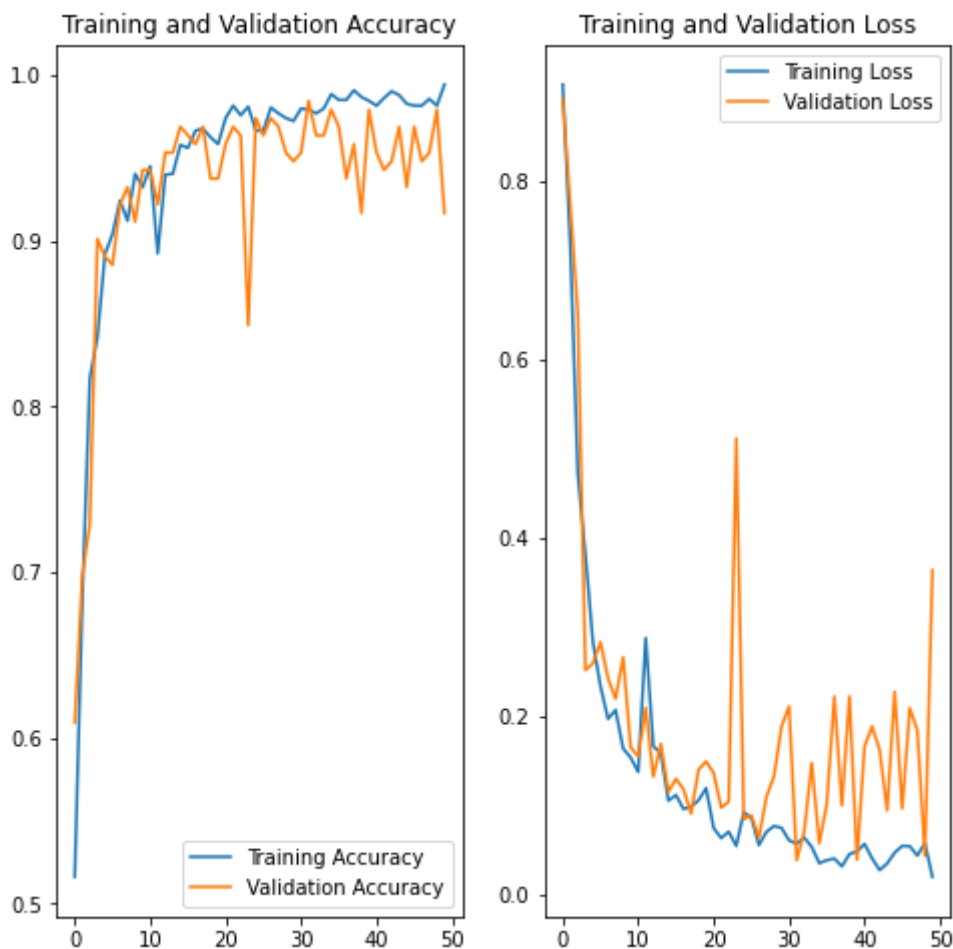
```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
```

```
loss = history.history['loss']
val_loss = history.history['val_loss']
```



```
plt.figure(figsize=(8, 8))
plt.subplot(1, 2, 1)
plt.plot(range(EPOCHS), acc, label='Training Accuracy')
plt.plot(range(EPOCHS), val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.subplot(1, 2, 2)
plt.plot(range(EPOCHS), loss, label='Training Loss')
plt.plot(range(EPOCHS), val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()
```



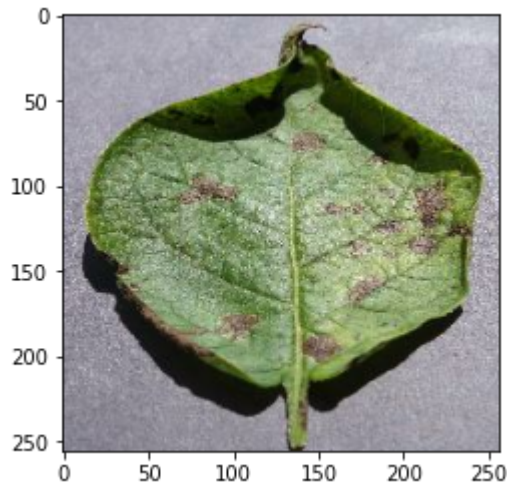
```
import numpy as np
for images_batch, labels_batch in test_ds.take(1):

    first_image = images_batch[0].numpy().astype('uint8')
    first_label = labels_batch[0].numpy()

    print("first image to predict")
    plt.imshow(first_image)
    print("actual label:", class_names[first_label])

    batch_prediction = model.predict(images_batch)
    print("predicted label:", class_names[np.argmax(batch_prediction[0])])
```

first image to predict
 actual label: Potato__Early_blight
 predicted label: Potato__Early_blight



```
def predict(model, img):
    img_array = tf.keras.preprocessing.image.img_to_array(images[i].numpy())
    img_array = tf.expand_dims(img_array, 0)

    predictions = model.predict(img_array)

    predicted_class = class_names[np.argmax(predictions[0])]
    confidence = round(100 * (np.max(predictions[0])), 2)
    return predicted_class, confidence
```

```
plt.figure(figsize=(15, 15))
for images, labels in test_ds.take(1):
    for i in range(9):
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))

        predicted_class, confidence = predict(model, images[i].numpy())
        actual_class = class_names[labels[i]]

        plt.title(f"Actual: {actual_class},\n Predicted: {predicted_class}.\n Confidence:

        plt.axis("off")
```

Actual: Potato__healthy,
Predicted: Potato__healthy.
Confidence: 99.72%



Actual: Potato__Late_blight,
Predicted: Potato__Late_blight.
Confidence: 97.7%



Actual: Potato__Late_blight,
Predicted: Potato__Late_blight.
Confidence: 100.0%



Actual: Potato__Late_blight,
Predicted: Potato__Late_blight.
Confidence: 99.99%



Actual: Potato__Early_blight,
Predicted: Potato__Early_blight.
Confidence: 99.99%



Actual: Potato__Late_blight,
Predicted: Potato__Early_blight.
Confidence: 94.89%



Actual: Potato__Early_blight,
Predicted: Potato__Early_blight.
Confidence: 100.0%



Actual: Potato__Early_blight,
Predicted: Potato__Early_blight.
Confidence: 99.93%



Actual: Potato__Early_blight,
Predicted: Potato__Early_blight.
Confidence: 100.0%



```
import os
# model_version=max([int(i) for i in os.listdir("../models") + [0]])+1
model_version = 1
model.save(f"models/{model_version}")
```

INFO:tensorflow:Assets written to: models/1/assets

```
model.save("potatoes.h5")
```

