













# MODEL BUILDING FOR VEGETABLE DISEASE PREDICTION

Team ID	PNT2022TMID51209
Project Name	Project - Fertilizers Recommendation System For Disease Prediction

 jupyter Model Building for vegetable disease prediction Last Checkpoint: 2 minutes ago (autosaved)  Logout

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

```
In [1]: #Image Augmentation
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 [2]: x_train=train_datagen.flow_from_directory(r'C:\Users\ELCOT\Desktop\IBM\Dataset Plant Disease\Veg-dataset\Veg-dataset\train_set', target_size=(180, 180))
x_test=test_datagen.flow_from_directory(r'C:\Users\ELCOT\Desktop\IBM\Dataset Plant Disease\Veg-dataset\Veg-dataset\test_set', target_size=(180, 180))

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

In [3]: #Import Library Files
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

In [4]: #Initialize the sequential model
model=Sequential()

In [5]: # Add the convolution layer
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))

In [6]: # Add the max pooling layer
model.add(MaxPooling2D(pool_size=(2,2)))
```

```
In [7]: # Add the Flatten Layer
model.add(Flatten())

In [8]: # Add the Hidden Layer 1
model.add(Dense(units=300, kernel_initializer='uniform', activation='relu'))

In [9]: # Add the Hidden Layer 2
model.add(Dense(units=150, kernel_initializer='uniform', activation='relu'))

In [10]: # Add the Hidden Layer 3
model.add(Dense(units=75, kernel_initializer='uniform', activation='relu'))


In [11]: # Add the Output Layer
model.add(Dense(units=9, kernel_initializer='uniform', activation='softmax'))

In [12]: # Compiling the model
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

In [13]: # Train model
model.fit(x_train, steps_per_epoch=89, epochs=20, validation_data=x_test, validation_steps=27)


Epoch 1/20
89/89 [=====] - 66s 690ms/step - loss: 2.1881 - accuracy: 0.1685 - val_loss: 194.3618 - val_accuracy: 0.1296
Epoch 2/20
89/89 [=====] - 59s 663ms/step - loss: 2.1182 - accuracy: 0.1236 - val_loss: 109.5233 - val_accuracy: 0.1296
Epoch 3/20
89/89 [=====] - 67s 755ms/step - loss: 2.0954 - accuracy: 0.1180 - val_loss: 80.7805 - val_accuracy: 0.1296
```

```
Epoch 4/20
89/89 [=====] - 73s 817ms/step - loss: 2.1350 - accuracy: 0.1854 - val_loss: 164.9028 - val_accuracy: 0.1111
Epoch 5/20
89/89 [=====] - 69s 776ms/step - loss: 2.1272 - accuracy: 0.1798 - val_loss: 74.0293 - val_accuracy: 0.1296
Epoch 6/20
89/89 [=====] - 65s 729ms/step - loss: 2.0987 - accuracy: 0.2079 - val_loss: 94.8363 - val_accuracy: 0.2037
Epoch 7/20
89/89 [=====] - 57s 643ms/step - loss: 2.0832 - accuracy: 0.2079 - val_loss: 32.7240 - val_accuracy: 0.0926
Epoch 8/20
89/89 [=====] - 56s 628ms/step - loss: 2.0565 - accuracy: 0.2584 - val_loss: 134.1384 - val_accuracy: 0.2963
Epoch 9/20
89/89 [=====] - 56s 628ms/step - loss: 2.1103 - accuracy: 0.1517 - val_loss: 61.7429 - val_accuracy: 0.1481
Epoch 10/20
89/89 [=====] - 55s 622ms/step - loss: 2.0565 - accuracy: 0.2416 - val_loss: 121.7573 - val_accuracy: 0.1481
Epoch 11/20
89/89 [=====] - 53s 598ms/step - loss: 2.1168 - accuracy: 0.1854 - val_loss: 30.3666 - val_accuracy: 0.1667
Epoch 12/20
89/89 [=====] - 57s 635ms/step - loss: 2.0713 - accuracy: 0.2079 - val_loss: 56.4360 - val_accuracy: 0.2778
Epoch 13/20
89/89 [=====] - 59s 665ms/step - loss: 1.9673 - accuracy: 0.2416 - val_loss: 66.7629 - val_accuracy: 0.3333
Epoch 14/20
89/89 [=====] - 55s 612ms/step - loss: 1.9854 - accuracy: 0.3202 - val_loss: 95.1238 - val_accuracy: 0.3148
```

 jupyter










Model Building for vegetable disease prediction

Last Checkpoint: 5 minutes ago (autosaved)

 Logout

FileEditViewInsertCellKernelWidgetsHelp

TrustedPython 3 (ipykernel)




Code

```
Epoch 15/20
89/89 [=====] - 55s 614ms/step - loss: 1.7150 - accuracy: 0.3933 - val_loss: 148.8870 - val_accuracy: 0.2037
Epoch 16/20
89/89 [=====] - 56s 624ms/step - loss: 1.6605 - accuracy: 0.3708 - val_loss: 292.1395 - val_accuracy: 0.2222
Epoch 17/20
89/89 [=====] - 55s 611ms/step - loss: 1.8242 - accuracy: 0.3427 - val_loss: 156.9938 - val_accuracy: 0.3704
Epoch 18/20
89/89 [=====] - 56s 628ms/step - loss: 1.7093 - accuracy: 0.3876 - val_loss: 161.5939 - val_accuracy: 0.2778
Epoch 19/20
89/89 [=====] - 66s 745ms/step - loss: 1.6356 - accuracy: 0.3933 - val_loss: 143.6697 - val_accuracy: 0.3704
Epoch 20/20
89/89 [=====] - 64s 714ms/step - loss: 1.8686 - accuracy: 0.3427 - val_loss: 70.6791 - val_accuracy: 0.4259


Out[13]: <keras.callbacks.History at 0x259831a6be0>

In [16]: # Save model
model.save(r'C:\Users\Elcot\project\flask\uploads\vegetable.h5')
```

 jupyter










Model Building for vegetable disease prediction

Last Checkpoint: 6 minutes ago (autosaved)

 Logout

FileEditViewInsertCellKernelWidgetsHelp

TrustedPython 3 (ipykernel)



Code

```
In [16]: # Save model
model.save(r'C:\Users\Elcot\project\flask\uploads\vegetable.h5')

In [17]: model.summary()

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
_____
```