SPRINT 2

Date	: 19 NOVEMBER 2022
Team ID	: PNT2022TMID52301
Project Name	: FERTILIZERS RECOMMENDATION SYSTEM FOR DISEASE PREDICTION

Import The Libraries

```
from keras.models import
Sequential from keras.layers import
Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
```

Initializing The Model

Keras has 2 ways to define a neural network:

- Sequential
- Function API

The Sequential class is used to define linear initializations of network layers which then, collectively, constitutea model.
We will use the Sequential constructor to create a model, which will then have layers added to it using the add ()method.Now, will initialize our model.
Initialize the neural network layer by creating a reference/object to the Sequential class.

```
model=Sequential()
```

ADD CNN Layers

We will be adding three layers for CNN

- Convolution layer
- Pooling layer
- Flattening layer

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

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

model.add(Flatten())
```

Add Dense Layers

This step is to add a dense layer (output layer) where you will be specifying the number of classes your dependent variable has, activation function, and weight initializer as the arguments. We use the add () method toadd dense layers. the output dimensions here is 6

```
model.add(Dense(40, 'relu'))

model.add(Dense(20, 'relu'))

model.add(Dense(6,  'softmax', ))
```

Train And Save The Model

Compile the model

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

Model.Summary()

Can be used to see all parameters and shapes in each layer in our models.

```
model.summary()
```

Fit and save the model

```
model.fit(x_train,epochs=20,steps_per_epoch=89,validation_data = x_test,
validation_steps = 27)
```

The weights are to be saved for future use. The weights are saved in as .h5 file using save().

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

Output:

```
____________________________________________________________________
___________
 Layer (type)                    Output Shape                Param #
==========================================================
================
 conv2d (Conv2D)              (None, 126, 126, 32)          896


 max_pooling2d (
MaxPooling2D )                (None, 63, 63, 32)             0



 flatten (Flatten)            (None, 127008)                 0
 dense (Dense)              (None, 300)              38102700

 dense_1 (Dense)             (None, 150)               45150

 dense_2 (Dense)             (None, 75)                11325

 dense_3 (Dense)             (None, 9)                  684

==================================================
===============
Total params: 38,160,755
Trainable params: 38,160,755
Non-trainable params: 0
_________________________________________________________
_________
Epoch 1/20
89/89 [==============================] - 52s 576ms/step -
loss: 2.4956 - accuracy: 0.2686 - val_loss: 246.3766 - val_accuracy:
0.3426
Epoch 2/20
89/89 [==============================] - 44s 498ms/step -
loss: 1.2983 - accuracy: 0.5468 - val_loss: 651.4410 - val_accuracy:
0.2894
Epoch 3/20
```

```
89/89 [==============================] - 42s 469ms/step -
loss: 0.9406 - accuracy: 0.6735 - val_loss: 1125.0737 - val_accuracy:0.2442

Epoch 4/20

89/89 [==============================] - 39s 440ms/step -
loss: 0.7779 - accuracy: 0.7300 - val_loss: 1022.7507 - val_accuracy:0.2847

Epoch 5/20

89/89 [==============================] - 41s 462ms/step -
loss: 0.7470 - accuracy: 0.7465 - val_loss: 1396.1002 - val_accuracy:0.2581

Epoch 6/20

89/89 [==============================] - 45s 510ms/step -
loss: 0.6462 - accuracy: 0.7718 - val_loss: 1383.3610 - val_accuracy:0.2616

Epoch 7/20

89/89 [==============================] - 34s 387ms/step -
loss: 0.5867 - accuracy: 0.7928 - val_loss: 1626.8010 - val_accuracy:0.1771

Epoch 8/20

89/89 [==============================] - 45s 504ms/step -
loss: 0.5461 - accuracy: 0.8058 - val_loss: 1733.9170 - val_accuracy:0.2014

Epoch 9/20

89/89 [==============================] - 55s 617ms/step -
loss: 0.4965 - accuracy: 0.8283 - val_loss: 2105.0442 - val_accuracy:0.2523

Epoch 10/20
```

```
89/89 [==============================] - 55s 617ms/step -
loss: 0.5316 - accuracy: 0.8125 - val_loss: 1585.0485 - val_accuracy:0.2766

Epoch 11/20

89/89 [==============================] - 52s 577ms/step -
loss: 0.5039 - accuracy: 0.8258 - val_loss: 1588.1725 - val_accuracy:0.3032

Epoch 12/20

89/89 [==============================] - 51s 571ms/step -
loss: 0.4196 - accuracy: 0.8546 - val_loss: 2111.2288 - val_accuracy:0.2824

Epoch 13/20

89/89 [==============================] - 52s 582ms/step -
loss: 0.4402 - accuracy: 0.8504 - val_loss: 1728.3689 - val_accuracy:0.2824

Epoch 14/20

89/89 [==============================] - 51s 568ms/step -
loss: 0.4035 - accuracy: 0.8560 - val_loss: 1953.9325 - val_accuracy:0.2477

Epoch 15/20

89/89 [==============================] - 52s 578ms/step -
loss: 0.3994 - accuracy: 0.8606 - val_loss: 1739.5107 - val_accuracy:0.2894

Epoch 16/20

89/89 [==============================] - 51s 575ms/step -
loss: 0.3509 - accuracy: 0.8754 - val_loss: 1912.0873 - val_accuracy:0.3252

Epoch 17/20
```

```
89/89 [==============================] - 50s 561ms/step -
loss: 0.3818 - accuracy: 0.8606 - val_loss: 1777.9532 - val_accuracy:0.3125
Epoch 18/20
89/89 [==============================] - 50s 565ms/step -
loss: 0.3416 - accuracy: 0.8810 - val_loss: 2017.1232 - val_accuracy:0.2801
Epoch 19/20
89/89 [==============================] - 51s 574ms/step -
loss: 0.3515 - accuracy: 0.8743 - val_loss: 1423.0455 - val_accuracy:0.3530
Epoch 20/20
89/89 [==============================] - 50s 560ms/step -
loss: 0.3514 - accuracy: 0.8761 - val_loss: 1466.1351 - val_accuracy:0.3218
```

Model Building For Vegetable Disease Prediction

```
from keras.models import
Sequentialfrom keras.layers import
Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import 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.add(Dense(300, 'relu'))

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

model.add(Dense(75, 'relu'))

model.add(Dense(9,  'softmax',

))

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

model.summary()

model.fit(x_train,epochs=20,steps_per_epoch=89,validation_data
=x_test, validation_steps = 27)

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

Output:

```
C:\Users\mumma\PycharmProjects\plant_disease_detection\venv\Sc
ripts\python.exe
C:\Users\mumma\PycharmProjects\plant_disease_detection\veg_m
odel.py

Found 11386 images belonging to 9 classes.

Found 3416 images belonging to 9 classes.

Model: "sequential"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 126, 126, 32)      896
 max_pooling2d
(MaxPooling2D)                 (None, 63, 63, 32)                0



 flatten (Flatten)              (None, 127008)                    0

 dense (Dense)                  (None, 300)                   38102700

 dense_1 (Dense)                (None, 150)                     45150

 dense_2 (Dense)                (None, 75)                      11325

 dense_3 (Dense)                (None, 9)                        684

=======================================================
================
Total params: 38,160,755
Trainable params: 38,160,755
Non-trainable params: 0
_________________________________________________________________
__________
Epoch 1/20
89/89 [==============================] - 42s 459ms/step -
loss: 2.3594 - accuracy: 0.3153 - val_loss: 194.4259 - val_accuracy:
0.4931
```

```
Epoch 2/20

89/89 [==============================] - 48s 540ms/step -
loss: 1.2180 - accuracy: 0.5829 - val_loss: 532.0151 - val_accuracy:
0.3472

Epoch 3/20

89/89 [==============================] - 47s 527ms/step -
loss: 0.8772 - accuracy: 0.6822 - val_loss: 699.0117 - val_accuracy:
0.3669

Epoch 4/20

89/89 [==============================] - 47s 531ms/step -
loss: 0.7295 - accuracy: 0.7468 - val_loss: 1188.0234 - val_accuracy:0.2616

Epoch 5/20

89/89 [==============================] - 46s 512ms/step -
loss: 0.6464 - accuracy: 0.7738 - val_loss: 1339.5924 - val_accuracy:0.3241

Epoch 6/20

89/89 [==============================] - 45s 510ms/step -
loss: 0.5844 - accuracy: 0.7949 - val_loss: 1505.8514 - val_accuracy:0.2477

Epoch 7/20

89/89 [==============================] - 46s 514ms/step -
loss: 0.5777 - accuracy: 0.7960 - val_loss: 1878.0940 - val_accuracy:0.2407

Epoch 8/20

89/89 [==============================] - 45s 509ms/step -
loss: 0.5296 - accuracy: 0.8146 - val_loss: 1235.0643 - val_accuracy:0.3264

Epoch 9/20
```

```
89/89 [================================] - 46s 518ms/step -
loss: 0.4724 - accuracy: 0.8332 - val_loss: 1428.7808 - val_accuracy:0.2662

Epoch 10/20

89/89 [================================] - 41s 460ms/step -
loss: 0.4665 - accuracy: 0.8350 - val_loss: 1449.1967 - val_accuracy:0.2951

Epoch 11/20

89/89 [================================] - 44s 491ms/step -
loss: 0.4025 - accuracy: 0.8592 - val_loss: 1373.3068 - val_accuracy:0.2581

Epoch 12/20

89/89 [================================] - 45s 508ms/step -
loss: 0.4034 - accuracy: 0.8574 - val_loss: 1854.9781 - val_accuracy:0.2292

Epoch 13/20

89/89 [================================] - 49s 551ms/step -
loss: 0.3759 - accuracy: 0.8648 - val_loss: 2173.3513 - val_accuracy:0.2188

Epoch 14/20

89/89 [================================] - 48s 539ms/step -
loss: 0.3696 - accuracy: 0.8687 - val_loss: 1862.3169 - val_accuracy:0.2431

Epoch 15/20

89/89 [================================] - 49s 555ms/step -
loss: 0.3470 - accuracy: 0.8834 - val_loss: 1886.7236 - val_accuracy:0.2245

Epoch 16/20
```

```
89/89 [==============================] - 48s 541ms/step -
loss: 0.2935 - accuracy: 0.8968 - val_loss: 2358.3191 - val_accuracy:0.2789

Epoch 17/20

89/89 [==============================] - 48s 544ms/step -
loss: 0.2905 - accuracy: 0.9010 - val_loss: 2385.9612 - val_accuracy:0.2338

Epoch 18/20

89/89 [==============================] - 45s 510ms/step -
loss: 0.3274 - accuracy: 0.8820 - val_loss: 2428.2671 - val_accuracy:0.2002

Epoch 19/20

89/89 [==============================] - 43s 482ms/step -
loss: 0.3007 - accuracy: 0.8975 - val_loss: 2702.1655 - val_accuracy:0.2280

Epoch 20/20

89/89 [==============================] - 35s 394ms/step -
loss: 0.3242 - accuracy: 0.8883 - val_loss: 2831.8833 - val_accuracy:0.1794
```