

```
In [1]: import numpy as np
import pickle
import cv2
from os import listdir
from sklearn.preprocessing import LabelBinarizer
from keras.models import Sequential
# from keras.layers.normalization import BatchNormalization
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.layers.core import Activation, Flatten, Dropout, Dense
from keras import backend as K
from keras.preprocessing.image import ImageDataGenerator

from keras.preprocessing import image
from keras.preprocessing.image import img_to_array
from sklearn.preprocessing import MultiLabelBinarizer
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import tensorflow
```

```
In [3]: import random
random.seed(123)
```

```
In [16]: import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
client_c7221d8fc30948e893df788ae165e488 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='062Yj24-tgVj3z94ag0kvgPGK-hGMewhW05ozMYswmxv',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

streaming_body_1 = client_c7221d8fc30948e893df788ae165e488.get_object(Bucket='healthcare1-c'

# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

```
In [17]: ...
# "Insert StreamingBody object" generated code
...
from io import BytesIO
import zipfile

zip_ref = zipfile.ZipFile(BytesIO(streaming_body_1.read()), 'r')
filepath = zip_ref.namelist()
#print(zip_ref.namelist())
#zip_ref.close()
```

```
In [18]: for path in filepath:  
    zip_ref.extract(path)  
  
In [19]: pwd  
Out[19]: '/home/wsuser/work'  
  
In [20]: TRAINING_DIR='/home/wsuser/work/fruit-dataset/train/'  
TEST_DIR = '/home/wsuser/work/fruit-dataset/test/'  
  
In [21]: from keras.preprocessing.image import ImageDataGenerator  
train_datagen = ImageDataGenerator(shear_range = 0.2,  
                                   zoom_range = 0.2,  
                                   horizontal_flip=True,  
                                   rescale=1./255,  
                                   validation_split=0.2  
)  
  
In [22]: x_train =train_datagen.flow_from_directory(TRAINING_DIR,  
                                                target_size=(128,128),  
                                                batch_size=32,  
                                                class_mode='categorical')  
x_test =train_datagen.flow_from_directory(TEST_DIR,  
                                         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 [23]: 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 [27]: from keras import backend as K  
K.clear_session()  
model = Sequential()  
model.add(Convolution2D(32, (3,3), activation='relu', input_shape=(128,128,3)))  
model.add(MaxPooling2D(2, 2))  
  
model.add(Convolution2D(32, (3,3), activation='relu', input_shape=(128,128,3)))  
model.add(MaxPooling2D(2, 2))  
  
model.add(Flatten())  
model.add(Dense(40, activation='relu'))  
model.add(Dense(20, activation='relu'))  
model.add(Dense(6, activation='softmax'))  
  
In [28]: model.compile(loss='categorical_crossentropy', optimizer = 'adam', metrics=['accuracy'])  
  
In [29]: CNN_model=model.fit_generator(x_train,steps_per_epoch = 89, epochs=30,validation_data=x_te  
/tmp/wsuser/ipykernel_164/1272244427.py:1: UserWarning: `Model.fit_generator` is deprecated  
and will be removed in a future version. Please use `Model.fit`, which supports generators.  
    CNN_model=model.fit_generator(x_train,steps_per_epoch = 89, epochs=30,validation_data=x_t  
est, validation_steps = 27).history
```

Epoch 1/30
89/89 [=====] - 42s 469ms/step - loss: 0.9781 - accuracy: 0.6615 -
val_loss: 0.4807 - val_accuracy: 0.8553
Epoch 2/30
89/89 [=====] - 42s 470ms/step - loss: 0.4456 - accuracy: 0.8529 -
val_loss: 0.3284 - val_accuracy: 0.8877
Epoch 3/30
89/89 [=====] - 41s 461ms/step - loss: 0.3462 - accuracy: 0.8817 -
val_loss: 0.2735 - val_accuracy: 0.9132
Epoch 4/30
89/89 [=====] - 42s 466ms/step - loss: 0.2556 - accuracy: 0.9077 -
val_loss: 0.2437 - val_accuracy: 0.9155
Epoch 5/30
89/89 [=====] - 42s 468ms/step - loss: 0.2343 - accuracy: 0.9182 -
val_loss: 0.2981 - val_accuracy: 0.8958
Epoch 6/30
89/89 [=====] - 42s 475ms/step - loss: 0.2312 - accuracy: 0.9196 -
val_loss: 0.3673 - val_accuracy: 0.8750
Epoch 7/30
89/89 [=====] - 43s 479ms/step - loss: 0.2436 - accuracy: 0.9133 -
val_loss: 0.2384 - val_accuracy: 0.9271
Epoch 8/30
89/89 [=====] - 42s 470ms/step - loss: 0.1962 - accuracy: 0.9350 -
val_loss: 0.1534 - val_accuracy: 0.9421
Epoch 9/30
89/89 [=====] - 41s 463ms/step - loss: 0.1452 - accuracy: 0.9508 -
val_loss: 0.1281 - val_accuracy: 0.9618
Epoch 10/30
89/89 [=====] - 42s 468ms/step - loss: 0.1541 - accuracy: 0.9442 -
val_loss: 0.1798 - val_accuracy: 0.9329
Epoch 11/30
89/89 [=====] - 41s 458ms/step - loss: 0.1813 - accuracy: 0.9394 -
val_loss: 0.2097 - val_accuracy: 0.9213
Epoch 12/30
89/89 [=====] - 41s 463ms/step - loss: 0.1413 - accuracy: 0.9501 -
val_loss: 0.1276 - val_accuracy: 0.9537
Epoch 13/30
89/89 [=====] - 42s 468ms/step - loss: 0.1273 - accuracy: 0.9557 -
val_loss: 0.1606 - val_accuracy: 0.9491
Epoch 14/30
89/89 [=====] - 41s 462ms/step - loss: 0.1405 - accuracy: 0.9494 -
val_loss: 0.2057 - val_accuracy: 0.9225
Epoch 15/30
89/89 [=====] - 41s 460ms/step - loss: 0.1060 - accuracy: 0.9642 -
val_loss: 0.1081 - val_accuracy: 0.9630
Epoch 16/30
89/89 [=====] - 42s 467ms/step - loss: 0.1099 - accuracy: 0.9625 -
val_loss: 0.1306 - val_accuracy: 0.9583
Epoch 17/30
89/89 [=====] - 41s 455ms/step - loss: 0.1029 - accuracy: 0.9653 -
val_loss: 0.0969 - val_accuracy: 0.9688
Epoch 18/30
89/89 [=====] - 41s 461ms/step - loss: 0.1109 - accuracy: 0.9614 -
val_loss: 0.1276 - val_accuracy: 0.9537
Epoch 19/30
89/89 [=====] - 42s 467ms/step - loss: 0.0783 - accuracy: 0.9747 -
val_loss: 0.1166 - val_accuracy: 0.9653
Epoch 20/30
89/89 [=====] - 41s 460ms/step - loss: 0.0994 - accuracy: 0.9703 -
val_loss: 0.1127 - val_accuracy: 0.9595
Epoch 21/30

```
89/89 [=====] - 45s 500ms/step - loss: 0.0856 - accuracy: 0.9695 -  
val_loss: 0.1106 - val_accuracy: 0.9641  
Epoch 22/30  
89/89 [=====] - 41s 456ms/step - loss: 0.0813 - accuracy: 0.9688 -  
val_loss: 0.1364 - val_accuracy: 0.9583  
Epoch 23/30  
89/89 [=====] - 41s 456ms/step - loss: 0.1076 - accuracy: 0.9659 -  
val_loss: 0.0649 - val_accuracy: 0.9838  
Epoch 24/30  
89/89 [=====] - 42s 467ms/step - loss: 0.0708 - accuracy: 0.9749 -  
val_loss: 0.1342 - val_accuracy: 0.9583  
Epoch 25/30  
89/89 [=====] - 41s 457ms/step - loss: 0.0770 - accuracy: 0.9720 -  
val_loss: 0.1557 - val_accuracy: 0.9560  
Epoch 26/30  
89/89 [=====] - 41s 465ms/step - loss: 0.0914 - accuracy: 0.9712 -  
val_loss: 0.1085 - val_accuracy: 0.9664  
Epoch 27/30  
89/89 [=====] - 42s 466ms/step - loss: 0.0696 - accuracy: 0.9784 -  
val_loss: 0.0959 - val_accuracy: 0.9664  
Epoch 28/30  
89/89 [=====] - 41s 460ms/step - loss: 0.0593 - accuracy: 0.9796 -  
val_loss: 0.1119 - val_accuracy: 0.9641  
Epoch 29/30  
89/89 [=====] - 41s 462ms/step - loss: 0.0854 - accuracy: 0.9716 -  
val_loss: 0.1436 - val_accuracy: 0.9514  
Epoch 30/30  
89/89 [=====] - 41s 456ms/step - loss: 0.0669 - accuracy: 0.9789 -  
val_loss: 0.0670 - val_accuracy: 0.9757
```

```
In [30]: model.save("fruit.h5")
```

```
In [31]: streaming_body_2 = client_c7221d8fc30948e893df788ae165e488.get_object(Bucket='healthcare1-c'  
  
# Your data file was loaded into a botocore.response.StreamingBody object.  
# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities.  
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/  
# pandas documentation: http://pandas.pydata.org/
```

```
In [32]: ...  
# "Insert StreamingBody object" generated code  
...  
from io import BytesIO  
import zipfile  
  
zip_ref = zipfile.ZipFile(BytesIO(streaming_body_2.read()), 'r')  
filepath = zip_ref.namelist()
```

```
In [37]: for path in filepath:  
    zip_ref.extract(path)
```

```
In [33]: pwd
```

```
Out[33]: '/home/wsuser/work'
```

```
In [38]: TRAINING_DIR='/home/wsuser/work/Veg-dataset/train_set/'  
TEST_DIR = '/home/wsuser/work/Veg-dataset/test_set/'
```

```
In [39]: from keras.preprocessing.image import ImageDataGenerator  
train_datagen = ImageDataGenerator(shear_range = 0.2,  
                                   zoom_range = 0.2,  
                                   horizontal_flip=True,  
                                   rescale=1./255,  
                                   validation_split=0.2  
)
```

```
In [40]: x_train =train_datagen.flow_from_directory(TRAINING_DIR,  
                                                target_size=(128,128),  
                                                batch_size=32,  
                                                class_mode='categorical')  
x_test =train_datagen.flow_from_directory(TEST_DIR,  
                                         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 [41]: from keras import backend as K  
K.clear_session()  
model = Sequential()  
model.add(Convolution2D(32, (3,3), activation='relu', input_shape=(128,128,3)))  
model.add(MaxPooling2D(2, 2))  
  
model.add(Convolution2D(32, (3,3), activation='relu', input_shape=(128,128,3)))  
model.add(MaxPooling2D(2, 2))  
  
model.add(Flatten())  
model.add(Dense(40, activation='relu'))  
model.add(Dense(20, activation='relu'))  
model.add(Dense(9, activation='softmax'))
```

```
In [42]: model.compile(loss='categorical_crossentropy', optimizer = 'adam', metrics=['accuracy'])
```

```
In [43]: CNN_model=model.fit_generator(x_train,steps_per_epoch = 89, epochs=50,validation_data=x_te
```

/tmp/wsuser/ipykernel_164/1020607867.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.

```
CNN_model=model.fit_generator(x_train,steps_per_epoch = 89, epochs=50,validation_data=x_test, validation_steps = 27).history
```

```
Epoch 1/50
89/89 [=====] - 43s 476ms/step - loss: 1.9226 - accuracy: 0.2978 -
val_loss: 1.6005 - val_accuracy: 0.4294
Epoch 2/50
89/89 [=====] - 42s 472ms/step - loss: 1.2868 - accuracy: 0.5700 -
val_loss: 1.1202 - val_accuracy: 0.6262
Epoch 3/50
89/89 [=====] - 41s 464ms/step - loss: 0.9918 - accuracy: 0.6678 -
val_loss: 0.8685 - val_accuracy: 0.7106
Epoch 4/50
89/89 [=====] - 42s 469ms/step - loss: 0.8304 - accuracy: 0.7096 -
val_loss: 0.7083 - val_accuracy: 0.7523
Epoch 5/50
89/89 [=====] - 42s 467ms/step - loss: 0.7039 - accuracy: 0.7549 -
val_loss: 0.6996 - val_accuracy: 0.7523
Epoch 6/50
89/89 [=====] - 42s 469ms/step - loss: 0.6713 - accuracy: 0.7685 -
val_loss: 0.7941 - val_accuracy: 0.7199
Epoch 7/50
89/89 [=====] - 41s 457ms/step - loss: 0.6011 - accuracy: 0.7971 -
val_loss: 0.5403 - val_accuracy: 0.7998
Epoch 8/50
89/89 [=====] - 42s 470ms/step - loss: 0.5631 - accuracy: 0.8041 -
val_loss: 0.6754 - val_accuracy: 0.7581
Epoch 9/50
89/89 [=====] - 43s 477ms/step - loss: 0.5405 - accuracy: 0.8097 -
val_loss: 0.5540 - val_accuracy: 0.8021
Epoch 10/50
89/89 [=====] - 42s 473ms/step - loss: 0.4970 - accuracy: 0.8297 -
val_loss: 0.6717 - val_accuracy: 0.7616
Epoch 11/50
89/89 [=====] - 42s 468ms/step - loss: 0.5522 - accuracy: 0.8090 -
val_loss: 0.4574 - val_accuracy: 0.8472
Epoch 12/50
89/89 [=====] - 41s 458ms/step - loss: 0.4888 - accuracy: 0.8248 -
val_loss: 0.4132 - val_accuracy: 0.8472
Epoch 13/50
89/89 [=====] - 41s 461ms/step - loss: 0.4183 - accuracy: 0.8553 -
val_loss: 0.3640 - val_accuracy: 0.8704
Epoch 14/50
89/89 [=====] - 41s 464ms/step - loss: 0.4331 - accuracy: 0.8522 -
val_loss: 0.4433 - val_accuracy: 0.8333
Epoch 15/50
89/89 [=====] - 41s 461ms/step - loss: 0.4054 - accuracy: 0.8547 -
val_loss: 0.4964 - val_accuracy: 0.8322
Epoch 16/50
89/89 [=====] - 41s 458ms/step - loss: 0.3880 - accuracy: 0.8607 -
val_loss: 0.3398 - val_accuracy: 0.8843
Epoch 17/50
89/89 [=====] - 41s 459ms/step - loss: 0.3598 - accuracy: 0.8783 -
val_loss: 0.3078 - val_accuracy: 0.8889
Epoch 18/50
89/89 [=====] - 42s 473ms/step - loss: 0.3508 - accuracy: 0.8838 -
val_loss: 0.3460 - val_accuracy: 0.8819
Epoch 19/50
89/89 [=====] - 42s 471ms/step - loss: 0.3960 - accuracy: 0.8607 -
val_loss: 0.3218 - val_accuracy: 0.8843
Epoch 20/50
89/89 [=====] - 41s 463ms/step - loss: 0.3410 - accuracy: 0.8768 -
val_loss: 0.3263 - val_accuracy: 0.8935
Epoch 21/50
```

```
89/89 [=====] - 41s 463ms/step - loss: 0.3176 - accuracy: 0.8927 -  
val_loss: 0.3229 - val_accuracy: 0.8877  
Epoch 22/50  
89/89 [=====] - 42s 467ms/step - loss: 0.2959 - accuracy: 0.8966 -  
val_loss: 0.3304 - val_accuracy: 0.8657  
Epoch 23/50  
89/89 [=====] - 41s 458ms/step - loss: 0.3557 - accuracy: 0.8789 -  
val_loss: 0.3892 - val_accuracy: 0.8588  
Epoch 24/50  
89/89 [=====] - 41s 455ms/step - loss: 0.2848 - accuracy: 0.9006 -  
val_loss: 0.2490 - val_accuracy: 0.9132  
Epoch 25/50  
89/89 [=====] - 41s 459ms/step - loss: 0.3051 - accuracy: 0.8887 -  
val_loss: 0.2666 - val_accuracy: 0.9132  
Epoch 26/50  
89/89 [=====] - 41s 458ms/step - loss: 0.2604 - accuracy: 0.9131 -  
val_loss: 0.2839 - val_accuracy: 0.8924  
Epoch 27/50  
89/89 [=====] - 42s 470ms/step - loss: 0.2980 - accuracy: 0.8961 -  
val_loss: 0.2667 - val_accuracy: 0.9097  
Epoch 28/50  
89/89 [=====] - 41s 456ms/step - loss: 0.2542 - accuracy: 0.9154 -  
val_loss: 0.2778 - val_accuracy: 0.9039  
Epoch 29/50  
89/89 [=====] - 42s 473ms/step - loss: 0.2666 - accuracy: 0.9061 -  
val_loss: 0.2851 - val_accuracy: 0.8947  
Epoch 30/50  
89/89 [=====] - 41s 463ms/step - loss: 0.2591 - accuracy: 0.8999 -  
val_loss: 0.2722 - val_accuracy: 0.9144  
Epoch 31/50  
89/89 [=====] - 41s 460ms/step - loss: 0.2728 - accuracy: 0.9050 -  
val_loss: 0.2697 - val_accuracy: 0.9109  
Epoch 32/50  
89/89 [=====] - 43s 478ms/step - loss: 0.2332 - accuracy: 0.9182 -  
val_loss: 0.2541 - val_accuracy: 0.9086  
Epoch 33/50  
89/89 [=====] - 41s 465ms/step - loss: 0.2477 - accuracy: 0.9199 -  
val_loss: 0.2749 - val_accuracy: 0.8993  
Epoch 34/50  
89/89 [=====] - 41s 462ms/step - loss: 0.2250 - accuracy: 0.9192 -  
val_loss: 0.2148 - val_accuracy: 0.9190  
Epoch 35/50  
89/89 [=====] - 42s 465ms/step - loss: 0.2268 - accuracy: 0.9224 -  
val_loss: 0.2230 - val_accuracy: 0.9317  
Epoch 36/50  
89/89 [=====] - 41s 459ms/step - loss: 0.2738 - accuracy: 0.9029 -  
val_loss: 0.1904 - val_accuracy: 0.9410  
Epoch 37/50  
89/89 [=====] - 41s 460ms/step - loss: 0.2369 - accuracy: 0.9245 -  
val_loss: 0.2017 - val_accuracy: 0.9363  
Epoch 38/50  
89/89 [=====] - 41s 457ms/step - loss: 0.2114 - accuracy: 0.9289 -  
val_loss: 0.1737 - val_accuracy: 0.9433  
Epoch 39/50  
89/89 [=====] - 41s 456ms/step - loss: 0.2296 - accuracy: 0.9219 -  
val_loss: 0.2032 - val_accuracy: 0.9352  
Epoch 40/50  
89/89 [=====] - 41s 462ms/step - loss: 0.1796 - accuracy: 0.9350 -  
val_loss: 0.1801 - val_accuracy: 0.9421  
Epoch 41/50  
89/89 [=====] - 41s 455ms/step - loss: 0.2102 - accuracy: 0.9280 -
```

```
val_loss: 0.1929 - val_accuracy: 0.9387
Epoch 42/50
89/89 [=====] - 41s 458ms/step - loss: 0.2007 - accuracy: 0.9335 -
val_loss: 0.2316 - val_accuracy: 0.9352
Epoch 43/50
89/89 [=====] - 41s 462ms/step - loss: 0.1911 - accuracy: 0.9350 -
val_loss: 0.1839 - val_accuracy: 0.9329
Epoch 44/50
89/89 [=====] - 41s 460ms/step - loss: 0.1940 - accuracy: 0.9321 -
val_loss: 0.1950 - val_accuracy: 0.9352
Epoch 45/50
89/89 [=====] - 41s 459ms/step - loss: 0.1660 - accuracy: 0.9431 -
val_loss: 0.1451 - val_accuracy: 0.9514
Epoch 46/50
89/89 [=====] - 41s 458ms/step - loss: 0.2088 - accuracy: 0.9282 -
val_loss: 0.3538 - val_accuracy: 0.8785
Epoch 47/50
89/89 [=====] - 41s 463ms/step - loss: 0.2116 - accuracy: 0.9254 -
val_loss: 0.2297 - val_accuracy: 0.9109
Epoch 48/50
89/89 [=====] - 41s 458ms/step - loss: 0.1842 - accuracy: 0.9326 -
val_loss: 0.2354 - val_accuracy: 0.9155
Epoch 49/50
89/89 [=====] - 41s 460ms/step - loss: 0.1505 - accuracy: 0.9480 -
val_loss: 0.1467 - val_accuracy: 0.9525
Epoch 50/50
89/89 [=====] - 41s 462ms/step - loss: 0.1443 - accuracy: 0.9544 -
val_loss: 0.1320 - val_accuracy: 0.9514
```

```
In [44]: model.save("Vegetable.h5")
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```
