# AI-powered Nutrition Analyzer for Fitness Enthusiasts

# Project Report

Report by Team :
PNT2022TMID23315

# 1. Introduction

## 1.1 Overview

Food is essential for human life and has been the concern of many healthcare conventions. Nowadays new dietary assessment and nutrition analysis tools enable more opportunities to help people understand their daily eating habits, exploring nutrition patterns and maintain a healthy diet. Nutritional analysis is the process of determining the nutritional content of food. It is a vital part of analytical chemistry that provides information about the chemical composition, processing, quality control and contamination of food.

## 1.2 purpose

The main aim of the project is to building a model which is used for classifying the fruit depends on the different characteristics like colour, shape, texture etc. Here the user can capture the images of different fruits and then the image will be sent the trained model. The model analyses the image and detect the nutrition based on the fruits like (Sugar, Fibre, Protein, Calories, etc.).

# 2. Literature Survey

## 2.1 Existing Problem

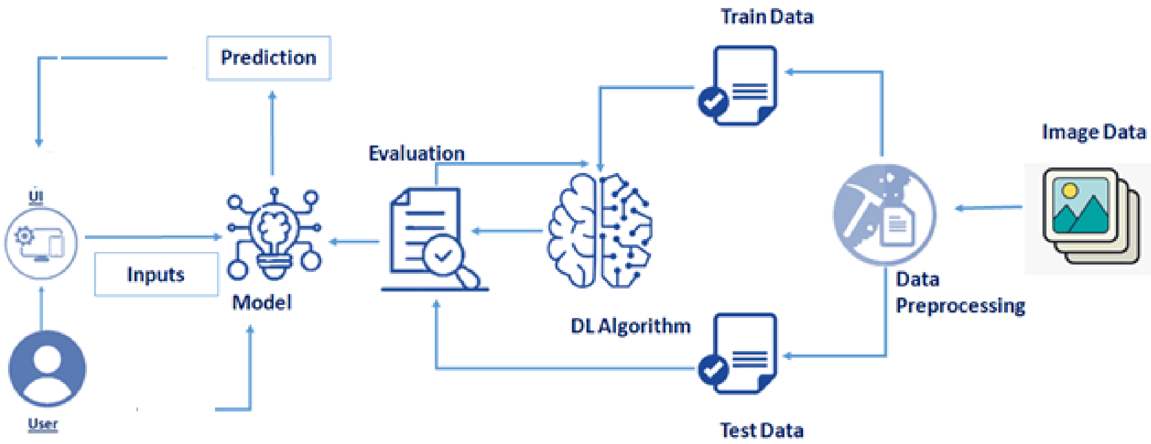
Nowadays new dietary assessment and nutrition analysis tools enable more opportunities to help people understand their daily eating habits, exploring nutrition patterns and maintain a healthy diet. Food is essential for human life and has been the concern of many healthcare conventions.

## 2.2 Proposed Solution

Nutritional analysis is the process of determining the nutritional content of food. It is a vital part of analytical chemistry that provides information about the chemical composition, processing, quality control and contamination of food. Here the user can capture the images of different fruits and then the image will be sent the trained model. The model analyses the image and detect the nutrition based on the fruits like (Sugar, Fibre, Protein, Calories, etc.).

# 3. Theoretical Analysis

## 3.1 Block Diagram




## 3.2 Hardware Software Design

### Software requirements

- Tensor flow
- Keras
- Flask
- Anaconda Navigator

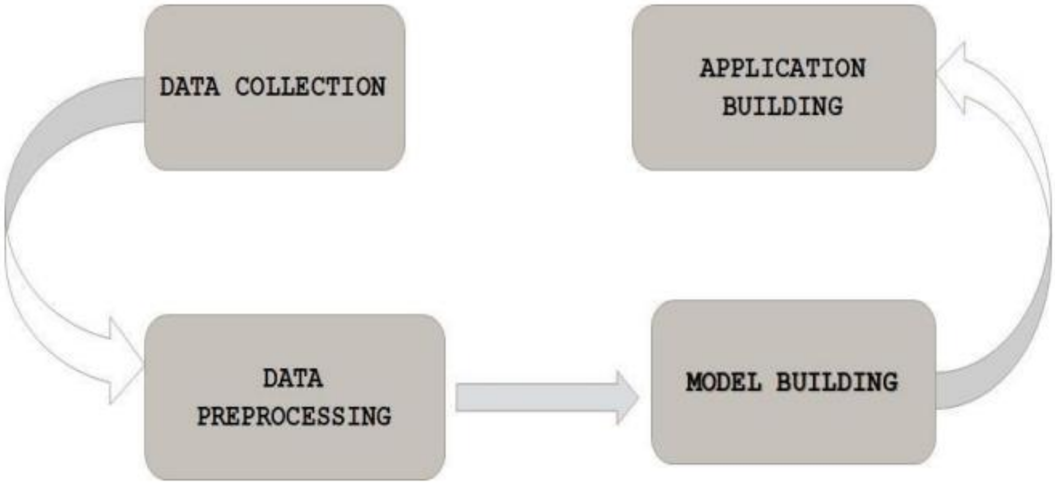
### Hardware requirements

- Processor: Intel Core i3
- Hard Disk Space: Min 100 GB
- Ram: 4 GB
- Display: 14.1 "Color Monitor (LCD, CRT or LED)
 Clock Speed: 1.67 GHz

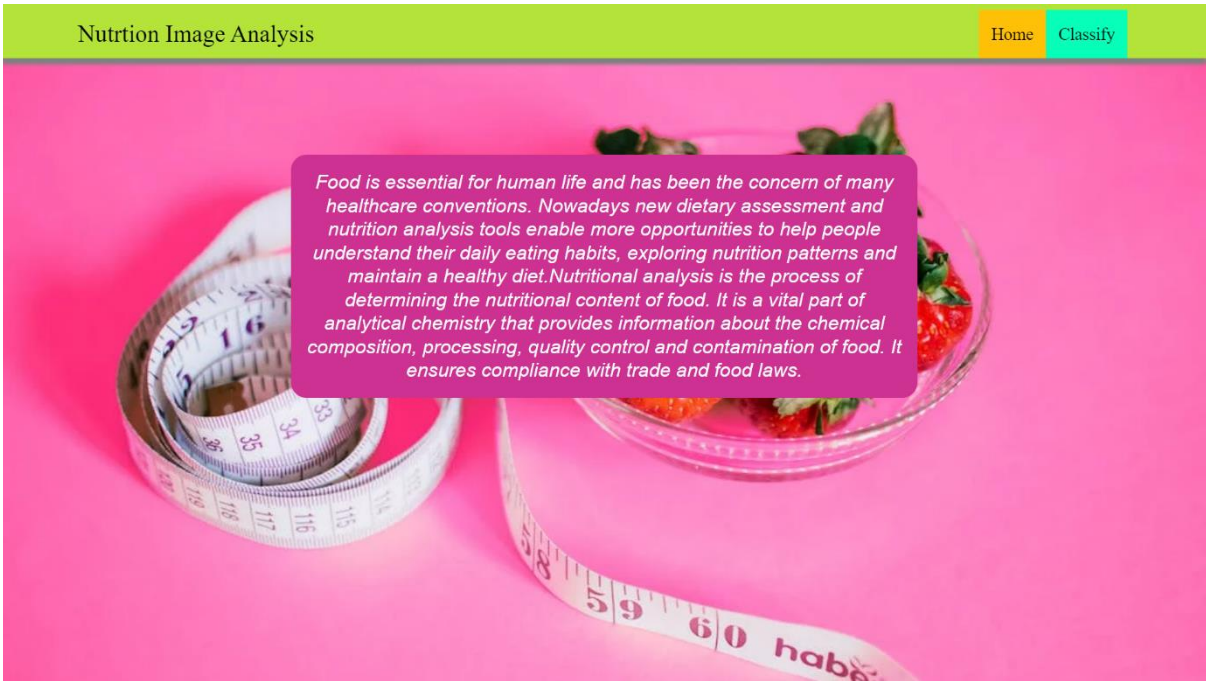
# 4. Experimental Investigations

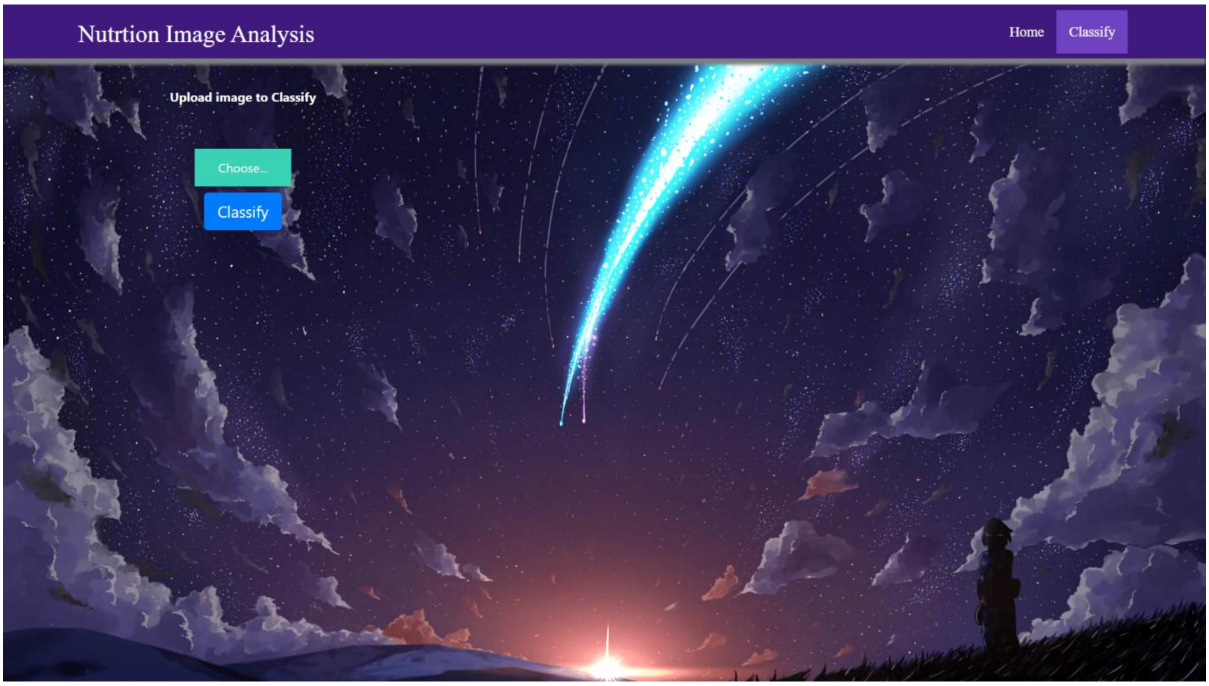
Study shows that it provides with different test images of food images, the model detects, nutrition prediction of uploaded image. When we choose an image and click in to the upload it then it will show the predicted output.

# 5. Flowchart

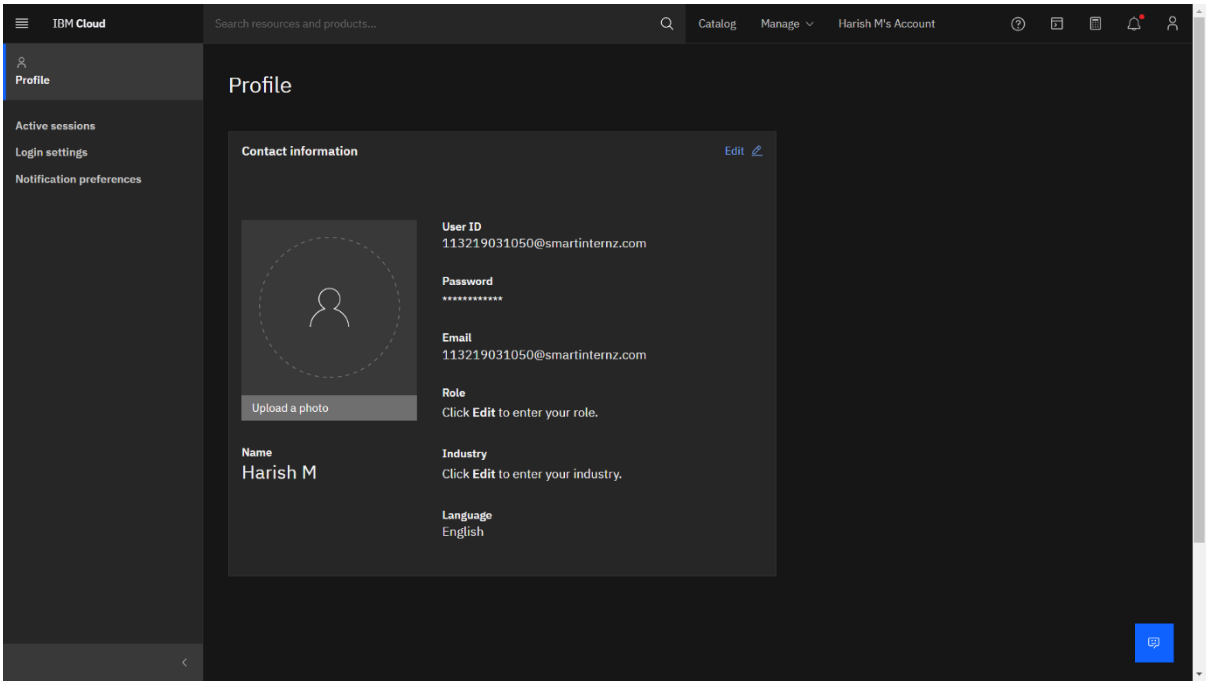


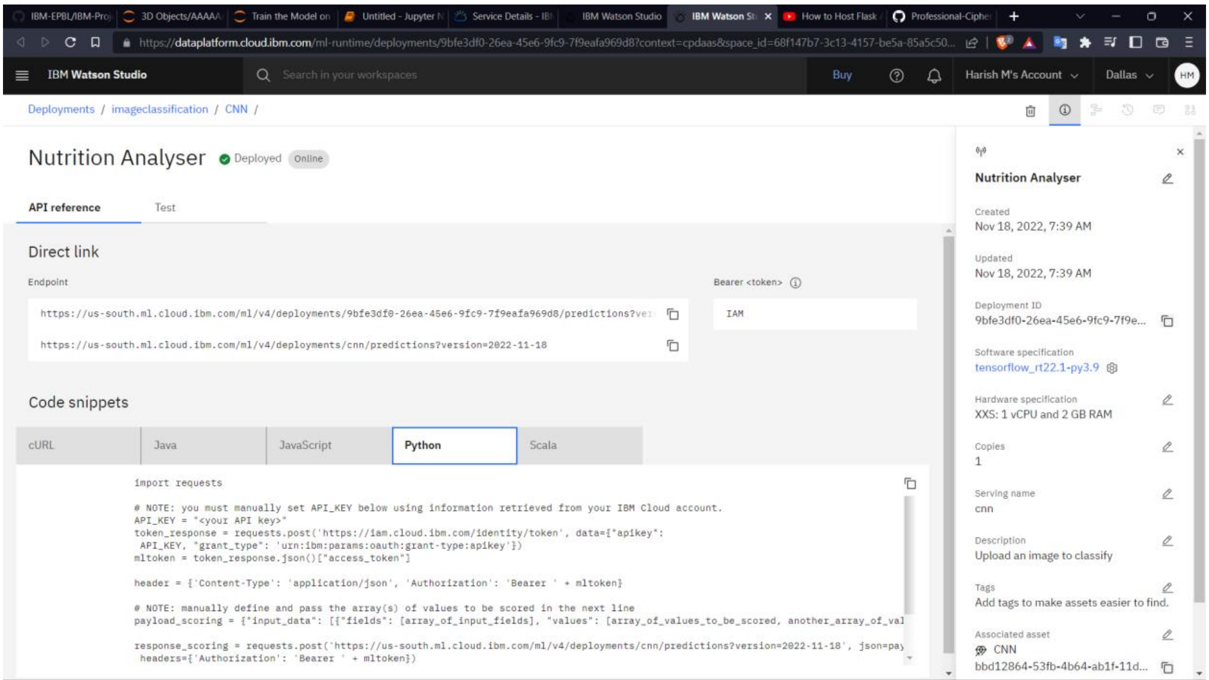

# 6. Result




Nutrtion Image Analysis
Home
Classify
Upload image to Classify
Choose...
Classify


Nutrtion Image Analysis
Home
Classify
Upload image to Classify
Choose...
Food Classified as:
APPLES
[{'sugar_g': 10.3, 'fiber_g': 2.4, 'serving_size_g': 100.0, 'sodium_mg': 1, 'name': 'apples', 'potassium_mg': 11, 'fat_saturated_g': 0.0, 'fat_total_g': 0.2, 'calories': 53.4, 'cholesterol_mg': 0, 'protein_g': 0.3, 'carbohydrates_total_g': 13.8}]


IBM Cloud
Search resources and products...
Catalog
Manage
Harish M's Account
Profile
Active sessions
Login settings
Notification preferences
Profile
Contact information
Edit
User ID
113219031050@smartinternz.com
Password
***********
Email
113219031050@smartinternz.com
Role
Click Edit to enter your role.
Upload a photo
Name
Harish M
Industry
Click Edit to enter your industry.
Language
English


https://dataplatform.cloud.ibm.com/ml-runtime/deployments/9bfe3df0-26ea-45e6-9fc9-7f9eafa969d8?context=cpdaas&space_id=68f147b7-3c13-4157-be5a-85a5c50...
IBM Watson Studio
Search in your workspaces
Buy
Harish M's Account
Dallas
Deployments / imageclassification / CNN /
Nutrition Analyser
Deployed
Online
API reference
Test
Direct link
Endpoint
https://us-south.ml.cloud.ibm.com/ml/v4/deployments/9bfe3df0-26ea-45e6-9fc9-7f9eafa969d8/predictions?ve
https://us-south.ml.cloud.ibm.com/ml/v4/deployments/cnn/predictions?version=2022-11-18
Bearer <token>
IAM
Code snippets
cURL
Java
JavaScript
Python
Scala
import requests
# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "<your API key>"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
# NOTE: manually define and pass the array(s) of values to be scored in the next line
payload_scoring = {"input_data": [{"fields": [array_of_input_fields], "values": [array_of_values_to_be_scored, another_array_of_val
response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/cnn/predictions?version=2022-11-18', json=pay
headers={'Authorization': 'Bearer ' + mltoken})
Nutrition Analyser
Created
Nov 18, 2022, 7:39 AM
Updated
Nov 18, 2022, 7:39 AM
Deployment ID
9bfe3df0-26ea-45e6-9fc9-7f9e...
Software specification
tensorflow_rt22.1-py3.9
Hardware specification
XXS: 1 vCPU and 2 GB RAM
Copies
1
Serving name
cnn
Description
Upload an image to classify
Tags
Add tags to make assets easier to find.
Associated asset
CNN
bbd12864-53fb-4b64-ab1f-11d...

# 7. Advantages & Disadvantages

## Advantages:

- Keeps track of the calorie intake into the body.
- Helps in maintaining the body mass index.

## Disadvantages:

- Data mining techniques does not help to provide effective decision making.

# 8. Applications

- Deep Learning technology is considered as one of the key technologies used in detection.
- It presents the results obtained by processing input from uploading image.

# 9. Conclusion

In this project, we have established the application to predict from uploaded image based on the IBM cloud application.

# 10. Future Scope

The project can be further enhanced by deploying the deep learning model obtained using a web application and larger dataset cloud be used for prediction to give higher accuracy and produce better result.

## Source Code

## App.py

```
from flask import Flask,render_template,request

# Flask-It is our framework which we are going to use to run/serve our application.
#request-for accessing file which was uploaded by the user on our application.

import os
import numpy as np #used for numerical analysis
from tensorflow.keras.models import load_model#to load our trained model
from tensorflow.keras.preprocessing import image
import requests


app = Flask(__name__,template_folder="templates") # initializing a flask app
# Loading the model
model=load_model('nutrition.h5')
print("Loaded model from disk")


@app.route('/')# route to display the home page
def home():
    return render_template('home.html')#rendering the home page

@app.route('/image1',methods=['GET','POST'])# routes to the index html
def image1():
    return render_template("image.html")



@app.route('/predict',methods=['GET', 'POST'])# route to show the predictions in a web UI
def launch():
    if request.method=='POST':
        f=request.files['file'] #requesting the file
        basepath=os.path.dirname('__file__')#storing the file directory
        filepath=os.path.join(basepath,"uploads",f.filename)#storing the file in uploads folder
        f.save(filepath)#saving the file

        img=image.load_img(filepath,target_size=(64,64)) #load and reshaping the image
        x=image.img_to_array(img)#converting image to an array
        x=np.expand_dims(x,axis=0)#changing the dimensions of the image
```

```
        pred=np.argmax(model.predict(x), axis=1)
        print("prediction",pred)#printing the prediction
        index=['APPLES','BANANA','ORANGE','PINEAPPLE','WATERMELON']

        result=str(index[pred[0]])

        x=result
        print(x)
        result=nutrition(result)
        print(result)

        return render_template("0.html",showcase=(result),showcase1=(x))
def nutrition(index):


    url = "https://calorieninjas.p.rapidapi.com/v1/nutrition"

    querystring = {"query":index}

    headers = {
        'x-rapidapi-key': "5d797ab107mshe668f26bd044e64p1ffd34jsnf47bfa9a8ee4",
        'x-rapidapi-host': "calorieninjas.p.rapidapi.com"
        }

    response = requests.request("GET", url, headers=headers, params=querystring)

    print(response.text)
    return response.json()['items']
if __name__ == "__main__":
   # running the app
    app.run(debug=False)
```

## Model Creation in IBM Cloud

```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Activation, Dense, Flatten, BatchNormalization,
Conv2D, MaxPool2D
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.metrics import categorical_crossentropy
from sklearn.metrics import confusion_matrix
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

IBM Watson Studio

Projects / image classification / img_proj

In [51]:
```
from keras.preprocessing.image import ImageDataGenerator
```

In [52]:
```
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
```

In [53]:
```
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.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='WImD1lYFgLr7ETzVUrUheKzW-cdRHnZBTZU5S4909gq3',
    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')

bucket = 'imageclassification-donotdelete-pr-ohyztlr8kisyqz'
object_key = 'Nutrition classifier_zipped.zip'

streaming_body_1 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

# 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 to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

In [54]:
```
if not hasattr(streaming_body_1,"__iter__"):streaming_body_1.__iter__ = types.MethodType(__iter__,streaming_body_1)
```

In [55]:
```
from io import BytesIO
from zipfile import ZipFile
unzip = ZipFile(BytesIO(streaming_body_1.read()),'r')
file_paths = unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

In [56]:
```
import os
filenames = os.listdir('/home/wsuser/work/Nutrition classifier/TEST_SET')
```

IBM Watson Studio

Projects / image classification / img_proj

```
file_paths = unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

In [56]:
```
import os
filenames = os.listdir('/home/wsuser/work/Nutrition classifier/TEST_SET')
```

In [57]:
```
x_train=train_datagen.flow_from_directory(r'/home/wsuser/work/Nutrition classifier/TRAIN_SET',target_size=(64,64),batch_size=5,color_mode='rgb',class_mode='sparse')
x_test = train_datagen.flow_from_directory(r'/home/wsuser/work/Nutrition classifier/TEST_SET',target_size=(64,64),batch_size=5,color_mode='rgb',class_mode='sparse')
```

```
Found 2626 images belonging to 5 classes.
Found 1055 images belonging to 5 classes.
```

In [58]:
```
print(x_test.class_indices)
```

```
{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3, 'WATERMELON': 4}
```

In [59]:
```
from collections import Counter as c
c(x_train.labels)
```

Out[59]:
```
Counter({0: 606, 1: 445, 2: 479, 3: 621, 4: 475})
```

In [60]:
```
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Dense,Flatten
from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout
from keras.preprocessing.image import ImageDataGenerator
```

In [61]:
```
model=Sequential()
classifier=Sequential()
```

In [62]:
```
classifier.add(Conv2D(32,(3,3),input_shape=(64,64,3),activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2,2)))
classifier.add(Conv2D(32,(3,3),activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2,2)))
classifier.add(Flatten())
classifier.add(Dense(units=128,activation='relu'))
classifier.add(Dense(units=5,activation='softmax'))
```

In [63]:
```
classifier.summary()
```

```
Model: "sequential_3"
```

```
classifier.add(Dense(units=128,activation='relu'))
classifier.add(Dense(units=5,activation='softmax'))
```

In [63]:
```
classifier.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 62, 62, 32)        896

 max_pooling2d_2 (MaxPooling  (None, 31, 31, 32)       0
 2D)

 conv2d_3 (Conv2D)           (None, 29, 29, 32)        9248

 max_pooling2d_3 (MaxPooling  (None, 14, 14, 32)       0
 2D)

 flatten_1 (Flatten)         (None, 6272)              0

 dense_2 (Dense)             (None, 128)               802944

 dense_3 (Dense)             (None, 5)                 645

=================================================================
Total params: 813,733
Trainable params: 813,733
Non-trainable params: 0
_________________________________________________________________
```

In [64]:
```
classifier.compile(optimizer='adam',loss='sparse_categorical_crossentropy',metrics=['accuracy'])
```

In [65]:
```
classifier.fit_generator(generator=x_train,steps_per_epoch=len(x_train),epochs= 2 ,validation_data=x_test,validation_steps=len(x_test))
```

```
/tmp/wsuser/ipykernel_164/3830403452.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  classifier.fit_generator(generator=x_train,steps_per_epoch=len(x_train),epochs= 2 ,validation_data=x_test,validation_steps=len(x_test))
Epoch 1/2
526/526 [==============================] - 32s 59ms/step - loss: 0.1204 - accuracy: 0.9539 - val_loss: 0.0825 - val_accuracy: 0.9611
Epoch 2/2
526/526 [==============================] - 32s 61ms/step - loss: 0.0455 - accuracy: 0.9863 - val_loss: 0.0702 - val_accuracy: 0.9754
```

Out[65]: <keras.callbacks.History at 0x7fc5ba836e80>

In [17]:
```
classifier.save('/home/wsuser/work/nutrition.h5')
```

IBM Watson Studio    Search in your workspaces    Buy    Harish M's Account    Dallas    HM
Projects / image classification / img_proj

```
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learning-client) (3.3)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (1.20.3)
Installing collected packages: watson-machine-learning-client
Successfully installed watson-machine-learning-client-1.0.391
```

In [21]:
```
from ibm_watson_machine_learning import APIClient
wml_credintials = {
    "url" : "https://us-south.ml.cloud.ibm.com",
    "apikey" : "3EMrL-7wESBZLFCk0abidBdj4Pnlz-7Hsiqd3E39NCQX"
}

client = APIClient(wml_credintials)
```

In [22]:
```
def guid_from_space_name(client,space_name):
    space = client.spaces.get_details()
    return(next(item for item in space['resources'] if item['entity']['name'] == space_name)['metadata']['id'])
```

In [23]:
```
space_uid = guid_from_space_name(client , 'imageclassification')
print("Space UID = "+ space_uid)
```

```
Space UID = 68f147b7-3c13-4157-be5a-85a5c50f29ee
```

In [24]:
```
client.set.default_space(space_uid)
```

Out[24]: 'SUCCESS'

In [25]:
```
software_spec_uid = client.software_specifications.get_uid_by_name("tensorflow_rt22.1-py3.9")
software_spec_uid
```

Out[25]: 'acd9c798-6974-5d2f-a657-ce06e986df4d'

In [26]:
```
client.software_specifications.list(limit = 100)
```

```
------------------------------  ------------------------------------  ----
NAME                            ASSET_ID                              TYPE
default_py3.6                   0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
kernel-spark3.2-scala2.12       020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt      069ea134-3346-5748-b513-49120e15d288  base
scikit-learn_0.20-py3.6         09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
spark-mllib_3.0-scala_2.12      09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
pytorch-onnx_rt22.1-py3.9       0b848dd4-e681-5599-be41-b5f6fccc6471  base
ai-function_0.1-py3.6           0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
shiny-r3.6                      0e6e79df-875e-4f24-8ae9-62dcc2148306  base
tensorflow_2.4-py3.7-horovod    1092590a-307d-563d-9b62-4eb7d64b3f22  base
```

IBM Watson Studio    Search in your workspaces    Buy    Harish M's Account    Dallas    HM

Projects / image classification / img_proj

```
autoai-kb_3.4-py3.8             da9b39c3-758c-5a4f-9cfd-457dd4d8c395  base
kernel-spark3.2-r3.6            db2fe4d6-d641-5d05-9972-73c654c60e0a  base
autoai-kb_rt22.1-py3.9          db6afe93-665f-5910-b117-d879897404d9  base
tensorflow_rt22.1-py3.9-horovod dda170cc-ca67-5da7-9b7a-cf84c6987fae  base
autoai-ts_1.0-py3.7             deef04f0-0c42-5147-9711-89f9904299db  base
tensorflow_2.1-py3.7-horovod    e384fce5-fdd1-53f8-bc71-11326c9c635f  base
default_py3.7                   e4429883-c883-42b6-87a8-f419d64088cd  base
do_22.1                         e51999ba-6452-5f1f-8287-17228b88b652  base
autoai-obm_3.2                  eae86aab-da30-5229-a6a6-1d0d4e368983  base
do_20.1                         f686cdd9-7904-5f9d-a732-01b0d6b10dc5  base
scikit-learn_0.19-py3.6         f963fa9d-4bb7-5652-9c5d-8d9289ef6ad9  base
tensorflow_2.4-py3.8            fe185c44-9a99-5425-986b-59bd1d2eda46  base
------------------------------  ------------------------------------  ----
```

In [27]:

```
model_details = client.repository.store_model(model = 'image-classification-model_new.tgz' , meta_props = {
    client.repository.ModelMetaNames.NAME:"CNN",
    client.repository.ModelMetaNames.TYPE : "tensorflow_2.7",
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID : software_spec_uid
})

model_id = client.repository.get_model_uid(model_details)
```

```
This method is deprecated, please use get_model_id()
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ibm_watson_machine_learning/repository.py:1453: UserWarning: This method is deprecated, please use get_model_id()
  warn("This method is deprecated, please use get_model_id()")
```

In [28]:

```
model_id
```

Out[28]: `'bbd12864-53fb-4b64-ab1f-11d45246f865'`

In [29]:

```
client.repository.download(model_id, 'nutrition_analyzer2_model.h5')
```

```
Successfully saved model content to file: 'nutrition_analyzer2_model.h5'
```

Out[29]: `'/home/wsuser/work/nutrition_analyzer2_model.h5'`