VOLUME ESTIMATION PROBLEM

ZAZA MARIA ELENA, PERNA GRAZIA

Abstract **This project aims to estimate volume, calories and weight from food images. We used the pre-trained model Faster RCNN from Tensorflow-hub to detect the type of food and the OneYuan coin in the image. The coin plays an important role because it helps us to calculate the volume of the food. During the process, GrabCut was used for cutting the box containing the food. Finally, we did the estimations. For doing a final evaluation, we compared our results with the scores contained on a file called "density.xls". The experiment results showed that our estimation method is pretty effective, but mostly for some category of food.**

I. INTRODUCTION

In 2022, the distribution of body-mass-index (BMI) across Italy varied greatly by region. According to the data, southern regions had a higher share of overweight and obese people compared to the national average. Overall, the overweight population in Italy is projected to reach 68 percent by 2025. The Italian regions with the highest share of people considered as having a normal weight in 2022 were Lazio, Trentino-South Tyrol and Veneto. Conversely, the region of Aosta Valley hosted the most underweight people in the country, in relative terms, with 5.7 percent. The number of individuals suffering from diabetes in Italy amounted to 3,888 in 2022. Although the risk factors related to type one diabetes are not fully known, among the risk factors for diabetes type 2, being overweight or obese are among the most common. Indeed, in 2021, almost 17 percent of obese women were also diabetic. This rate lowers to 14.1 percent for men.

Also Childhood obesity is becoming an issue in the country, with the share of overweight and obese children growing every year. Indeed, Italy has become one of the European countries with the highest obesity rate among children. This tendency is more prevalent among young boys, with 29.8 percent of male minors overweight between 2019 and 2021, compared to 24 percent of females.

For these situations, careful monitoring of the food consumed by the individual is necessary, in order to minimize the risks and to give the possibility to medical-health personnel to improve the conditions of a patient's life. These considerations have led to the development of assisted and automated monitoring strategies through the use of portable and wearable devices equipped with artificial vision algorithms. However, looking for solutions for the automatic recognition of foods, the estimation of the volume, weight and the subsequent calculation of the calories represents a problem that is anything but trivial, given the extremely variable nature of the food presented once it has been processed, cooked and served.

A computer vision project that estimates the volume of foods could help fight obesity and related health problems in several ways. For example:

- Estimating the volume of foods can help people be more aware of their calorie intake. This can be especially helpful for those dieting or trying to maintain a healthy body weight.
- Obesity is often associated with eating too many portions. A computer vision application that informs people that they are consuming too large portions can be an educational tool to promote portion control.
- People with specific dietary needs, such as diets based on calorie restrictions or specific nutritional needs, can benefit from an app that tracks their food choices precisely and provides personalized suggestions.
- In clinical settings, an app that estimates food volume can be used by doctors and nutritionists to monitor patients' eating behavior remotely, offering advice and support when needed.

For this reason, our work has focused on the volume and weight estimation problem and on the calculation of calories. From an algorithmic point of view, the first step is to understand what food is represented and then dedicate the study on the estimation of the volume. Research on this last issue were therefore conducted.

II. RELATED WORK

This section reviews three relevant studies that have employed different methodologies and models for volume estimation problem.

One important work has been done by Yanchao Liang and Jianhua Li supported by National Natural Science Foundation of China. They created a very useful dataset for volume estimation problem called "ECUSTFD" which has 19 types of food: apple, banana, bread, bun, doughnut, egg, fired dough nut, grape, lemon, litchi, mango, mooncake, orange, peach, pear, plum, qiwi, sachima, tomato and the number of food images is 2978. Each image contains a coin, which is the calibration object, and they provide food's volume and quality information useful to test the final results. Their results show that for most types of food in their experiment, the estimation volume is closer to reference volume. The mean error between estimation volume and true volume does not exceed ±20% except banana, grape, mooncake.

Another important work has been done by Dario Allegra et al. who has proposed an approach focused on distinguishing between images depicting food and those depicting other types

of content. This distinction is treated as a binary classification problem using the "One-Class Classification" paradigm to test the effectiveness of the proposed approaches. Two datasets were collected: one containing 4805 images of food and the other containing 8005 images representing other contents.

The problem of recognizing and classifying food is complex due to the great variety of ingredients, shapes, textures and colors present in food images After proving different approaches, starting from “Texton” they have reached the best results using “Anti-Texton” that captures the spatial relationships between Textons. This approach resulted in an accuracy of 92.60% in classifying scope types and 86.27% in classifying suggested cutlery types.

Manika Puri et al has proposed a system that improves accuracy of food intake assessment using computer vision technique. Their solution is to use a mobile phone to capture images of foods, recognize food types, estimate their respective volumes and finally return quantitative nutrition information. The use of low-quality images from mobile phones also makes 3D reconstruction difficult. This article addresses these challenges by combining different vision techniques, including visual recognition and 3D reconstruction, and provides experimental evaluations of the obtained results. Their dataset includes 150 common food types, and they are working to expand it. The mean error by volume is 5.75(± 3.75)% across all sets.

III. DATASET

The dataset used for this work is “ECUSTFD” dataset, which comprises 19 types of food, including apple, banana, bread, bun, doughnut, egg, fried doughnut, grape, lemon, litchi, mango, mooncake, orange, peach, pear, plum, qiwi, sachima, and tomato. The dataset contains a total of 2978 food images. Each type of food is represented by varying numbers of images and objects.

For a single food portion, multiple pairs of images were captured using smartphones. Each group of images includes a top view and a side view of the food, in **Figure 1 and Figure 2** there are two examples of photos. Each image contains a One Yuan coin, that is as we said before is the calibration object.

Figure 1 Apple top view

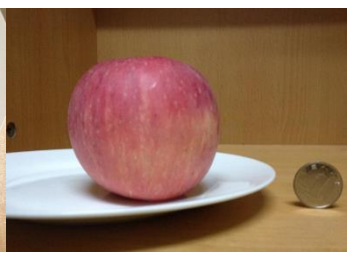

Figure 2 Apple side view

Two datasets are provided: one containing original images and another with resized images. In the resized dataset, each image is less than 1000×1000 pixels in size. Another important information supplied is a table which contains the number of images, the number of objects, density (g/cm^3) and energy content (kcal/g) for each food type in the ECUSTFD dataset.

Additionally, the dataset includes the following information for each image:

- Annotation: Bounding boxes are provided for each object in the images, with separate annotations for resized images (original images were not annotated due to their high resolution).
- Mass: Mass information for each food item, measured using an electronic scale, is included.
- Volume: Volume information is provided as a reference. Volumes were measured using the drainage method.
- Density and Energy: To estimate calorie content, density and energy information is provided. Density is calculated using volume and mass data from ECUSTFD. Energy values for each food type are obtained from a nutrition table.

IV. LIBRARIES

Here's an explanation of the libraries used in our code:

- *Tensorflow* (tf) is an open-source framework for machine learning and deep learning. It is used to train and implement neural networks.
- *Tensorflow_hub* (hub) is a library that provides pre-trained machine learning models and model parts (called modules) for specific purposes. These modules can be reused in several learning transfer applications. It is useful for loading pre-trained models, in our case is used for Faster RCNN model.
- *Matplotlib.pyplot* (plt) is a Python visualization library. "pyplot" is a Matplotlib-specific module that offers a wide range of functions for creating graphs and plots. It is used to display images.
- *Numpy* (np) is a fundamental library for scientific computing in Python. It provides data structures for representing multidimensional arrays and functions for operating on them. It is often used in conjunction with TensorFlow for data manipulation.
- *PIL* (*Python Imaging Library)* is used for image manipulation and processing. The specific modules listed offer functionality for creating, manipulating and drawing on images. For example, ImageOps can be used for image processing operations such as cropping or color conversion.

- *Time* is a standard Python library used for time management.
- *tempfile* provides functions for creating temporary files and folders. It is often used when it is necessary to manipulate temporary data during program execution.
- *cv2 (OpenCV)* is a widely used open-source library for computer vision and image processing. It is used for image processing operations, such as loading images or processing them before analysis.

V. FASTER RCNN: FOOD DETECTION

To detect food in images, we use Faster Region-based Convolutional Neural Networks (Faster R-CNN) and Grab Cut as segmentation algorithms.

Faster RCNN is a framework based on deep convolutional networks. It includes a Region Proposal Network (RPN)and an Object Detection Network. When we put an image as input, we will get a series of bounding boxes. For each bounding box created by Faster R-CNN its class is judged. It's an Object detection model trained on Open Images V4 with ImageNet pre-trained Inception Resnet V2 as image feature extractor. The maximal number of detections outputted is 100. Detections are outputted for 600 boxable categories.

The output dictionary contains:

- *detection_boxes*: a tf.float32 tensor of shape [N, 4] containing bounding box coordinates in the following order: [ymin, xmin, ymax, xmax].
- *detection_class_entities*: a tf.string tensor of shape [N] containing detection class names as Freebase MIDs.
- *detection_class_names*: a tf.string tensor of shape [N] containing human-readable detection class names.
- *detection_class_labels*: a tf.int64 tensor of shape [N] with class indices.
- *detection_scores*: a tf.float32 tensor of shape [N] containing detection scores.

To estimate calories, the user should insert a top view and a side view of the food. At the end, the volume of each food is calculated based on the calibration objects in the images and then the calories of each food is obtained by using the values in the density and nutrition table.

In this first part of our project, we implemented these main functions for food detection:

- *resize_image*: this function resizes the image using specified height and weight. It returns the resized image.
- *draw_bounding_box_on_image*: this funtion draws bounding box, a box around the detected object.
- *run_detector*: this function takes the pre-trained model and the path of the image we need to analyze.

VI. GRABCUT: IMAGE SEGMENTATION

For what concerns the image segmentation, to identify the food's bounding box contour in the photos, we decided to use the Grabcut algorithm.

This algorithm takes in input the image with detections and bounding boxes, provided by the Faster-RCNN, and then shows the cut image representing only the food detected. The image segmentation is an important step because we take from the function *run_grabcut* the coordinates of the bounding box. These dimensions (ymin, xmin, ymax, xmax) are utile in the last part of our project for the estimation problem.

This method is applied to both the side and the top images.

VII. VOLUME AND CALORIES ESTIMATION

In the last part of our project we compute the volume, the weight and the calories of the detected food using the bounding box coordinates.

This part converts from pixels to millimeters using a known reference point (in this case the diameter of the currency).

The variables *coin_diameter_mm* and *coin_diameter_pixel* represent the diameter of a coin in millimeters (25.0 mm) and the corresponding diameter in pixels on an image (94.49 pixels), respectively (***Figure 3***). These values are used as a reference point for converting the object's dimensions.

Figure 1 One Yuan Coin

Then *top_coin_height_mm* and *side_coin_height_mm* were calculated. They represent the height of an object in millimeters in the two views. The height is calculated by multiplying the difference between the object's ymax and ymin

by the coin's diameter in millimeters and then dividing by the coin's diameter in pixels. This calculation converts the height of the object from pixels to millimeters.

The conversion factors *conversion_factor_top* and *conversion_factor_side* are used to convert other dimensions of the regions of interest (bounding boxes) into millimeters, making the measurement results more meaningful.

The conversion factors were useful to calculate *top_view_box_width_mm* and *top_view_box_height_mm*, which represent the size (in millimeters) of the bounding box area in the top view and in the side view (*side_view_box_width_mm* and *side_view_box_height_mm*). To calculate the approximated volume of the object we multiplied the dimensions by each other (*top_view_box_width_mm, side_view_box_height_mm, top_view_box_height_mm*) and then proceeded with dividing it by 1000 to get the volume in cubic centimeters (cm^3).

Instead, the weight estimation is given by multiplying the volume by the specific weight of the food. The result represents the estimated weight of the object in grams (g).

For the energy calculation we take the value of the energy associated with the object (the calorie content of the food) and we multiply it by the weight.

In the table (***Figure 4***) the values of density and energy for each food are shown. In this project we used only 10 type of foods: Banana, Apple, Doughnut, Egg, Grape, Lemon, Mango, Orange, Peach and Pear because in the label map used ("*oid_v4_label_map.pbtxt*") doesn't contain all the types of food present in our dataset.

Food Type	The number of images	The number of objects	Density (g/cm^3)	Energy $(kcal/g)$
apple	296	19	0.78	0.52
banana	178	15	0.91	0.89
bread	66	7	0.18	3.15
bun	90	8	0.34	2.23
doughnut	210	9	0.31	4.34
egg	104	7	1.03	1.43
fired dough twist	124	7	0.58	24.16
grape	58	2	0.97	0.69
lemon	148	4	0.96	0.29
litchi	78	5	1.00	0.66
mango	220	10	1.07	0.60
mix	108	14	/	/
mooncake	134	6	0.96	18.83
orange	254	15	0.90	0.63
peach	126	5	0.96	0.57
pear	166	6	1.02	0.39
plum	176	4	1.01	0.46
qiwi	120	8	0.97	0.61
sachima	150	5	0.22	21.45
tomato	172	4	0.98	0.27

Figure 2 Food information

VIII. RESULTS

The results achieved are very close to the real ones. To check the validity of these results we used the file *"density"*, provided by the project of Liang-yc in which is showed the volume and the weight of each food.

During our research , we found that in the file the unity of measure of volume is not mm^3, as shown in the table, but is in cm^3 because our results in cm^3 correspond to the ones in mm^3. For some type of food, like banana, grape (because the pretrained model considers only a grape, not the whole fruit), the estimation is not correct as the other ones due to the food's shape that differs a lot from the box's shape. The error between estimation volume and true volume does not exceed ±20% except banana, grape and doughnut.

IX. DEMO

In addition to the image analysis, a very simple graphical user interface (GUI) has been implemented in order to facilitate the input of images for object detection and subsequent volume, weight and calorie estimation. The GUI provides a user-friendly way to upload images (contained in the Demo\JPEGImages\T and Demo\JPEGImages\S) and start the object detection process. Users can simply select the images and obtain information about the detected objects.

The Demo's script is inside the Demo folder.

Note that it is very important to select sequentially the top and the side view (or in reverse).

REFERENCES

(https://iplab.dmi.unict.it/ragusa/pdf/Ital_IA__Food.pdf)

(https://ieeexplore.ieee.org/abstract/document/5403087)

(https://github.com/Liang-yc/ECUSTFD-resized-)

(https://pan.baidu.com/s/1dF866Ut#list/path=%2Fcalorie%20estimation%2FECUSTFD_origin&parentPath=%2Fcalorie%20estimation)

(https://arxiv.org/pdf/1705.07632v3.pdf)

(https://www.statista.com/statistics/727910/distribution-of-body-mass-index-by-region-italy/#:~:text=In%202022%2C%20the%20distribution%20of,reach%2068%20percent%20by%202025)

(https://www.statista.com/statistics/727910/distribution-of-body-mass-index-by-region-italy/#:~:text=In%202022%2C%20the%20distribution%20of,reach%2068%20percent%20by%202025)