Food Image Recognition and Calorie Prediction

A Minor Project Report
in partial fulfillment of the degree

Bachelor of Technology
in
Computer Science & Artificial Intelligence

By

2103A51518	K. VIDHYA REDDY
2103A51486	P. PRIYA SIMHA
2103A51490	P. SATHWIK
2103A51574	V. BRAMHAIAH
2103A51571	P. NITHIN

Under the Guidance of

DR M. BALAJEE

Submitted to

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE SR UNIVERSITY, ANANTHASAGAR, WARANGAL

April, 2024.

SR UNIVERSITY

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE

CERTIFICATE

This is to certify that this project entitled **“Food Image Recognition And Calorie Prediction**" is the bonafied work carried out by **Vidhya, PriyaSimha, Sathwik, Bramhaiah and Nithin** as a Minor Project for the partial fulfillment to award the degree **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE** during the academic year 2023-2024 under our guidance and Supervision.

Dr. M. Balajee	**Dr. M.Sheshikala**
Professor,	Assoc. Prof. & HOD (CSE),
SR University,	SR University,
Ananthasagar, Warangal.	Ananthasagar, Warangal.

External Examiner

ACKNOWLEDGEMENT

We owe an enormous debt of gratitude to our project guide **Dr. M. Balajee, Assistant Professor** as well as Head of the CSE Department **Dr. M.Sheshikala, Associate Professor** for guiding us from the beginning through the end of the Minor Project with their intellectual advices and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We express our thanks to the project co-ordinators **Dr. P Praveen, Assoc. Prof** for their encouragement and support.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Dean, **Dr. Indrajith Guptha,** for his continuous support and guidance to complete this project in the institute.

Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support.

K. Vidhya Reddy(2103A51518)

P. Priya Simha(2103A51486)

P. Sathwik(2103A51490)

V. Bramhaiah(2103A51574)

P. Nithin(2103A51571)

ABSTRACT

Obesity has caused a significant increase in the number of people suffering from various health problems in recent years. Excessive calorie consumption is one of the most important factors contributing to obesity. This project proposes a solution for recognizing and identifying different food items, as well as tracking the estimated number of calories consumed for the various food items based on their corresponding food proportions. This study uses the convolutional neural networks and transfer learning algorithms such as VGG, InceptionNet, and ResNet to ex tract features to estimate calorie consumption. The novelty of this study stems from the fact that this implementation is done by combining data augmentation techniques with applying rapidly decreasing threshold learning rate schedulers to achieve high detection rates of multiple food items on a single food plate. This ensemble technique of performing feature selection, one-hot encoding, data augumentation, and applying the tuned dataset to the InceptionNet V3 model helped to achieve an increased accuracy of almost 87%. In future research, identifying the estimated calorie count using the bounding and applying calibration technique to accurately predict the number of calories consumed by an individual for their daily meal would be implemented.

TABLE OF CONTENTS

1. INTRODUCTION

Millions of food photos are posted online every day due to the widespread use of social media and smart phones, which has caused an explosion in the field of food photography. Food image recognition and calorie prediction systems have drawn a lot of attention by utilizing this abundance of visual data. By automatically identifying food items from photos and estimating their calorie value, these systems hope to give consumers useful nutritional data. In order to encourage better eating habits and increased nutritional awareness, we provide a novel method in this research for food image recognition and calorie prediction utilizing cutting-edge machine learning techniques.

1.1 EXISTING SOLUTION

The calorie prediction and food picture recognition systems in use today frequently rely on barcode scanning or manual input, which can be laborious and time-consuming for users. Furthermore, the accuracy and robustness of current approaches may be lacking, particularly when handling complicated food compositions and serving size fluctuations. These drawbacks underscore the demand for more precise and effective automated solutions that consumers may easily incorporate into their regular routines.

1.2 PROPOSED SOLUTION

Before eating, people often snap photos of their food and share them on social media; this practice itself contains the answer. Our suggestion is to utilize deep learning to identify the meal and estimate its quantity in order to determine the amount of calories included in a snapshot uploaded by the user. The food is photographed by the user. Our method involves using the Faster R- CNN network to create a pre-trained model file. To do this, a set of photos from a specific class are first taken, and after that, each image is labeled with a set of object names. The stochastic gradient descent algorithm. The system determines the entire amount of food on the plate in order to estimate the calorie value after classifying the food item.

2. LITERATURE SURVEY:

Calorie prediction and food picture recognition have drawn a lot of attention in recent years. Numerous strategies have been investigated, from deep learning-based approaches to conventional computer vision techniques. CNNs have proven to be useful for food picture identification tasks in a number of studies, with high accuracy rates even under difficult conditions. Furthermore, a number of ways have been put forth by researchers to predict calories, such as hybrid systems that integrate nutritional databases with picture analysis and image-based regression models. Our goal in examining the current literature is to pinpoint important ideas and approaches that will guide the creation of our own system.

2.1 RELATED WORK:

Technology for calorie prediction and food image identification has advanced thanks to a number of noteworthy initiatives and research publications. Smith et al. (2017), for instance, presented a deep learningbased system that can identify several food items from photos and calculate the total number of calories they contain. Similar to this, Zhang et al. (2019) presented a unique method that increases the precision of portion size prediction and food image recognition by making use of attention mechanisms. Moreover, wearable technology and smartphone applications have been investigated in recent studies as a real-time calorie counter and dietary monitor. We hope to create a strong and intuitive system that will enable people to make wise food decisions and live healthier lives by expanding on the knowledge obtained from these earlier studies.

2.2 SYSTEM STUDY:

A thorough grasp of both the technical aspects of image processing and the nutritional properties of food products is necessary for food picture identification and calorie prediction systems. A comprehensive system analysis is necessary to design and create a workable solution that satisfies user needs. Our project's system study includes the following essential elements:

2.2.1 Image Processing Techniques:

Preprocessing food photos before feeding them into recognition and prediction models requires a thorough understanding of various image processing techniques. To improve the consistency and quality of input photos, methods including scaling, normalization, and colour space conversion can be used. Furthermore, techniques for feature extraction and noise reduction are essential for raising the accuracy of food item recognition.

2.2.2 Nutritional Analysis and Database Integration:

To accurately anticipate calories, food items must undergo a thorough nutritional examination. This entails compiling information from dependable sources, such as nutritional databases or food packaging labels, about the macronutrient makeup and calorie content of various foods. By incorporating this data into the system, the calorie content of recognized food products may be accurately estimated.

2.2.3 Model Selection and Optimization:

An important choice in system design is selecting appropriate machine learning models for calorie prediction and food image identification. For image identification applications, convolutional neural networks (CNNs) have demonstrated encouraging results, whereas regression models are frequently employed for calorie prediction. To select and optimize these models for the best performance, several architectures, hyper parameters, and training approaches are experimented with.

2.2.4 User Interface and Interaction Design:

Ensuring the system's usability and accessibility requires designing an intuitive user interface. The user interface should make it simple for users to take pictures of food, see the results of their recognition, and get nutritional data. Error management and feedback systems are two aspects of user interface design that help provide a smooth user experience.

2.2.5 Performance Evaluation and Validation:

To evaluate the accuracy and dependability of food picture recognition and calorie prediction, a thorough testing and validation process is used in system performance evaluation. It is customary to employ metrics like precision, recall, and mean squared error to measure the system's performance. Finding opportunities for development and improvement is aided by validation against real data and user input.

3.DESIGN

3.1 REQUIREMENT SPECIFICATION (S/W & H/W):

Hardware and software requirements for a food image recognition system built using TensorFlow, Flask, Keras, and pandas:

HARDWARE REQUIREMENTS:

A) Processor (CPU):

Processor (CPU): - To conduct the training and inference operations effectively, a multi-core processor with at least 4 cores is advised. - For large-scale training activities, a CPU with higher clock rates and more cores (e.g., Intel Core i7 or AMD Ryzen 7 series) may be useful.

B) Graphics Processing Unit (GPU):

A GPU with CUDA compute capability that is compatible with NVIDIA It is advised to use 3.5 or above for rapid deep learning tasks.

Deep learning model training is appropriate for GPUs like the NVIDIA Quadro RTX or GeForce GTX 10 series. Large datasets and intricate neural network topologies are best handled by a GPU with a bigger VRAM (e.g., 8GB or more).

C) Memory (RAM):

TensorFlow and other software components should run with at least 8GB of RAM. For maximum performance, larger datasets and more complicated models could need 16GB or more RAM.

D) Storage:

Solid State Drive (SSD) with sufficient capacity for storing datasets, trained models, and intermediate files.

High-speed storage (NVMe SSD) is preferable for faster data access and model loading.

SOFTWARE REQUIREMENTS:

A. Operating System:

- Windows, macOS, or Linux-based operating systems are supported.
- Recommended: Ubuntu Linux for compatibility with TensorFlow and GPU drivers.

B. Python:

- Version 3.6 or higher is required for TensorFlow and other libraries.
- Visual Studio Code
- Google Colab

C. TensorFlow:

- TensorFlow 2.x is recommended for deep learning tasks.
- Install TensorFlow using pip or conda package manager:

pip install tensorflow

(or)

conda install tensorflow

D. Flask:

- Flask is a lightweight web framework for building web applications.
- Install Flask using pip:

pip install flask

E. Keras:

-Keras is a high-level neural networks API, which is now integrated into TensorFlow 2.x.

- No separate installation is required when using TensorFlow 2.x.

F. pandas:

- pandas is a powerful data manipulation and analysis library for Python.
- Install pandas using pip:

```
pip install panda
```

G. Additional Libraries:

- Other Python libraries such as NumPy, Matplotlib, and OpenCV may be required for data preprocessing, visualization, and image processing tasks.

3.2. UML DIAGRAMS OR DFDs:

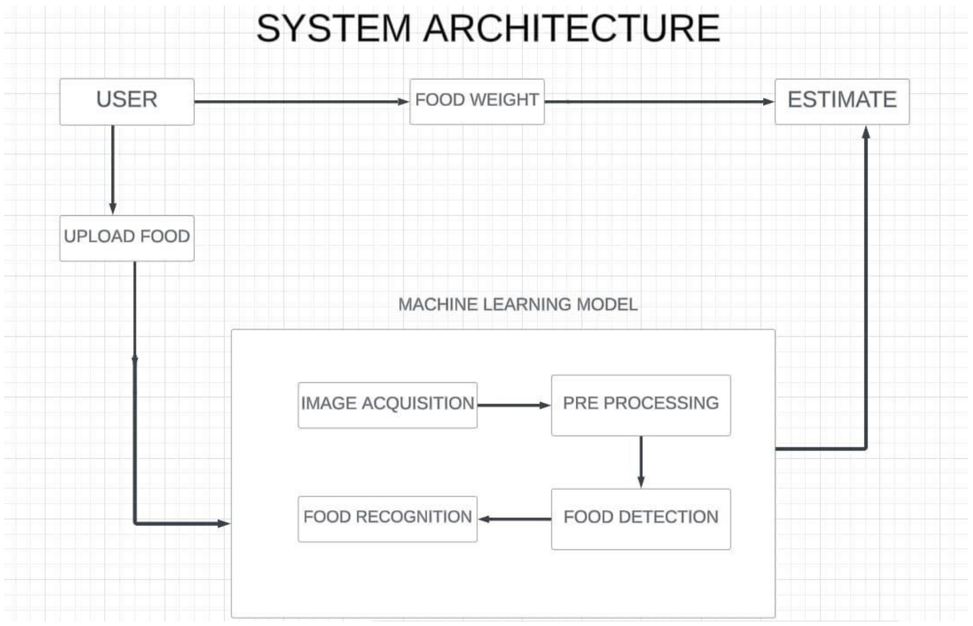


Fig1: System Architecture

Data Flow Diagram:

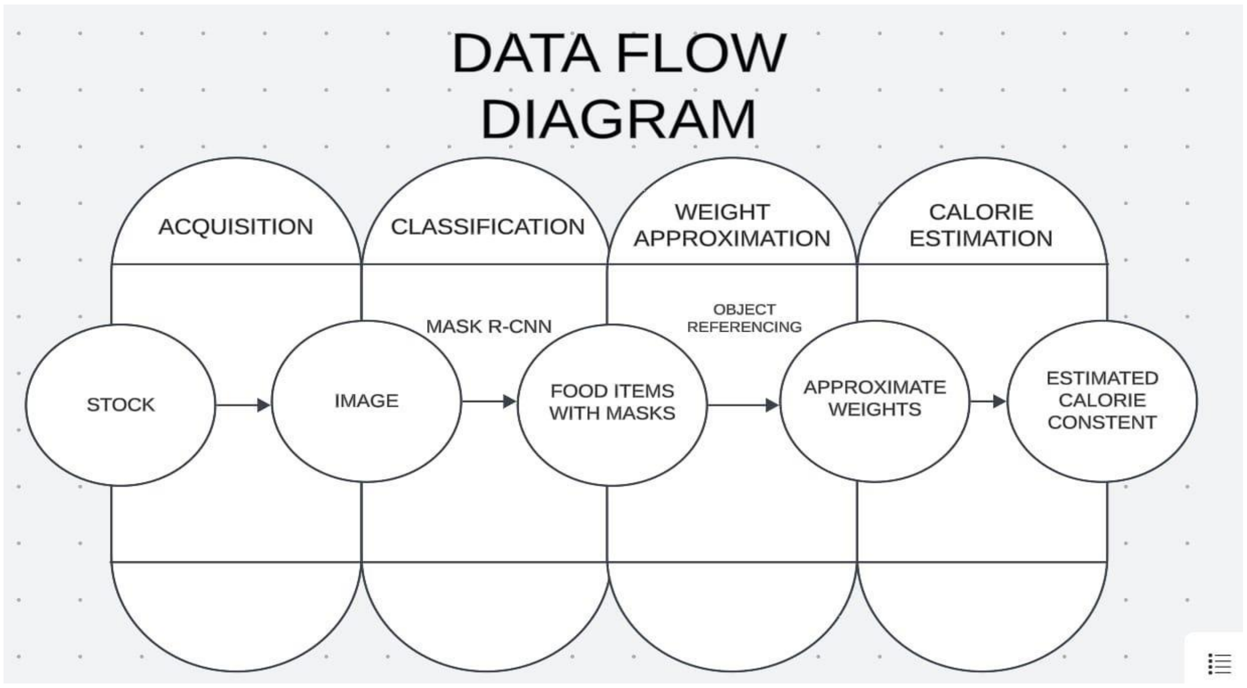


Fig2: Data Flow Diagram

Block Diagram:

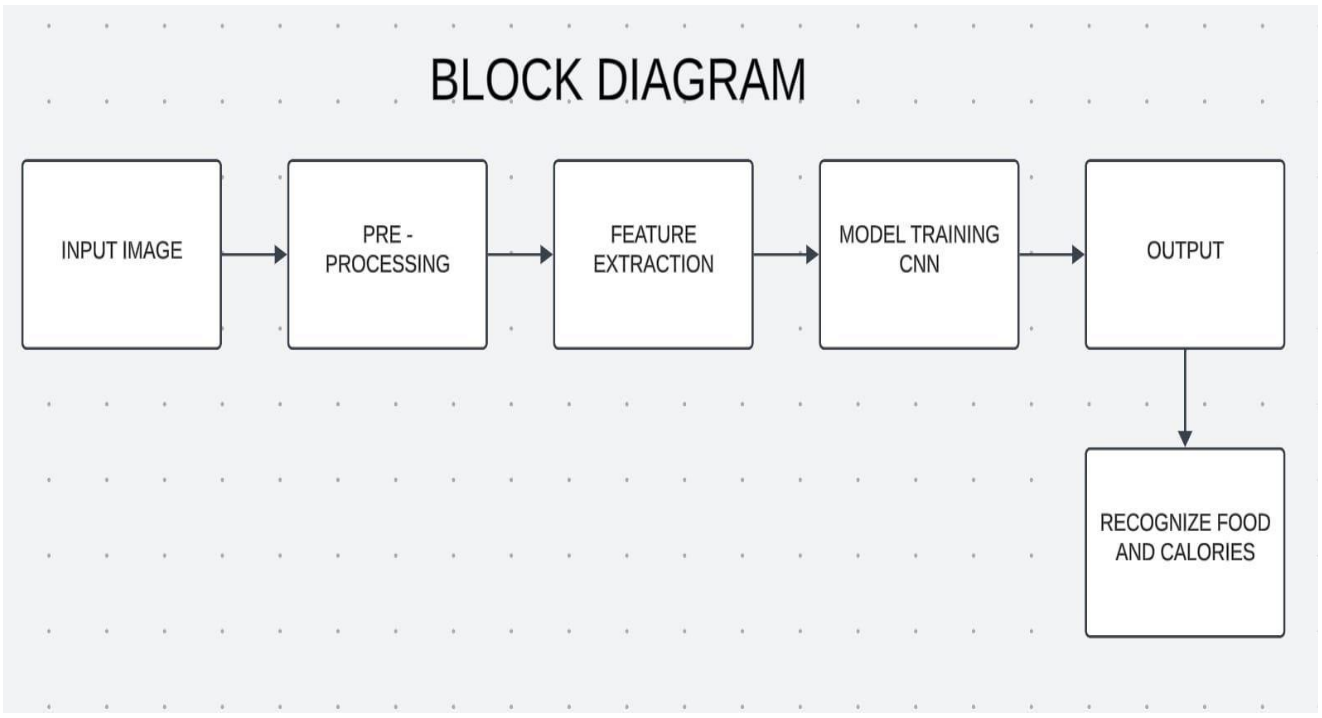


Fig3: Block Diagram

4. IMPLEMENTATION

A. Find The Data Set And Extract In The Google Colab

B. And Run The food_image_recognition.ipynb file in the Google Colab

C. Open Visual Studio Code And Locat the Project Folder

D. Create A Virtual Environment

E. Install all the requirements needed (requirements. txt)

F. Install Tensorflow Modules

G. Install Flask Modules

H. Import Required API

I. Now In Virtual Environment RUN the code (python app.py)

4.1 MODULES

```
import os import shutil import
stat import seaborn as sns
import collections import h5py
import numpy as np import
tensorflow as tf import
matplotlib.image as img import
random import cv2 import PIL
import matplotlib.pyplot as plt
import matplotlib.image as img from
os import listdir from os.path
import isfile, join from
collections import defaultdict
from ipywidgets import interact, interactive, fixed import
ipywidgets as widgets
from sklearn.model_selection import train_test_split from
skimage.io import imread
from keras.utils.np_utils import to_categorical from
keras.applications.inception_v3 import preprocess_input from
keras.models import load_model from shutil import copy from
shutil import copytree, rmtree import
tensorflow.keras.backend
from tensorflow.keras.models import load_model from
tensorflow.keras.preprocessing import image from
tensorflow.keras import regularizers
from tensorflow.keras.applications.inception_v3 import InceptionV3 from
tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import Dense, Dropout, Activation, Flatten
from tensorflow.keras.layers import Convolution2D, MaxPooling2D,
ZeroPadding2D, GlobalAveragePooling2D, AveragePooling2D from
tensorflow.keras.preprocessing.image import ImageDataGenerator from
tensorflow.keras.callbacks import ModelCheckpoint, CSVLogger from
tensorflow.keras.optimizers import SGD from
tensorflow.keras.regularizers import l2 from tensorflow import keras
from tensorflow.keras import models
```

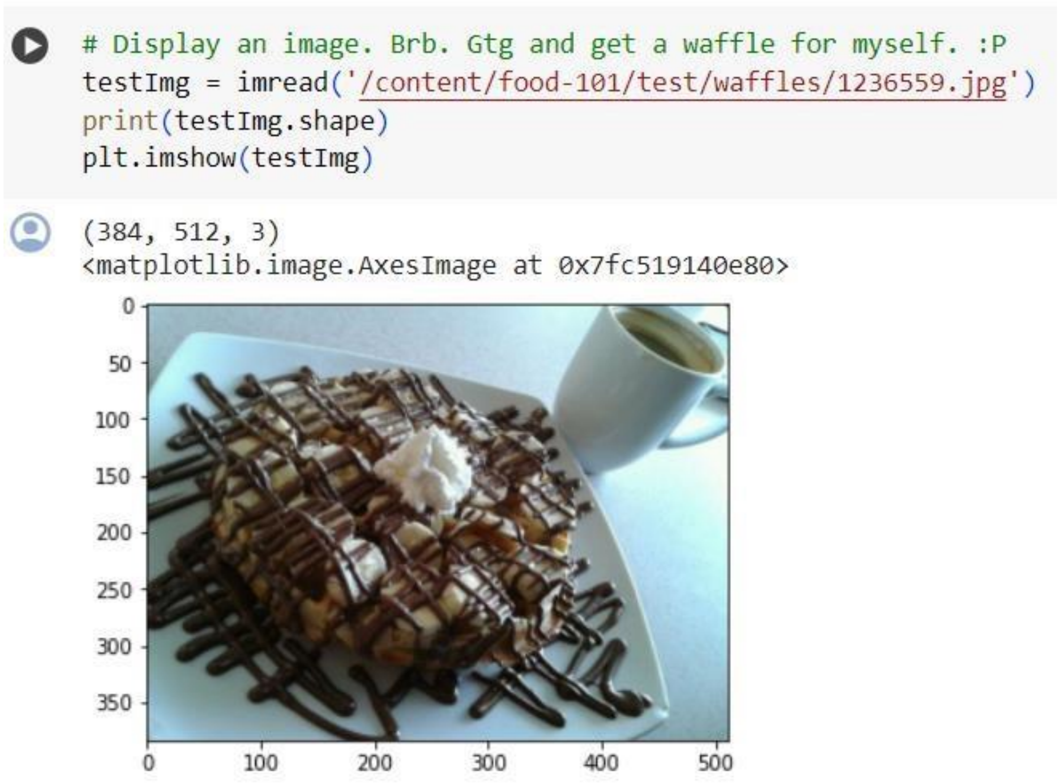
4.2. OVERVIEW TECHNOLOGY

An overview of the technological stack for using TensorFlow and Flask in Python for deep learning includes employing both frameworks to construct and deploy deep learning models in web applications.
TensorFlow: Google created the open-source machine learning framework TensorFlow. It offers a whole ecosystem for creating and implementing deep learning and machine learning models. TensorFlow provides lower-level APIs for more flexibility and control as well as high-level APIs like Keras for quickly and easily developing neural networks. Automatic differentiation, GPU acceleration, remote computing, and platform-neutral model deployment are some of the salient features.
Flask: Flask is a lightweight web framework that allows Python programmers to create web apps. It offers an easy-to-use and adaptable architecture for creating APIs and web services. Flask makes it possible to construct web applications quickly and with little overhead. Routing, request handling, templating , and support for extensions for extra functionality are important characteristics.

5. TESTING

5.1. TEST CASES:

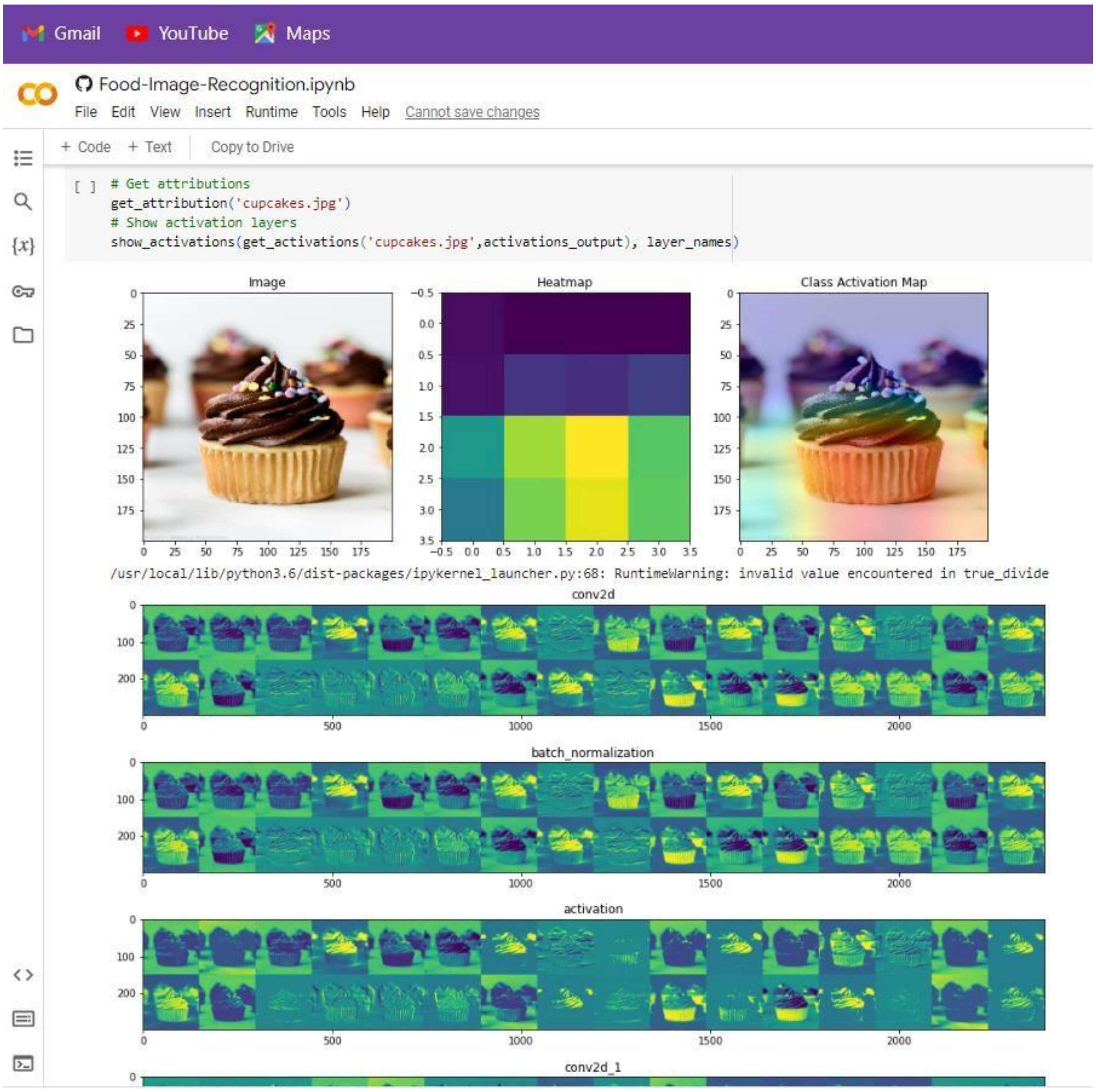
CASE1



```
# Display an image. Brb. Gtg and get a waffle for myself. :P
testImg = imread('/content/food-101/test/waffles/1236559.jpg')
print(testImg.shape)
plt.imshow(testImg)
```

```
(384, 512, 3)
<matplotlib.image.AxesImage at 0x7fc519140e80>
```

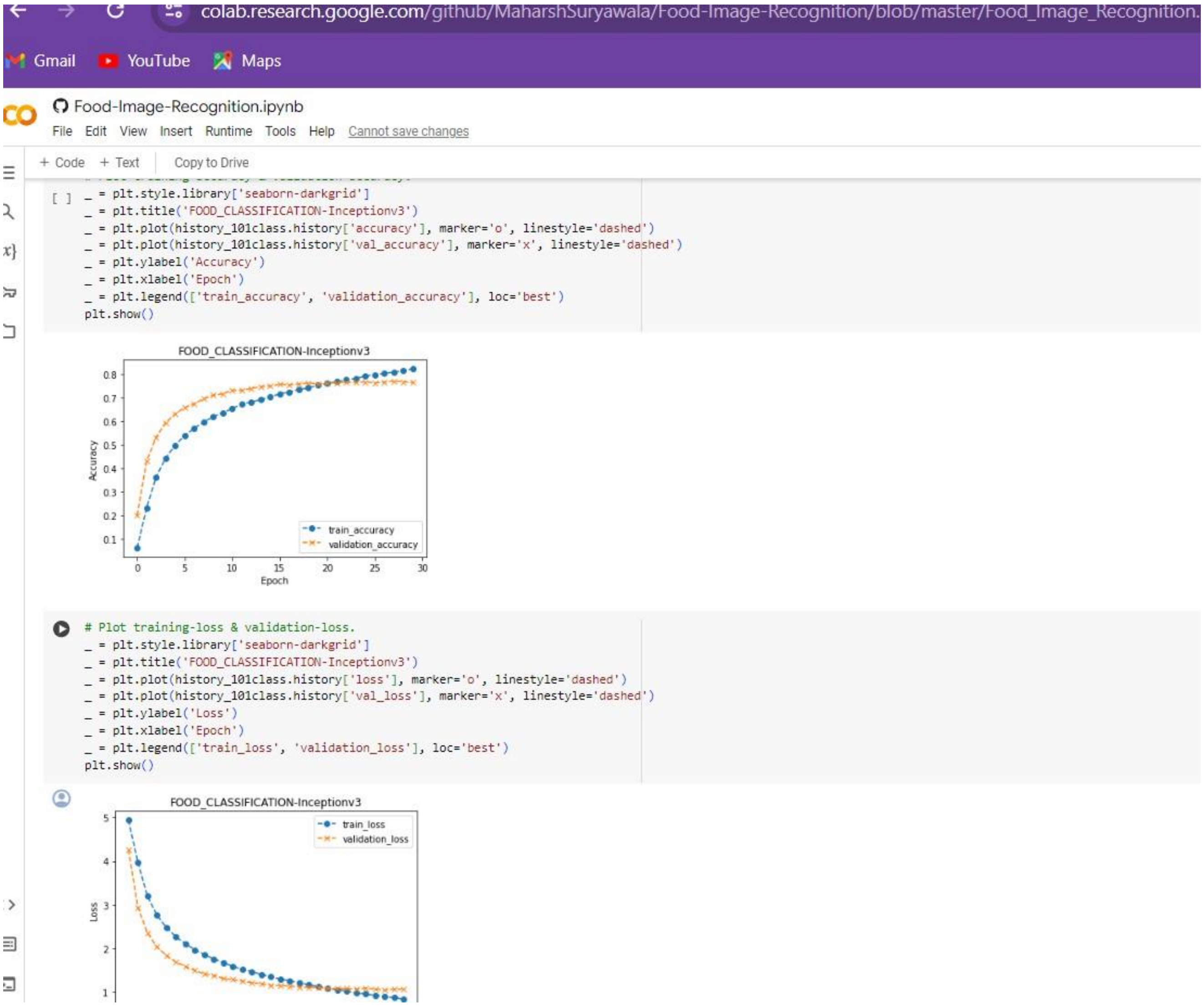
CASE 2


Gmail
YouTube
Maps
Food-Image-Recognition.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
Image
Heatmap
Class Activation Map
conv2d
batch_normalization
activation
conv2d_1

```
# Get attributions
get_attribution('cupcakes.jpg')
# Show activation layers
show_activations(get_activations('cupcakes.jpg',activations_output), layer_names)
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:68: RuntimeWarning: invalid value encountered in true_divide
```

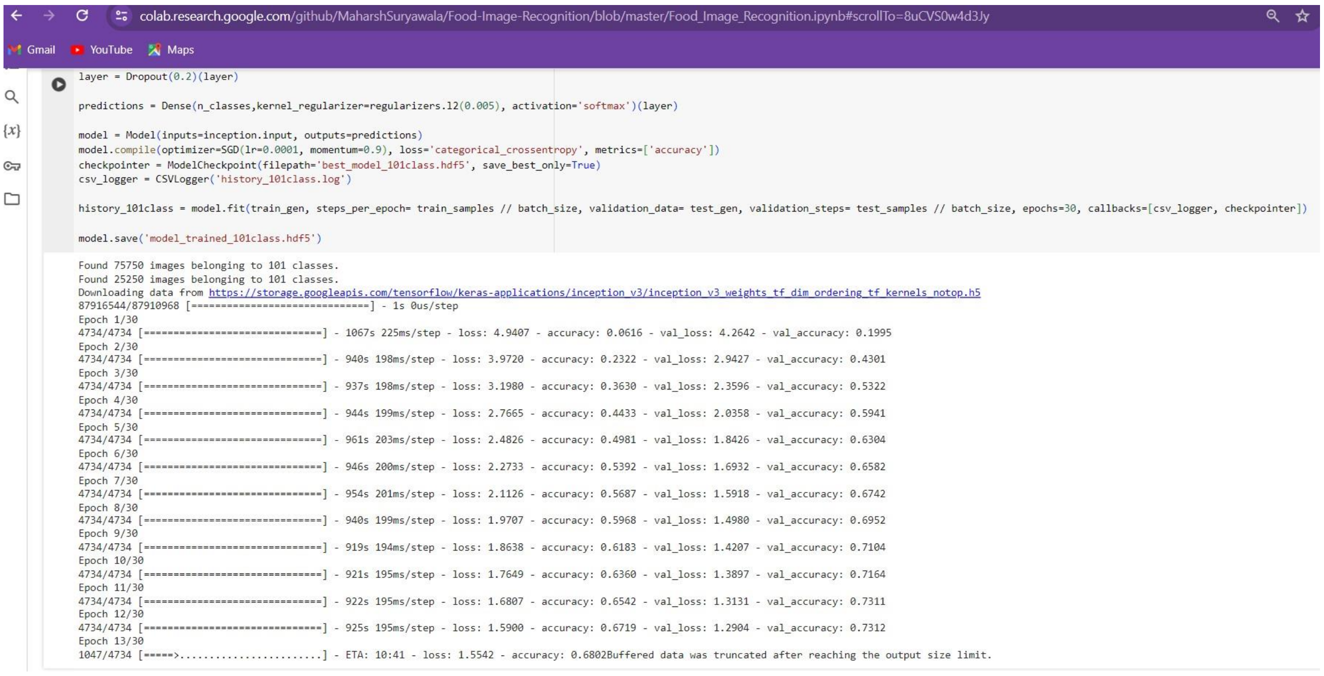
CASE 3


colab.research.google.com/github/MaharshSuryawala/Food-Image-Recognition/blob/master/Food_Image_Recognition.
Gmail
YouTube
Maps
Food-Image-Recognition.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
FOOD_CLASSIFICATION-Inceptionv3
Accuracy
Epoch
train_accuracy
validation_accuracy
Loss
train_loss
validation_loss

```
_ = plt.style.library['seaborn-darkgrid']
_ = plt.title('FOOD_CLASSIFICATION-Inceptionv3')
_ = plt.plot(history_101class.history['accuracy'], marker='o', linestyle='dashed')
_ = plt.plot(history_101class.history['val_accuracy'], marker='x', linestyle='dashed')
_ = plt.ylabel('Accuracy')
_ = plt.xlabel('Epoch')
_ = plt.legend(['train_accuracy', 'validation_accuracy'], loc='best')
plt.show()
```

```
# Plot training-loss & validation-loss.
_ = plt.style.library['seaborn-darkgrid']
_ = plt.title('FOOD_CLASSIFICATION-Inceptionv3')
_ = plt.plot(history_101class.history['loss'], marker='o', linestyle='dashed')
_ = plt.plot(history_101class.history['val_loss'], marker='x', linestyle='dashed')
_ = plt.ylabel('Loss')
_ = plt.xlabel('Epoch')
_ = plt.legend(['train_loss', 'validation_loss'], loc='best')
plt.show()
```

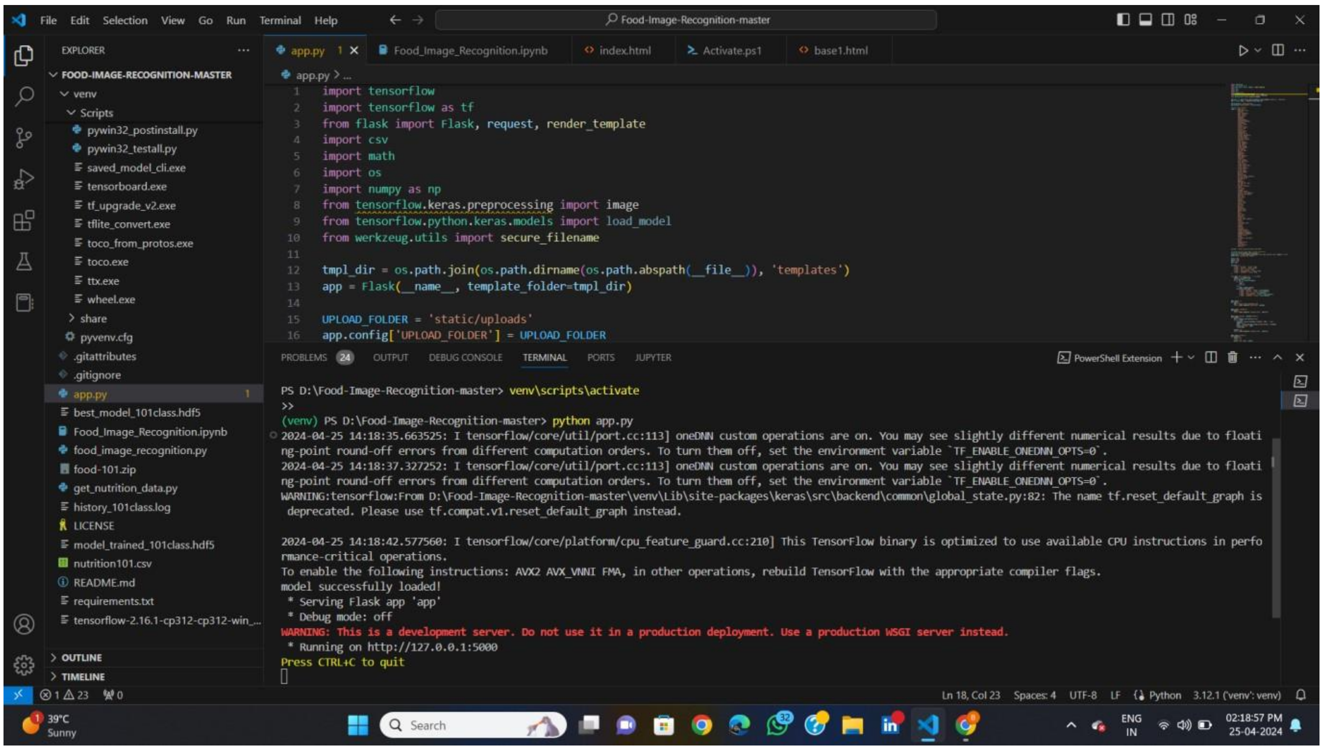
CASE 4


colab.research.google.com/github/MaharshSuryawala/Food-Image-Recognition/blob/master/Food_Image_Recognition.ipynb#scrollTo=8uCVS0w4d3Jy
Gmail
YouTube
Maps

```python
layer = Dropout(0.2)(layer)

predictions = Dense(n_classes,kernel_regularizer=regularizers.l2(0.005), activation='softmax')(layer)

model = Model(inputs=inception.input, outputs=predictions)
model.compile(optimizer=SGD(lr=0.0001, momentum=0.9), loss='categorical_crossentropy', metrics=['accuracy'])
checkpointer = ModelCheckpoint(filepath='best_model_101class.hdf5', save_best_only=True)
csv_logger = CSVLogger('history_101class.log')

history_101class = model.fit(train_gen, steps_per_epoch= train_samples // batch_size, validation_data= test_gen, validation_steps= test_samples // batch_size, epochs=30, callbacks=[csv_logger, checkpointer])

model.save('model_trained_101class.hdf5')
```

```
Found 75750 images belonging to 101 classes.
Found 25250 images belonging to 101 classes.
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/inception_v3/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5
87916544/87910968 [==============================] - 1s 0us/step
Epoch 1/30
4734/4734 [==============================] - 1067s 225ms/step - loss: 4.9407 - accuracy: 0.0616 - val_loss: 4.2642 - val_accuracy: 0.1995
Epoch 2/30
4734/4734 [==============================] - 940s 198ms/step - loss: 3.9720 - accuracy: 0.2322 - val_loss: 2.9427 - val_accuracy: 0.4301
Epoch 3/30
4734/4734 [==============================] - 937s 198ms/step - loss: 3.1980 - accuracy: 0.3630 - val_loss: 2.3596 - val_accuracy: 0.5322
Epoch 4/30
4734/4734 [==============================] - 944s 199ms/step - loss: 2.7665 - accuracy: 0.4433 - val_loss: 2.0358 - val_accuracy: 0.5941
Epoch 5/30
4734/4734 [==============================] - 961s 203ms/step - loss: 2.4826 - accuracy: 0.4981 - val_loss: 1.8426 - val_accuracy: 0.6304
Epoch 6/30
4734/4734 [==============================] - 946s 200ms/step - loss: 2.2733 - accuracy: 0.5392 - val_loss: 1.6932 - val_accuracy: 0.6582
Epoch 7/30
4734/4734 [==============================] - 954s 201ms/step - loss: 2.1126 - accuracy: 0.5687 - val_loss: 1.5918 - val_accuracy: 0.6742
Epoch 8/30
4734/4734 [==============================] - 940s 199ms/step - loss: 1.9707 - accuracy: 0.5968 - val_loss: 1.4980 - val_accuracy: 0.6952
Epoch 9/30
4734/4734 [==============================] - 919s 194ms/step - loss: 1.8638 - accuracy: 0.6183 - val_loss: 1.4207 - val_accuracy: 0.7104
Epoch 10/30
4734/4734 [==============================] - 921s 195ms/step - loss: 1.7649 - accuracy: 0.6360 - val_loss: 1.3897 - val_accuracy: 0.7164
Epoch 11/30
4734/4734 [==============================] - 922s 195ms/step - loss: 1.6807 - accuracy: 0.6542 - val_loss: 1.3131 - val_accuracy: 0.7311
Epoch 12/30
4734/4734 [==============================] - 925s 195ms/step - loss: 1.5900 - accuracy: 0.6719 - val_loss: 1.2904 - val_accuracy: 0.7312
Epoch 13/30
1047/4734 [=====>........................] - ETA: 10:41 - loss: 1.5542 - accuracy: 0.6802Buffered data was truncated after reaching the output size limit.
```

CASE 5

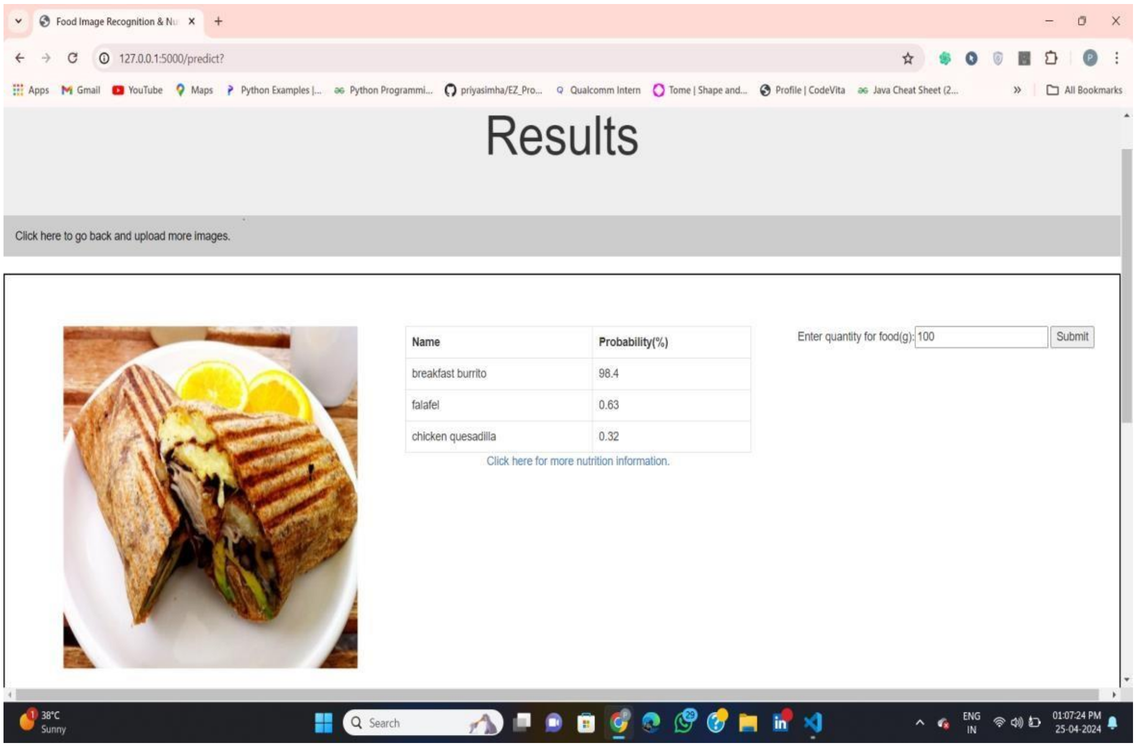


```python
import tensorflow
import tensorflow as tf
from flask import Flask, request, render_template
import csv
import math
import os
import numpy as np
from tensorflow.keras.preprocessing import image
from tensorflow.python.keras.models import load_model
from werkzeug.utils import secure_filename

tmpl_dir = os.path.join(os.path.dirname(os.path.abspath(__file__)), 'templates')
app = Flask(__name__, template_folder=tmpl_dir)

UPLOAD_FOLDER = 'static/uploads'
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
```

```
PS D:\Food-Image-Recognition-master> venv\scripts\activate
>>
(venv) PS D:\Food-Image-Recognition-master> python app.py
2024-04-25 14:18:35.663525: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to floati
ng-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-04-25 14:18:37.327252: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to floati
ng-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
WARNING:tensorflow:From D:\Food-Image-Recognition-master\venv\Lib\site-packages\keras\src\backend\common\global_state.py:82: The name tf.reset_default_graph is
 deprecated. Please use tf.compat.v1.reset_default_graph instead.

2024-04-25 14:18:42.577560: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in perfo
rmance-critical operations.
To enable the following instructions: AVX2 AVX_VNNI FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
model successfully loaded!
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

6. RESULTS

Case1:



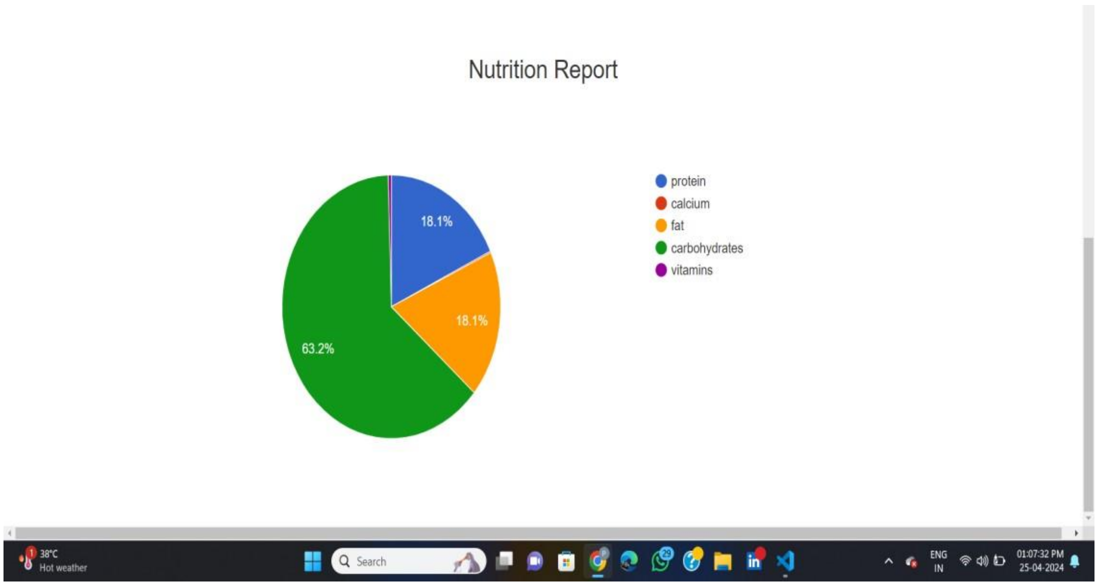
Results

Click here to go back and upload more images.

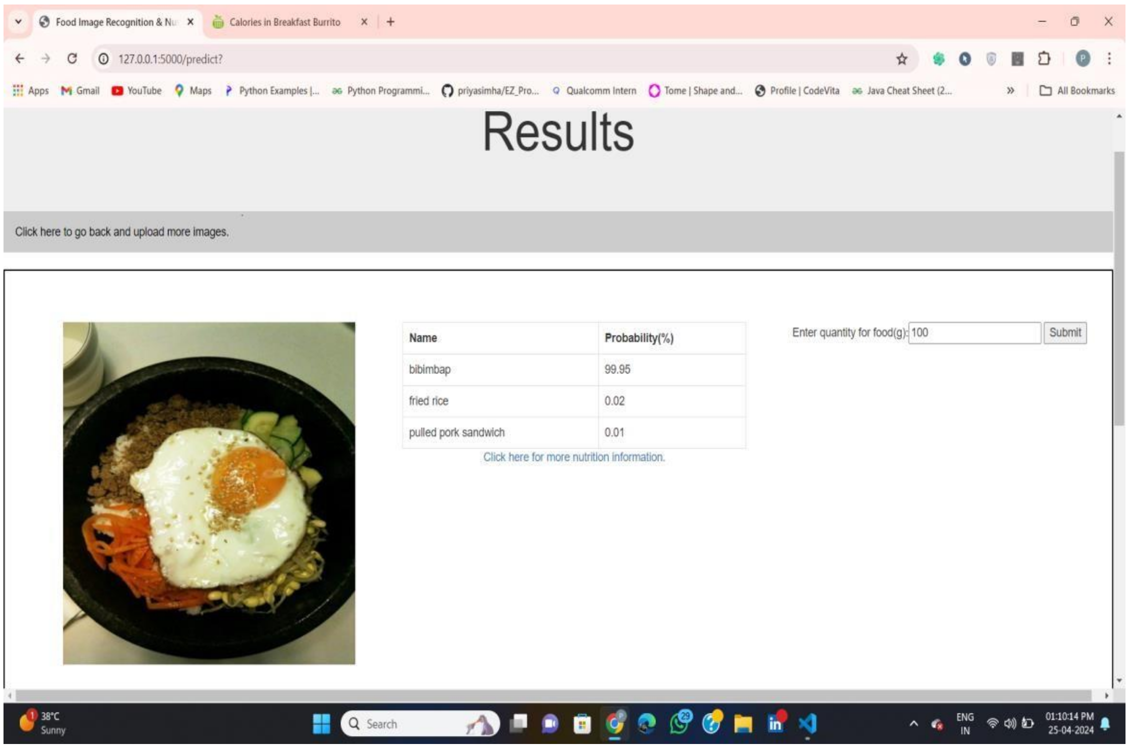
Name	Probability(%)
breakfast burrito	98.4
falafel	0.63
chicken quesadilla	0.32

Click here for more nutrition information.

Enter quantity for food(g): 100 Submit



Case2:



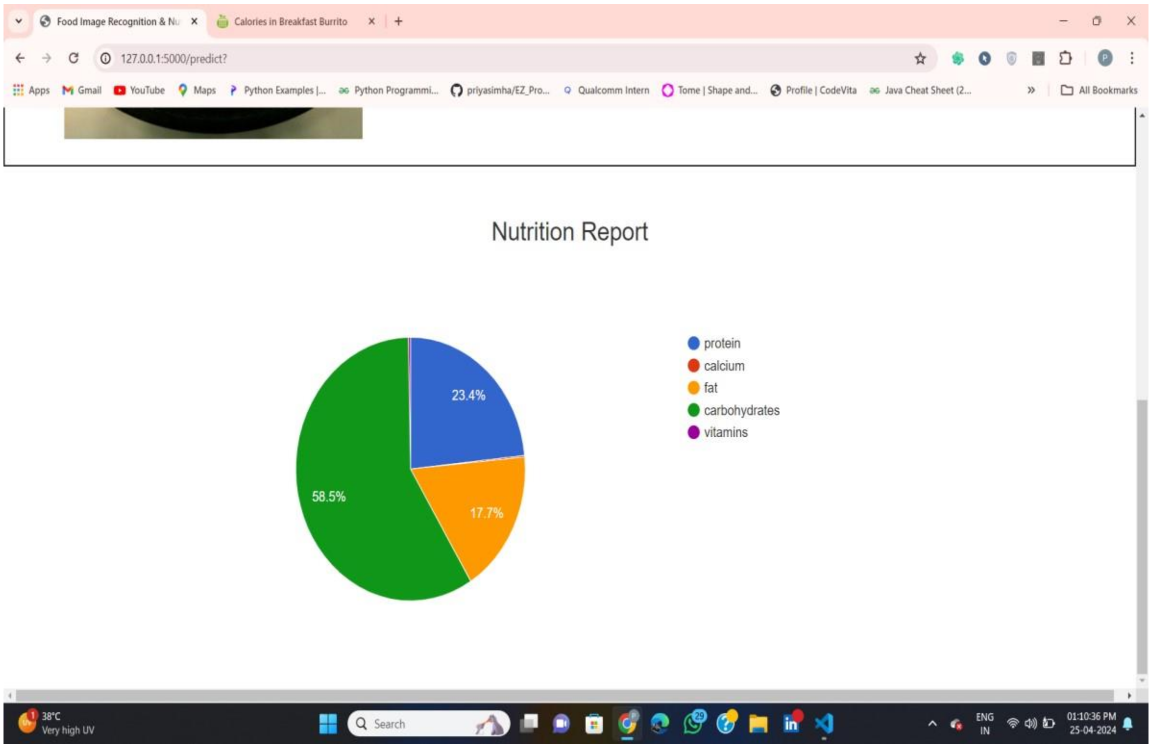
Results

Click here to go back and upload more images.

Name	Probability(%)
bibimbap	99.95
fried rice	0.02
pulled pork sandwich	0.01

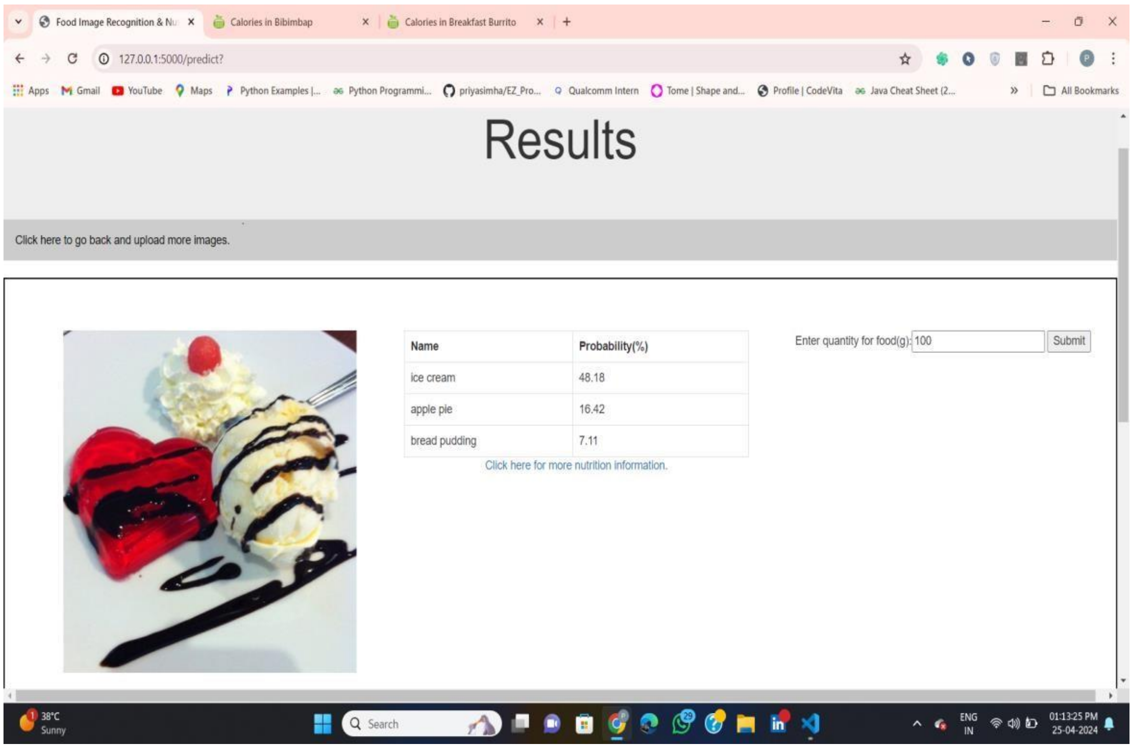
Click here for more nutrition information.

Enter quantity for food(g): 100 Submit



Nutrition Report

Case3:



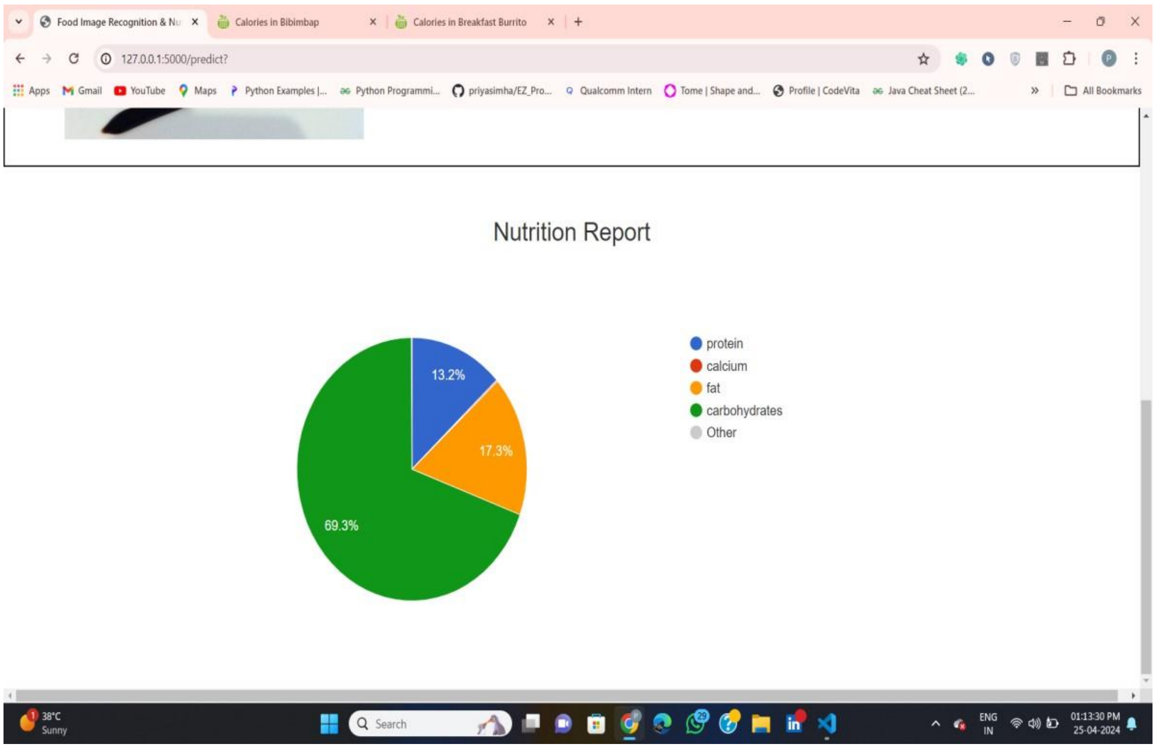
Results

Click here to go back and upload more images.

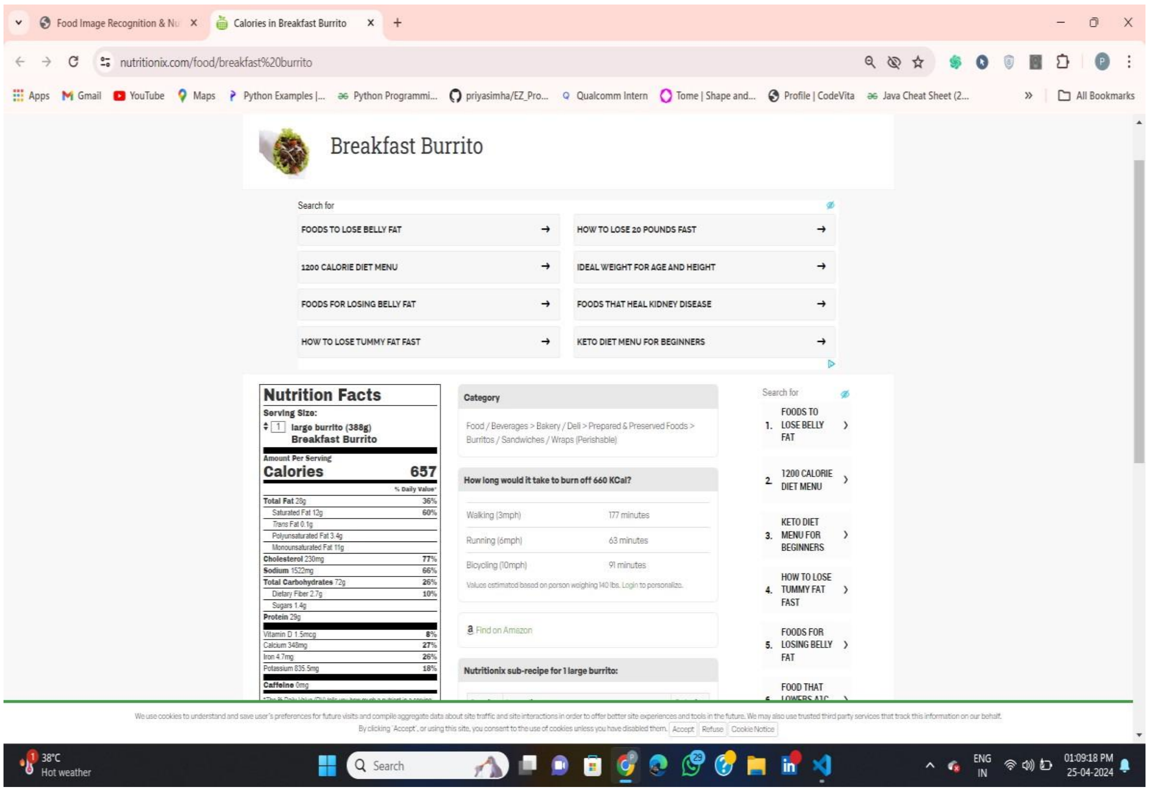
Name	Probability(%)
ice cream	48.18
apple pie	16.42
bread pudding	7.11

Click here for more nutrition information.

Enter quantity for food(g): 100 Submit



Nutrition Report



Food Image Recognition & Nu | Calories in Breakfast Burrito

nutritionix.com/food/breakfast%20burrito

Apps | Gmail | YouTube | Maps | Python Examples |... | Python Programmi... | priyasimha/EZ_Pro... | Qualcomm Intern | Tome | Shape and... | Profile | CodeVita | Java Cheat Sheet (2... | All Bookmarks

Breakfast Burrito

Search for

FOODS TO LOSE BELLY FAT →	HOW TO LOSE 20 POUNDS FAST →
1200 CALORIE DIET MENU →	IDEAL WEIGHT FOR AGE AND HEIGHT →
FOODS FOR LOSING BELLY FAT →	FOODS THAT HEAL KIDNEY DISEASE →
HOW TO LOSE TUMMY FAT FAST →	KETO DIET MENU FOR BEGINNERS →

Nutrition Facts

Serving Size: 1 **large burrito (388g)**
Breakfast Burrito

Amount Per Serving

Calories	**657**
	% Daily Value*
Total Fat 28g	**36%**
Saturated Fat 12g	**60%**
Trans Fat 0.1g	
Polyunsaturated Fat 3.4g	
Monounsaturated Fat 11g	
Cholesterol 230mg	**77%**
Sodium 1522mg	**66%**
Total Carbohydrates 72g	**26%**
Dietary Fiber 2.7g	**10%**
Sugars 1.4g	
Protein 29g	
Vitamin D 1.5mcg	**8%**
Calcium 348mg	**27%**
Iron 4.7mg	**26%**
Potassium 835.5mg	**18%**
Caffeine 0mg	

Category

Food / Beverages > Bakery / Deli > Prepared & Preserved Foods > Burritos / Sandwiches / Wraps (Perishable)

How long would it take to burn off 660 KCal?

Walking (3mph)	177 minutes
Running (6mph)	63 minutes
Bicycling (10mph)	91 minutes

Values estimated based on person weighing 140 lbs. Login to personalize.

Find on Amazon

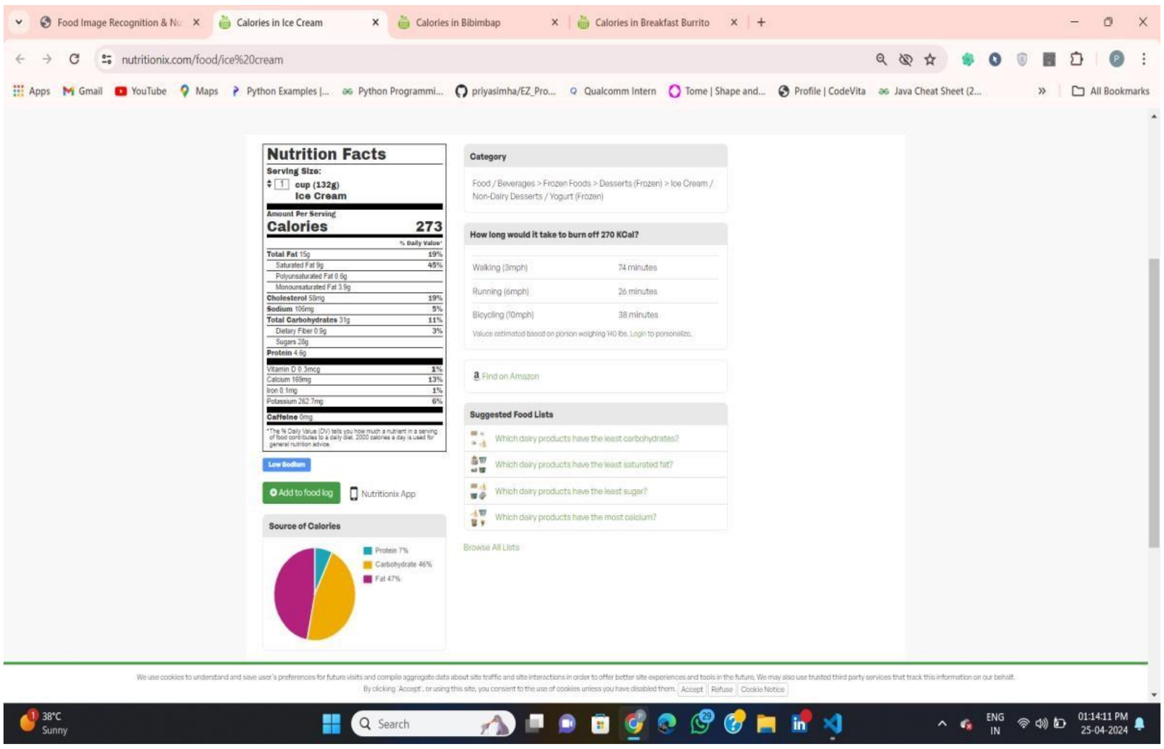
Nutritionix sub-recipe for 1 large burrito:

Search for

1. FOODS TO LOSE BELLY FAT ›
2. 1200 CALORIE DIET MENU ›
3. KETO DIET MENU FOR BEGINNERS ›
4. HOW TO LOSE TUMMY FAT FAST ›
5. FOODS FOR LOSING BELLY FAT ›
6. FOOD THAT LOWERS A1C ›

We use cookies to understand and save user's preferences for future visits and compile aggregate data about site traffic and site interactions in order to offer better site experiences and tools in the future. We may also use trusted third party services that track this information on our behalf. By clicking 'Accept', or using this site, you consent to the use of cookies unless you have disabled them. Accept Refuse Cookie Notice

38°C Hot weather | Search | ENG IN | 01:09:18 PM 25-04-2024



Food Image Recognition & Nu | Calories in Ice Cream | Calories in Bibimbap | Calories in Breakfast Burrito

nutritionix.com/food/ice%20cream

Apps | Gmail | YouTube | Maps | Python Examples |... | Python Programmi... | priyasimha/EZ_Pro... | Qualcomm Intern | Tome | Shape and... | Profile | CodeVita | Java Cheat Sheet (2... | All Bookmarks

Nutrition Facts

Serving Size: 1 **cup (132g)**
Ice Cream

Amount Per Serving

Calories	**273**
	% Daily Value*
Total Fat 15g	**19%**
Saturated Fat 9g	**45%**
Polyunsaturated Fat 0.6g	
Monounsaturated Fat 3.9g	
Cholesterol 58mg	**19%**
Sodium 106mg	**5%**
Total Carbohydrates 31g	**11%**
Dietary Fiber 0.9g	**3%**
Sugars 28g	
Protein 4.6g	
Vitamin D 0.3mcg	**1%**
Calcium 169mg	**13%**
Iron 0.1mg	**1%**
Potassium 262.7mg	**6%**
Caffeine 0mg	

*The % Daily Value (DV) tells you how much a nutrient in a serving of food contributes to a daily diet. 2000 calories a day is used for general nutrition advice.

Low Sodium

Add to food log | Nutritionix App

Source of Calories

🟦 Protein 7%
🟨 Carbohydrate 46%
🟪 Fat 47%

Category

Food / Beverages > Frozen Foods > Desserts (Frozen) > Ice Cream / Non-Dairy Desserts / Yogurt (Frozen)

How long would it take to burn off 270 KCal?

Walking (3mph)	74 minutes
Running (6mph)	26 minutes
Bicycling (10mph)	38 minutes

Values estimated based on person weighing 140 lbs. Login to personalize.

Find on Amazon

Suggested Food Lists

- Which dairy products have the least carbohydrates?
- Which dairy products have the least saturated fat?
- Which dairy products have the least suger?
- Which dairy products have the most calcium?

Browse All Lists

We use cookies to understand and save user's preferences for future visits and compile aggregate data about site traffic and site interactions in order to offer better site experiences and tools in the future. We may also use trusted third party services that track this information on our behalf. By clicking 'Accept', or using this site, you consent to the use of cookies unless you have disabled them. Accept Refuse Cookie Notice

38°C Sunny | Search | ENG IN | 01:14:11 PM 25-04-2024

7. CONCLUSION

Empower the user with a convenient, intelligent, and accurate system that helps them become sensible about their calorie intake. We employed a rather unique combination of graph- cut segmentation and deep learning neural networks as a means of accurately classifying and recognizing food items. The combination of those two methods provides a powerful instrument to achieve 100% accuracy in food recognition in our system. The implementation of the virtualization approach of the application allows us to benefit from cloud-based resources. Future work is to increase our database of images and use the approach presented in this paper to test mixed food portions.

8.FUTURE SCOPE

Future studies will use the bounding box technique to properly estimate an individual's daily calorie intake. To calculate the calories of each meal constituent, crop the areas and calibrate the identified portions based on distance. The next implementation phase will use a calibration technique to accurately calculate the number of calories consumed. This will result in a more efficient and self-sustaining system for use in the health sector.

1. Fine-Grained Recognition
2. Multimodal Integration
3. Portion Size Estimation
4. Personalization and Adaptation
5. Real-Time Feedback
6. Integration with Wearable Devices
7. Healthcare Applications
8. Global Cuisine Coverage
9. Ethical and Privacy Considerations
10. Collaborative Platforms