

# SPRINT-1

## TRAINING THE DATASETS:

### FRUIT DATASET:

```
jupyter Fruit-Training (autosaved) Python 3 (ipykernel) Logout

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

In [3]: import numpy as np
import tensorflow as tf
from tensorflow.keras import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Convolution2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
import matplotlib.pyplot as plt

In [4]: from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale = 1./255, shear_range = 0.2, zoom_range = 0.2, horizontal_flip = True)
test_datagen = ImageDataGenerator(rescale = 1)

In [26]: x_train = train_datagen.flow_from_directory('fruit-dataset/fruit-dataset/train', target_size = (128,128), batch_size = 32, class_mode = 'categorical')
x_test = test_datagen.flow_from_directory('fruit-dataset/fruit-dataset/test', target_size = (128,128), batch_size = 32, class_mode = 'categorical')

Found 5384 images belonging to 6 classes.
Found 1686 images belonging to 6 classes.

In [7]: model=Sequential()

In [8]: model.add(Convolution2D(32,(3,3),input_shape = (128,128,3),activation = 'relu'))
```

```
jupyter Fruit-Training (autosaved) Python 3 (ipykernel) Logout

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

In [10]: model.add(Flatten())

In [20]: model.add(Dense(units = 40 ,kernel_initializer = 'uniform',activation = 'relu'))
model.add(Dense(units = 20 ,kernel_initializer = 'random_uniform',activation = 'relu'))
model.add(Dense(units = 6 ,activation = 'softmax',kernel_initializer = 'random_uniform'))

In [21]: model.compile(loss = 'categorical_crossentropy',optimizer = "adam",metrics = ["accuracy"])

In [23]: model.fit_generator(x_train, steps_per_epoch = 168, epochs = 3, validation_data = x_test, validation_steps = 52)

C:\Users\sriya\AppData\Local\Temp\ipykernel_5172\4242861645.py:1: UserWarning: "Model.fit_generator" is deprecated and will be removed in a future version. Please use "Model.fit", which supports generators.
  model.fit_generator(x_train, steps_per_epoch = 168, epochs = 3, validation_data = x_test, validation_steps = 52)

Epoch 1/3
168/168 [=====] - 41s 230ms/step - loss: 0.9263 - accuracy: 0.6532 - val_loss: 100.6599 - val_accuracy: 0.7326
Epoch 2/3
168/168 [=====] - 36s 213ms/step - loss: 0.4262 - accuracy: 0.8436 - val_loss: 196.0316 - val_accuracy: 0.7043
Epoch 3/3
168/168 [=====] - 37s 218ms/step - loss: 0.3098 - accuracy: 0.8881 - val_loss: 215.5389 - val_accuracy: 0.7145

Out[23]: <keras.callbacks.History at 0x1afc5b7caf0>

In [24]: model.save("fruit.h5")
```

## VEGETABLE DATASET:

```
jupyter Vegetable-training (autosaved) Logout
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

In [1]: import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Convolution2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
import matplotlib.pyplot as plt

In [2]: from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale = 1./255, shear_range = 0.2, zoom_range = 0.2, horizontal_flip = True)
test_datagen = ImageDataGenerator(rescale = 1)

In [6]: x_train = train_datagen.flow_from_directory('Veg-dataset/Veg-dataset/train_set', target_size = (128,128), batch_size = 32, class_mode = 'categorical')
x_test = test_datagen.flow_from_directory('Veg-dataset/Veg-dataset/test_set', target_size = (128,128), batch_size = 32, class_mode = 'categorical')

Found 11386 images belonging to 9 classes.
Found 3416 images belonging to 9 classes.

In [7]: model=Sequential()

In [8]: model.add(Convolution2D(32,(3,3),input_shape = (128,128,3),activation = 'relu'))

In [9]: model.add(MaxPooling2D(pool_size = (2,2)))
```

```
jupyter Vegetable-training (autosaved) Logout
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

In [9]: model.add(MaxPooling2D(pool_size = (2,2)))

In [10]: model.add(Flatten())

In [12]: model.add(Dense(units = 300 ,kernel_initializer = 'uniform',activation = 'relu'))
model.add(Dense(units = 150 ,kernel_initializer = 'uniform',activation = 'relu'))
model.add(Dense(units = 75 ,activation = 'relu',kernel_initializer = 'uniform'))
model.add(Dense(units = 9 ,activation = 'softmax',kernel_initializer = 'uniform'))

In [13]: model.compile(loss = 'categorical_crossentropy',optimizer = "adam",metrics = ["accuracy"])

In [14]: model.fit_generator(x_train, steps_per_epoch = 89, epochs = 20,validation_data = x_test, validation_steps = 27)

Epoch 1/20
C:\Users\sriya\AppData\Local\Temp\ipykernel_25452\1643366923.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
model.fit_generator(x_train, steps_per_epoch = 89, epochs = 20,validation_data = x_test, validation_steps = 27)
89/89 [=====] - 35s 387ms/step - loss: 2.1018 - accuracy: 0.1791 - val_loss: 101.1147 - val_accuracy: 0.1273
Epoch 2/20
89/89 [=====] - 34s 380ms/step - loss: 2.0854 - accuracy: 0.1692 - val_loss: 34.7438 - val_accuracy: 0.1782
Epoch 3/20
89/89 [=====] - 34s 379ms/step - loss: 2.0405 - accuracy: 0.1847 - val_loss: 62.4292 - val_accuracy: 0.2720
Epoch 4/20
```

```
jupyter Vegetable-training (autosaved) Logout
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

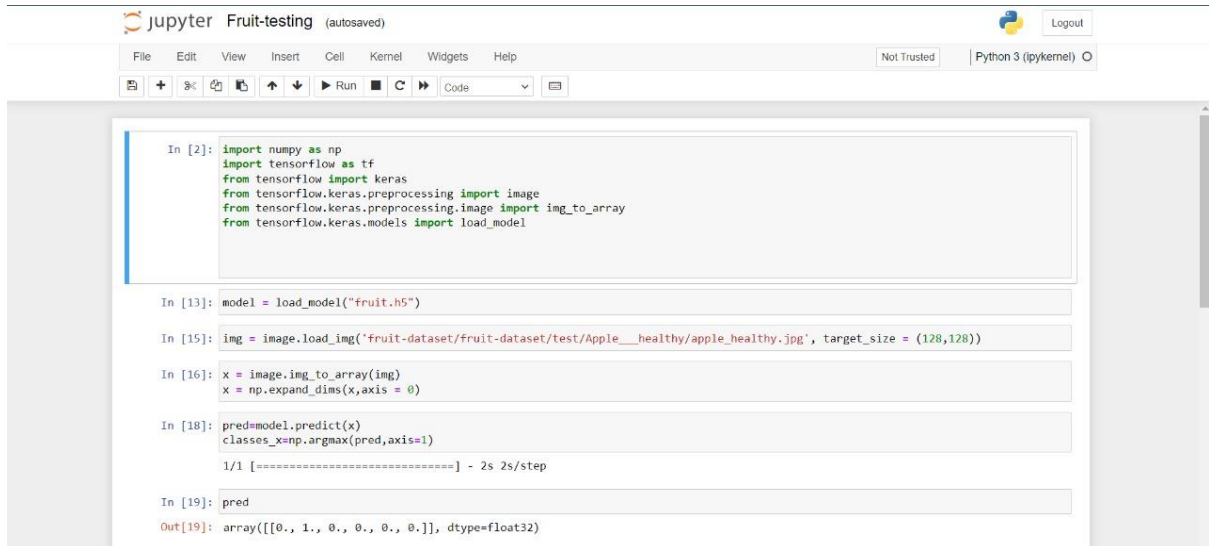
Epoch 5/20
89/89 [=====] - 44s 421ms/step - loss: 1.0551 - accuracy: 0.7744 - val_loss: 0.7300 - val_loss: 723.0001 - val_accuracy: 0.4016
Epoch 14/20
89/89 [=====] - 49s 548ms/step - loss: 0.6738 - accuracy: 0.7658 - val_loss: 964.8778 - val_accuracy: 0.3785
Epoch 15/20
89/89 [=====] - 42s 475ms/step - loss: 0.7187 - accuracy: 0.7581 - val_loss: 600.6293 - val_accuracy: 0.4225
Epoch 16/20
89/89 [=====] - 47s 530ms/step - loss: 0.6361 - accuracy: 0.7840 - val_loss: 759.7274 - val_accuracy: 0.3889
Epoch 17/20
89/89 [=====] - 43s 485ms/step - loss: 0.6651 - accuracy: 0.7791 - val_loss: 864.9741 - val_accuracy: 0.3299
Epoch 18/20
89/89 [=====] - 44s 500ms/step - loss: 0.5702 - accuracy: 0.8072 - val_loss: 968.4995 - val_accuracy: 0.4190
Epoch 19/20
89/89 [=====] - 39s 436ms/step - loss: 0.6014 - accuracy: 0.7960 - val_loss: 1045.8334 - val_accuracy: 0.3519
Epoch 20/20
89/89 [=====] - 40s 454ms/step - loss: 0.5043 - accuracy: 0.8329 - val_loss: 1101.1173 - val_accuracy: 0.4097

Out[14]: <keras.callbacks.History at 0x24dc570b80>

In [15]: model.save("vegetable.h5")
```

# TESTING THE DATASETS:

## FRUIT DATASET:



The image shows a Jupyter Notebook titled "Fruit-testing" with a toolbar at the top containing icons for file operations, editing, and running code. The notebook contains several code cells. The first cell (In [2]) imports necessary libraries: numpy, tensorflow, keras, image, img\_to\_array, and load\_model. The second cell (In [13]) loads a model named "fruit.h5". The third cell (In [15]) loads an image from the "fruit-dataset/test/Apple\_\_\_healthy/apple\_healthy.jpg" file with a target size of (128,128). The fourth cell (In [16]) converts the image to a numpy array and expands its dimensions. The fifth cell (In [18]) uses the loaded model to predict the class of the image. The output of this cell is a progress bar showing 1/1 completed in 2 seconds. The sixth cell (In [19]) prints the prediction result, which is an array of class probabilities.

```
In [2]: import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model

In [13]: model = load_model("fruit.h5")

In [15]: img = image.load_img('fruit-dataset/fruit-dataset/test/Apple___healthy/apple_healthy.jpg', target_size = (128,128))

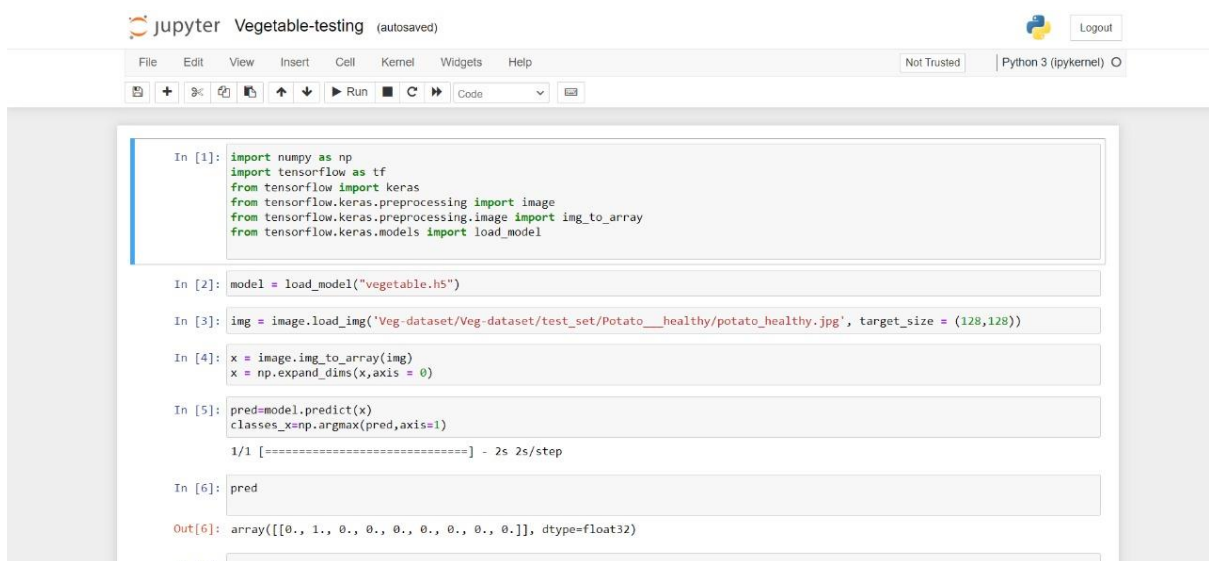
In [16]: x = image.img_to_array(img)
x = np.expand_dims(x,axis = 0)

In [18]: pred=model.predict(x)
classes_x=np.argmax(pred,axis=1)

1/1 [=====] - 2s 2s/step

In [19]: pred
Out[19]: array([[0., 1., 0., 0., 0., 0.]], dtype=float32)
```

## VEGETABLE DATASET:



The image shows a Jupyter Notebook titled "Vegetable-testing" with a toolbar at the top containing icons for file operations, editing, and running code. The notebook contains several code cells. The first cell (In [1]) imports necessary libraries: numpy, tensorflow, keras, image, img\_to\_array, and load\_model. The second cell (In [2]) loads a model named "vegetable.h5". The third cell (In [3]) loads an image from the "Veg-dataset/Veg-dataset/test\_set/Potato\_\_\_healthy/potato\_healthy.jpg" file with a target size of (128,128). The fourth cell (In [4]) converts the image to a numpy array and expands its dimensions. The fifth cell (In [5]) uses the loaded model to predict the class of the image. The output of this cell is a progress bar showing 1/1 completed in 2 seconds. The sixth cell (In [6]) prints the prediction result, which is an array of class probabilities.

```
In [1]: import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model

In [2]: model = load_model("vegetable.h5")

In [3]: img = image.load_img('Veg-dataset/Veg-dataset/test_set/Potato___healthy/potato_healthy.jpg', target_size = (128,128))

In [4]: x = image.img_to_array(img)
x = np.expand_dims(x,axis = 0)

In [5]: pred=model.predict(x)
classes_x=np.argmax(pred,axis=1)

1/1 [=====] - 2s 2s/step

In [6]: pred
Out[6]: array([[0., 1., 0., 0., 0., 0., 0., 0., 0.]], dtype=float32)
```