```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_datagen=ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)

test_datagen=ImageDataGenerator(rescale=1./255)

x_train=train_datagen.flow_from_directory('veg-dataset/v_train',
                                          target_size=(128,128),batch_size=24,class_mode='categorical')

x_test=test_datagen.flow_from_directory('veg-dataset/v_test',
                                        target_size=(128,128),batch_size=24,class_mode='categorical')

Found 11386 images belonging to 9 classes.

Found 3416 images belonging to 9 classes.

x_train.class_indices

{'Pepper,bell__Bacterial_spot': 0,
 'Pepper,bell__healthy': 1,
 'Potato___Early_blight': 2,
 'Potato___Late_blight': 3,
 'Potato___healthy': 4,
 'Tomato___Bacterial_spot': 5,
 'Tomato___Late_blight': 6,
 'Tomato___Leaf_Mold': 7,
 'Tomato___Septoria_leaf_spot': 8}

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten

model=Sequential()

model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))

model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Flatten())

model.summary()

Model: "sequential_2"

_________________________

 Layer (type)                Output Shape              Param #

=================================================================
```

conv2d_2 (Conv2D) (None, 126, 126, 32) 896

max_pooling2d_2 (MaxPooling (None, 63, 63, 32) 0
2D)

flatten_2 (Flatten) (None, 127008) 0

===

Total params: 896

Trainable params: 896

Non-trainable params: 0

_____

model.add(Dense(300,activation='relu'))

model.add(Dense(150,activation='relu'))

model.add(Dense(9,activation='softmax'))

model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

model.fit(x_train,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test),
epochs=5)

Epoch 1/5

475/475 [==============================] - 209s 438ms/step - loss: 1.3413 - accuracy: 0.5973 -
val_loss: 0.9746 - val_accuracy: 0.6970

Epoch 2/5

475/475 [==============================] - 204s 429ms/step - loss: 0.5945 - accuracy: 0.7962 -
val_loss: 0.5676 - val_accuracy: 0.7857

Epoch 3/5

475/475 [==============================] - 202s 425ms/step - loss: 0.4401 - accuracy: 0.8478 -
val_loss: 0.3148 - val_accuracy: 0.8934

Epoch 4/5

475/475 [==============================] - 203s 428ms/step - loss: 0.3859 - accuracy: 0.8676 -
val_loss: 0.2264 - val_accuracy: 0.9259

Epoch 5/5

475/475 [==============================] - 203s 427ms/step - loss: 0.3254 - accuracy: 0.8878 -
val_loss: 0.3815 - val_accuracy: 0.8589

```
score,acc=model.evaluate(x_test,batch_size=128,verbose=2)
acc
143/143 - 8s - loss: 0.3815 - accuracy: 0.8589 - 8s/epoch - 57ms/step
0.8588992953300476
model.save('veg.h5')
```