

FINAL DELIVERABLES

FINAL CODE

Date	15 November 2022
Team ID	PNT2022TMID04276
Project Name	Project - AI-Powered Nutrition Analyzer for Fitness Enthusiasts

```
# -*- coding: utf-8 -*-
```

```
"""Model Building.ipynb
```

Automatically generated by Colaboratory.

Original file is located at

<https://colab.research.google.com/drive/1kvKID8OQQLMsU1rTD8SI1U2EQCfIRTes>

```
**Import The ImageDataGenerator Library**
```

```
"""
```

```
import keras
```

```
from keras.preprocessing.image import ImageDataGenerator
```

```
*****Configure ImageDataGenerator Class*****
```

```
train_datagen=ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
```

```
test_datagen=ImageDataGenerator(rescale=1./255)
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
*****Apply Image DataGenerator Functionality To Trainset And Testset*****
```

```
x_train =
```

```
train_datagen.flow_from_directory(r'/content/drive/MyDrive/Project/Dataset/TRAIN_SET', target_size=(64, 64), batch_size=5, color_mode='rgb', class_mode='sparse')
```

```

x_test =
test_datagen.flow_from_directory(r'/content/drive/MyDrive/Project/Dataset/TRAIN_SET',ta
rget_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')

print(x_train.class_indices)

print(x_test.class_indices)

from collections import Counter as c

c(x_train .labels)

"""# Model Building

**Importing The Model Building Libraries**
"""

from keras.preprocessing.image import ImageDataGenerator
import numpy as np
import tensorflow
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

*****Initializing The Model*****

model=Sequential()

*****Adding CNN Layers*****

classifier = Sequential()
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())

*****Adding Dense Layers**

"""

```

```

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

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

classifier.summary()

*****Configure The Learning Process*****

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

*****Train The Model*****

classifier.fit_generator(generator=x_train,steps_per_epoch =
len(x_train),epochs=20,validation_data=x_test,validation_steps = len(x_test))

*****Save The Model*****

classifier.save('model/nutrition.h5')

*****Test The Model*****

import numpy as np

from tensorflow.keras.models import load_model
from keras.preprocessing import image

from tensorflow.keras.preprocessing import image

model = load_model("model/nutrition.h5")

img =image.load_img(r"/content/drive/MyDrive/Nutrition Image Analysis using CNN and
Rapid API/Dataset/Data Collection/Data set/ORANGE/102_100.jpg",grayscale=False,
target_size= (64,64))

x = image.img_to_array(img)

x = np.expand_dims(x,axis=0)

pred =model.predict(x)
pred

labels=['APPLES', 'BANANA', 'ORANGE','PINEAPPLE','WATERMELON']

```

```
labels[np.argmax(pred)]
```

```
*****flask*****
```

```
from flask import Flask,render_template,request
```

```
import os
```

```
import numpy as np
```

```
from tensorflow.keras.models import load_model
```

```
from tensorflow.keras.preprocessing import image
```

```
import requests
```

```
app=Flask(__name__,template_folder="templates")
```

```
model=load_model('model/nutrition.h5')
```

```
print("loaded model from the disk")
```

```
@app.route('/',methods=['POST'], endpoint='/')
```

```
def home():
```

```
    return render_template('final home.html')
```

```
@app.route('/image1',methods=['GET','POST'])
```

```
def image1():
```

```
    return render_template('o.html')
```

```
@app.route('/predict',methods=['GET','POST'])
```

```
def launch():
```

```
    if request.method=='POST':
```

```
        f=request.files['file']
```

```
        basepath=os.path.dirname('__file__')
```

```
        filepath=os.path.join(base,"uploads",f.filename)
```

```
        f.save(filepath)
```

```
        img=image.load_img(filepath,target_size=(64,64))
```

```
        x=image.img_to_array(img)
```

```
        x=np.expand_dims(x,axis=0)
```

```
        pred=np.argmax(model.predict(x),axis=1)
```

```
        print("prediction",pred)
```

```
        index=['APPLES','BANANA','ORANGE','PINEAPPLE','WATERMELON']
```

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

```
        x=result
```

```
        print(x)
```

```
        result=nutrition(result)
```

```
print(result)

return render_template("o.html",showcase=("result"),showcase1=(x))

def nutrition(index):
    url = "https://calorieninjas.p.rapidapi.com/vi/nutrition"
    querystring= {"query": index}
    headers = {
        'x-rapidapi-key': "5d797ab187mshe668f26bd044e64p1ffd34jsnf47bfa9a8ee4",
        '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__":
    app.run(debug=False)
```