

The Jupyter Notebook interface shows a file explorer at the top with the following contents:

File Name	Size	Type
01-08-2022 18:05	<DIR>	Fruit-dataset
01-08-2022 19:48	111,319,456	Fruits.H5
01-08-2022 18:06	<DIR>	Veg-dataset
01-08-2022 13:39	111,324,768	Vegetables.H5
2 File(s)	222,644,216	Bytes
4 Dir(s)	61,699,586,176	Bytes

The code cells show the following commands and outputs:

```
In [13]: x_train=train_datagen.flow_from_directory('C:\\Users\\admin\\Downloads\\Project\\Veg-dataset\\train_set',target_size=(64,64),
4
Found 11300 images belonging to 9 classes.

In [14]: x_test=test_datagen.Flow_from_directory('C:\\Users\\admin\\Downloads\\Project\\Veg-dataset\\test_set',target_size=(64,64),cl
4
Found 3416 images belonging to 9 classes.

In [15]: x_train.class_indices
Out[15]: {'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}
```

**Figure 1 Python coding for plant disease detection in jupyter notebook**

The Jupyter Notebook interface shows the following code cells:

```
In [19]: model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

In [20]: len(x_train)
Out[20]: 356

In [21]: len(x_test)
Out[21]: 167

In [22]: model.fit(x_train,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test),epochs=5)
356/356 [=====] - 929s 2s/step - loss: 1.1485 - accuracy: 0.5988 - val_loss: 0.6079 - val_accuar
y: 0.7857
Out[22]: <keras.callbacks.History at 0x22588991a38>

In [28]: ls
Volume in drive C has no label.
Volume Serial Number is 8869-58E3

Directory of C:\\Users\\admin\\Downloads\\Project

01-08-2022 20:38 <DIR> .
01-08-2022 20:38 <DIR> ..
01-08-2022 18:05 <DIR> fruit-dataset
01-08-2022 18:06 <DIR> Veg-dataset
```

**(a) Single epoch**

```
jupyter Project Last Checkpoint: Last Wednesday at 10:04 PM (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) C

In [22]: len(x_train)
Out[22]: 358

In [23]: len(x_test)
Out[23]: 187

In [24]: model.fit(x_train, steps_per_epoch=len(x_train), validation_data=x_test, validation_steps=len(x_test), epochs=5)


Epoch 1/5
356/356 [=====] - 744s 2s/step - loss: 1.0807 - accuracy: 0.6195 - val_loss: 0.7210 - val_accuracy: 0.7387
Epoch 2/5
356/356 [=====] - 307s 2s/step - loss: 0.5580 - accuracy: 0.8808 - val_loss: 0.4729 - val_accuracy: 0.8387
Epoch 3/5
356/356 [=====] - 510s 2s/step - loss: 0.4823 - accuracy: 0.8388 - val_loss: 0.5110 - val_accuracy: 0.8249
Epoch 4/5
356/356 [=====] - 573s 2s/step - loss: 0.3760 - accuracy: 0.8603 - val_loss: 0.2675 - val_accuracy: 0.9119
Epoch 5/5
356/356 [=====] - 727s 2s/step - loss: 0.2880 - accuracy: 0.8981 - val_loss: 0.2882 - val_accuracy: 0.8955


Out[24]: <keras.callbacks.History at 0x1a6bb2a3a30>
```

(b) five epochs

Figure 2 Accuracy of tested dataset by CNN

```
jupyter Project Last Checkpoint: Last Wednesday at 10:04 PM (unsaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) C

In [33]: img_image.load_img("C:\\Users\\adwin\\Downloads\\Project\\Veg-dataset\\test_set\\Pepper_bell_healthy\\b45d8a2-3dc1-4120-8f1-4120-8f1.jpg")
Out[33]: 

In [34]: act\\Veg-dataset\\test_set\\Pepper_bell_healthy\\ff80d1a7-1150-4c8a-b095-1a28028cfac5_18_HL_8266.jpg", target_size=(64, 64))
Out[34]: 


In [35]: img
Out[35]: 
```

Figure 3 Original and reframed image in the model

```

In [40]: x=np.expand_dims(x,axis=0)

In [41]: x
Out[41]: array([[[[210., 212., 226.],
                  [210., 212., 226.],
                  [210., 212., 226.],
                  ...,
                  [197., 193., 207.],
                  [205., 197., 213.],
                  [200., 198., 210.]],
                  [[217., 213., 227.],
                  [218., 215., 229.],
                  [217., 213., 227.],
                  ...,
                  [193., 187., 203.],
                  [187., 183., 197.],
                  [186., 182., 190.]],
                  [[220., 216., 230.],
                  [221., 217., 231.],
                  [220., 218., 230.],
                  ...,
                  [188., 185., 190.],
                  [202., 198., 212.],
                  [192., 188., 202.]]]])

```

**Figure 4 Image array structure built in python coding**

```

Index=[ 'Pepper_bell_Bacterial_spot', 'Pepper_bell_healthy', 'Potato_Early_blight', 'Potato_Late_blight', 'Potato_healthy', 'Tomato_Sepioria_leaf_spot' ]
index[y[0]]
1/1 [=====] - 8s 31ms/step

Out[48]: 'Potato_Early_blight'

In [49]: img=image.load_img('C:\\Users\\admin\\Downloads\\Project\\Veg-dataset\\test_set\\tomato_Late_blight\\MS_Late.B-8624.jpg',target_size=(224,224))
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
Index=[ 'Pepper_bell_Bacterial_spot', 'Pepper_bell_healthy', 'Potato_Early_blight', 'Potato_Late_blight', 'Potato_healthy', 'Tomato_Sepioria_leaf_spot' ]
index[y[0]]
1/1 [=====] - 0s 31ms/step

Out[49]: 'Potato_Early_blight'

In [50]: img=image.load_img('C:\\Users\\admin\\Downloads\\Project\\Veg-dataset\\test_set\\tomato_Sepioria_leaf_spot\\Keller-St_CG_189.jpg',target_size=(224,224))
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
Index=[ 'Pepper_bell_Bacterial_spot', 'Pepper_bell_healthy', 'Potato_Early_blight', 'Potato_Late_blight', 'Potato_healthy', 'Tomato_Sepioria_leaf_spot' ]
index[y[0]]
1/1 [=====] - 0s 18ms/step

Out[50]: 'tomato_Sepioria_leaf_spot'

```

**Figure 5 Test results for plant disease detection using CNN**

```

jupyter Project-Fruit disease detection Last Checkpoint: Last Wednesday at 10:04 PM (autosaved)
Python 3 (ipykernel)

File Edit View Insert Cell Kernel Widgets Help

In [10]: try:
          x_train = ImageDataset('C:\\Users\\admin\\Downloads\\Project\\fruit-dataset\\train', target_size=(64,64), class_mode='categorical', batch_size=32)
          print('Found 5384 Images belonging to 6 classes.')
        except:
          pass

In [11]: try:
          x_test = ImageDataset('C:\\Users\\admin\\Downloads\\Project\\fruit-dataset\\test', target_size=(64,64), class_mode='categorical', batch_size=32)
          print('Found 1680 Images belonging to 6 classes.')
        except:
          pass

In [12]: x_train.class_indices
Out[12]: {'Apple_Black_rot': 0,
          'Apple_healthy': 1,
          'Corn_(maize)_Northern_leaf_Blight': 2,
          'Corn_(maize)_healthy': 3,
          'Peach_Bacterial_spot': 4,
          'Peach_healthy': 5}

CNN

```

Figure 6 Coding for fruit disease detection using CNN

```

jupyter Project-Fruit disease detection Last Checkpoint: Last Wednesday at 10:04 PM (autosaved)
Python 3 (ipykernel)

File Edit View Insert Cell Kernel Widgets Help

y: 0.9087
Epoch 2/10
160/160 [=====] - 230s 1s/step - loss: 0.2781 - accuracy: 0.9079 - val_loss: 0.2232 - val_accuracy: 0.9199
y: 0.9199
Epoch 3/10
160/160 [=====] - 311s 2s/step - loss: 0.2037 - accuracy: 0.9313 - val_loss: 0.3566 - val_accuracy: 0.9472
y: 0.9472
Epoch 4/10
160/160 [=====] - 313s 2s/step - loss: 0.1704 - accuracy: 0.9382 - val_loss: 0.3904 - val_accuracy: 0.9359
y: 0.9359
Epoch 5/10
160/160 [=====] - 261s 1s/step - loss: 0.1419 - accuracy: 0.9482 - val_loss: 0.2007 - val_accuracy: 0.9348
y: 0.9348
Epoch 6/10
160/160 [=====] - 258s 1s/step - loss: 0.1384 - accuracy: 0.9507 - val_loss: 0.1134 - val_accuracy: 0.9533
y: 0.9533
Epoch 7/10
160/160 [=====] - 240s 1s/step - loss: 0.0825 - accuracy: 0.9725 - val_loss: 0.0000 - val_accuracy: 0.9698
y: 0.9698
Epoch 8/10
160/160 [=====] - 239s 1s/step - loss: 0.0998 - accuracy: 0.9604 - val_loss: 0.3745 - val_accuracy: 0.9448
y: 0.9448
Epoch 9/10
160/160 [=====] - 257s 2s/step - loss: 0.0932 - accuracy: 0.9600 - val_loss: 0.1030 - val_accuracy: 0.9349
y: 0.9349
Epoch 10/10
160/160 [=====] - 244s 1s/step - loss: 0.0833 - accuracy: 0.9694 - val_loss: 0.0036 - val_accuracy: 0.9757
y: 0.9757

Out[24]: keras.callbacks.History at 0x258a790a30e

```

Figure 7 Training results for fruit disease detection using CNN

```

jupyter Project-Fruit disease detection Last Checkpoint: Last Wednesday at 10:04 PM (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help Python 3 (ipykernel)

In [45]: index=[ 'Apple_black_rot', 'Apple_healthy', 'Corn_(maize)_Northern_Leaf_Blight', 'Corn_(maize)_healthy', 'Peach_Bacterial_spot', 'Peach_healthy' ]
Out[45]:
In [46]: index[y[0]]
Out[46]: 'Apple_healthy'

In [47]: img=image.load_img("C:\\Users\\admin\\Downloads\\Project\\fruit-dataset\\test\\Apple_black_rot\\8bc4b6c3-6a81-480e-a22f-867e-811d38d-4.jpg")
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
index=[ 'Apple_black_rot', 'Apple_healthy', 'Corn_(maize)_Northern_Leaf_Blight', 'Corn_(maize)_healthy', 'Peach_Bacterial_spot', 'Peach_healthy' ]
index[y[0]]
Out[47]: 'Apple_black_rot'

In [48]: img=image.load_img("C:\\Users\\admin\\Downloads\\Project\\fruit-dataset\\test\\Corn_(maize)_Northern_Leaf_Blight\\1a11d38d-4.jpg")
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
index=[ 'Apple_black_rot', 'Apple_healthy', 'Corn_(maize)_Northern_Leaf_Blight', 'Corn_(maize)_healthy', 'Peach_Bacterial_spot', 'Peach_healthy' ]
index[y[0]]
Out[48]: 'Corn_(maize)_healthy'

```

Figure 8 Test results 1 for fruit disease detection using CNN

```

jupyter Project-Fruit disease detection Last Checkpoint: Last Wednesday at 10:04 PM (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help Python 3 (ipykernel)

In [49]: index=[ 'Apple_black_rot', 'Apple_healthy', 'Corn_(maize)_Northern_Leaf_Blight', 'Corn_(maize)_healthy', 'Peach_Bacterial_spot', 'Peach_healthy' ]
index[y[0]]
Out[49]: 'Corn_(maize)_healthy'

In [50]: img=image.load_img("C:\\Users\\admin\\Downloads\\Project\\fruit-dataset\\test\\Peach_Bacterial_spot\\8bc4b6c3-6a81-480e-a22f-867e-811d38d-4.jpg")
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
index=[ 'Apple_black_rot', 'Apple_healthy', 'Corn_(maize)_Northern_Leaf_Blight', 'Corn_(maize)_healthy', 'Peach_Bacterial_spot', 'Peach_healthy' ]
index[y[0]]
Out[50]: 'Peach_Bacterial_spot'

In [51]: img=image.load_img("C:\\Users\\admin\\Downloads\\Project\\fruit-dataset\\test\\Corn_(maize)_healthy\\1a11d38d-4.jpg")
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x),axis=1)
index=[ 'Apple_black_rot', 'Apple_healthy', 'Corn_(maize)_Northern_Leaf_Blight', 'Corn_(maize)_healthy', 'Peach_Bacterial_spot', 'Peach_healthy' ]
index[y[0]]
Out[51]: 'Corn_(maize)_healthy'

```

Figure 9 Test results 2 for fruit disease detection using CNN

```

jupyter Project-Fruit disease detection -IBM Last Checkpoint: Yesterday at 2:08 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) C

In [79]: software_space_uid=client.software_specifications.get_uid_by_name('tensorflow-PT2.1-py3.8')

In [80]: software_space_uid

Out[80]: 'ac9c798-8974-5d2f-a657-ce86e86d7ad'

In [81]: !tar -xvf Project.tar.gz Fruits.hs
# Fruits.hs

In [82]: model_details=client.repository.store_model(model='Project.tar.gz',
                                                    meta_props={
                                                        client.repository.ModelMetaNames.NAME:'Fruit disease detection',
                                                        client.repository.ModelMetaNames.TYPE:'tensorflow 2.7',
                                                        client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_uid
                                                    })

In [86]: model_id=client.repository.get_model_id(model_details)

In [87]: model_id

Out[87]: '8cb7eab0-879d-4119-8b2f-68a8bde6e01a'

In [88]: client.repository.download(model_id,'fruits.tar.gz')
Successfully saved model content to file: 'Fruits.tar.gz'

Out[88]: 'C:\\Users\\admin\\Downloads\\Project\\Fruits.tar.gz'

```

Figure 10 IBM deployment in local system

```

jupyter Deployment Fruit Disease model Last Checkpoint: Yesterday at 2:08 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Connected Trusted Python 3 (ipykernel) 15

In [2]: from ibm_watson_machine_learning import APIClient
        uwl_credentials={
            "url":"https://us-south.ml.cloud.ibm.com",
            "apikey":"VMg0fYLRz07iaFVWmECS7k2Wfo-3mS-6Agr3ixV3c"
        }

In [3]: client=APIClient(uwl_credentials)

In [4]: def guid_space_name(client,Fruit_disease_deploy):
        space=client.spaces.get_details()
        return(next(item for item in space['resources'] if item['entity']['name']==Fruit_disease_deploy)['metadata']['id'])

In [5]: space_uid=guid_space_name(client,'Fruit_disease_deploy')
        print(space_uid)
2bdb3278-a38e-45d1-8363-a72118439743

In [6]: client.set_default_space(space_uid)

Out[6]: 'SUCCESS'

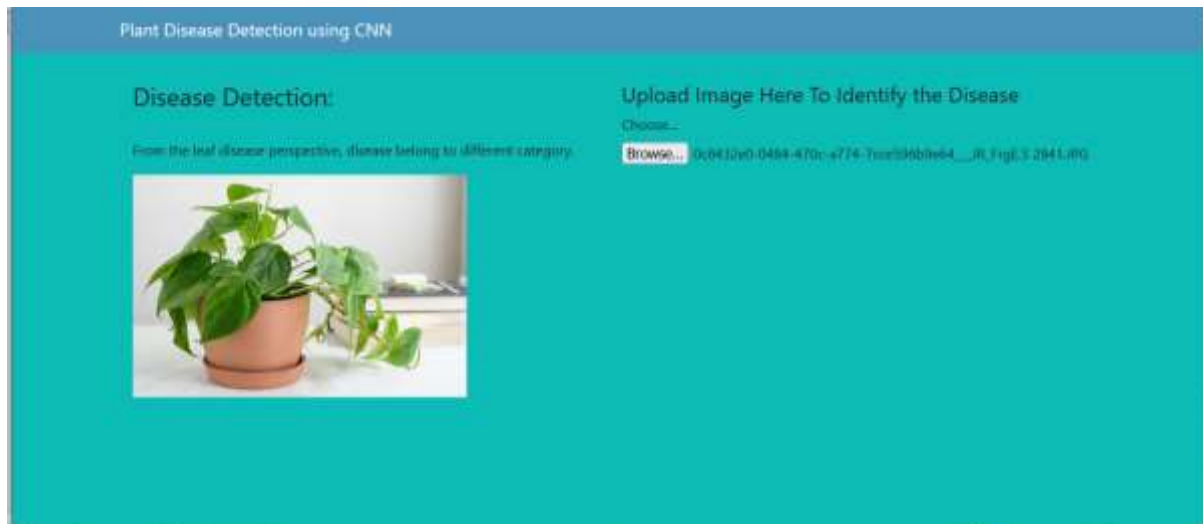
In [7]: client.repository.download('8cb7eab0-879d-4119-8b2f-68a8bde6e01a','fruits.tar.gz')
Successfully saved model content to file: 'fruits.tar.gz'

Out[7]: 'C:\\Users\\admin\\Downloads\\Flask_Deployment\\Fruits.tar.gz'

```

Select items to perform actions on them			Upload	More
<input type="checkbox"/>	Name	Last Modified	File size	
<input type="checkbox"/>	Fruit	15 minutes ago		
<input type="checkbox"/>	Fruits	6 hours ago		
<input checked="" type="checkbox"/>	Deployment Fruit Disease model app	Running 6 hours ago	5.35 KB	
<input type="checkbox"/>	Fruits.hs	a day ago	111 MB	
<input type="checkbox"/>	Fruits.tar.gz	6 hours ago	81.1 KB	

Figure 11 Flask deployment in local system



**Figure 12 Html deployment of plant disease detection using CNN**