#### APPENDIX SOURCE CODE

from flask import Flask, render\_template, flash, request, session,send\_file from flask import render\_template, redirect, url\_for, request

import warnings import datetime import cv2

app = Flask( name ) app.config['DEBUG']

app.config['SECRET\_KEY'] = '7d441f27d441f27567d441f2b6176a' @app.route("/")

def homepage():

return render\_template('index.html') @app.route("/Test")

def Test():

return render\_template('NewUser.html') @app.route("/testimage", methods=['GET', 'POST']) def testimage():

if request.method == 'POST': file = request.files['fileupload'] file.save('static/Out/Test.jpg')

img = cv2.imread('static/Out/Test.jpg')

if img is None:

print('no data')

img1 = cv2.imread('static/Out/Test.jpg') print(img.shape)

img = cv2.resize(img, ((int)(img.shape[1] / 5), (int)(img.shape[0] / 5))) original = img.copy()

neworiginal = img.copy() cv2.imshow('original', img1)

gray = cv2.cvtColor(img1, cv2.COLOR\_BGR2GRAY) img1S = cv2.resize(img1, (960, 540)) cv2.imshow('Original image', img1S)

grayS = cv2.resize(gray, (960, 540)) cv2.imshow('Gray image', grayS) gry = 'static/Out/gry.jpg' cv2.imwrite(gry, grayS)

from PIL import ImageOps,Image im = Image.open(file)

im\_invert = ImageOps.invert(im) inv = 'static/Out/inv.jpg' im\_invert.save(inv, quality=95)

dst = cv2.fastNlMeansDenoisingColored(img1, None, 10, 10, 7, 21) cv2.imshow("Nosie Removal", dst)

noi = 'static/Out/noi.jpg' cv2.imwrite(noi, dst) import warnings

warnings.filterwarnings('ignore')

import tensorflow as tf

classifierLoad = tf.keras.models.load\_model('model.h5')

import numpy as np

from keras.preprocessing importimage

test\_image = image.load\_img('static/Out/Test.jpg', target\_size=(200, 200)) img1 = cv2.imread('static/Out/Test.jpg')

# test\_image = image.img\_to\_array(test\_image) test\_image = np.expand\_dims(test\_image, axis=0) result = classifierLoad.predict(test\_image) print(result)

out = ''

fer = ''

if result[0][0] == 1: out = "APPLES"

fer = '52 calories/1per' elif result[0][1] == 1: out = "BANANA"

fer = '100 calories/1per' elif result[0][2] == 1: out = "ORANGE"

fer = '50 calories/1per' elif result[0][3] == 1: out = "PINEAPPLE"

fer = '60 calories/1per' elif result[0][4] == 1:

out = "WATERMELON"

fer = '400 calories/1per' org = 'static/Out/Test.jpg'

return render\_template('NewUser.html',fer=fer,result=out,org=org) if name == ' main ':

app.run(debug=True, use\_reloader=True)