Group 42:
Food recognition and leftover estimation

Summer Project - Computer Vision 22/23

Nicola Lorenzon[1, 2], Daniele Moschetta[1, 3], and Carmine Graniello[1, 4]

[1]Information Engineering Department (DEI) - University of Padua
[2]nicola.lorenzon.3@studenti.unipd.it
[3]daniele.moschetta@studenti.unipd.it
[4]carmine.graniello@studenti.unipd.it

October 9, 2024

Contents

1 Introduction

In this Section, it is described the main purpose of this project and how the developed system has been evaluated.

1.1 Problem Specifics

The aim of this project was to develop a Computer Vision system able to perform an analysis of some canteen's food trays in order to detect, segment and estimate the amount of leftovers of thirteen types of foods: pasta with pesto, pasta with tomato sauce, pasta with meat sauce, pasta with clams and mussels, pilaw rice with peppers and peas, grilled pork cutlet, fish cutlet, rabbit, seafood salad, beans, basil potatoes, salad, bread.
It has been developed a system that, given in input two string representing: the path of the food_image and the path of the leftover, localize and recognize all possible food items in the two images, segment each localized food item, computes the food leftover amount.
Problem specifics:
There was given to us a dataset "Food_leftover_dataset" which consists of 8 different trays of food, each containing a first course, second course, side dish, and possibly salad and bread. For each tray, a food_image and several leftover images are provided. The benchmark dataset contains three leftover images for each tray, the three leftover images represent a level of difficulty:

1. Dishes and objects in the same position on the tray for the food_image and leftover images
2. Dishes and objects in a different order on the tray between the food_image and leftover images, but food only partially eaten (i.e., looking very similar between the two images)
3. Dishes and objects in a different order on the tray between the food_image and leftover images and minimal leftover food.

For each image in the dataset, we have:

- *.png file containing the segmentation masks previously found.
- *.txt file listing all the bounding boxes of each food item, saved with a specific text format.

The system produces, for each food_image and leftover images, the segmentation mask image and the bounding boxes file and saves them. The format of the files is the following:

- Bounding boxes file: every food item is found inside a rectangle defined by 4 parameters [x, y, width, height], where (x,y) are the top-left corner coordinates and width and height are the bounding box main dimensions; such parameters are listed in a row, one food item per row; each row contains also an additional parameter which describes the food category ID.
- Segmentation mask: each pixel is assigned a food category ID.

It also computes the Leftover Estimation as explained in Section 2.3.

1.2 Performance evaluation

The system is tested with three different metrics on thirty-two test images.
The metrics used are:

- Food localization (mAP), explained in Section 3

- Food segmentation (mIoU), explained in Section 3
- Food Leftover Estimation Difference (LED)

2 Methodology

In this Section, we will describe the methodology used in our solution.

The program takes in input two strings representing the paths of the images of trays. For each pair of trays, is it's been done the following steps:

1. food_image
   (a) Detect the plates as described in chapter 2.1.1.
   (b) For each plate:
      i. Find the food patch inside as described in Section 2.2.1.
      ii. Segment the food patch as described in Section 2.2.2.
      iii. Find bread as described in Section 2.1.3.
      iv. Segment bread based on the previous detection as described in Section 2.2.4.
      v. Find salad bowl as described in Section 2.1.2
      vi. Segment salad inside the bowl as described in Section 2.2.3.
   (c) Create the final food_image segmentation mask and save it as **.jpg* gray scale image.
   (d) Computes the bounding boxes for each label in the segmentation mask and saves the coordinates in a **.txt* file.
2. leftover
   (a) Detect the plates as described in Section 2.1.1.
   (b) For each plate:
      i. Find the food patch inside as described in Section 2.2.1.
      ii. Segment the food patch as described in Section 2.2.2, knowing the expected foods.
      iii. Find bread as described in Section 2.1.3, knowing whether the bread was present in the food_image tray.
      iv. Segment bread based on the previous detection as described in Section 2.2.4.
      v. Find salad bowl as described in Section 2.1.2.
      vi. Segment salad inside the bowl as described in Section 2.2.3.
   (c) Create the final tray segmentation mask and save it as **.jpg* grayscale image.
   (d) Computes the bounding boxes for each label in the segmentation mask and saves the coordinates in a **.txt* file.
3. Computes the leftover estimation as described in Section 2.3.

2.1 Food Finding

2.1.1 Plates

To find plates it's been exploited the HoughCircles method from OpenCV to develop `FoodFinder::findSaladBowl`.

The pipeline to detect plates is the following one:

1. Compute `circlesPlate`, `circlesSalad`, `refineSalad`.
   - circlesPlate: compute a rough over-estimation of circles plate with parameters A_1, detecting all plates and more circles referring to salads.
   - circlesSalad: compute an underestimation of the circles of salads with parameters A_2, detecting not all of them.
   - refineSalad: compute a more accurate detection of salads with parameters A_3, with a focus on the smaller and distorted ones.
2. Refine circlesPlate removing circlesSalad
3. If the number of plates left (after step 2) is greater than two, then it means that there are other salads plate to remove. The function removes then the plates in refineSalad.

The results on tray1, tray2, tray3, tray4 is the following:

Figure 1: Result of plate detection in first four trays. From the bottom tray1, tray2, tray3, tray4. From right to left food_image, leftover1, leftover2, leftover3.

2.1.2 Salad

To find plates circles it's been developed `FoodFinder::findSaladBowl` exploiting the `HoughCircles` method from OpenCV.
The pipeline to detect plates is the following one:

1. Compute the grayscale image of the source image for which we want to detect the salads
2. Compute `circlesSalad`, a fist detection of the bowls with `HoughCircles` with parameters B_1.

- If it is found one plate or, if the method is computed on the leftover image (after meal) and in the food_image (before meal) it is not found any bowl, then it returns `circlesSalad`.
- Else, it means that the `HoughCircle` found more than one plate of that it found no plates but in the food_image there was. A more coarse `HoughCircle` is computed to find the plates. The found plates are then removed from the previous plate vector.

The result on tray1, tray2, tray3, tray4 is the following:

Figure 2: Result of salad bowl detection in first four trays. From the bottom tray1, tray2, tray3, tray4. From right to left food_image, leftover1, leftover2, leftover3.

2.1.3 Bread

To find plates it's been exploited a series of methods and thresholds from OpenCV to develop `FoodFinder::findBread`.
This method has been developed to create a segmentation mask from which it is possible to compute a rectangular bounding box for the following bread segmentation method.
The pipeline to detect bread is the following one:

1. Fill the plates circles, found with `FoodFinder::findSaladBowl` and `FoodFinder::findPlates` increased with a factor of 1.4.
2. Converts the image to YUV color space.
3. Thresholds the U chrominance blue-difference channel.
4. Makes a closure with an 11x11 rectangular kernel.
5. Computes components (blobs) statistics and keeps the largest component only.

6. Computes Canny Image of the source image inside the mask found in the previous point.
7. Computes the convolution of the Canny Image and a 7x7 kernel and threshold the convoluted image based on the number of edges point found.
8. Performs a series of dilation, closure and erosion with `cv::morphologyEx` function and rectangular kernels.
9. Converts the image to HSV color space.
10. Thresholds the S saturation channel.
11. Performs an erosion and a dilation.
12. Computes components (blobs) statistics and thresholds the components to a given area value. Keeps the largest component.

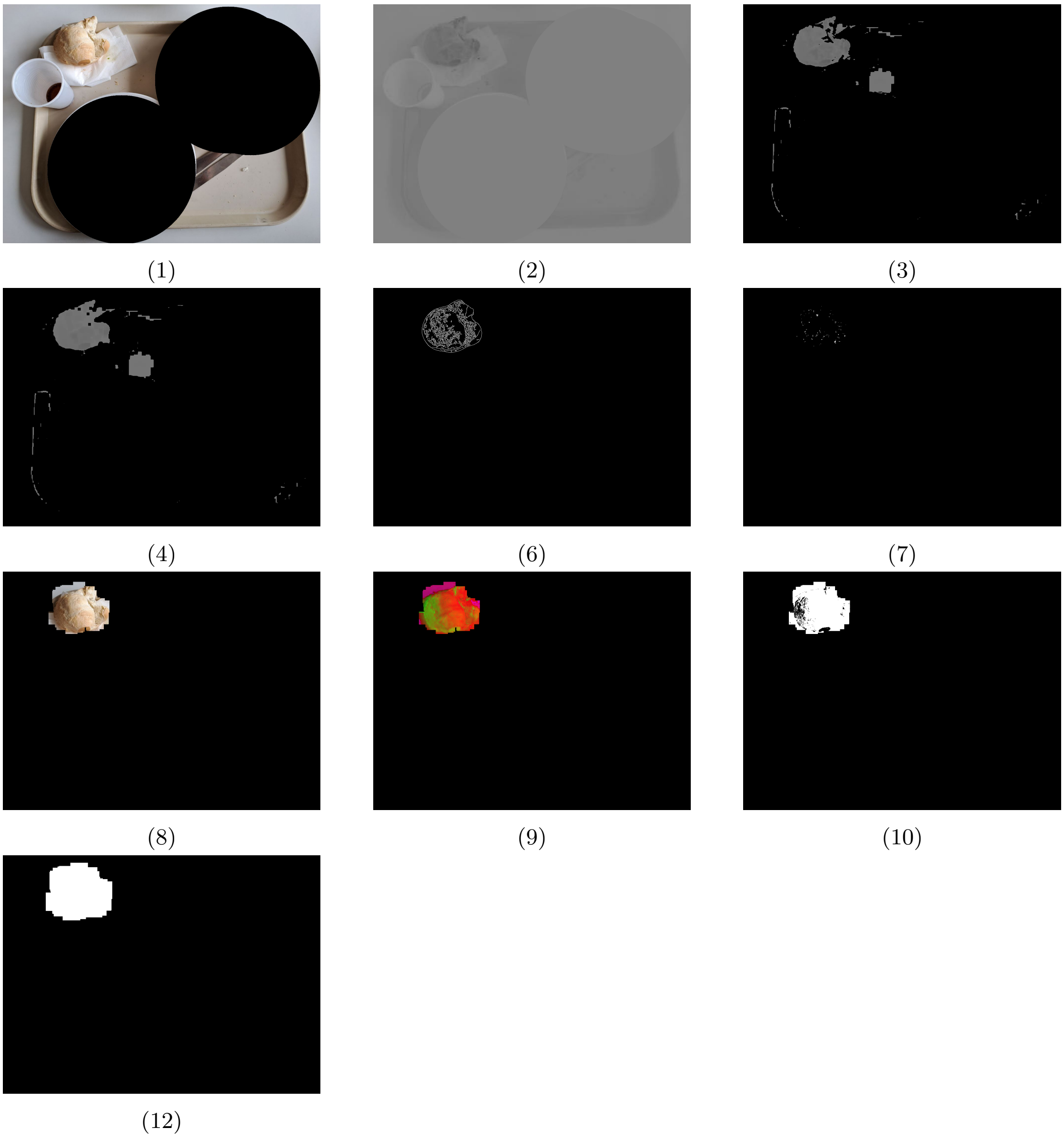


Figure 3: Bread finder pipeline.

The function returns then the `cv::Mat` with the bread rough detection.
The result on the trays with the bread is the following:

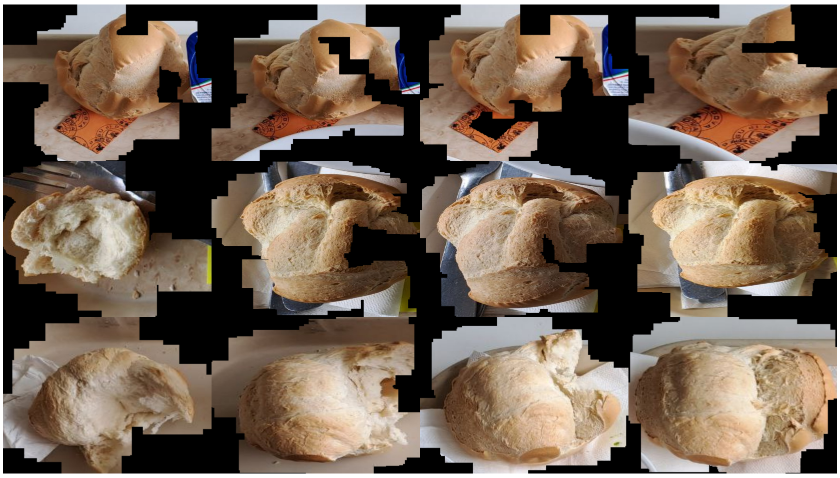


Figure 4: Result of bread detection in all the trays with bread.

2.2 Food segmenting

In this section, we will describe how the food is segmented after having found the areas of the tray where it is present. The segmenting functionalities are implemented in the `FoodSegmenter` and `HistogramComparator` libraries.

2.2.1 Plates

The `FoodSegmenter` library allows to segment food masks from the plates by means of the `FoodSegmenter::getFoodMaskFromPlate` method.

The `FoodSegmenter::getFoodMaskFromPlate` method, given the tray image and the position of the plate, masks the food discriminating its pixels from the ones of the plate itself. In particular, the process is composed of the following steps:

1. The image is pre-processed using Gaussian blur;

2. The image is converted to HSV color space and the pixels inside the plate are scanned and added to the food mask if they satisfy a >80 threshold on the Saturation channel (to ignore plate pixels) or a >20 and <25 threshold on the Hue channel (to avoid lighter basil potatoes areas);

3. The mask contours are scanned in order to check if there is a big contour and discard contours having a negligible area with respect to the plate area;

4.
   a. If a big contour has been found, only that contour is kept on the mask;
   b. Otherwise, the two biggest contours are kept if they are not too far from the mask center with respect to the plate radius;

5. Since the obtained mask can have a lot of holes, the closing morphological operator is applied;

2.2.2 Food in plates

The segmenting process of the plate areas is exploited by methods of the `FoodSegmenter` and `HistogramThresholder` libraries that will be described in this Section.

HistogramThresholder The `HistogramThresholder` library allows to compare a food patch with manually defined thresholds in order to discriminate the food present in the given patch.

In order to discriminate the different dishes it has been used the HSV color space, evaluating the Hue channel of the images. The information given by the Hue channel allows to represent with one single value the color present in a certain pixel, making it easier to understand which is the overall pixel color distribution of an image patch. The range of values of this channel in OpenCV is shown in Figure 5.

Figure 5: Hue OpenCV value range

The `HistogramThresholder::getImageHistogram` and `HistogramThresholder::getHue-Histogram` are in charge of evaluating the Hue histogram for a certain image patch.

To manually set the thresholds on the Hue histograms evaluated with the functions previously described, it has been evaluated and plotted the Hue histograms of the food, we wanted to discriminate and then manually estimated a representative histogram for each food label.

The manually defined histograms have been saved in the `labels_histograms.txt` file, which is read by the system and used to store the threshold. The format of this file is shown in Listing 1.

```
<number of labels>
<number of histogram bins>
<first label histogram>
<...>
<last label histogram>
```

Listing 1: Thresholds file structure

In Figure 6 we have reported the manually estimated Hue distribution for labels 1 (pasta with pesto) and 7 (fish cutlet). It can be seen from the plots that label 1 tends more toward the green hue, while label 7 toward the orange hue.

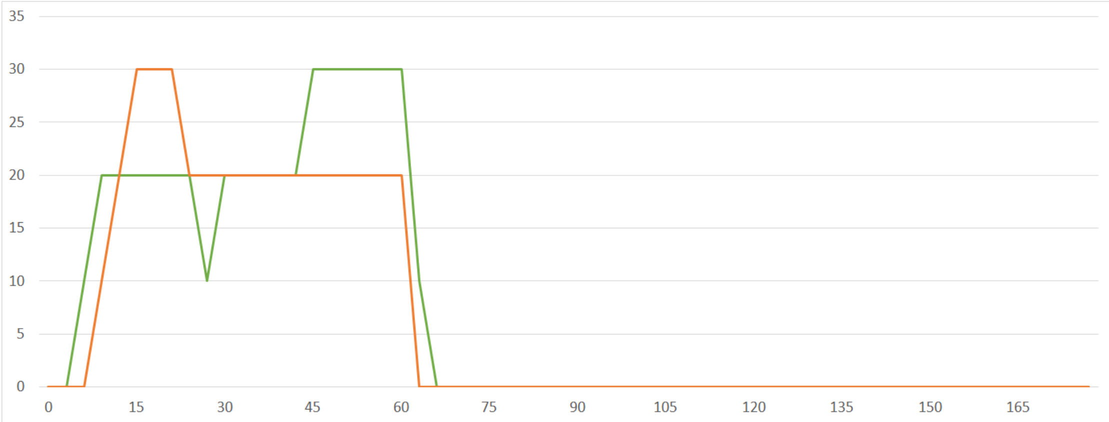


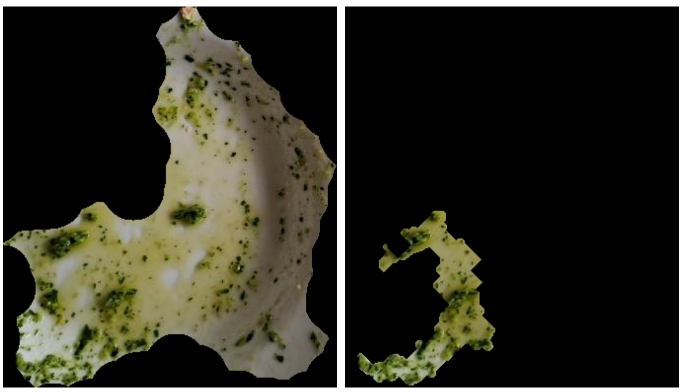
Figure 6: Manually estimated Hue histograms of label 1 (in green) and label 7 (in orange)

The `HistogramThresholder::getLabelDistances` method allows to compare the Hue values histogram of an image patch with the ones manually computed that represent the different food labels, retrieving for each possible label the estimated distance with respect to the given patch.

FoodSegmenter The `FoodSegmenter` library allows to identify the foods present in the segmented food masks by means of the `FoodSegmenter::refineMask` and `FoodSegmenter::getFoodMaskFromPlates` methods.

The `FoodSegmenter::refineMask` method is in charge of refining the food masks after they have been labeled. In particular, we have implemented specific refinition methods:

- `FoodSegmenter::refinePestoPasta` which removes the pesto sauce leftover and some remaining part of the plates, as shown in Figure 7.
  This is done with an analysis of the HSV image, made with `cv::cvtColor(src,dest,cv::COLOR_BGR2HSV)`, with a focus on the Saturation channel.
  A series of morphological operators is performed to the thresholded mask, and only the largest connected component is kept.



(a) Not refined (b) Refined

Figure 7: Refine Pasta al Pesto

- `FoodSegmenter::refinePilawRice` which removes the small rice leftover and some remaining part of plates, as shown in Figure 8. This is done with an analysis of an HSV image made with `cv::cvtColor(src,dest,cv::COLOR_BGR2HSV)` with a focus on saturation channel. After the threshold on the Saturation channel, only the largest connected component is kept.

(a) Not refined (b) Refined

Figure 8: Refine Pasta al Pesto

- `FoodSegmenter::refinePorkCutlet`, which uses the closing morphological operator to fill the holes in the pork cutlets previously miss classified due to the lighter areas that were marked before as plate areas, as shown in Figure 9.

(a) Not refined (b) Refined

Figure 9: Refine pork cutlet

The `FoodSegmenter::getFoodMaskFromPlates` method is in charge of choosing the best labels for the food masks found and splitting the masks containing more foods. In particular, the process is composed of the following steps:

1. For each plate, the food mask is evaluated using the `FoodSegmenter::getFoodMaskFrom` method. From the obtained area of the tray image, the histogram is evaluated using the `HistogramThresholder::getImageHistogram` method and the distances from the found histogram and all the labels histograms are evaluated using the `HistogramThresholder::getLabelDistances` method;

2. a. If the tray is a food_image (before the meal):
   Assuming there can be at most two food plates, if there is one plate, the nearest label is assigned.
   If there are two plates, firstly two labels per plate are found: one associated with a first dish and one associated with a second dish. Exploiting the tray composition rules (there can be at most one first and one second dish in a tray), the possible combinations of the food found are two:

$$(plate_1 = firstDish, \, plate_2 = secondDish)$$
$$(plate_1 = secondDish, \, plate_2 = firstDish)$$

   To choose which one is the best tuple, for each plate has been computed a loss as

$$plateNormalizedLoss = \frac{|plateFirstDishDistance - plateSecondDishDistance|}{plateWorstDishDistance}$$

   The plate that receives the highest score chooses whether to be a first or second dish, while the other plate is forced to be what is left.

   b. If the tray is a leftover (after the meal):
   The same algorithm is applied, but the labels' histograms distances are computed only between the food previously found.

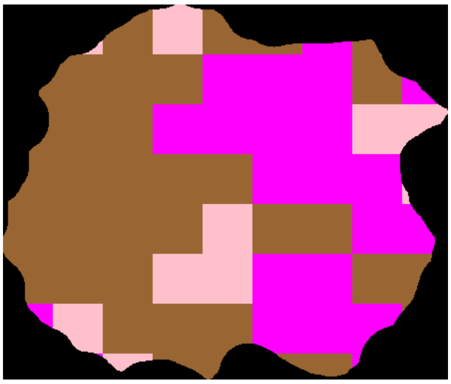
3. For each plate:

   a. If it is a first dish, the mask is added to the segmentation mask;

   b. Otherwise, the mask may need to be split in second and side dishes.

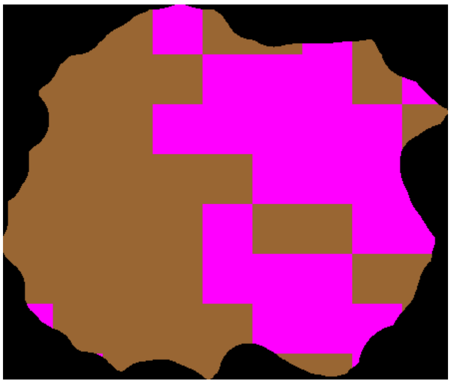
   To do so:

   i. A window is slid over the food mask, for each non-empty window the `HistogramThresholder::getImageHistogram` and `HistogramThresholder::getLabelDistances` methods are used to get the least distant label;

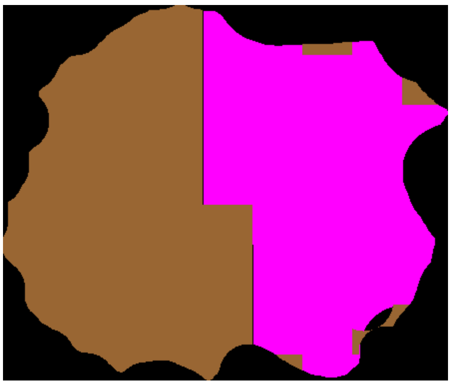


ii. The best second dish label and the two best side dish labels are kept, if they are present in the found label list;

iii. A smaller window is slid over the food mask, for each non-empty window the `HistogramThresholder::getImageHistogram` and `HistogramThresholder::g-etLabelDistances` methods are used to get the least distant label among the kept ones;

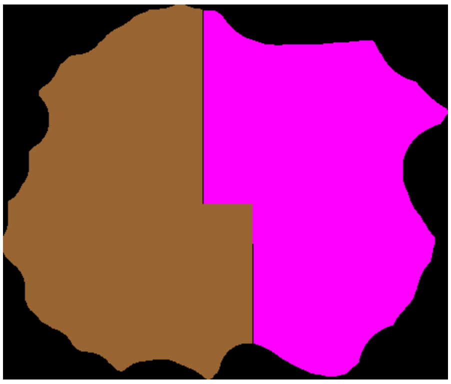


iv. The label of each smaller window is re-assigned by computing the most frequent one in its 8-neighboring windows, in order to get more uniform areas;



v. The mask windows having the same labels are merged together;

vi. If two foods have been found in the plate, for each mask only the biggest contour is kept (and also the second biggest one if it is just slightly smaller than the first), leaving all the smaller contours to the other label mask;



vii. The masks are refined using the `FoodSegmenter::refineMask` method;

viii. The masks are added to the segmentation mask;

2.2.3 Salad

The segmenting process of the plate areas is exploited by the method `FoodSegmenter::getSaladMaskFromBowl`. This method, given the tray image and the position of the salad bowl, masks the salad discriminating its pixels from the ones of the bowl itself. In particular, this process is composed of the following steps:

1. the image is pre-processed using Gaussian blur;

2. the image is converted to HSV color space and the pixels inside the bowl are scanned and added to the salad mask if they satisfy a 175 threshold on the Saturation channel (to ignore bowl pixels) or a 245 threshold on the Value channel (to avoid discarding white radish areas);

3. since the obtained mask can have a lot of holes, the closing morphological operator is applied;

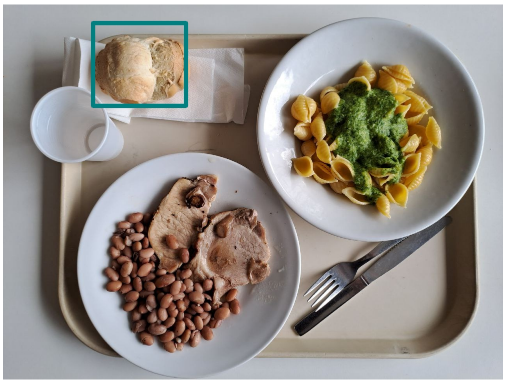
4. mask contours are scanned in order to discard contours having a negligible area with respect to the bowl area.

2.2.4 Bread

The segmenting process of the plate areas is exploited by the method `FoodSegmenter::getBreadMask`. This method, given the tray image and the bread area detected with `FoodFinder::findBread` 2.1.3 and the `cv::Mat` in which to save, performs the following steps:

1. Erodes the passed mask with a 5x5 elliptical kernel iterating the erosion five times.
2. Computes the rectangular bound box of the binary mask with `cv::boundingRect`.
3. Using `grabCut` [1] it refines the segmentation mask [1].
4. Since there might happen that `grabCut` returns more than one blob, the bigger blob is chosen among the one returned by `grabCut`.

The before and after an image is the following:



(a) Result of bread detection.

(b) Result of bread segmentation

The result of the segmentation of all breads is the following:

[1] grabCut documentation: documentation.

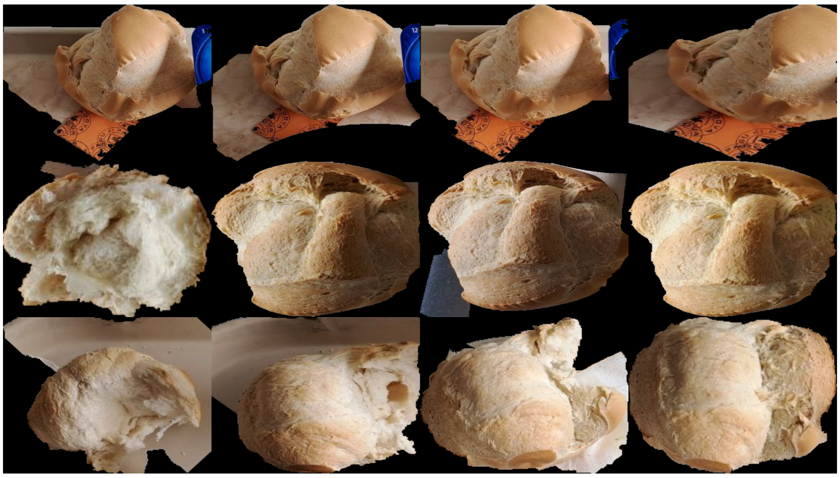


Figure 11: Result of bread segmentation in all the trays with bread.

2.3 Leftover estimation

To estimate the food leftovers after the meal we have compared the segmentation masks obtained for the two images respectively. For each food found in the before image, we have computed the food amount (i.e. number of pixels) present in the before and after tray, evaluating the leftover amount as the difference between the two.

An example of the system output obtained for tray2 leftover1 is reported in Listing 2.

```
food_image:
    2. pasta with tomato sauce
    7. fish cutlet
    11. basil potatoes
    12. salad
leftover:
    2. pasta with tomato sauce
    7. fish cutlet
    11. basil potatoes
    12. salad
Food quantities:
2. pasta with tomato sauce
    Before amount = 116795
    After amount = 77147
    Leftover amount = 39648
7. fish cutlet
    Before amount = 35856
    After amount = 17939
    Leftover amount = 17917
11. basil potatoess
    Before amount = 38175
    After amount = 37503
    Leftover amount = 672
12. salad
    Before amount = 87716
    After amount = 60448
    Leftover amount = 27268
```

Listing 2: Tray output example for tray2 leftover1

3 Metrics

In this Section, we will describe the metrics we have computed to evaluate the developed system.

3.1 Introduction

To establish the system's performance it has been used the following metrics:

- mean Average Precision (mAP): used to assess the quality of the Food Detection. The mAP has been calculated at Intersection over Union (IoU) threshold 0.5;
- mean Intersection over Union (mIoU): used to assess the quality of Food Segmentation, is the average of the IoU computed for each food item;
- Difference between leftover estimation based on the system segmentation and the leftover estimation based on the ground truth segmentation called Leftover Estimation Difference (LED). The quantity of food leftover is defined as the ratio R_i (in terms of pixels) between the segmentation mask in the leftover image (after meal) and the segmentation mask in the food_image (before meal).

3.2 mean Average Precision (mAP)

The mAP metric is defined as the mean of all the Average Precision (AP) computed for each possible class of the dataset (or category).
The AP could be considered as a measure of "how much a system is able to correctly classify the inputs". For its computation, it has been used the function `MetricsCalculation::calculateAP`.

This function can compute the mAP of a class singularly and the mAP of the system as well. This function gives the AP computed for a class taking in input:

- A vector of Predicition's reference. Each Prediction object is an ensemble of:
  - The identified class (by ID)
  - Confidence score of the detection
  - Boolean value which is true if the match is True Positive (TP) (iff confidence score > 0.50), false otherwise
- The class for which it must compute the AP
- The number of objects belonging to the ground truth (regarding the above-specified class)

The confidence score for a given match is based on IoU of the two bounding boxes, the IoU threshold to discriminate a good match from a bad match is 0.50. A match that has a confidence score greater or equal than 0.50, will be considered as a TP detection.
The threshold value defines how well the detection must be to be considered a TP.
Given a food item match, an IoU threshold closer to one imposes that the detections must be close to perfect to assess the item TP one, as it requires almost perfect detections; while an IoU threshold closer to zero (not zero) would be more flexible, considering even small overlaps as valid TP detections.
To briefly sum up, `MetricsCalculation::calculateAP` calculates the AP using the "PASCAL VOC 11 point interpolation method" for one single object class.

- This function takes the whole set of predictions made by our system. It takes only the ones with `classID` code and sorts them in decreasing order of confidence score
- After that it computes the precision and recall values for each match of `classID`. Those two values will be coordinate points in a reference system precision-recall, where the x-axis representing the recall is subdivided into 11 points: $\{0.0, 0.1, \ldots, 1.0\}$. It will end up having 11 points, each one has the maximum precision score among the nearest recall points.

$$currentprecision = \frac{cumulativeTP}{cumulativeTP+cumulativeFP}$$

$$currentrecall = \frac{cumulativeTP}{gTNumberItemsPerClass}$$

- The AP is computed using the 11-point interpolation technique: Graphically in this reference system, each precision value is replaced with the maximum precision value to the right of that recall level. As it is possible to see from the image below, the orange line is transformed into the green line and the curve will decrease monotonically instead of the zigzag pattern.

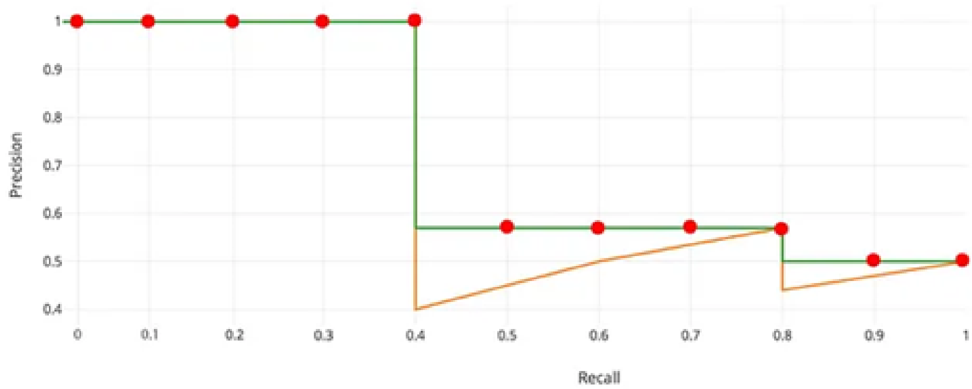


Figure 12: Image from: https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173

- To compute the final value, the sum of all 11 precision values after interpolation is normalized to 11. In this toy example

$$AP = (5 \times 1.0 + 4 \times 0.57 + 2 \times 0.5)/11 = 0.7527$$

To conclude, mAP can be considered as the mean of all the AP's computed for each detectable class of our food recognition system.

Remember that:
Precision: is the capability of a system to identify only relevant objects.

$$P = \frac{TP}{TP+FP}$$

Recall: is the capability of a system to find all cases of interest. It is the ratio of correct true predictions over all possible ground truth items.

$$R = \frac{TP}{TP+FN}$$

To state how good is a system to classify the inputs, it is possible to consider:
An $AP = 1$ means that it achieved perfection in object localization. On the other hand, an $AP = 0$ could mean several things: Each detected object has a confidence score below the desired IoU threshold, or even zero matches were achieved (worst case scenario).

3.3 mean Intersection over Union (mIoU)

Intersection over Union (IoU) is a metric used to measure "how good the object detection is".
Intersection over Union (IoU) considers the predicted and the ground truth bounding boxes, overlapping them and seeing how much of the total area they share. The bigger the intersection over the union area of these two rectangles, the higher IoU will be.
IoU ranges in $[0, 1]$.
A perfect match ($IoU = 1$) occurs when the area and location of the predicted and ground-truth boxes are the same. Instead, an $IoU = 0$ means that the detection is incorrect, or even a non-detection case of the desired object (worst case possible).
Taking as examples the images below, it has been performed a visible example: Food_image and leftovers (1 and 2) of pasta al pesto's Tray1. The Blu area is the intersection of the red area (detected by the proposed algorithm) and the green area (ground truth bounding box). The first two IoU computed (0.959773 and 0.959388) show us how well we have made these detections. The third IoU is 0.877863: not a perfect score, but still a very good detection.

(a) Food Image (b) Leftover1 (c) Leftover2

Figure 13: Example of IoU, in blue their intersection areas

All of this is computed by the function `metricsCalculations::singlePlateFoodSegmentationIoUMetric`.

Given two vectors of integers, representing two bounding box coordinates for a food item, this function returns the IoU of the food item detection. In detail, it:

- Computes the rectangle representing the intersection of those two boxes
- Computes its area, in pixels (intersectionPixels).

- Computes the area of the two boxes' union, in pixels (unionPixels).
- Returns the ratio intersectionPixels/unionPixels as IoU of a single food item.

Notice that a single bounding box (BB) coordinates record (e.g: $[x, y, w, h]$) is composed of x and y for top left pixel corner coordinates, w and h respectively for width and height of the BB itself.

3.4 Leftover Estimation (LE)

The last part of the metrics implemented by the system is LE.
LE is composed by three parts:

1. Computes ground truth's leftover as the ratio of food leftover pixels for a given food and food_image pixels for the same food, called R_i
2. Computes predicted leftover as the ratio of food leftover pixels for a given food and food_image pixels for the same food, called R_i

   The quantity of food leftover is expressed as the ratio R_i (in terms of pixels) between the segmentation mask in the "after" image and the segmentation mask in the "before" :

   $$R_i = \frac{\#\text{pixels for food } i \text{ in the "after image"}}{\#\text{pixels for food } i \text{ in the "before image"}}$$

3. Compute the difference, in absolute value, between these R_i's just found.

For points 1. and 2. are print out those numbers directly in `OneImageSegmentationMetric-Calculations`. In the leftover analysis part of this function, whenever there's a detection match of a food item between the food image and the leftover image, it is elaborated getting two masks (before and after). This is done firstly with ground truth's segmentation masks, and then with the one computed by the algorithm proposed.

For point 3. it's used `singlePlateLeftoverEstimationMetric` which takes in input two masks, respectively the food item in Food_image (image before the meal) and the food item in leftover (image after meal). It compares the ratio of the pixel numbers of those masks and returns R_i.
The leftover part concludes printing all the information; Here there's an example of leftover estimation metrics computed for Tray1's leftover1:
Leftover: 1(Tray 1)
Food item 1 : our R_i = 0.838043 Food item 1 : gT R_i = 0.74199
Food item 1 : abs diff = 0.0960533
Food item 6 : our R_i = 0.372595 Food item 6 : gT R_i = 0.817095
Food item 6 : abs diff = 0.4445
Food item 10 : our R_i = 1.68914 Food item 10 : gT R_i = 1.28539
Food item 10 : abs diff = 0.403754
Food item 13 : our R_i = 1.06191 Food item 13 : gT R_i = 0.869276
Food item 13 : abs diff = 0.192636
To state whether the system is good it is possible to make the following considerations:

- The lower is "ABS of the difference", the better the segmentation mask (compared to the ground truths one)

- The lower is R_i , the more consumed was the food item.

The number of pixels of leftover and the ratio between leftover and food_image could be used for comparison with other leftovers to see which was the most consumed food and which was the food with the highest number of leftovers. In a hypothetical scenario of a business company's canteen, those numbers could be useful to understand which types of food the clients like the most, and which food is the least eaten.
It's been implemented the absolute value of the difference as well, as a metric of comparison between our segmentation leftover masks and the ones from the ground truth. The higher is that difference, the further the system is from the ground truth case.

The `Test::testTheSystem` function gathers all the trays contained in a vector, scans through it and produces all the metrics for a specific Tray (before, after) and the overall system ones.

4 Results

In this Section, it has been reported the results obtained evaluating the system using the metrics presented in Section 3. In Tables 1, 2, 3 are reported the results obtained on all images, all trays and the system overall respectively.
Please notice that the mIoU and mAP scores are not reported in Table 1, since it was requested to compute only the leftover estimation, which is reported in Table 2.

Tray	Image	True labels	Detected labels	mIoU	mAP@50
1	food_image	1, 6, 10, 13	1, 6, 10, 13	0.941	1.000
	leftover_1	1, 6, 10, 13	1, 6, 10, 13	0.767	0.750
	leftover_2	1, 6, 10, 13	1, 6, 10, 13	0.738	1.000
	leftover_3	10, 13	1, 10, 13	-	-
2	food_image	2, 7, 11, 12	2, 7, 11, 12	0.830	1.000
	leftover_1	2, 7, 11, 12	2, 7, 11, 12	0.853	1.000
	leftover_2	2, 7, 11, 12	2, 7, 11, 12	0.760	1.000
	leftover_3	7, 11	2, 7, 11, 12	-	-
3	food_image	2, 8, 12	2, 8, 12	0.887	1.000
	leftover_1	2, 8, 12	2, 8, 12	0.708	0.667
	leftover_2	2, 8, 12	2, 8, 12	0.762	1.000
	leftover_3	8	2, 8, 12	-	-
4	food_image	5, 7, 11, 12, 13	5, 7, 11, 12, 13	0.838	1.000
	leftover_1	5, 7, 11, 12, 13	5, 7, 11, 12, 13	0.780	1.000
	leftover_2	5, 7, 11, 13	5, 7, 11, 13	0.813	1.000
	leftover_3	5, 11, 12, 13	5, 11, 12, 13	-	-
5	food_image	3, 8, 10, 13	3, 8, 10, 13	0.759	1.000
	leftover_1	3, 8, 10, 13	3, 8, 10, 13	0.546	0.500
	leftover_2	8, 10, 13	8, 10, 13	0.644	0.667
	leftover_3	8, 10, 13	3, 8, 10, 13	-	-
6	food_image	4, 6, 10, 12	4, 6, 10, 12	0.894	1.000
	leftover_1	4, 6, 10, 12	4, 6, 10, 12	0.880	1.000
	leftover_2	4, 6, 10, 12	4, 6, 10, 12	0.833	1.000
	leftover_3	4, 12	4, 6, 12	-	-
7	food_image	4, 7, 11, 12	4, 7, 11, 12	0.741	0.750
	leftover_1	4, 7, 11, 12	4, 7, 11, 12	0.810	1.000
	leftover_2	4, 7, 11, 12	4, 7, 11, 12	0.846	1.000
	leftover_3	4, 11	4, 11	-	-
8	food_image	4, 9, 10, 11, 12	4, 9, 10, 11, 12	0.596	0.600
	leftover_1	4, 9, 10, 12	4, 9, 11, 12	0.582	0.500
	leftover_2	4, 9, 10, 12	4, 9, 10, 11, 12	0.660	0.600
	leftover_3	4	4, 11, 12	-	-

Table 1: Results on all images

Tray	Leftover	mIoU	mAP@50	mLED
1	1	0.854	0.886	0.266
	2	0.839	1.000	0.117
	3	0.941	1.000	0.288
2	1	0.842	1.000	0.051
	2	0.795	1.000	0.208
	3	0.830	1.000	0.017
3	1	0.798	0.848	0.220
	2	0.825	1.000	0.197
	3	0.887	1.000	0.210
4	1	0.809	1.000	0.135
	2	0.827	1.000	0.123
	3	0.838	1.000	0.047
5	1	0.653	0.773	0.216
	2	0.636	0.886	0.207
	3	0.762	1.000	0.223
6	1	0.887	1.000	0.179
	2	0.864	1.000	0.083
	3	0.894	1.000	0.105
7	1	0.775	0.886	0.098
	2	0.793	0.886	0.142
	3	0.741	0.750	0.144
8	1	0.523	0.509	0.297
	2	0.625	0.600	0.402
	3	0.596	0.600	0.171

Table 2: Results on all trays

	mIoU	mAP@50	mLED
Overall	0.777	0.916	0.173

Table 3: Overall system results

From the overall results shown in Table 3 it can be seen that the proposed localization and segmentation system achieved satisfactory performances over the considered metrics. The mean Leftover Estimation Difference (mLED) value shows that comparing our leftover estimations with the ground truth's ones, there is a mean error around 17.3%. As it can be seen from the results on single images shown in Table 1, this error is due to the fact that in some cases, in the leftover_3 images, some detected foods were marked as totally consumed according to the ground truth. The reason is that some plates are still containing some leftovers, for example, pasta sauce, that is still detected as food from our system.
The mIoU value shows that the system achieves an accuracy around 77.7% in detecting food bounding boxes, which leads to the mAP 91.6% score, since the IoU metric has been used as a confidence score for the evaluation of the TPs.

5 Project structure

The project has been developed using the CMake build system and it is organized with the following structure:

- `include` directory, contains the header source code files;
- `src` directory, contains the implementation source code files;
- `data` directory, contains the `labels_histograms.txt` file;
- `input` directory, contains the Food_leftover_dataset;
- `output` directory, contains the output bounding boxes and masks files generated in output;
- `CMakeLists.txt` file, contains CMake instructions to organize the build of the project.

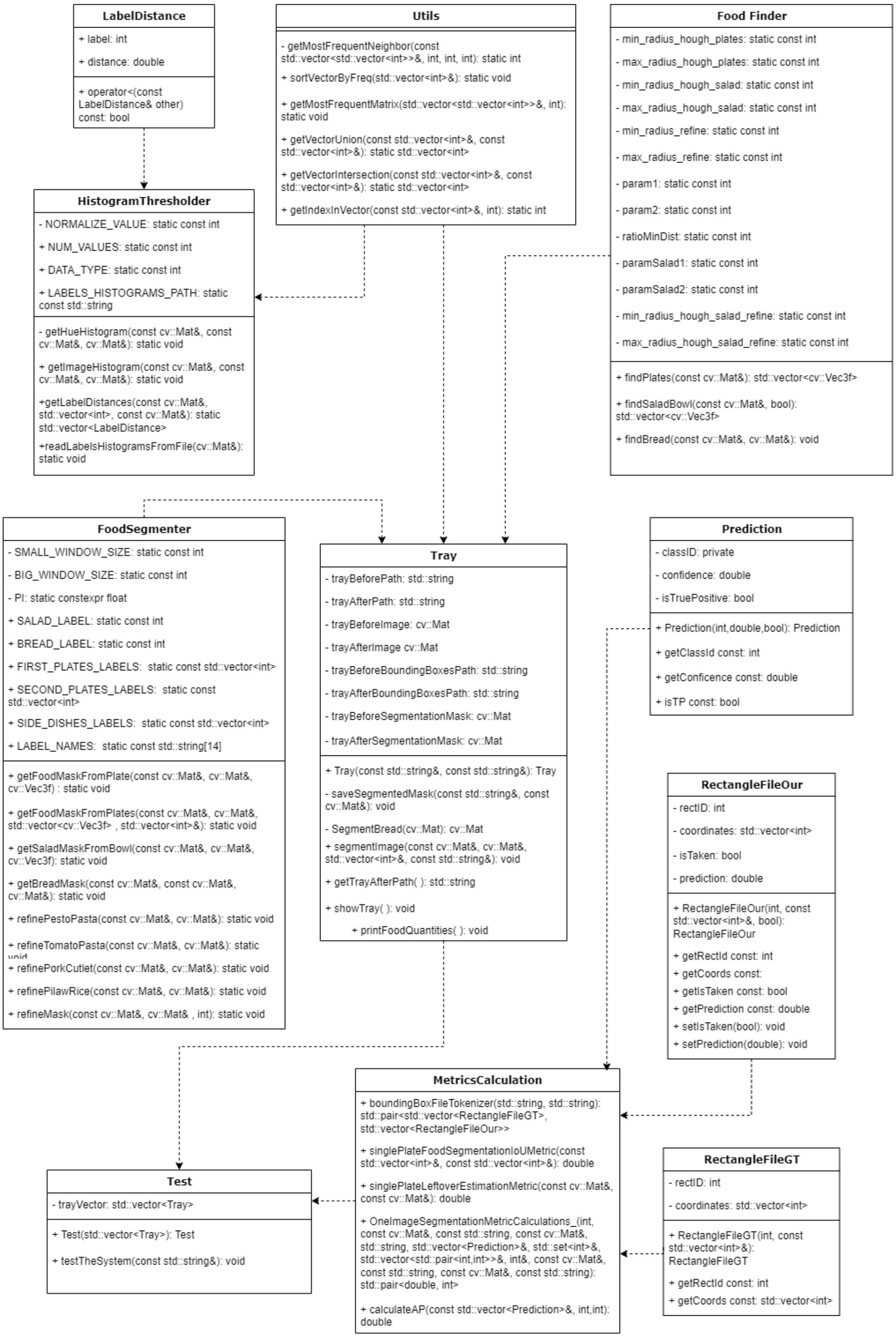


Figure 14: Diagram of the project libraries

6 Experimental setup

The system has been tested under the following environment:

- C++11[2];
- CMake v2.8.12[3] minimum required version;
- OpenCV v4.5.2[4];
- CPU Apple M1 Pro, 16GB RAM.

In Table 4 are reported the amounts of time taken by the elaboration of each tray under the described setup.
As it is possible to see in Table 4, Tray1, Tray4 and Tray5 take around two seconds more than the other trays because of the grabCut execution, used to segment brad (more detail in Section 2.2.4).

Tray	**Leftover**	**Time** (ms)
1	1	3246
	2	3473
	3	3119
2	1	1224
	2	1241
	3	1210
3	1	1162
	2	1134
	3	1209
4	1	3299
	2	3160
	3	3359
5	1	3503
	2	3413
	3	3818
6	1	1659
	2	1633
	3	1657
7	1	1267
	2	1261
	3	1059
8	1	1319
	2	1555
	3	1197

Table 4: Execution time

[2] `https://en.cppreference.com/w/cpp/11`
[3] `https://cmake.org/cmake/help/v2.8.12/cmake.html`
[4] `https://docs.opencv.org/4.5.2/`

7 Work split

In this Section, we show how the work has been split between the authors. In Tables 5 and 6 are reported the work loads and splits over the project and report respectively.

Project		
Task	**Owner of Task**	**Hours**
Find Plates	Nicola Lorenzon	45
Find Salads	Nicola Lorenzon	10
Find Bread	Nicola Lorenzon	30
Labels Histogram Estimation	Daniele Moschetta	15
Segment Food In Plates	Daniele Moschetta	60
Segment Salads	Daniele Moschetta	3
Segment Bread	Nicola Lorenzon	2
Refine Pasta al Pesto	Nicola Lorenzon	5
Refine Pilaw Rice	Nicola Lorenzon	2
Refine Pork Cutlet	Daniele Moschetta	2
Leftover Estimation	Carmine Graniello	25
Metrics	Carmine Graniello	65

Table 5: Project work load and split

Report		
Task	**Owner of Task**	**Hours**
Introduction	Carmine Graniello	2
Approach Explanation	Nicola Lorenzon	2
Food Finding: plates, salad bowl, bread	Nicola Lorenzon	4
Food Segmenting: plates, food in plates, salad	Daniele Moschetta	6
Food Segmenting: bread	Nicola Lorenzon	2
Leftover Estimation	Carmine Graniello	3
Metrics	Carmine Graniello	8
Results	Carmine Graniello	6
Conclusion	Carmine Graniello	1

Table 6: Report work load and split

8 Conclusions and Future work

The developed system has achieved satisfactory performance in food localization, food segmentation and leftover estimation, especially considering that it is exploiting only traditional computer vision techniques without the use of Machine Learning or Deep Learning.

The project could be further improved by refining all the different foods in the dataset, since, due to time constraints, refining techniques have been implemented just for pasta with pesto, pilaw rice and pork cutlet.

In conclusion, we really think that this project goes beyond just being a standalone effort. The development of this kind of systems can have a significant impact on everyday life, especially by reducing the amount of food and money wasted. We hope that with the implementation of leftover estimation systems, people will become more aware of the value of food and start consuming it more responsibly for the greater good.

References

[1] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": Interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, page 309–314, New York, NY, USA, 2004. Association for Computing Machinery.

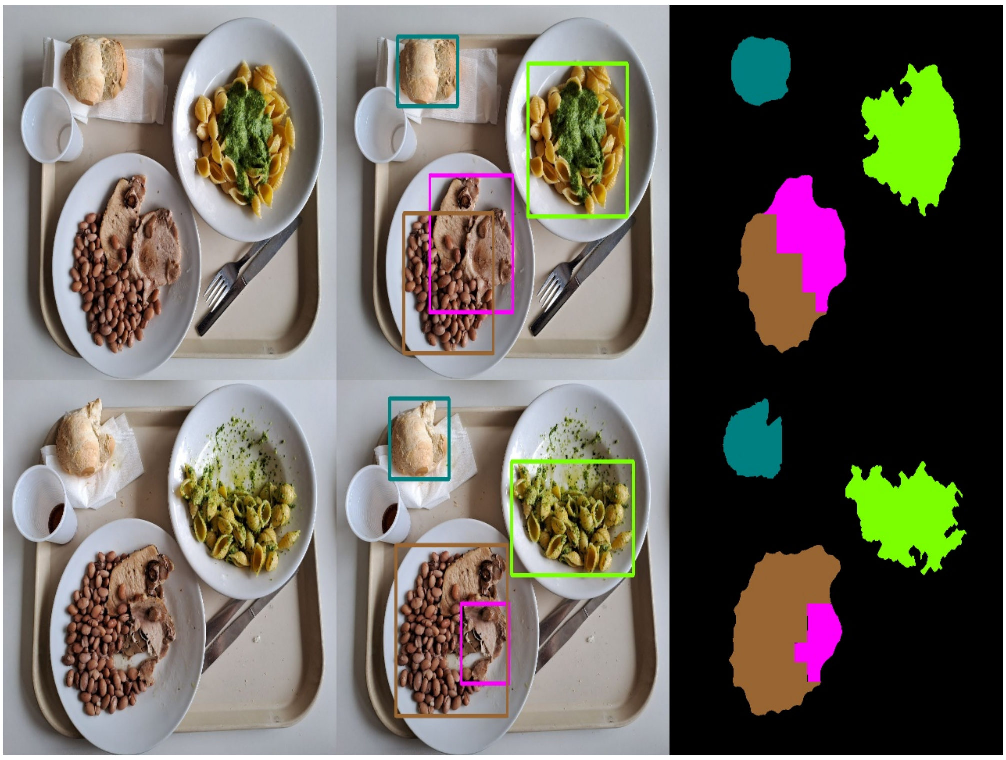
Appendices

A Full output of the system

The colors referred to the labels in the output images are:

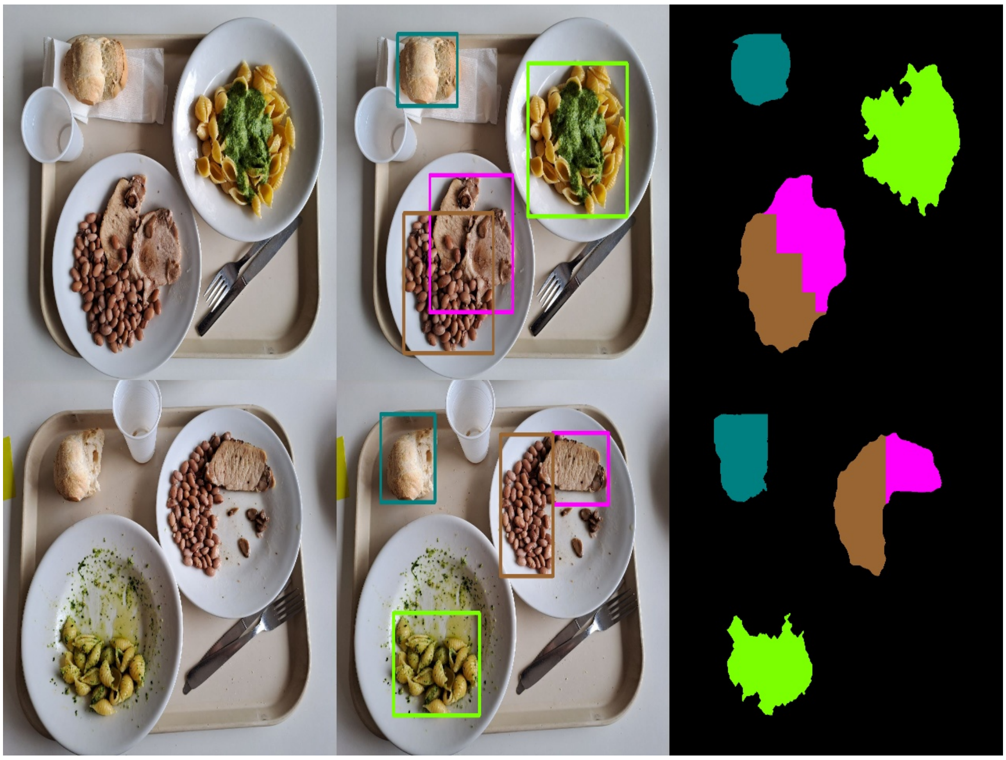
0. Background: Black
1. Pasta with pesto: Green
2. Pasta with tomato sauce: Red
3. Pasta with meat sauce: Blue
4. Pasta with clams and mussels: Yellow
5. Pilaw rice with peppers and peas: Cyan
6. Grilled pork cutlet: Magenta
7. Fish cutlet: Orange
8. Rabbit: Purple
9. Seafood salad: Pink
10. Beans: Brown
11. Basil potatoes: Gray
12. Salad: White
13. Bread: Olive

- Tray 1 Leftover 1



```
food_image:
          1. pasta with pesto
          6. grilled pork cutlet
          10. beans
          13. bread
leftover:
          1. pasta with pesto
          6. grilled pork cutlet
          10. beans
          13. bread
1. pasta with pesto
          Before amount = 93846
          After amount = 78647
          Leftover amount = 15199
6. grilled pork cutlet
          Before amount = 58573
          After amount = 21824
          Leftover amount = 36749
10. beans
          Before amount = 67963
          After amount = 114799
          Leftover amount = −46836
13. bread
          Before amount = 32724
          After amount = 34750
          Leftover amount = −2026
```

- Tray 1 Leftover 2



```
food_image:
          1. pasta with pesto
          6. grilled pork cutlet
          10. beans
          13. bread

leftover:
          1. pasta with pesto
          6. grilled pork cutlet
          10. beans
          13. bread

1. pasta with pesto
          Before amount = 93846
          After amount = 52712
          Leftover amount = 41134
6. grilled pork cutlet
          Before amount = 58573
          After amount = 25250
          Leftover amount = 33323
10. beans
          Before amount = 67963
          After amount = 52168
          Leftover amount = 15795
13. bread
          Before amount = 34678
          After amount = 39781
          Leftover amount = -5103
```

- Tray 1 Leftover 3

```
food_image:
          1. pasta with pesto
          6. grilled pork cutlet
          10. beans
          13. bread

leftover:
          1. pasta with pesto
          10. beans
          13. bread

1. pasta with pesto
          Before amount = 93846
          After amount = 8822
          Leftover amount = 85024
6. grilled pork cutlet
          Before amount = 58573
          After amount = 0
          Leftover amount = 58573
10. beans
          Before amount = 67963
          After amount = 63843
          Leftover amount = 4120
13. bread
          Before amount = 35190
          After amount = 35448
          Leftover amount = -258
```

- Tray 2 Leftover 1

```
food_image:
        2. pasta with tomato sauce
        7. fish cutlet
        11. basil potatoes
        12. salad

leftover:
        2. pasta with tomato sauce
        7. fish cutlet
        11. basil potatoes
        12. salad

2. pasta with tomato sauce
        Before amount = 116795
        After amount = 117097
        Leftover amount = −302
7. fish cutlet
        Before amount = 35856
        After amount = 40204
        Leftover amount = −4348
11. basil potatoes
        Before amount = 38175
        After amount = 40914
        Leftover amount = −2739
12. salad
        Before amount = 87716
        After amount = 95973
        Leftover amount = −8257
```

- Tray 2 Leftover 2

```
food_image:
        2. pasta with tomato sauce
        7. fish cutlet
        11. basil potatoes
        12. salad

leftover:
        2. pasta with tomato sauce
        7. fish cutlet
        11. basil potatoes
        12. salad

2. pasta with tomato sauce
        Before amount = 116795
        After amount = 77147
        Leftover amount = 39648
7. fish cutlet
        Before amount = 35856
        After amount = 17939
        Leftover amount = 17917
11. basil potatoes
        Before amount = 38175
        After amount = 37503
        Leftover amount = 672
12. salad
        Before amount = 87716
        After amount = 60448
        Leftover amount = 27268
```

- Tray 2 Leftover 3

```
food_image:
        2. pasta with tomato sauce
        7. fish cutlet
        11. basil potatoes
        12. salad

leftover:
        2. pasta with tomato sauce
        7. fish cutlet
        11. basil potatoes
        12. salad

2. pasta with tomato sauce
        Before amount = 116795
        After amount = 91903
        Leftover amount = 24892
7. fish cutlet
        Before amount = 35856
        After amount = 12110
        Leftover amount = 23746
11. basil potatoes
        Before amount = 38175
        After amount = 9927
        Leftover amount = 28248
12. salad
        Before amount = 87716
        After amount = 9293
        Leftover amount = 78423
```

- Tray 3 Leftover 1

```
food_image:
          2. pasta with tomato sauce
          8. rabbit
          12. salad

leftover:
          2. pasta with tomato sauce
          8. rabbit
          12. salad

2. pasta with tomato sauce
          Before amount = 108577
          After amount = 85692
          Leftover amount = 22885
8. rabbit
          Before amount = 46177
          After amount = 42723
          Leftover amount = 3454
12. salad
          Before amount = 83703
          After amount = 57680
          Leftover amount = 26023
```

- Tray 3 Leftover 2

```
food_image:
        2. pasta with tomato sauce
        8. rabbit
        12. salad

leftover:
        2. pasta with tomato sauce
        8. rabbit
        12. salad

2. pasta with tomato sauce
        Before amount = 108577
        After amount = 70785
        Leftover amount = 37792
8. rabbit
        Before amount = 46177
        After amount = 48071
        Leftover amount = −1894
12. salad
        Before amount = 83703
        After amount = 61537
        Leftover amount = 22166
```

- Tray 3 Leftover 3

```
food_image:
        2. pasta with tomato sauce
        8. rabbit
        12. salad

leftover:
        2. pasta with tomato sauce
        8. rabbit
        12. salad

2. pasta with tomato sauce
        Before amount = 108577
        After amount = 84012
        Leftover amount = 24565
8. rabbit
        Before amount = 46177
        After amount = 50945
        Leftover amount = −4768
12. salad
        Before amount = 83703
        After amount = 2382
        Leftover amount = 81321
```

- Tray 4 Leftover 1

```
food_image:
        5. pilaw rice with peppers and peas
        7. fish cutlet
        11. basil potatoes
        12. salad
        13. bread
leftover:
        5. pilaw rice with peppers and peas
        7. fish cutlet
        11. basil potatoes
        12. salad
        13. bread
5. pilaw rice with peppers and peas
        Before amount = 106766
        After amount = 93286
        Leftover amount = 13480
7. fish cutlet
        Before amount = 51430
        After amount = 26499
        Leftover amount = 24931
11. basil potatoes
        Before amount = 47077
        After amount = 34716
        Leftover amount = 12361
12. salad
        Before amount = 86588
        After amount = 87892
        Leftover amount = −1304
13. bread
        Before amount = 48204
        After amount = 80351
        Leftover amount = −32147
```

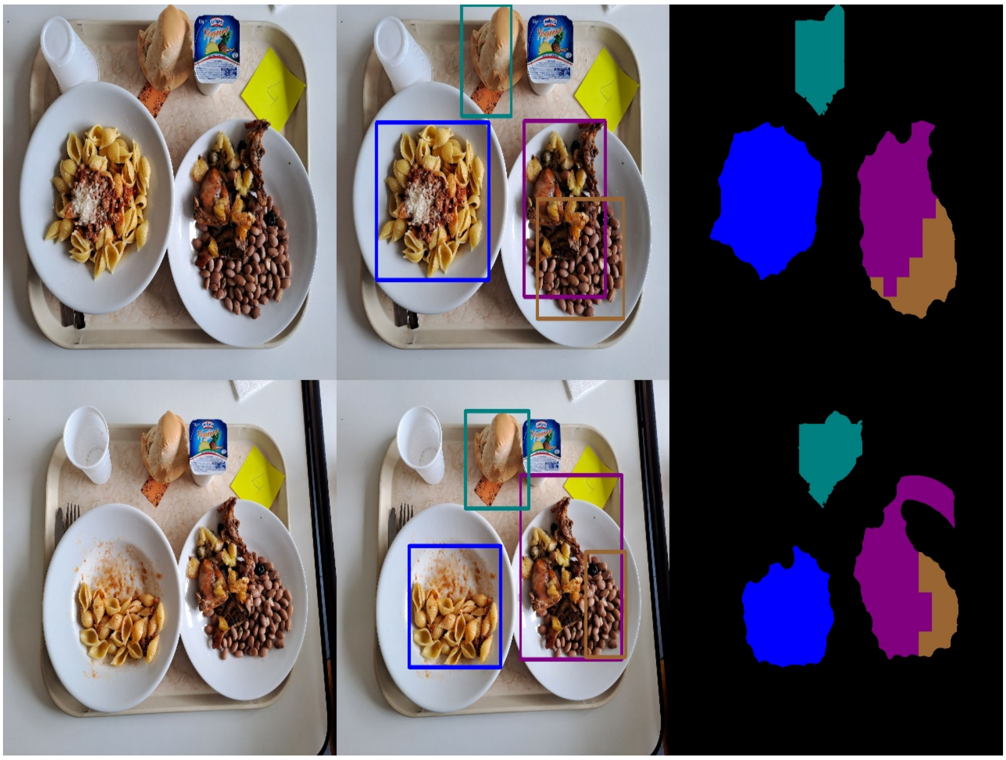
- Tray 4 Leftover 2

```
food_image:
        5. pilaw rice with peppers and peas
        7. fish cutlet
        11. basil potatoes
        12. salad
        13. bread

leftover:
        5. pilaw rice with peppers and peas
        7. fish cutlet
        11. basil potatoes
        13. bread

5. pilaw rice with peppers and peas
        Before amount = 106766
        After amount = 93735
        Leftover amount = 13031
7. fish cutlet
        Before amount = 51430
        After amount = 23996
        Leftover amount = 27434
11. basil potatoes
        Before amount = 47077
        After amount = 38719
        Leftover amount = 8358
12. salad
        Before amount = 86588
        After amount = 0
        Leftover amount = 86588
13. bread
        Before amount = 52249
        After amount = 62404
        Leftover amount = −10155
```

- Tray 4 Leftover 3

```
food_image:
        5. pilaw rice with peppers and peas
        7. fish cutlet
        11. basil potatoes
        12. salad
        13. bread

leftover:
        5. pilaw rice with peppers and peas
        11. basil potatoes
        12. salad
        13. bread

5. pilaw rice with peppers and peas
        Before amount = 106766
        After amount = 22627
        Leftover amount = 84139
7. fish cutlet
        Before amount = 51430
        After amount = 0
        Leftover amount = 51430
11. basil potatoes
        Before amount = 47077
        After amount = 4914
        Leftover amount = 42163
12. salad
        Before amount = 86588
        After amount = 50816
        Leftover amount = 35772
13. bread
        Before amount = 52397
        After amount = 18232
        Leftover amount = 34165
```

- Tray 5 Leftover 1

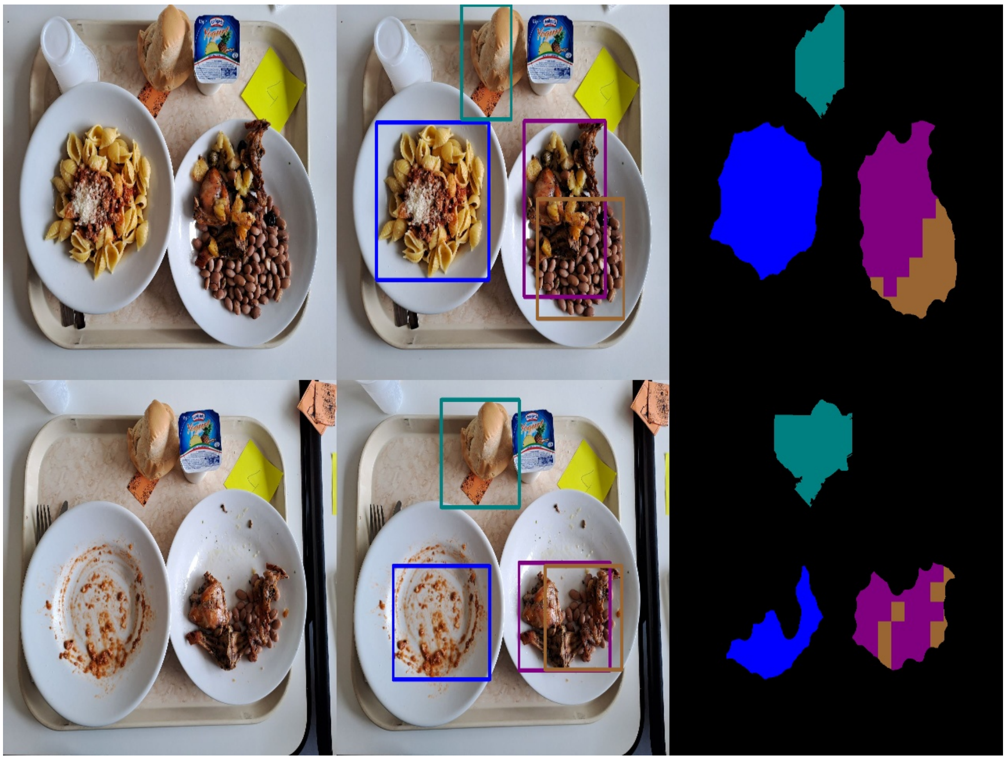


```
food_image:
        3. pasta with meat sauce
        8. rabbit
        10. beans
        13. bread

leftover:
        3. pasta with meat sauce
        8. rabbit
        10. beans
        13. bread

3. pasta with meat sauce
        Before amount = 118970
        After amount = 78085
        Leftover amount = 40885
8. rabbit
        Before amount = 90696
        After amount = 90142
        Leftover amount = 554
10. beans
        Before amount = 42607
        After amount = 25369
        Leftover amount = 17238
13. bread
        Before amount = 41503
        After amount = 39432
        Leftover amount = 2071
```

- Tray 5 Leftover 2

```
food_image:
          3. pasta with meat sauce
          8. rabbit
          10. beans
          13. bread

leftover:
          8. rabbit
          10. beans
          13. bread

3. pasta with meat sauce
          Before amount = 118970
          After amount = 0
          Leftover amount = 118970
8. rabbit
          Before amount = 90696
          After amount = 66111
          Leftover amount = 24585
10. beans
          Before amount = 42607
          After amount = 10723
          Leftover amount = 31884
13. bread
          Before amount = 38224
          After amount = 58542
          Leftover amount = -20318
```

- Tray 5 Leftover 3

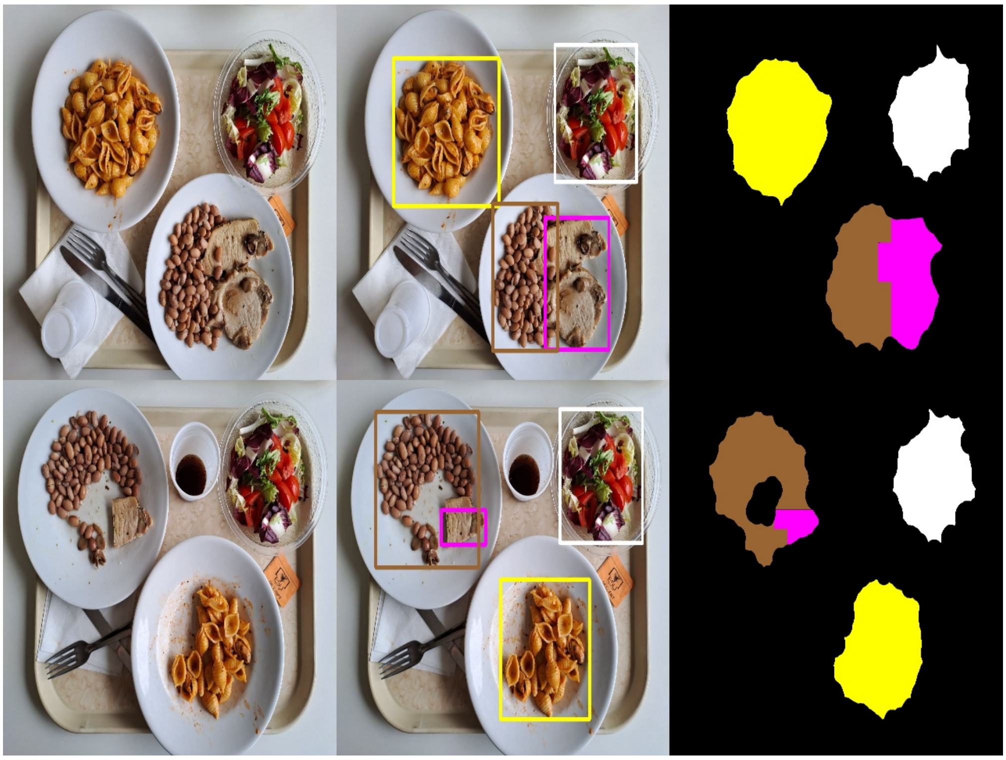


```
food_image:
          3. pasta with meat sauce
          8. rabbit
          10. beans
          13. bread

leftover:
          3. pasta with meat sauce
          8. rabbit
          10. beans
          13. bread

3. pasta with meat sauce
          Before amount = 118970
          After amount = 38963
          Leftover amount = 80007
8. rabbit
          Before amount = 90696
          After amount = 51527
          Leftover amount = 39169
10. beans
          Before amount = 42607
          After amount = 15915
          Leftover amount = 26692
13. bread
          Before amount = 38185
          After amount = 51430
          Leftover amount = −13245
```

- Tray 6 Leftover 1

```
food_image:
        4. pasta with clams and mussels
        6. grilled pork cutlet
        10. beans
        12. salad

leftover:
        4. pasta with clams and mussels
        6. grilled pork cutlet
        10. beans
        12. salad

4. pasta with clams and mussels
        Before amount = 104775
        After amount = 82070
        Leftover amount = 22705
6. grilled pork cutlet
        Before amount = 55376
        After amount = 18555
        Leftover amount = 36821
10. beans
        Before amount = 67942
        After amount = 61353
        Leftover amount = 6589
12. salad
        Before amount = 72538
        After amount = 68020
        Leftover amount = 4518
```

- Tray 6 Leftover 2



```
food_image:
          4. pasta with clams and mussels
          6. grilled pork cutlet
          10. beans
          12. salad

leftover:
          4. pasta with clams and mussels
          6. grilled pork cutlet
          10. beans
          12. salad

4. pasta with clams and mussels
          Before amount = 104775
          After amount = 83537
          Leftover amount = 21238
6. grilled pork cutlet
          Before amount = 55376
          After amount = 10335
          Leftover amount = 45041
10. beans
          Before amount = 67942
          After amount = 81863
          Leftover amount = −13921
12. salad
          Before amount = 72538
          After amount = 68460
          Leftover amount = 4078
```

- Tray 6 Leftover 3

```
food_image:
        4. pasta with clams and mussels
        6. grilled pork cutlet
        10. beans
        12. salad

leftover:
        4. pasta with clams and mussels
        6. grilled pork cutlet
        12. salad

4. pasta with clams and mussels
        Before amount = 104775
        After amount = 49264
        Leftover amount = 55511
6. grilled pork cutlet
        Before amount = 55376
        After amount = 49997
        Leftover amount = 5379
10. beans
        Before amount = 67942
        After amount = 0
        Leftover amount = 67942
12. salad
        Before amount = 72538
        After amount = 31804
        Leftover amount = 40734
```

- Tray 7 Leftover 1

```
food_image:
        4. pasta with clams and mussels
        7. fish cutlet
        11. basil potatoes
        12. salad

leftover:
        4. pasta with clams and mussels
        7. fish cutlet
        11. basil potatoes
        12. salad

4. pasta with clams and mussels
        Before amount = 105293
        After amount = 70368
        Leftover amount = 34925
7. fish cutlet
        Before amount = 46537
        After amount = 16227
        Leftover amount = 30310
11. basil potatoes
        Before amount = 42854
        After amount = 26617
        Leftover amount = 16237
12. salad
        Before amount = 63683
        After amount = 60847
        Leftover amount = 2836
```

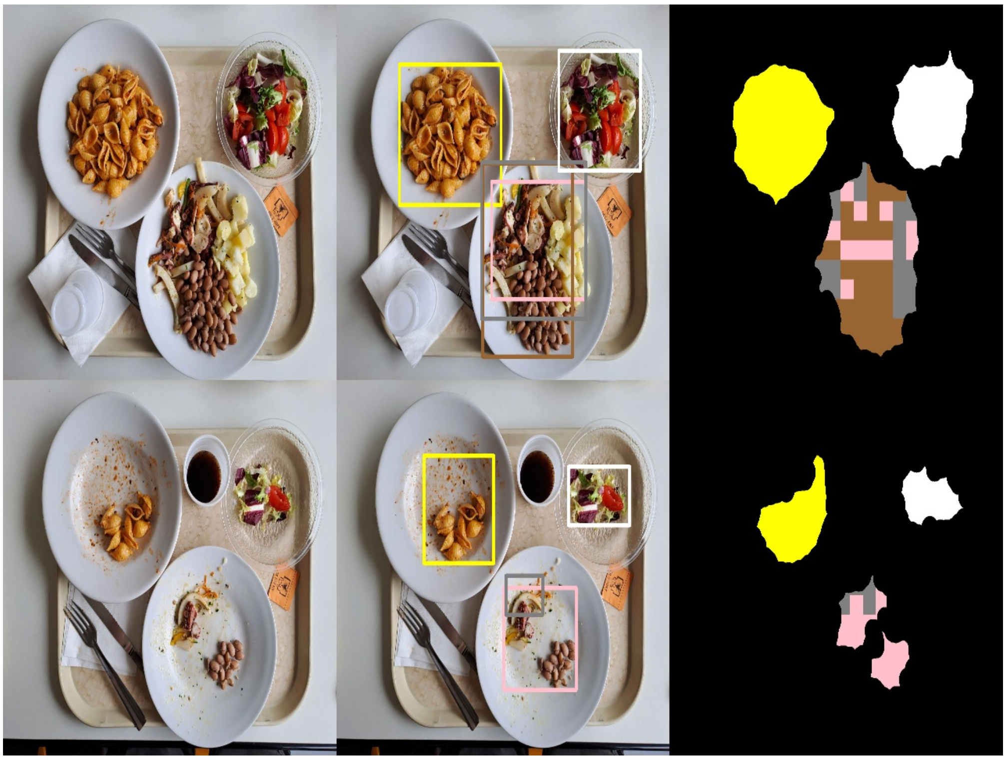
- Tray 7 Leftover 2

```
food_image:
        4. pasta with clams and mussels
        7. fish cutlet
        11. basil potatoes
        12. salad

leftover:
        4. pasta with clams and mussels
        7. fish cutlet
        11. basil potatoes
        12. salad

4. pasta with clams and mussels
        Before amount = 105293
        After amount = 75970
        Leftover amount = 29323
7. fish cutlet
        Before amount = 46537
        After amount = 16676
        Leftover amount = 29861
11. basil potatoes
        Before amount = 42854
        After amount = 32042
        Leftover amount = 10812
12. salad
        Before amount = 63683
        After amount = 66652
        Leftover amount = −2969
```

- Tray 7 Leftover 3

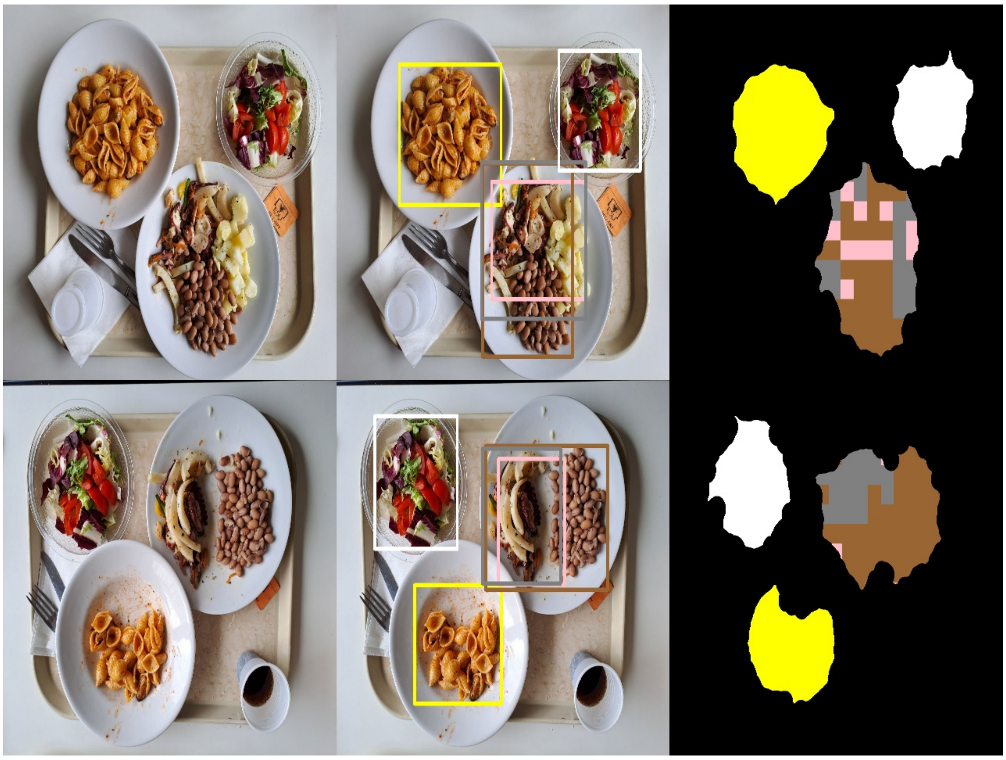
```
food_image:
        4. pasta with clams and mussels
        7. fish cutlet
        11. basil potatoes
        12. salad

leftover:
        4. pasta with clams and mussels
        11. basil potatoes

4. pasta with clams and mussels
        Before amount = 105293
        After amount = 48609
        Leftover amount = 56684
7. fish cutlet
        Before amount = 46537
        After amount = 0
        Leftover amount = 46537
11. basil potatoes
        Before amount = 42854
        After amount = 2856
        Leftover amount = 39998
12. salad
        Before amount = 63683
        After amount = 0
        Leftover amount = 63683
```

- Tray 8 Leftover 1



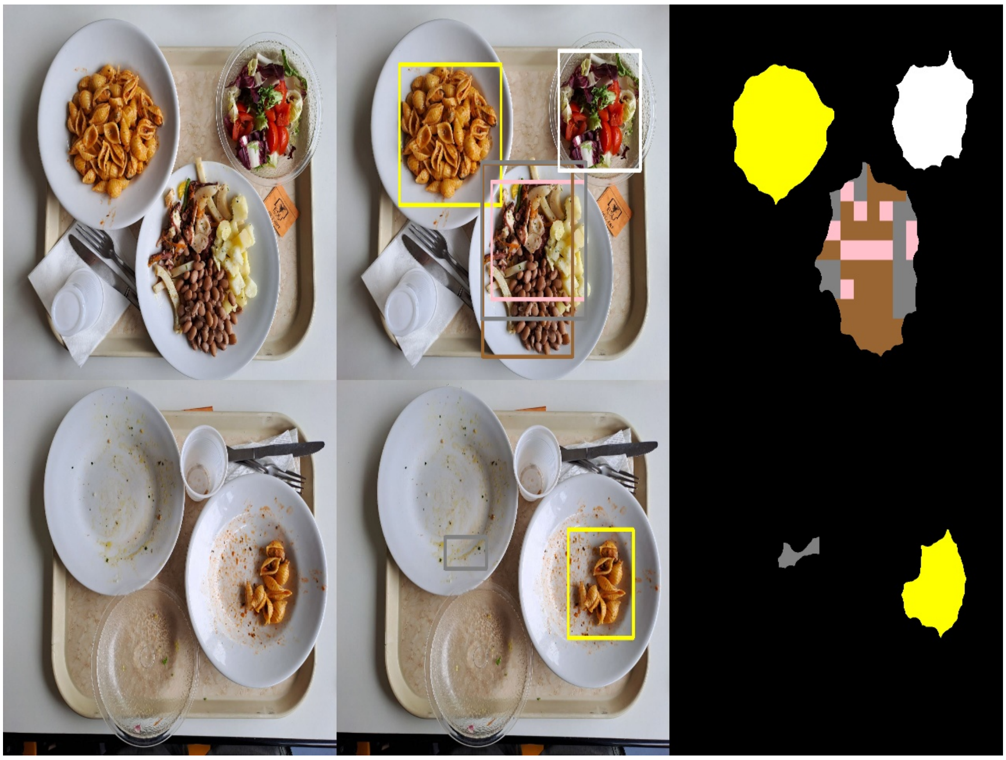
```
food_image:
        4. pasta with clams and mussels
        9. seafood salad
        10. beans
        11. basil potatoes
        12. salad

leftover:
        4. pasta with clams and mussels
        9. seafood salad
        11. basil potatoes
        12. salad

4. pasta with clams and mussels
        Before amount = 94917
        After amount = 38121
        Leftover amount = 56796
9. seafood salad
        Before amount = 26414
        After amount = 25172
        Leftover amount = 1242
10. beans
        Before amount = 70622
        After amount = 0
        Leftover amount = 70622
11. basil potatoes
        Before amount = 35671
        After amount = 6118
        Leftover amount = 29553
12. salad
        Before amount = 64321
        After amount = 21637
        Leftover amount = 42684
```

- Tray 8 Leftover 2



```
food_image:
          4. pasta with clams and mussels
          9. seafood salad
          10. beans
          11. basil potatoes
          12. salad
leftover:
          4. pasta with clams and mussels
          9. seafood salad
          10. beans
          11. basil potatoes
          12. salad
4. pasta with clams and mussels
          Before amount = 94917
          After amount = 66003
          Leftover amount = 28914
9. seafood salad
          Before amount = 26414
          After amount = 887
          Leftover amount = 25527
10. beans
          Before amount = 70622
          After amount = 82466
          Leftover amount = −11844
11. basil potatoes
          Before amount = 35671
          After amount = 36305
          Leftover amount = −634
12. salad
          Before amount = 64321
          After amount = 66914
          Leftover amount = −2593
```

- Tray 8 Leftover 3



```
food_image:
          4. pasta with clams and mussels
          9. seafood salad
          10. beans
          11. basil potatoes
          12. salad

leftover:
          4. pasta with clams and mussels
          11. basil potatoes

4. pasta with clams and mussels
          Before amount = 94917
          After amount = 40970
          Leftover amount = 53947
9. seafood salad
          Before amount = 26414
          After amount = 0
          Leftover amount = 26414
10. beans
          Before amount = 70622
          After amount = 0
          Leftover amount = 70622
11. basil potatoes
          Before amount = 35671
          After amount = 5549
          Leftover amount = 30122
12. salad
          Before amount = 64321
          After amount = 0
          Leftover amount = 64321
```