Final Report Computer Vison
Food Recognizer and Leftover Estimation

Osti Simone, Russo Christian Francesco, Spinato Matteo

General structure of the project

Folders of the project

- **FinalProjectCV:**
    - **images:**
      Contains the dataset of images provided for this project, where images are subdivided in subfolders *trayN*, each one containing *ground truth bounding boxes* (in .txt file), *ground truth masks* and *source images*.
    - **include:**
      Contains all the *header files* defined for this project.
    - **model:**
      Contains all the *trained models*, created using an additional Python script, and the label classes files used for this project.
    - **src:**
      Contains all the *source files* defined for this project.
- **PythonYoloFoodDetection:**
    - **train-yolov5-and-onnx-covert.ipynb:**
      Jupiter notebook used for training the models and for performing the conversion from pt to onnx.
    - **The dataset CV_group_projects.v40-tot_final.yolov5pytorch:**
      491 images divided in 441 for training set, 42 for validation set and 9 for test set, all multiplied by 3 in augmentation phase for a total of 1373 labelled images.
      (loaded on https://cloud.dei.unipd.it/index.php/s/qKSpRsWRyrAfGbF# for the upload of the project)

Classes of the project

- **Main.cpp:**
    - **Description:**
      Is the main file of the project.
    - **Main features:**
        - Reads the input parameters:
            - *root_path* of the project (e.g., “D:\\FinalProjectCV”)
            - *trayN* the set of images to elaborate (e.g., “Tray1”)
        - Creates an instance of the *FoodRecognizer_and_LeftoverEstimator* class and initialize it with the provided input parameters, then calls the train-of-functions:

        - *process_before_meal()*
        - *process_after_meal()*
        - *estimate_leftovers()*
        - *evaluate_performances()*

        which represent the core functions of the project.

- **FoodRecognizer_and_LeftoverEstimator.h/.cpp:**
    - **Description:**
      Is the core class of the project. It is the class that implements and handles all the main operation performed in the project.
    - **Main features:**
        - **Constructor (initialization):**
          Calls the *load_and_store_img_from_dataset()* function to load all the source and the ground truth images (bounding boxes and masks) from the dataset of images, for the selected *TrayN* and store them in the *images_table*.
        - **Main data structures:**
            - ***food_table***: is a table defined as follows:
              map<MealModality, map<FoodCategory, Food>> foods_table
            - ***images_table***: is a table defined as follows:
              map<MealModality, Images> images_table

          where *MealModality* is an enumeration which defines the 4 meal modality of our problem: before meal, after meal leftover 1, after meal leftover 2, after meal leftover 3, *FoodCategory* is an enumeration which defines the labels of the classes of foods that can be present in each image, *Food* is a struct which allows to store the images (ROIs, bounding boxes and masks) of each predicted and ground truth food, and *Images* is a struct which allows to store the images (bounding boxes and masks): source, predicted and ground truth containing all the foods present in the source image.
        - **Main public functions:**
            - ***process_before_meal():*** calls the following functions for the *MealModality* before meal:
                - *detect_foods()*
                - *segment_foods()*
                - *waitKey_next_meal()* (allows to pause the computation and to reset the plots between one *MealModality* and the other)

              Note: the previous functions receive as input a boolean parameter that allow to plot, or not, all the intermediate plots of the computations performed in the program.
            - ***process_after_meal():*** calls the following functions for the *MealModality* after meal:
                - *detect_foods()*
                - *segment_foods()*
                - *waitKey_next_meal()* (allows to pause the computation and to reset the plots between one *MealModality* and the other)

Note: the previous functions receive as input a boolean parameter that allow to plot, or not, all the intermediate plots of the computations performed in the program.

- ***estimate_leftovers():***
    - Calls the functions: *build_and_store_pred_boundingboxes_mask*() and *build_and_store_pred_mask*() to create a mask containing the bounding boxes and a mask containing the segmentations of all the foods predicted and store the resulting images in the *images_table*.
    - Plots all the images in the *foods_table* and all the images in the *images_table* using the functions *plot_foods_table*(), *plot_images_table_masks(), plot_images_table_overlapped_ROIs(), plot_images_table_filtered_ROIs()*
    - Calls the functions: *compute_and_store_leftover_estimations()* and *print_leftovers_table()*
- ***evaluate_performances():*** compute the mAP and the mIoU for every *MealModality*. The two performance estimations work only if the predicted mask and the ground truth mask both exist:
    - If the predicted mask is empty, but the ground truth exists, the performance score is set to 0, since the system did not recognize the food.
    - If the ground truth mask does not exist, skip the current iteration in the loop, since we do not evaluate a class if wrongly predicted, i.e., if there is a miss-matching (classification) w.r.t. its corresponding ground truth mask.
    - If predicted mask and ground truth mask both exist, the performance score is computed. This schema holds also for bounding boxes, and not only for masks.

▪ **Main private functions:**

- ***detect_foods():***
    - Finds the 3 ROIs: first dish ROI, second dish ROI, and the side dish ROI (tray without dishes).
    - For each kind of ROI, applies the most suitable classifier (among the 3 that we have created), in order to find the first meal foods, the second meal foods and the side foods and side dishes.
    - For each predicted/detected food, stores them in the *foods_table* (food class label and bounding box) using the *store_detected_food()* function.
- ***segment_foods():***
    - For each food detected, performs the segmentation of the area inside the bounding box.
    - Store the mask obtained for each food in the *foods_table* using the *store_segmented_food()* function.

Note: here it is important to notice that the foods in the *MealModality* after meal are segmented and stored only if they have been detected with the same class label in the *MealModality* before meal. Therefore, if the bounding boxes of the same food are detected both before and after meal, but in one of the two cases they are wrongly classified, this is considered as a non-matching.

- ***compute_and_store_leftover_estimations():***
    - For each food detected before meal, estimates the leftovers in each after meal food image as the ratio between the foreground pixels in the mask after meal and the foreground pixels in the mask before meal, as follows:

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

    - In addition, for each *MealModality* after meal, estimates, in the same way as in the previous point, the total foods leftovers over the masks of all foods and store the result in the *images_table.*
- Other functions were defined to handle the storage and the plotting of all the images stored in the two tables *food_table* and *images_table*.
- Other helper functions were defined to solve recurrent problems in the previous functions.

- **ROIsDetection.h/.cpp:**
    - **Description:**
      Provides a set of functions which allow to find the ROIs in the source image which might potentially contain some food.
    - **Main features:**
        - **Main public functions:**
            - ***detect_tray()*:** detects the tray ROI in the image.
              Note: for more details on the approaches adopted to solve this problem see the section ROIsDetection.
            - ***detect_dishes()*:** detects the dishes ROIs in the image.
              Note: for more details on the approach adopted to solve this problem see the section ROIsDetection.
            - ***get_first_dish_ROI()*:** returns an image containing the ROI of the first dish (the biggest one).
            - ***get_second_dish_ROI()*:** returns an image containing the ROI of the second dish (the smaller one).
            - ***get_sidedish_and_sidefood_ROI()*:** returns an image containing the ROI of the tray without the dishes. This ROI is used for finding side dishes and side foods.
        - Other helper functions were defined to solve recurrent problems in the previous functions.
- **FoodsDetection.h/.cpp:**
    - **Description:**

Provides a set of functions which allow to detect the set of foods contained in the ROIs provided.

- **Main features:**
  - **Constructor (initialization):**
    Calls the *load_YOLO_model()* function to load the most suitable model for the kind of foods we want to detect (i.e., first course food, second course food, or side dish and side food).
  - **Main public functions:**
    - ***detect_foods()***: detects all the foods present in the input ROI.
      Note: for more details on the approach adopted to solve this problem see the section FoodsDetection.
    - ***get_foods_map():*** returns a map containing as keys the labels of the classes of foods detected, and as corresponding values the pairs (food bounding box, food ROI).
  - Other helper functions were defined to solve recurrent problems in the previous functions.

- **FoodsSegmentation.h/.cpp:**
  - **Description:**
    Provides a set of functions which allow to segment and produce a mask for each food detected.
  - **Main features:**
    - **Main public functions:**
      - ***segment_food():*** segments and produces a mask of the food detected contained within the provided bounding box. This method is used only for the *MealModality* before meal.
      - ***segment_food_in_color_ranges():*** segments and produces a mask of the food detected contained in the provided bounding box. This method is used for the *MealModality* after meal.
      - ***get_food_mask():*** returns the mask of the food segmented.
      - ***get_food_color_ranges():*** returns the set of colors collected for the food segmented in the *MealModality* before meal, to be used for segmenting the corresponding food in the Meal modality after meal.
    - **Main private functions:**
      Novel method 1: (first weak segmentation method proposed)
      - ***extraction_of_color_ranges():*** this method implements the 1st step of the 1st and 2nd novel methods.
      - ***thresholding_with_color_ranges():*** this method implements the 2nd step of the 1st novel method.
      - ***mask_enhancement_general_before():*** this method implements the 3rd step of the 1st novel method (part before meal).
      - ***mask_enhancement_general_after():*** this method implements the 3rd step of the 1st novel method (part after meal).

- ***first_novel_method_segmentation():*** this method just calls the aforementioned functions to perform the 3 steps of the 1st novel segmentation method proposed (for the *MealModality* before meal), in order to produce and store the first weak segmented mask.
- ***first_novel_method_segmentation_in_color_ranges():*** this method just calls the aforementioned functions to perform the 3 steps of the 1st novel segmentation method proposed (for the *MealModality* after meal), in order to produce and store the first weak segmented mask.

Novel method 2: (second weak segmentation method proposed)

- ***fill_contours_with_color_ranges():*** this method implements the 2nd step of the 2nd novel method.
- ***mask_enhancement_morph_close_before():*** this method implements the 3rd step of the 2nd novel method (part before meal).
- ***mask_enhancement_morph_open_after():*** this method implements the 3rd step of the 2nd novel method (part after meal).
- ***second_novel_method_segmentation():*** this method just calls the aforementioned functions to perform the 3 steps of the 2nd novel segmentation method proposed (for the *MealModality* before meal), in order to produce and store the first weak segmented mask.
- ***second_novel_method_segmentation_in_color_ranges():*** this method just calls the aforementioned functions to perform the 3 steps of the 2nd novel segmentation method proposed (for the *MealModality* after meal), in order to produce and store the first weak segmented mask.

Ensamble: (strong segmentation method)

- ***ensamble_segmentation():*** combines the two weak masks, using a bitwise AND operation, and produces the final mask of the segmented food.

Note: for more details on the two novel methods proposed see the section

FoodsSegmentation.

- **PerformanceEvaluation.h/.cpp:**
  - **Description:**
    Provides a set of functions for evaluating the overall performance of the program.
  - **Main features:**
    - **Main public functions:**
      - ***IoU_evaluation():*** evaluate the IoU score of a predicted mask w.r.t. its corresponding ground truth mask.
- **DatasetCategories.h/.cpp:**
  - **Description:**
    Provides a set of enumerations and the relative helper functions for covering all possible classes of entities that we can find in the dataset of images and in the definition of our problem.

- **Main features:**
  - **Enumerations defined:**
    - ***Tray:*** *TRAY_1, …, TRAY_8.*
    - ***ImagesType:*** SOURCE, BOUNDING_BOX, MASK.
    - ***Kind:*** PREDICTED, GROUND_TRUTH.
    - ***MealModality:*** BEFORE_MEAL, …, AFTER_MEAL_LEFTOVER_3.
    - ***FoodCourse***: *FIRST_COURSE, SECOND_COURSE, SIDE_DISH_AND_SIDE_FOODS.*
    - ***FoodCategory:*** BACKGROUND, PASTA_WITH_PESTO, …, BREAD.
  - **Main helper functions:** (common to all the enumerations)
    - *toString()*
    - *index()*
    - *to<Enum>()*

    Other more specific helper functions were defined.

- **library/Util.h/.cpp:**
  - **Description:**

    It is an util library class which provides general purposes utility functions.
  - **Main util functions provided:**
    - *type2str()*: converts the image type to a string representing the corresponding OpenCV data type.
    - *fullfile()*: implements the fullfile function analogous to the one defined in MATLAB, which allows to concatenate the paths of the files in a cross-platforms way.
    - *subplot()*: implements the subplot function analogous to the one defined in MATLAB, which allows to create and plot a grid of images (independently by their size and their number of channels), with any number of rows and columns, and allows to associate to each plotted image a label which describe it.

Approaches adopted to solve the problem

ROIsDetection

- **Goal:**
    - Identify the tray ROI (without dishes)
    - Identify the dishes ROIs (separately)
- **Motivations:**
  Identifying the ROIs which might possibly contain some food is important in order to:
    - Use a specific food detector approach depending on the kind of food we are searching for (i.e., first course food, second course food, or side dish and side food).
    - Allow to enhance the confidence of the detection.
- **Challenges:**
    - Developing an approach that can work and adapt well to all the images, i.e., for many different perspectives, dimensions of the objects, viewpoints, illumination conditions, etc., i.e., for the different kind of situations that we can find in the dataset of images.
    - Tuning all the parameters (which are many) in such a way our approach could work well in all the aforementioned situations.
- **Adopted solutions:**
    - **Tray:** steps of *detect_tray()*:
        - Input: a source image.
        - Extract the marginal part of the image, by adding a central rectangular mask → to obtain a region in which search for the lines of the edges of the tray.
        - Convert the image to grayscale → to reduce image complexity.
        - Apply gaussian blur → to reduce noise and details in the image.
        - Apply Canny edge detector → to find the edges of the tray in the image.
        - Find lines in the marginal part of the image using the Hough transform.
        - Filter almost horizontal and almost vertical lines.
        - Find the tray vertices, by computing the intersection points of all the lines and by picking out the nearest ones to each corner of the image.
        - Build the polygon containing the tray, by linking together the four vertices of the tray found in the previous step.
        - Create a mask for identifying the area of the image containing the tray.
        - Extract the ROI of the image containing the tray by applying the previous mask on the source image.

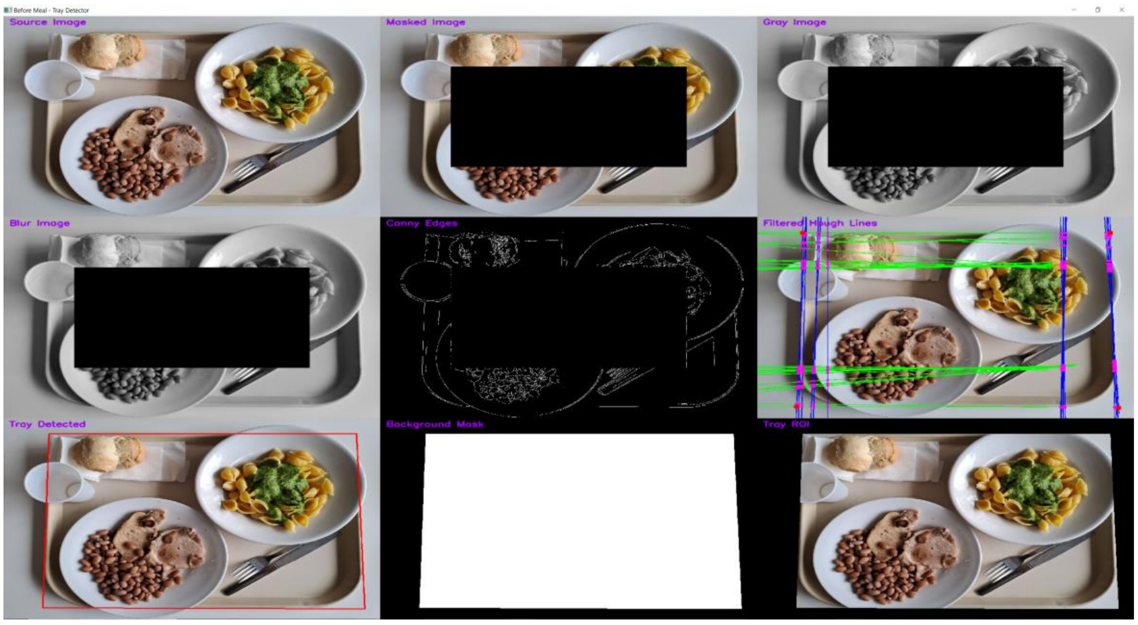


Figure 1 - Example of Tray Detection

- **Dishes:** steps of *detect_dishes()*:
    - Input: a source image.
    - Convert the image to grayscale → to reduce image complexity.
    - Apply gaussian blur → to reduce noise and details in the image.
    - Apply Canny edge detector → to find the edges of the dishes in the image.
    - Find the circles of the dishes in the image in adaptive way (with adaptive parameters):
        - Termination condition: two dishes are found or maximum number of iterations.
        - Find circles in the image using the Hough circles transform with the current parameters of the Hough circles transform.
        - Filter the circles found:
            - Delete circles that are too much overlapped
            - Delete circles that are only partially contained in the image
        - Update the adaptive parameters of the Hough circles transform for the next iteration of the algorithm.
    - Create a mask for identifying the area of the image containing the two dishes.
    - Extract the ROI of the image containing the two dishes by applying the previous mask on the source image.

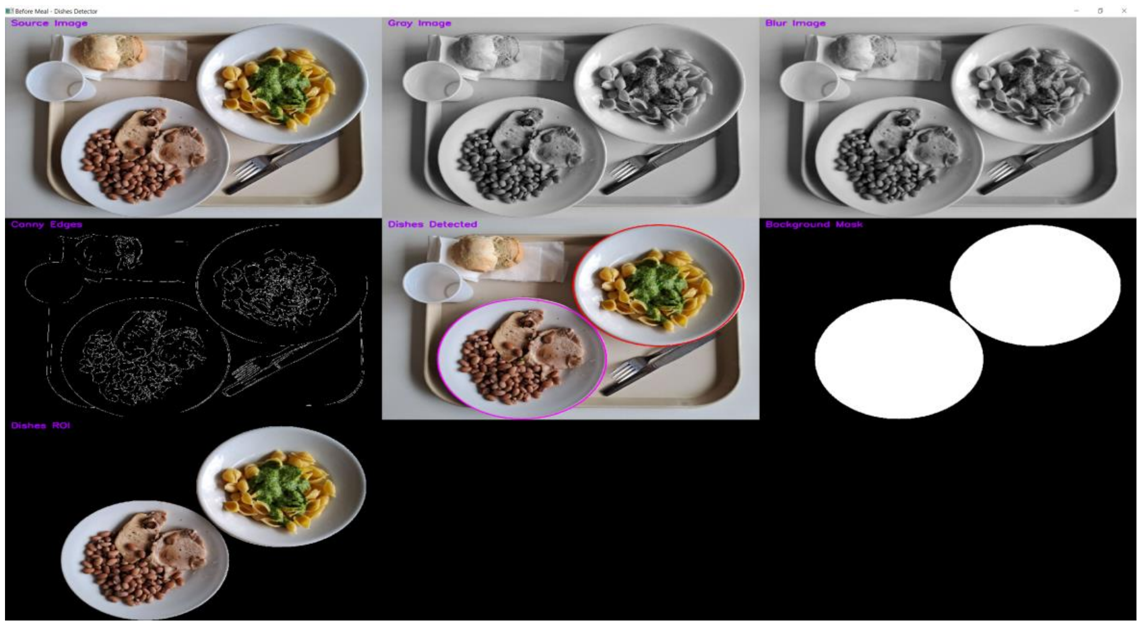


Figure 2 - Example of Dishes Detection

- **Other solutions tested:**
  - **Tray:**
    - We have tested the Hough transform method to identify the rectangle of the tray in the image.
    - We have tested the Hough transform method to identify the rounded rectangle of the tray in the image.

    These solutions did not work in our case, since the pictures where shotted using different perspective, therefore the tray has often the shape of a generic trapezoid. Another problem is that the edges of the tray are very often obliterated by many other objects, such as the dishes, or the food present on the tray.
  - **Dishes:**
    - We have tested the Hough circles transform method to identify the dishes in the image without using adaptive parameters.

    Finding parameters that work well for all the images was really hard.
    Fortunately, unlike the tray detection problem, in this case we observed that almost all the images (except one) contain 2 dishes, hence this condition is sufficient to establish a termination condition for our adaptive algorithm.
    In the case in which the image contains only one dish, then a false positive dish is found in addition to the true positive dish. This is not a problem, since it only represents an additional ROI in which perform the food detection (with no food matching).

FoodsDetection

- **Goal:**
  - Detect the food
  - Classify the food
- **Approach:** deep learning method: we trained a YOLOv5 system to detect and classify foods in the tray, after defining a dataset. Our decision was due to the advantage of having a system able

to detect and classify the foods at the same time. The dataset was created using the online platform Roboflow, and was populated by 491 images accurately selected from other datasets in the platform or outside, or from the web.

- **Challenges:**
  - Accuracy: with few samples, and with only a classifier, the classification was very poor. This problem is typical of deep learning methods. We need a bigger dataset in order to get a good level of accuracy.
  - Leftovers: the detection of leftover food did not work well, since we used a dataset of images before meal (i.e., with no foods leftovers).
- **Adopted solutions:**
  - We solved the accuracy problem by reducing the number of foods that can be detected by a single classifier, using multiple classifiers class-specialized. In this way, each classifier has less food categories to worry about. In practice, we developed 3 different classifiers, for first course food, second course food, and side food. For each dish detected in order first detect if there is a first course, then based on the answer look for a second course and their associated side dish. Then finally search in the tray if there is a side dish like bread or salad. In this way there is no possibility of misclassifying a first course on a second course or on a side.
  - We use three thresholds (confidence, score and non-maximum suppression threshold), by combining them we can keep only the highest scoring results and manage bounding box overlap. The thresholds have been tuned for each classifier and for each moment of the meal (before, leftover1, leftover2, leftover3).
  - Then, we used data augmentation to increment the cardinality of the training set and get better performances.

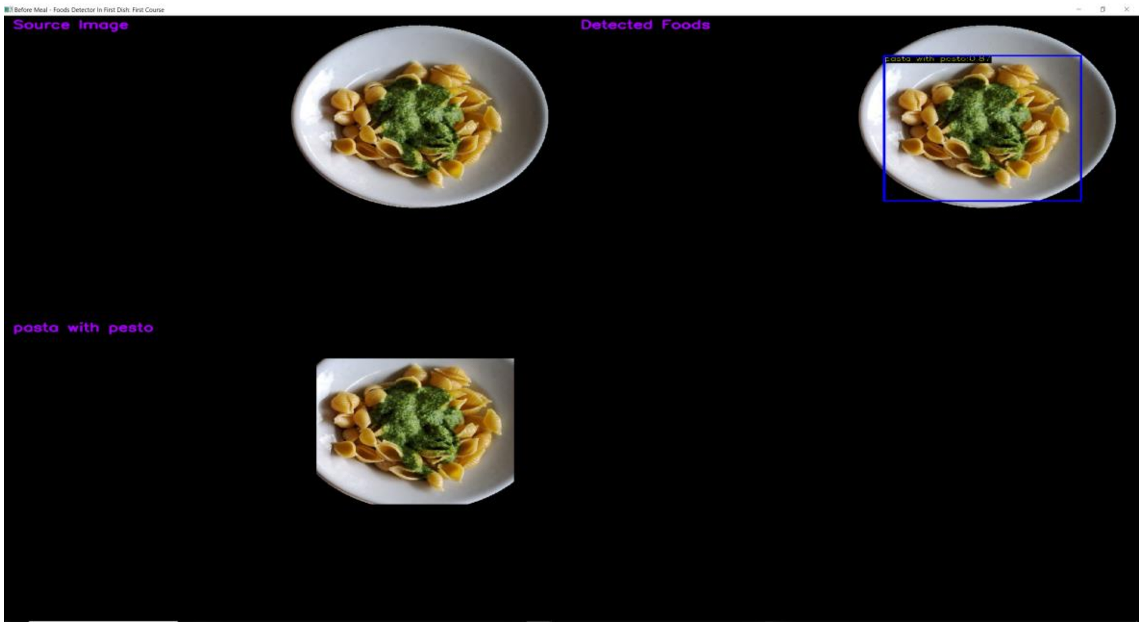


Figure 3 - Example of Foods Detection

FoodsSegmentation

- **Goal:**
  - Develop a good method for segmenting a wide range of different foods in an image.
  - Manage the different food-situations before and after meal.
  - Tuning all the parameters in such a way our approach could work well in all possible different situations.
- **Challenges:**
  - Develop an approach that can work and adapt well to all the foods. This is very hard task, since we have a lot of different foods in our dataset, and since foods are soft objects, the same food can be present in very different arrangements, shapes, and colors on the plate.
  - Manage the different cases after and before meal for each food. In the *MealModality* before meal, obtaining a good segmentation of each food is a hard task. However, segmenting food in the *MealModality* after meal is much harder, this is due to the fact that the foods are much more mixed with other foods and present different shapes and colors than before. Moreover, in these last cases, a lot of leftovers sprinkles and a lot of dirty left by foods are spread all over the dish, thus, it becomes really hard to discriminate dirty from food.
- **Adopted solutions:**
  - **Novel method 1:** steps of *first_novel_method_segmentation()* and *first_novel_method_segmentation_in_color_ranges():*
    - ***extraction_of_color_ranges():***
      - Creates an elliptical pattern of circles (as shown in Figure 4) placed exactly in the center of the bounding box area containing the food.
      - The shape and the size of the ellipse and the size of each circle adapt automatically to the size of the bounding box.
      - Collects and stores the food colors extracted from each circle.

      This step is useful for sampling the image colors (that will be used in the next steps of the proposed methods).
      Note: this step uses as starting hypothesis the fact that the bounding box of the food is placed exactly around it, so we are sure to extract exactly the ranges colors of that food.
    - ***thresholding_with_color_ranges():***
      - Performs a thresholding operation for filtering all the pixels whose colors in contained in the range of colors collected for that food.
    - ***mask_enhancement_general_before() and mask_enhancement_general_after():***
      - They perform a sequence of morphological operations and of contours refinements, in order to remove the dirty left by food, remove holes or imperfections in the final segmentation mask due to shadows and better define the borders of food.

- Note: this step is performed only for the *MealModality* before meal, this because in such case the food is more uniformly distributed within the bounding box and it is more unlikely (less probable) that one of the circles falls over the dish, including in this way also the color of the dish among the range of colors collected for that food, and leading in this way to a very bad segmentation!
  The range of colors collected in the *MealModality* before meal is used for segmenting that food for both the *MealModality* before meal and after meal.

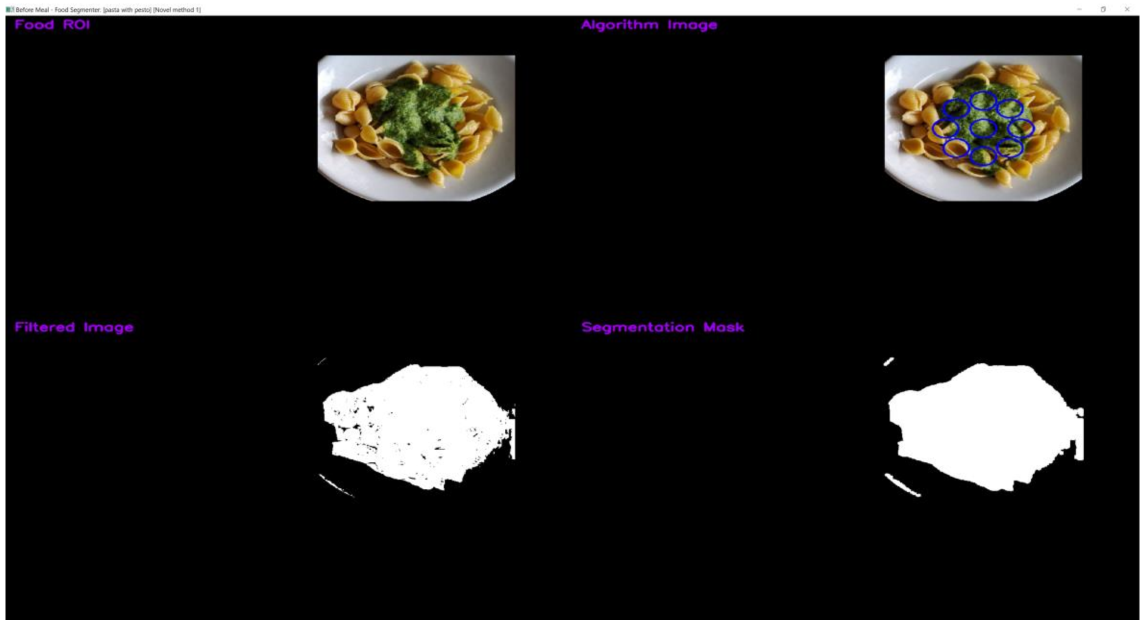

Before Meal - Food Segmenter: [pasta with pesto] [Novel method 1]
Food ROI
Algorithm Image
Filtered Image
Segmentation Mask

Figure 4 - Example of (weak) Food Segmentation with 1st Novel Method

- **Novel method 2:** steps of *second_novel_method_segmentation()* and *second_novel_method_segmentation_in_color_ranges():*
  - ***fill_contours_with_color_ranges():***
    - Uses the OpenCV function *findContours()* for finding the food in the image, in particular for each range of color extracted by each circle (in 1st step), creates a mask and stores it in a vector.
    - Merges together all the masks in the vector using the following ranking rule: for each pixel in the final mask, if that pixel is a foreground pixel in at least min_votes (parameter) masks stored in the vector, then it will be a set as aforeground pixel in the final mask too, otherwise it will be set as a background pixel.
  - ***mask_enhancement_morph_close_before() and mask_enhancement_morph_open_after():***
    - They perform a sequence of morphological operations in order to remove the dirty left by food, remove holes or imperfections in the final segmentation mask due to shadows and better define the borders of food.

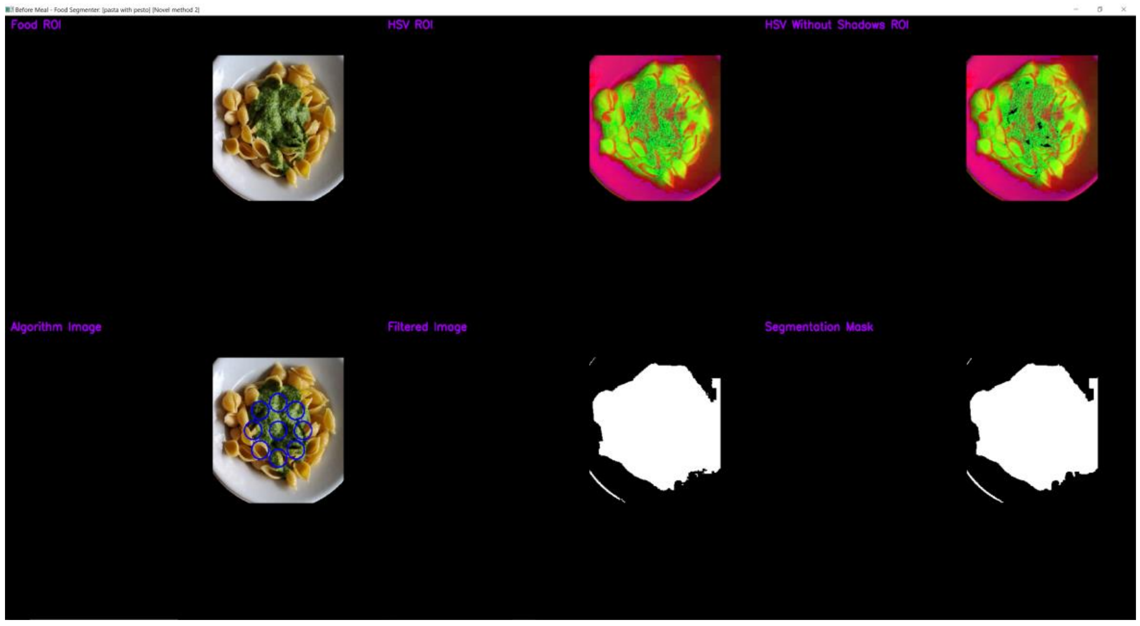


Figure 5 - Example of (weak) Food Segmentation with 2nd Novel Method

- **Ensemble:** function *ensemble_segmentation()*:
  - Combines the two weak masks obtained, using a bitwise AND operation, and produces the final mask of the segmented food.

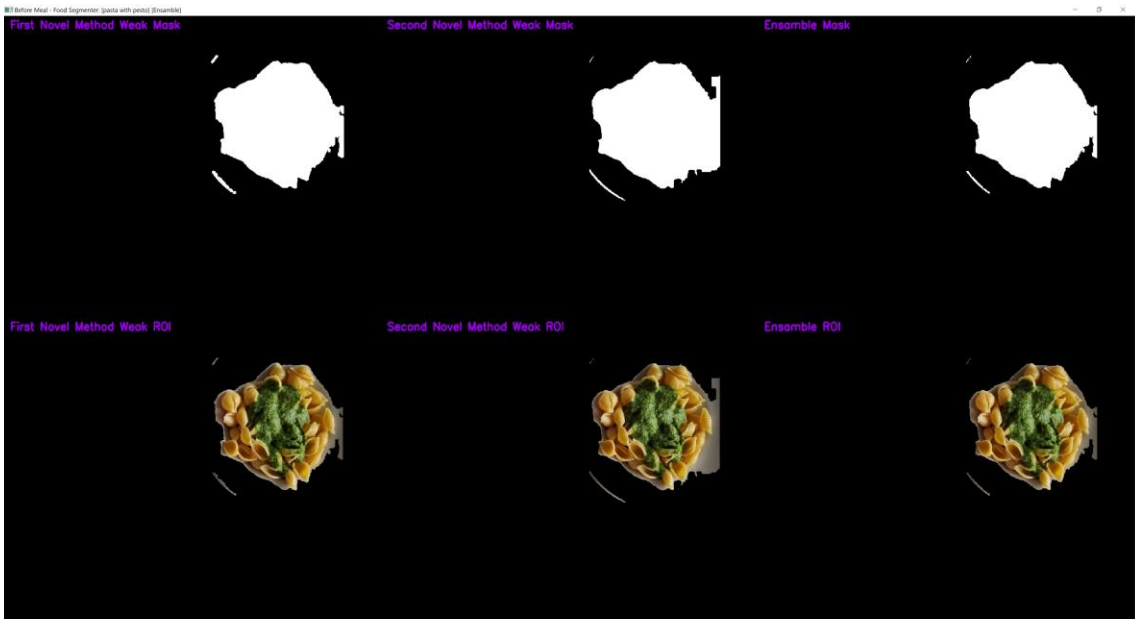

Before Meal - Food Segmenter: [pasta with pesto] [Ensemble]
First Novel Method Weak Mask
Second Novel Method Weak Mask
Ensemble Mask
First Novel Method Weak ROI
Second Novel Method Weak ROI
Ensemble ROI

Figure 6 - Example of (strong) Food Segmentation with Ensemble

Note: all the operations described for the 1st and the 2nd novel method are performed in the HSV (Hue, Saturation, Value) color space, instead of the BGR space. This because we performed many tests with many different approaches for trying to implement shadows cancellation (without noteworthy results), and in the end we observed better performance in this color space.

- **Other solutions tested:**
  - The first segmentation approach that we tested was k-means segmentation method, but it did not perform very well, this was also due to the parameter k needed a priori.

- The second segmentation approach that we tested was Canny edge detector + morphological closing. If the Canny detector produces a lot of edges, a segmentation mask can be created by closing all the edges. This approach was discharged since, in order to erase the white pixels of the dish, it was used a threshold on the colours, erasing also part of the foods, and other edges were taken into consideration, not allowing the filling of the mask.
- We have tried many different approaches in HSV color space for trying to implement foods shadows cancellation, in order to better define the borders of the foods in the final segmentation mask, but with poor results. This was due to the fact that many shadows are projected by the food shapes over the food itself, so if we remove shadows, we risk to remove part of the food, and it is really difficult to tune a good threshold to avoid this behaviour while removing marginal shadows.
- Lastly, the very last moment, as suggested by other colleagues, we tested a further approach using the OpenCV *grabcut()* function, but we obtained really poor results.

PerformanceEstimation

We implemented the requested performance estimations, but we made a simplification on the mean Average Precision calculation. In our classification problem, we have for sure a single occurrence of a food class in each tray, no more than one. In this way, we don't need to build a table of confidence scores of multiple occurrences of the same class, as reported in the linked paper, so the calculation of Precision and Recall can be simplified, since we do not consider the cumulative element. Then, Precision and Recall are equal to 0 or 1 both, and the Precision-Recall graph has area 0 or 1.

Consequently, the formula AP = 1/11 * Sum(11 point interpolated precision) is equal to 0 or 1. In conclusion, the AP formula can be simplified by just looking at the IoU of the bounding boxes: if the IoU score is bigger than the threshold 0.5, we have an AP score equal to 1, otherwise equal to 0. Confirmation of this fact can be seen in the paper example, where the classes of "sheep" and "truck" are composed by only one occurrence, and their AP is 0 or 1.

In practice, the calculation of AP and IoU is performed for segmentation and localization if there are both predicted mask (or bounding box) and ground truth mask (or bounding box) of the same food, or if there is the ground truth mask without the predicted mask. In this last case, the score computed is equal to 0, since the system did not detect the food.
If the system detected a food not belonging to the ground truth list, we don't evaluate the wrong class prediction, and this wrong label is not considered for the mean of IoU or AP.

DeepLeaning models (Python)

- **Dataset:** the dataset has been made with Roboflow, an online platform that allows to manage dataset, upload images, label them, copy images from other datasets, apply data augmentation and download created datasets. Unfortunately, the task of detection and classification of food it's very specific, with a wide range of food types, dishes and very high intraclass variance this causes great difficulty in finding suitable images for the dataset (even other images taken from Italian university cafeterias are often not good).

In a real world case, every institution that wants to apply such a system to check the amount of food leftover should first build its own dataset with images of its own foods, this would greatly improve the model performance.

The dataset consists of 491 images divided into 441 for training set, 42 for validation set and 9 in the test set. Then we tried several data augmentation types, such as rotation, crop, saturation, brightness, exposure, noise, also applied to bounding boxes (crop). In our tests flip, rotation and brightness change turned out to be better, so we augmented the dataset to 1373 images.

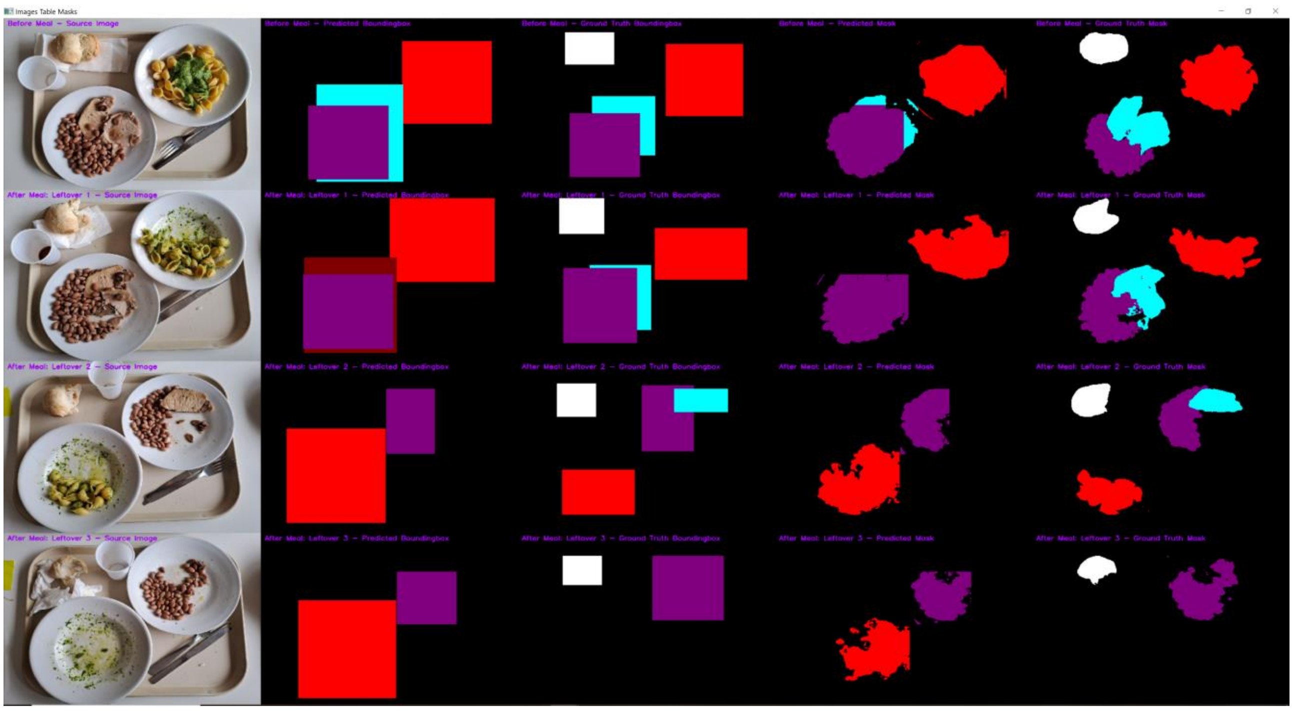
- **Training:** the training has been done in Google Colab in order to have some free gpu time for the computation, il the provided Jupiter notebook there is the code used for training and the conversion from a model.pt, pytorch type of model, to our model.onnx more suitable for OpencCV.
- **Classifiers:** after many tests whit a single model to detect all the dishes we notice that the detection of first course tends to be better than other the detection of food type so built a system whit two classifier, one for first course and one for other, that seek for a second or a side dish only new doesn't find a first course.

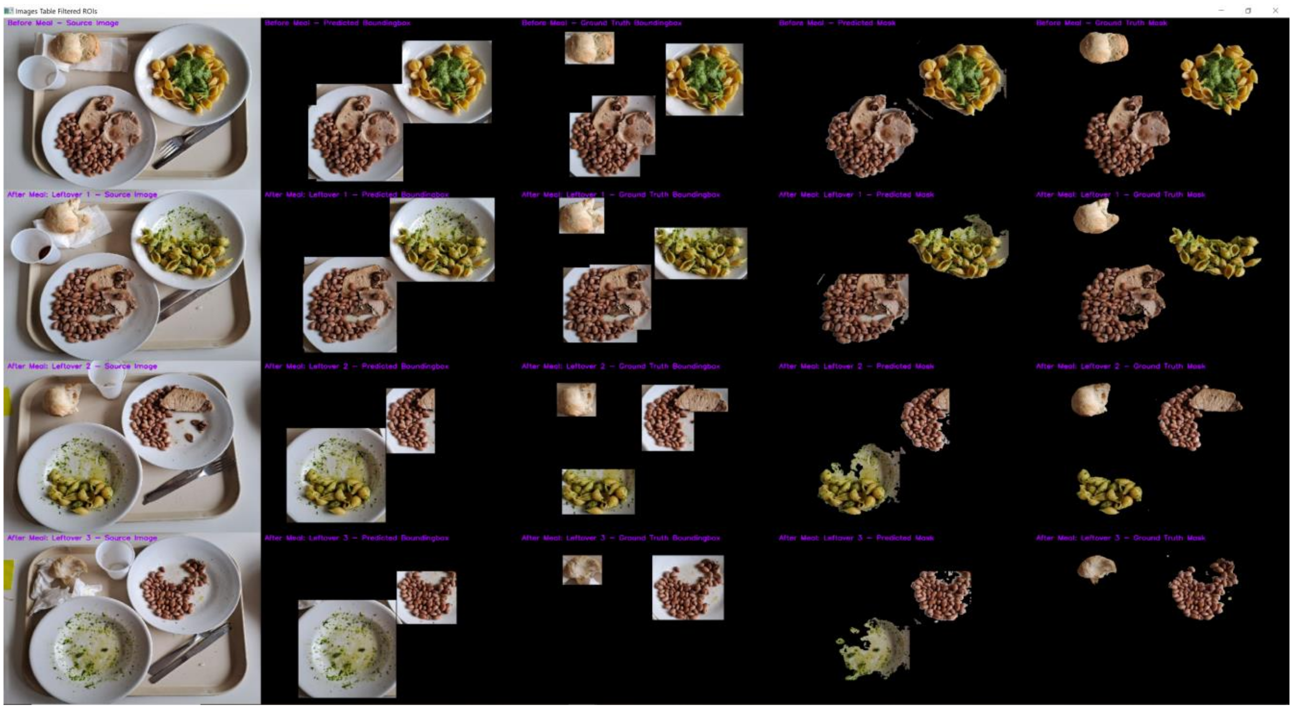
  After many further tests, some with two classifiers (one for food in the dishes and one for food in the tray), we split again the classifiers to three that first seek for a first course in a dish at time, then if doesn't find it seek for a second course on it and then the last classifier seek for side food firstly in the dishes then in the tray.

Results

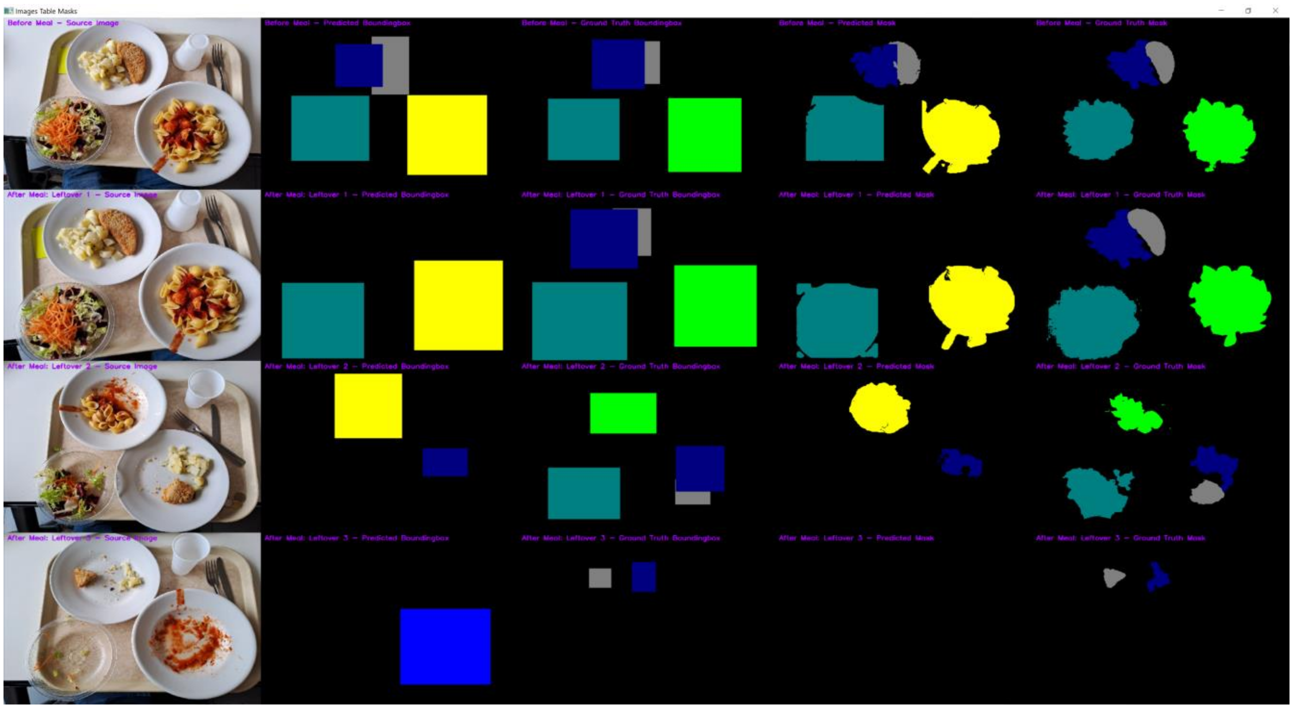
Images

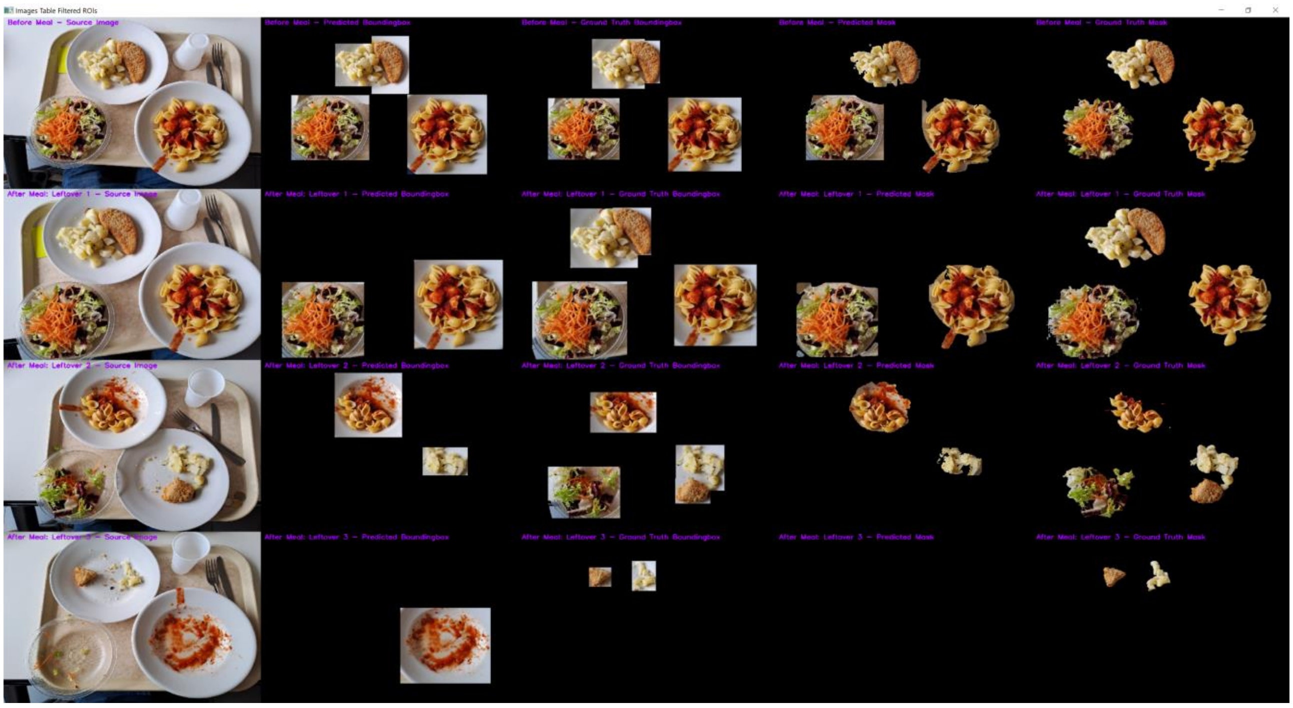
Tray 1


Images Table Masks
Before Meal – Source Image
Before Meal – Predicted Boundingbox
Before Meal – Ground Truth Boundingbox
Before Meal – Predicted Mask
Before Meal – Ground Truth Mask
After Meal: Leftover 1 – Source Image
After Meal: Leftover 1 – Predicted Boundingbox
After Meal: Leftover 1 – Ground Truth Boundingbox
After Meal: Leftover 1 – Predicted Mask
After Meal: Leftover 1 – Ground Truth Mask
After Meal: Leftover 2 – Source Image
After Meal: Leftover 2 – Predicted Boundingbox
After Meal: Leftover 2 – Ground Truth Boundingbox
After Meal: Leftover 2 – Predicted Mask
After Meal: Leftover 2 – Ground Truth Mask
After Meal: Leftover 3 – Source Image
After Meal: Leftover 3 – Predicted Boundingbox
After Meal: Leftover 3 – Ground Truth Boundingbox
After Meal: Leftover 3 – Predicted Mask
After Meal: Leftover 3 – Ground Truth Mask

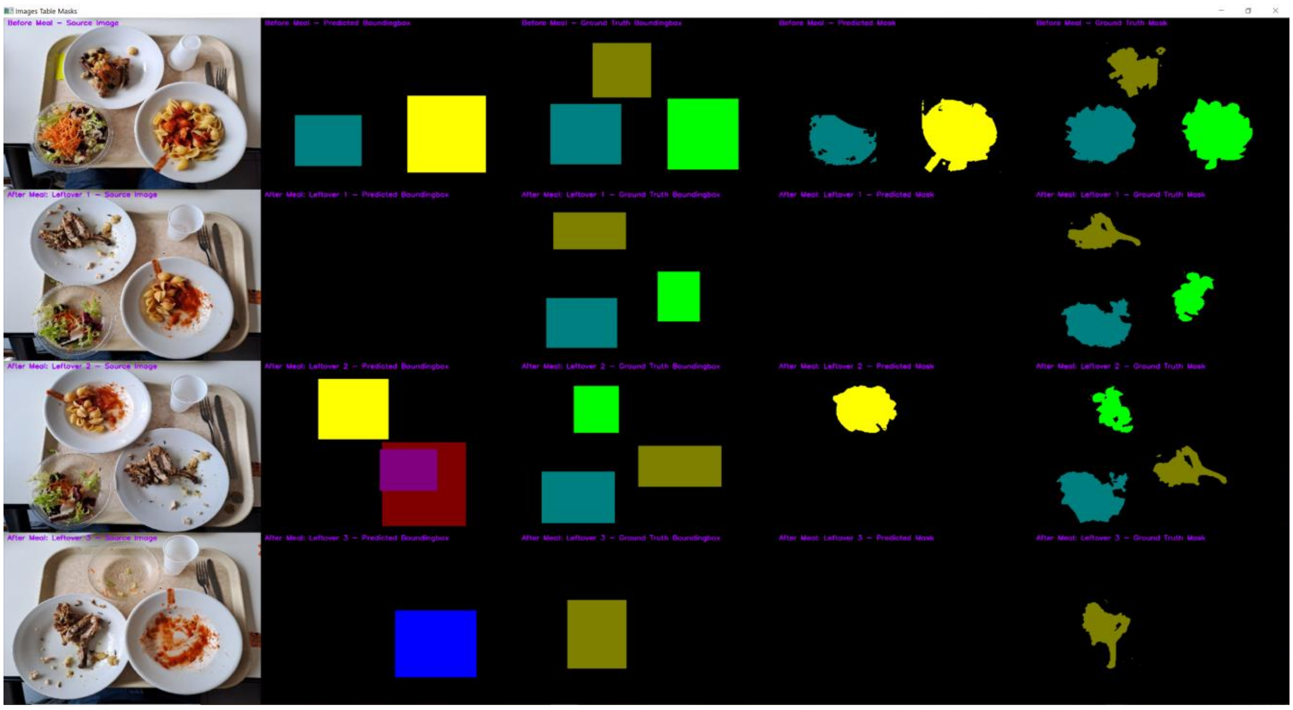

Images Table Filtered ROIs
Before Meal – Source Image
Before Meal – Predicted Boundingbox
Before Meal – Ground Truth Boundingbox
Before Meal – Predicted Mask
Before Meal – Ground Truth Mask
After Meal: Leftover 1 – Source Image
After Meal: Leftover 1 – Predicted Boundingbox
After Meal: Leftover 1 – Ground Truth Boundingbox
After Meal: Leftover 1 – Predicted Mask
After Meal: Leftover 1 – Ground Truth Mask
After Meal: Leftover 2 – Source Image
After Meal: Leftover 2 – Predicted Boundingbox
After Meal: Leftover 2 – Ground Truth Boundingbox
After Meal: Leftover 2 – Predicted Mask
After Meal: Leftover 2 – Ground Truth Mask
After Meal: Leftover 3 – Source Image
After Meal: Leftover 3 – Predicted Boundingbox
After Meal: Leftover 3 – Ground Truth Boundingbox
After Meal: Leftover 3 – Predicted Mask
After Meal: Leftover 3 – Ground Truth Mask

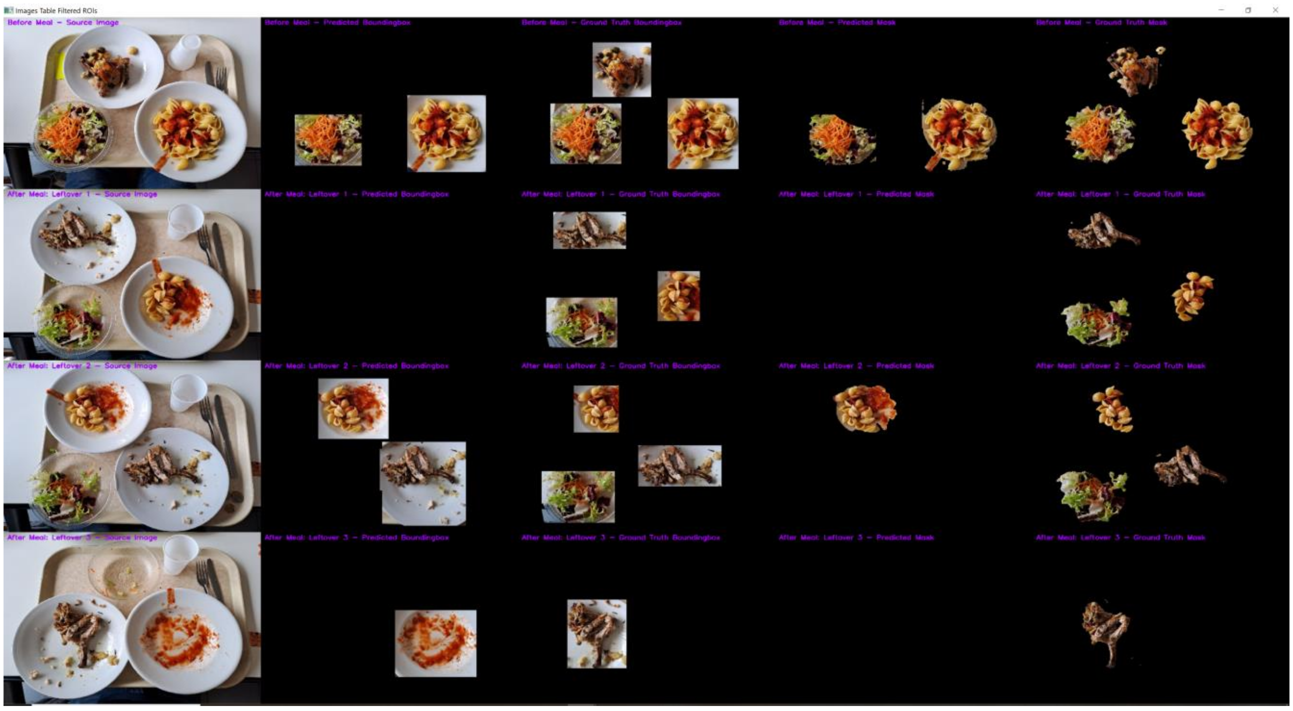
Tray 2


Images Table Masks
Before Meal – Source Image
Before Meal – Predicted Boundingbox
Before Meal – Ground Truth Boundingbox
Before Meal – Predicted Mask
Before Meal – Ground Truth Mask
After Meal: Leftover 1 – Source Image
After Meal: Leftover 1 – Predicted Boundingbox
After Meal: Leftover 1 – Ground Truth Boundingbox
After Meal: Leftover 1 – Predicted Mask
After Meal: Leftover 1 – Ground Truth Mask
After Meal: Leftover 2 – Source Image
After Meal: Leftover 2 – Predicted Boundingbox
After Meal: Leftover 2 – Ground Truth Boundingbox
After Meal: Leftover 2 – Predicted Mask
After Meal: Leftover 2 – Ground Truth Mask
After Meal: Leftover 3 – Source Image
After Meal: Leftover 3 – Predicted Boundingbox
After Meal: Leftover 3 – Ground Truth Boundingbox
After Meal: Leftover 3 – Predicted Mask
After Meal: Leftover 3 – Ground Truth Mask

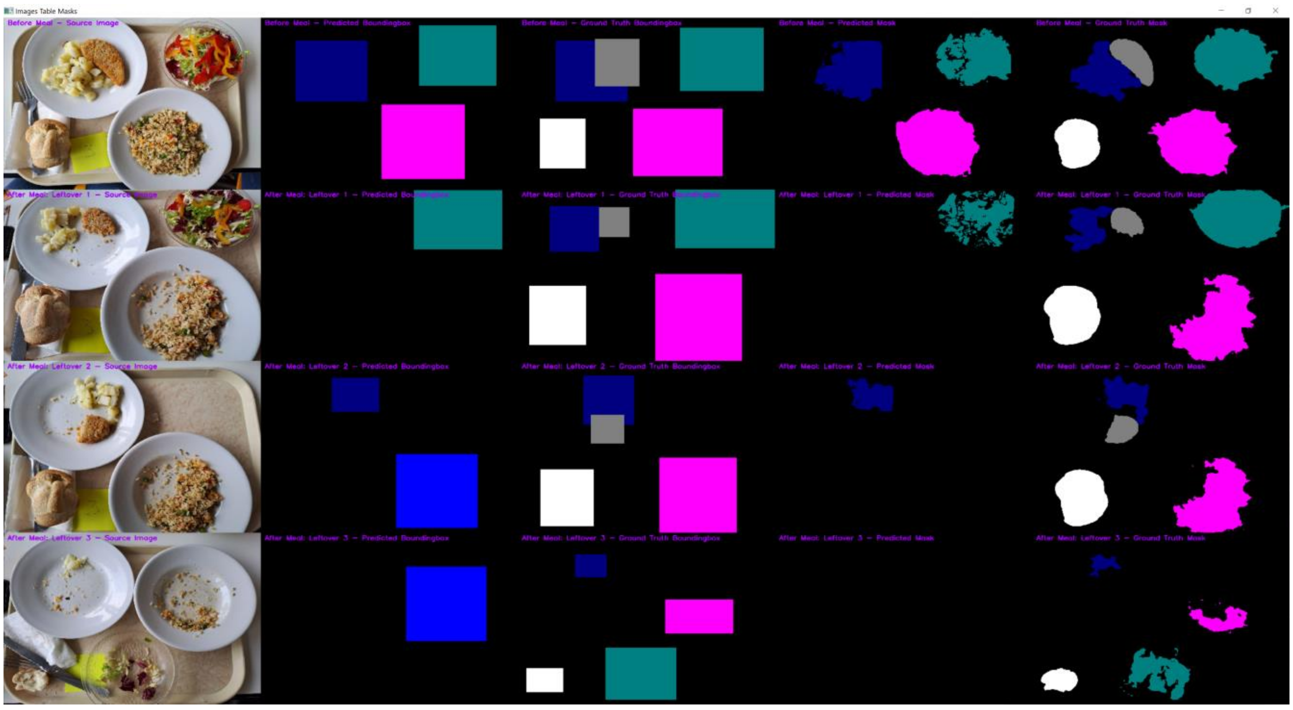

Images Table Filtered ROIs
Before Meal – Source Image
Before Meal – Predicted Boundingbox
Before Meal – Ground Truth Boundingbox
Before Meal – Predicted Mask
Before Meal – Ground Truth Mask
After Meal: Leftover 1 – Source Image
After Meal: Leftover 1 – Predicted Boundingbox
After Meal: Leftover 1 – Ground Truth Boundingbox
After Meal: Leftover 1 – Predicted Mask
After Meal: Leftover 1 – Ground Truth Mask
After Meal: Leftover 2 – Source Image
After Meal: Leftover 2 – Predicted Boundingbox
After Meal: Leftover 2 – Ground Truth Boundingbox
After Meal: Leftover 2 – Predicted Mask
After Meal: Leftover 2 – Ground Truth Mask
After Meal: Leftover 3 – Source Image
After Meal: Leftover 3 – Predicted Boundingbox
After Meal: Leftover 3 – Ground Truth Boundingbox
After Meal: Leftover 3 – Predicted Mask
After Meal: Leftover 3 – Ground Truth Mask

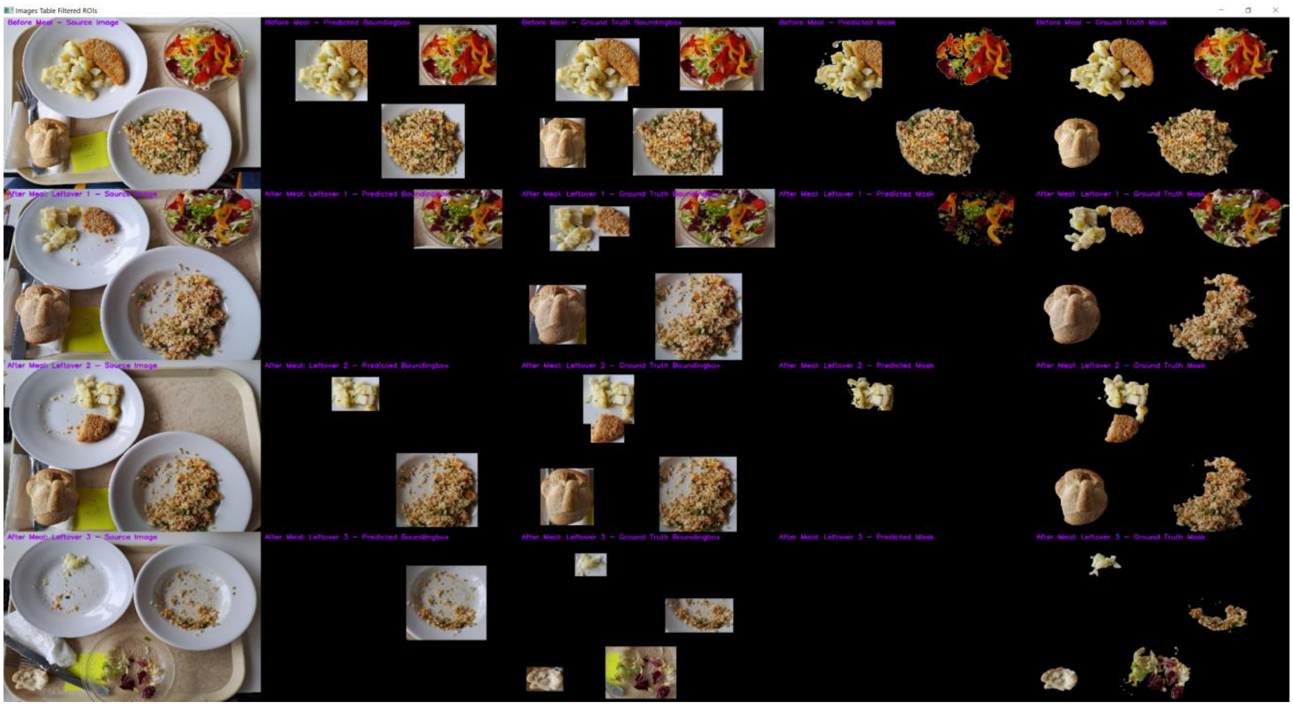
Tray 3


Images Table Masks
Before Meal – Source Image
Before Meal – Predicted Boundingbox
Before Meal – Ground Truth Boundingbox
Before Meal – Predicted Mask
Before Meal – Ground Truth Mask
After Meal: Leftover 1 – Source Image
After Meal: Leftover 1 – Predicted Boundingbox
After Meal: Leftover 1 – Ground Truth Boundingbox
After Meal: Leftover 1 – Predicted Mask
After Meal: Leftover 1 – Ground Truth Mask
After Meal: Leftover 2 – Source Image
After Meal: Leftover 2 – Predicted Boundingbox
After Meal: Leftover 2 – Ground Truth Boundingbox
After Meal: Leftover 2 – Predicted Mask
After Meal: Leftover 2 – Ground Truth Mask
After Meal: Leftover 3 – Source Image
After Meal: Leftover 3 – Predicted Boundingbox
After Meal: Leftover 3 – Ground Truth Boundingbox
After Meal: Leftover 3 – Predicted Mask
After Meal: Leftover 3 – Ground Truth Mask

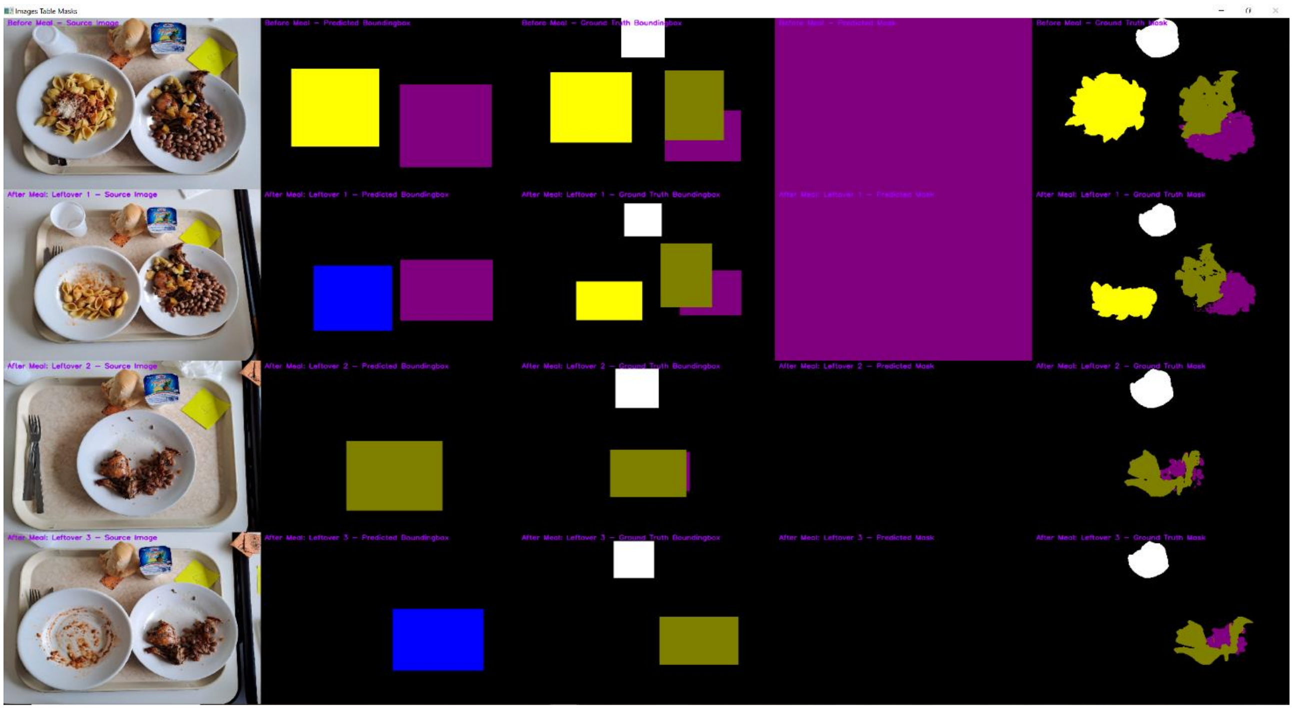

Images Table Filtered ROIs
Before Meal – Source Image
Before Meal – Predicted Boundingbox
Before Meal – Ground Truth Boundingbox
Before Meal – Predicted Mask
Before Meal – Ground Truth Mask
After Meal: Leftover 1 – Source Image
After Meal: Leftover 1 – Predicted Boundingbox
After Meal: Leftover 1 – Ground Truth Boundingbox
After Meal: Leftover 1 – Predicted Mask
After Meal: Leftover 1 – Ground Truth Mask
After Meal: Leftover 2 – Source Image
After Meal: Leftover 2 – Predicted Boundingbox
After Meal: Leftover 2 – Ground Truth Boundingbox
After Meal: Leftover 2 – Predicted Mask
After Meal: Leftover 2 – Ground Truth Mask
After Meal: Leftover 3 – Source Image
After Meal: Leftover 3 – Predicted Boundingbox
After Meal: Leftover 3 – Ground Truth Boundingbox
After Meal: Leftover 3 – Predicted Mask
After Meal: Leftover 3 – Ground Truth Mask

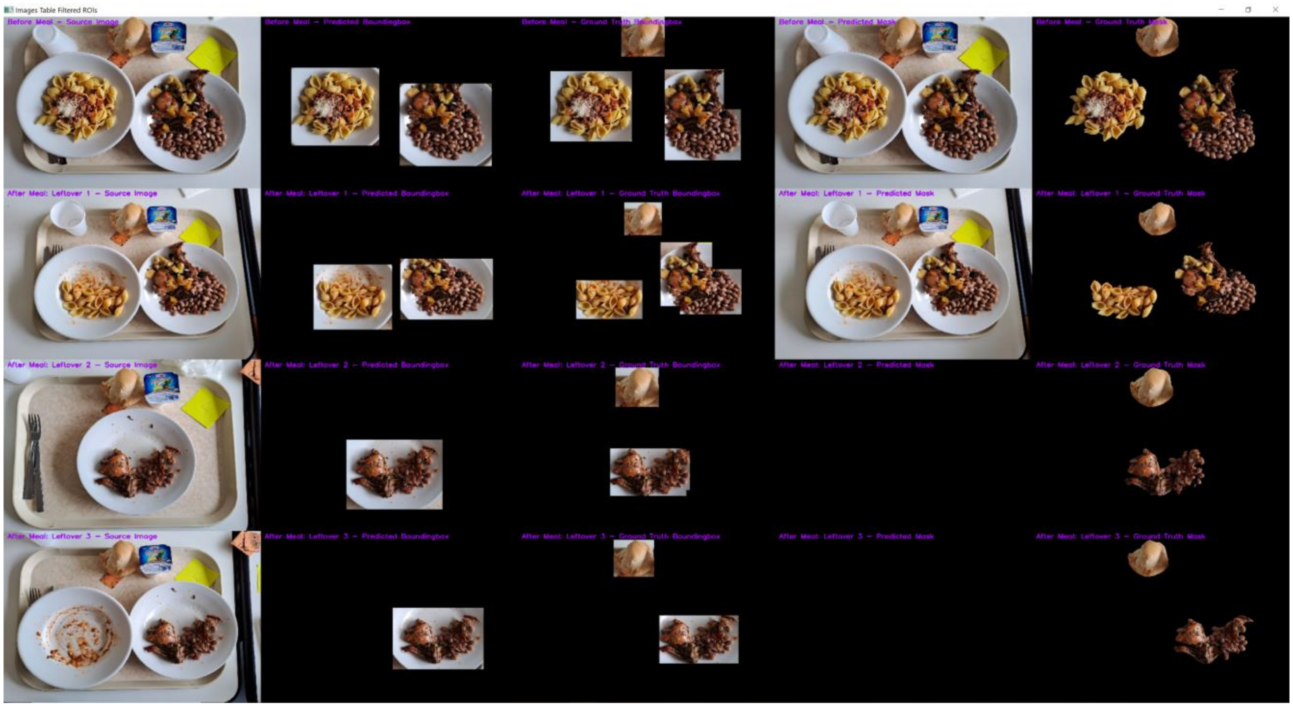
Tray 4


Images Table Masks
Before Meal - Source Image
Before Meal - Predicted Boundingbox
Before Meal - Ground Truth Boundingbox
Before Meal - Predicted Mask
Before Meal - Ground Truth Mask
After Meal: Leftover 1 - Source Image
After Meal: Leftover 1 - Predicted Boundingbox
After Meal: Leftover 1 - Ground Truth Boundingbox
After Meal: Leftover 1 - Predicted Mask
After Meal: Leftover 1 - Ground Truth Mask
After Meal: Leftover 2 - Source Image
After Meal: Leftover 2 - Predicted Boundingbox
After Meal: Leftover 2 - Ground Truth Boundingbox
After Meal: Leftover 2 - Predicted Mask
After Meal: Leftover 2 - Ground Truth Mask
After Meal: Leftover 3 - Source Image
After Meal: Leftover 3 - Predicted Boundingbox
After Meal: Leftover 3 - Ground Truth Boundingbox
After Meal: Leftover 3 - Predicted Mask
After Meal: Leftover 3 - Ground Truth Mask

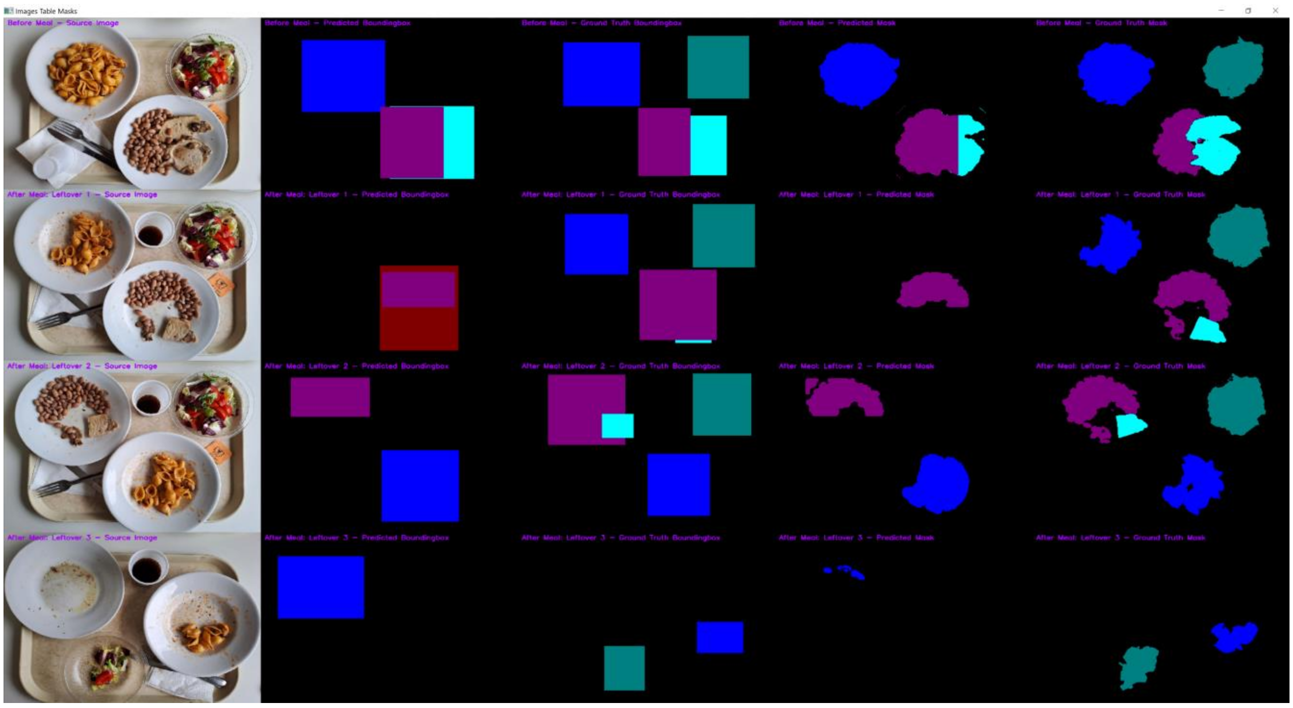

Images Table Filtered ROIs
Before Meal - Source Image
Before Meal - Predicted Boundingbox
Before Meal - Ground Truth Boundingbox
Before Meal - Predicted Mask
Before Meal - Ground Truth Mask
After Meal: Leftover 1 - Source Image
After Meal: Leftover 1 - Predicted Boundingbox
After Meal: Leftover 1 - Ground Truth Boundingbox
After Meal: Leftover 1 - Predicted Mask
After Meal: Leftover 1 - Ground Truth Mask
After Meal: Leftover 2 - Source Image
After Meal: Leftover 2 - Predicted Boundingbox
After Meal: Leftover 2 - Ground Truth Boundingbox
After Meal: Leftover 2 - Predicted Mask
After Meal: Leftover 2 - Ground Truth Mask
After Meal: Leftover 3 - Source Image
After Meal: Leftover 3 - Predicted Boundingbox
After Meal: Leftover 3 - Ground Truth Boundingbox
After Meal: Leftover 3 - Predicted Mask
After Meal: Leftover 3 - Ground Truth Mask

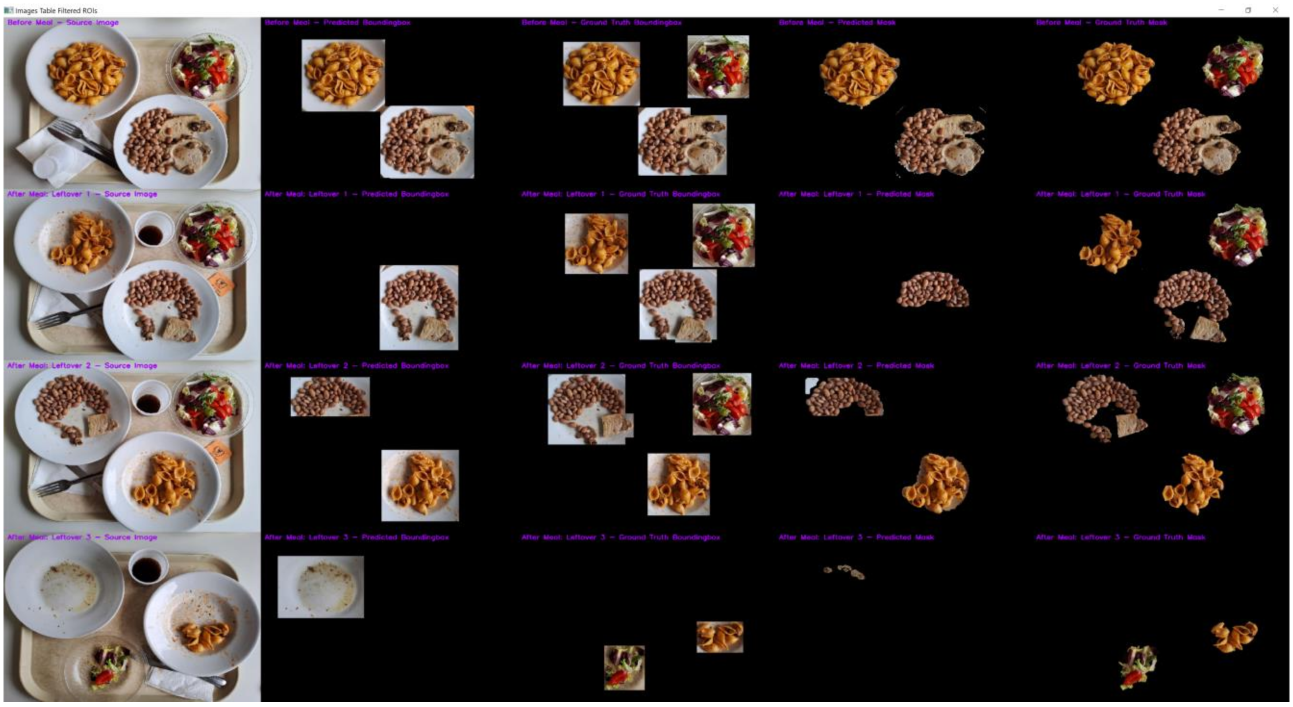
Tray 5


Images Table Masks
Before Meal – Source Image
Before Meal – Predicted Boundingbox
Before Meal – Ground Truth Boundingbox
Before Meal – Predicted Mask
Before Meal – Ground Truth Mask
After Meal: Leftover 1 – Source Image
After Meal: Leftover 1 – Predicted Boundingbox
After Meal: Leftover 1 – Ground Truth Boundingbox
After Meal: Leftover 1 – Predicted Mask
After Meal: Leftover 1 – Ground Truth Mask
After Meal: Leftover 2 – Source Image
After Meal: Leftover 2 – Predicted Boundingbox
After Meal: Leftover 2 – Ground Truth Boundingbox
After Meal: Leftover 2 – Predicted Mask
After Meal: Leftover 2 – Ground Truth Mask
After Meal: Leftover 3 – Source Image
After Meal: Leftover 3 – Predicted Boundingbox
After Meal: Leftover 3 – Ground Truth Boundingbox
After Meal: Leftover 3 – Predicted Mask
After Meal: Leftover 3 – Ground Truth Mask

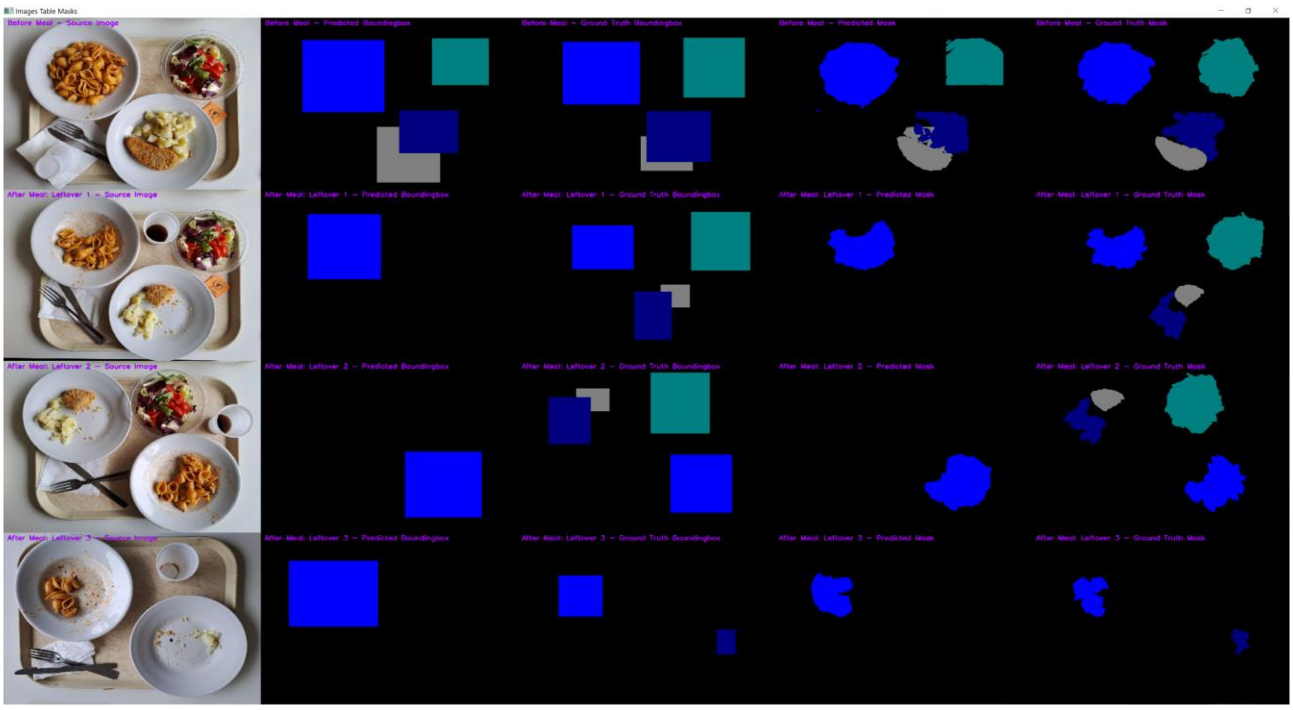

Images Table Filtered ROIs
Before Meal – Source Image
Before Meal – Predicted Boundingbox
Before Meal – Ground Truth Boundingbox
Before Meal – Predicted Mask
Before Meal – Ground Truth Mask
After Meal: Leftover 1 – Source Image
After Meal: Leftover 1 – Predicted Boundingbox
After Meal: Leftover 1 – Ground Truth Boundingbox
After Meal: Leftover 1 – Predicted Mask
After Meal: Leftover 1 – Ground Truth Mask
After Meal: Leftover 2 – Source Image
After Meal: Leftover 2 – Predicted Boundingbox
After Meal: Leftover 2 – Ground Truth Boundingbox
After Meal: Leftover 2 – Predicted Mask
After Meal: Leftover 2 – Ground Truth Mask
After Meal: Leftover 3 – Source Image
After Meal: Leftover 3 – Predicted Boundingbox
After Meal: Leftover 3 – Ground Truth Boundingbox
After Meal: Leftover 3 – Predicted Mask
After Meal: Leftover 3 – Ground Truth Mask

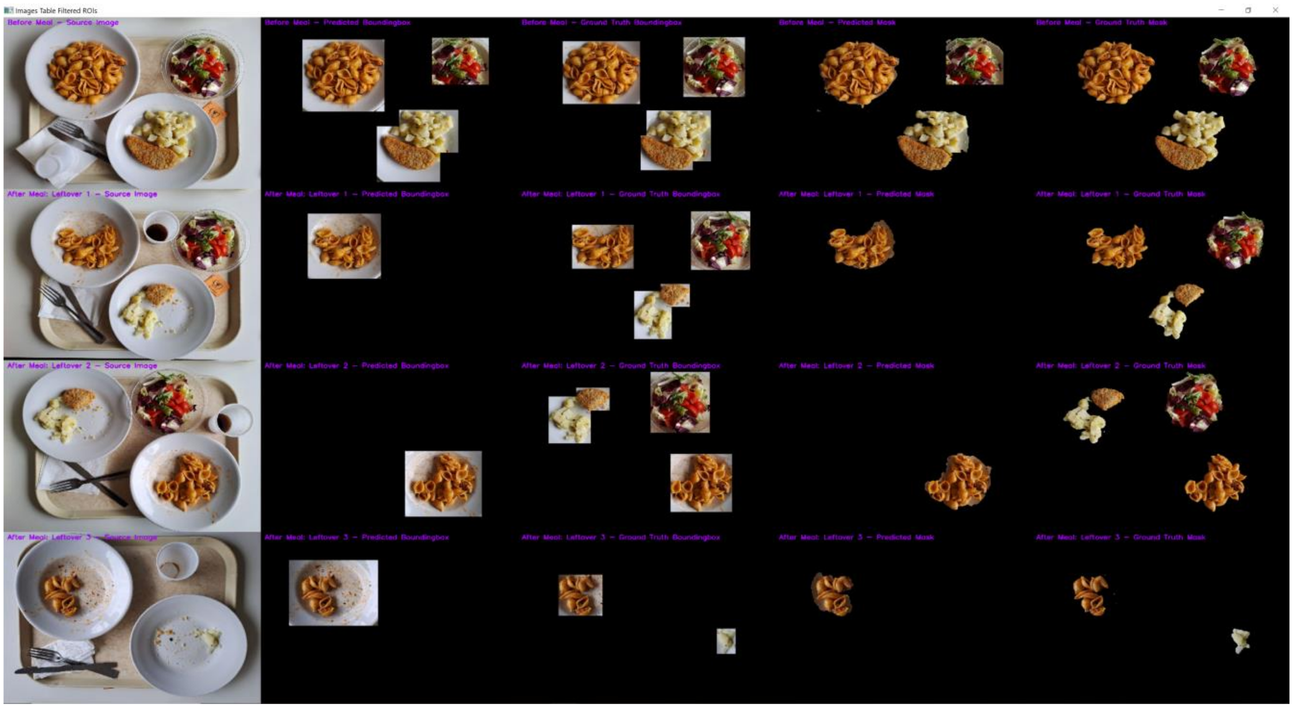
Tray 6


Images Table Masks
Before Meal – Source Image
Before Meal – Predicted Boundingbox
Before Meal – Ground Truth Boundingbox
Before Meal – Predicted Mask
Before Meal – Ground Truth Mask
After Meal: Leftover 1 – Source Image
After Meal: Leftover 1 – Predicted Boundingbox
After Meal: Leftover 1 – Ground Truth Boundingbox
After Meal: Leftover 1 – Predicted Mask
After Meal: Leftover 1 – Ground Truth Mask
After Meal: Leftover 2 – Source Image
After Meal: Leftover 2 – Predicted Boundingbox
After Meal: Leftover 2 – Ground Truth Boundingbox
After Meal: Leftover 2 – Predicted Mask
After Meal: Leftover 2 – Ground Truth Mask
After Meal: Leftover 3 – Source Image
After Meal: Leftover 3 – Predicted Boundingbox
After Meal: Leftover 3 – Ground Truth Boundingbox
After Meal: Leftover 3 – Predicted Mask
After Meal: Leftover 3 – Ground Truth Mask

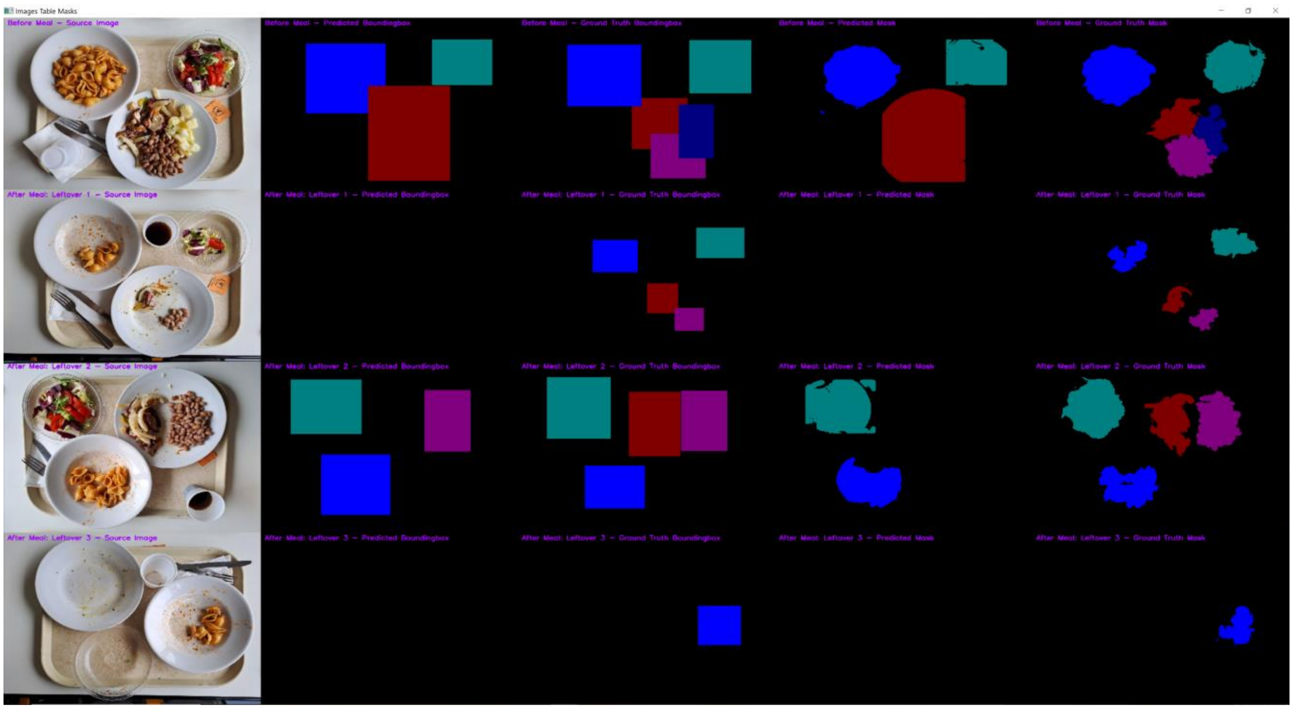

Images Table Filtered ROIs
Before Meal – Source Image
Before Meal – Predicted Boundingbox
Before Meal – Ground Truth Boundingbox
Before Meal – Predicted Mask
Before Meal – Ground Truth Mask
After Meal: Leftover 1 – Source Image
After Meal: Leftover 1 – Predicted Boundingbox
After Meal: Leftover 1 – Ground Truth Boundingbox
After Meal: Leftover 1 – Predicted Mask
After Meal: Leftover 1 – Ground Truth Mask
After Meal: Leftover 2 – Source Image
After Meal: Leftover 2 – Predicted Boundingbox
After Meal: Leftover 2 – Ground Truth Boundingbox
After Meal: Leftover 2 – Predicted Mask
After Meal: Leftover 2 – Ground Truth Mask
After Meal: Leftover 3 – Source Image
After Meal: Leftover 3 – Predicted Boundingbox
After Meal: Leftover 3 – Ground Truth Boundingbox
After Meal: Leftover 3 – Predicted Mask
After Meal: Leftover 3 – Ground Truth Mask

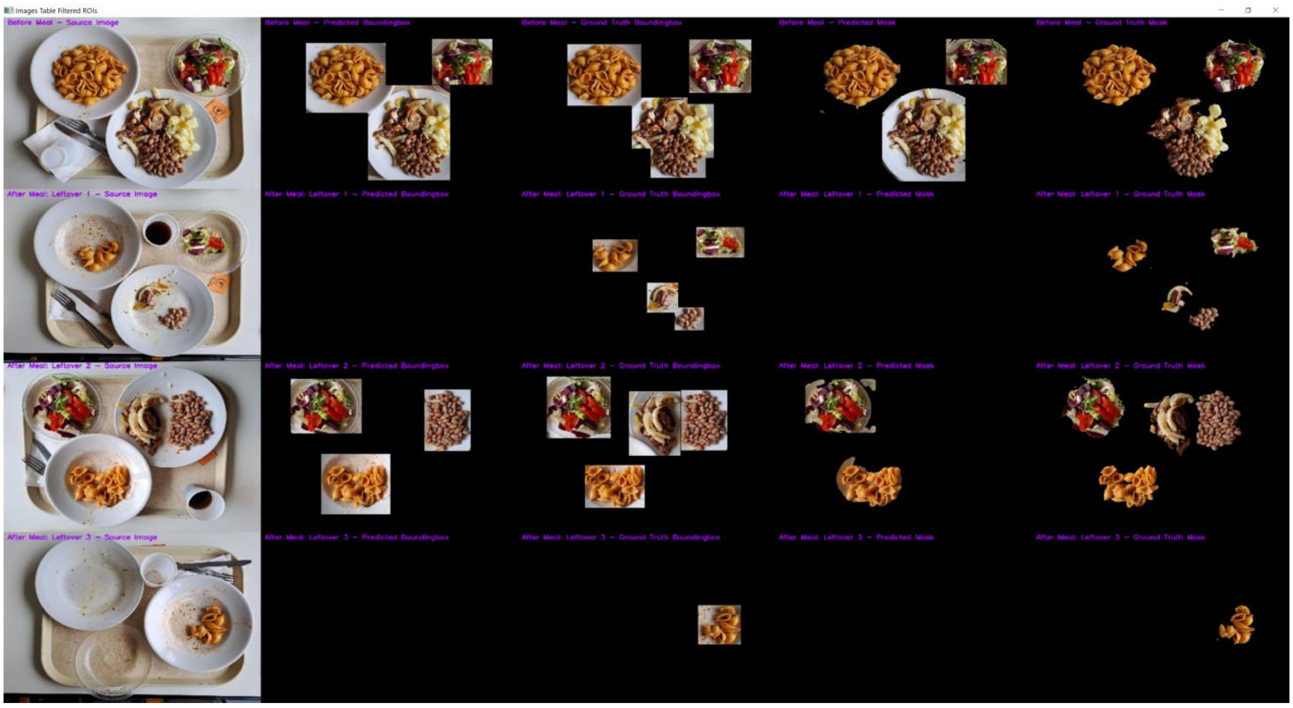
Tray 7


Images Table Masks
Before Meal – Source Image
Before Meal – Predicted Boundingbox
Before Meal – Ground Truth Boundingbox
Before Meal – Predicted Mask
Before Meal – Ground Truth Mask
After Meal: Leftover 1 – Source Image
After Meal: Leftover 1 – Predicted Boundingbox
After Meal: Leftover 1 – Ground Truth Boundingbox
After Meal: Leftover 1 – Predicted Mask
After Meal: Leftover 1 – Ground Truth Mask
After Meal: Leftover 2 – Source Image
After Meal: Leftover 2 – Predicted Boundingbox
After Meal: Leftover 2 – Ground Truth Boundingbox
After Meal: Leftover 2 – Predicted Mask
After Meal: Leftover 2 – Ground Truth Mask
After Meal: Leftover 3 – Source Image
After Meal: Leftover 3 – Predicted Boundingbox
After Meal: Leftover 3 – Ground Truth Boundingbox
After Meal: Leftover 3 – Predicted Mask
After Meal: Leftover 3 – Ground Truth Mask


Images Table Filtered ROIs
Before Meal – Source Image
Before Meal – Predicted Boundingbox
Before Meal – Ground Truth Boundingbox
Before Meal – Predicted Mask
Before Meal – Ground Truth Mask
After Meal: Leftover 1 – Source Image
After Meal: Leftover 1 – Predicted Boundingbox
After Meal: Leftover 1 – Ground Truth Boundingbox
After Meal: Leftover 1 – Predicted Mask
After Meal: Leftover 1 – Ground Truth Mask
After Meal: Leftover 2 – Source Image
After Meal: Leftover 2 – Predicted Boundingbox
After Meal: Leftover 2 – Ground Truth Boundingbox
After Meal: Leftover 2 – Predicted Mask
After Meal: Leftover 2 – Ground Truth Mask
After Meal: Leftover 3 – Source Image
After Meal: Leftover 3 – Predicted Boundingbox
After Meal: Leftover 3 – Ground Truth Boundingbox
After Meal: Leftover 3 – Predicted Mask
After Meal: Leftover 3 – Ground Truth Mask

Tray 8


Images Table Masks
Before Meal – Source Image
Before Meal – Predicted Boundingbox
Before Meal – Ground Truth Boundingbox
Before Meal – Predicted Mask
Before Meal – Ground Truth Mask
After Meal: Leftover 1 – Source Image
After Meal: Leftover 1 – Predicted Boundingbox
After Meal: Leftover 1 – Ground Truth Boundingbox
After Meal: Leftover 1 – Predicted Mask
After Meal: Leftover 1 – Ground Truth Mask
After Meal: Leftover 2 – Source Image
After Meal: Leftover 2 – Predicted Boundingbox
After Meal: Leftover 2 – Ground Truth Boundingbox
After Meal: Leftover 2 – Predicted Mask
After Meal: Leftover 2 – Ground Truth Mask
After Meal: Leftover 3 – Source Image
After Meal: Leftover 3 – Predicted Boundingbox
After Meal: Leftover 3 – Ground Truth Boundingbox
After Meal: Leftover 3 – Predicted Mask
After Meal: Leftover 3 – Ground Truth Mask


Images Table Filtered ROIs
Before Meal – Source Image
Before Meal – Predicted Boundingbox
Before Meal – Ground Truth Boundingbox
Before Meal – Predicted Mask
Before Meal – Ground Truth Mask
After Meal: Leftover 1 – Source Image
After Meal: Leftover 1 – Predicted Boundingbox
After Meal: Leftover 1 – Ground Truth Boundingbox
After Meal: Leftover 1 – Predicted Mask
After Meal: Leftover 1 – Ground Truth Mask
After Meal: Leftover 2 – Source Image
After Meal: Leftover 2 – Predicted Boundingbox
After Meal: Leftover 2 – Ground Truth Boundingbox
After Meal: Leftover 2 – Predicted Mask
After Meal: Leftover 2 – Ground Truth Mask
After Meal: Leftover 3 – Source Image
After Meal: Leftover 3 – Predicted Boundingbox
After Meal: Leftover 3 – Ground Truth Boundingbox
After Meal: Leftover 3 – Predicted Mask
After Meal: Leftover 3 – Ground Truth Mask

Leftovers

Tray 1						
Food	After meal leftover 1		After meal leftover 2		After meal leftover 3	
	Leftover estimated	Leftover expected	Leftover estimated	Leftover expected	Leftover estimated	Leftover expected
pasta with pesto	92.15%	74.18%	88.07%	48.70%	63.96%	0.00%
grilled pork cullet	0.00%	81.70%	0.00%	43.10%	0.00%	0.00%
beans	111.05%	128.52%	50.82%	84.50%	48.56%	0.00%
bread	0.00%	86.97%	0.00%	81.48%	0.00%	56.08%

Tray 2						
Food	After meal leftover 1		After meal leftover 2		After meal leftover 3	
	Leftover estimated	Leftover expected	Leftover estimated	Leftover expected	Leftover estimated	Leftover expected
pasta with tomato sauce	0.00%	100.45%	0.00%	34.21%	0.00%	0.00%
fish cutlet	0.00%	106.75%	0.00%	74.72%	0.00%	35.77%
basil potatoes	0.00%	107.65%	53.33%	73.26%	0.00%	24.71%
salad	94.88%	123.64%	0.00%	70.76%	0.00%	0.00%

Tray 3						
Food	After meal leftover 1		After meal leftover 2		After meal leftover 3	
	Leftover estimated	Leftover expected	Leftover estimated	Leftover expected	Leftover estimated	Leftover expected
pasta with tomato sauce	0.00%	34.44%	0.00%	31.90%	0.00%	0.00%
rabbit	0.00%	77.05%	0.00%	84.37%	0.00%	89.36%
salad	0.00%	74.85%	0.00%	79.59%	0.00%	0.00%

Tray 4						
Food	After meal leftover 1		After meal leftover 2		After meal leftover 3	
	Leftover estimated	Leftover expected	Leftover estimated	Leftover expected	Leftover estimated	Leftover expected
pilaw rice with peppers and peas	0.00%	108.73%	0.00%	86.86%	0.00%	20.38%
fish cutlet	0.00%	59.88%	0.00%	58.30%	0.00%	0.00%
basil potatoes	0.00%	59.00%	36.14%	55.61%	0.00%	15.69%

salad	75.36%	116.59%	0.00%	0.00%	0.00%	48.03%
bread	0.00%	145.64%	0.00%	128.94%	0.00%	36.88%

Tray 5						
Food	**After meal leftover 1**		**After meal leftover 2**		**After meal leftover 3**	
	Leftover estimated	**Leftover expected**	**Leftover estimated**	**Leftover expected**	**Leftover estimated**	**Leftover expected**
pasta with meat sauce	0.00%	47.78%	0.00%	0.00%	0.00%	0.00%
rabbit	0.00%	69.83%	0.00%	67.90%	0.00%	62.04%
beans	100.00%	66.28%	0.00%	25.44%	0.00%	26.65%
bread	0.00%	70.98%	0.00%	96.32%	0.00%	83.70%

Tray 6						
Food	**After meal leftover 1**		**After meal leftover 2**		**After meal leftover 3**	
	Leftover estimated	**Leftover expected**	**Leftover estimated**	**Leftover expected**	**Leftover estimated**	**Leftover expected**
pasta with clams and mussels	0.00%	60.64%	72.35%	61.74%	4.72%	26.70%
grilled pork cutlet	0.00%	27.85%	0.00%	23.53%	0.00%	0.00%
beans	54.72%	123.78%	65.39%	124.13%	0.00%	0.00%
salad	0.00%	108.45%	0.00%	100.90%	0.00%	44.58%

Tray 7						
Food	**After meal leftover 1**		**After meal leftover 2**		**After meal leftover 3**	
	Leftover estimated	**Leftover expected**	**Leftover estimated**	**Leftover expected**	**Leftover estimated**	**Leftover expected**
pasta with clams and mussels	57.47%	51.25%	65.04%	56.19%	35.46%	27.00%
fish cutlet	0.00%	41.61%	0.00%	48.83%	0.00%	0.00%
basil potatoes	0.00%	49.44%	0.00%	53.76%	0.00%	16.33%
salad	0.00%	91.21%	0.00%	97.88%	0.00%	0.00%

Tray 8						
Food	**After meal leftover 1**		**After meal leftover 2**		**After meal leftover 3**	
	Leftover estimated	**Leftover expected**	**Leftover estimated**	**Leftover expected**	**Leftover estimated**	**Leftover expected**
pasta with clams and mussels	0.00%	22.72%	63.12%	55.43%	0.00%	26.08%

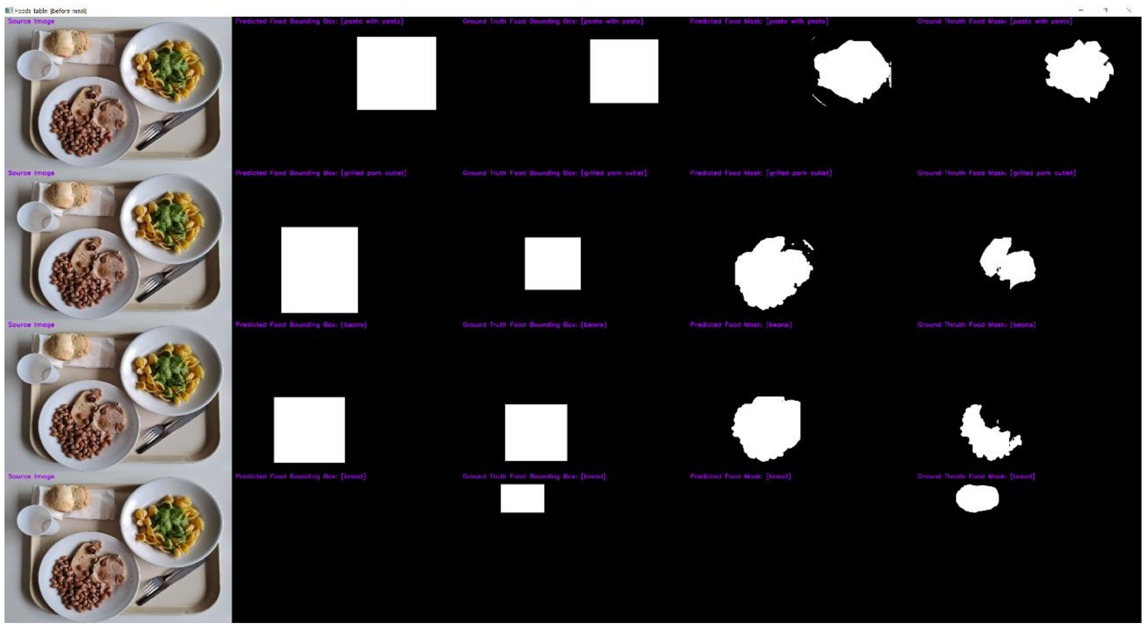
seafood salad	0.00%	31.71%	0.00%	131.14%	0.00%	0.00%
beans	0.00%	25.91%	0.00%	127.52%	0.00%	0.00%
salad	0.00%	41.72%	122.80%	112.04%	0.00%	0.00%

Performance

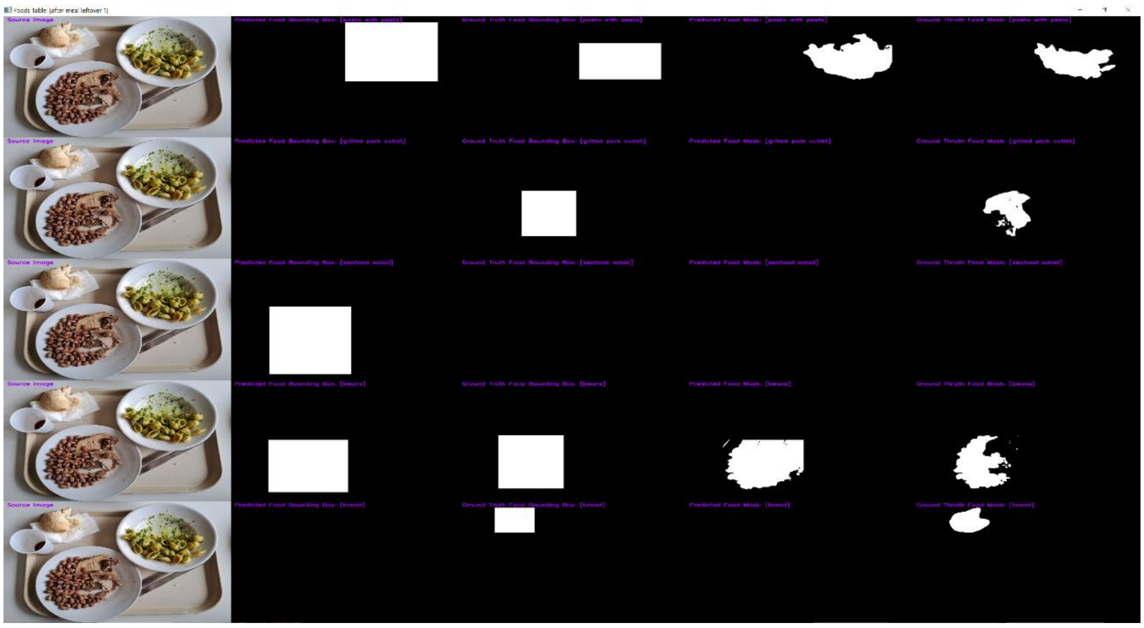
Tray number	mAP score			mIoU score		
	Before meal	After meal 1	After meal 2	Before meal	After meal 1	After meal 2
Tray 1	0.50	0.50	0.25	0.45	0.30	0.31
Tray 2	0.75	0.25	0.25	0.53	0.21	0.15
Tray 3	0.33	0.00	0.00	0.21	0.00	0.00
Tray 4	0.60	0.20	0.25	0.47	0.11	0.17
Tray 5	0.50	0.00	0.33	0.03	0.01	0.00
Tray 6	0.75	0.00	0.50	0.50	0.18	0.37
Tray 7	1.00	0.25	0.25	0.74	0.20	0.20
Tray 8	0.40	0.00	0.75	0.36	0.00	0.37

Example of final plots [for tray1]

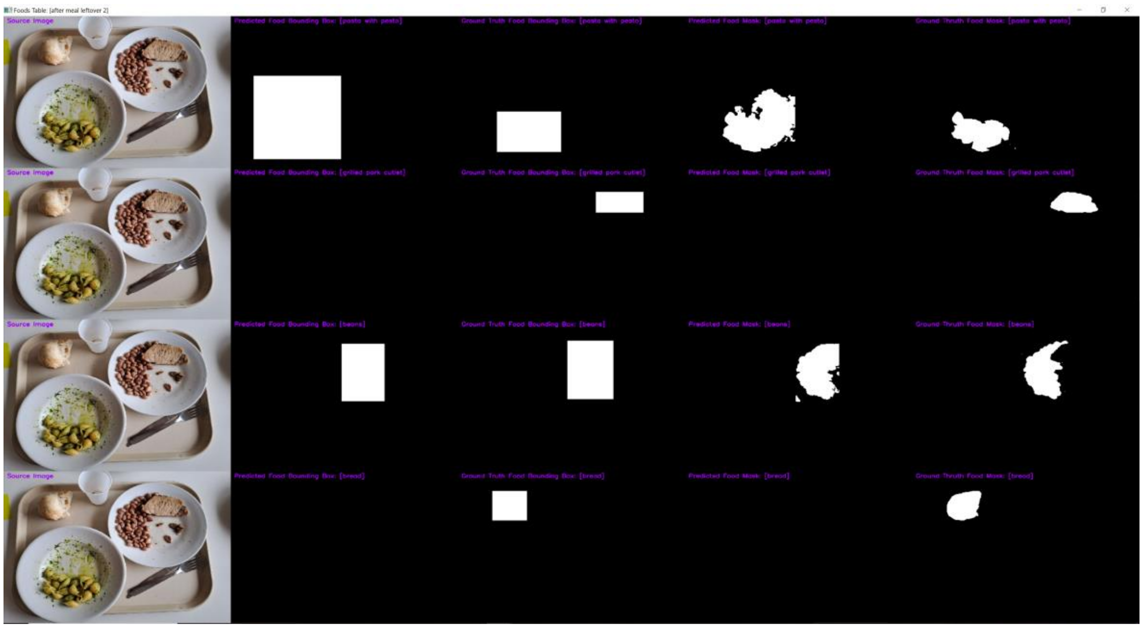
foods_table before meal


Foods Table [before meal]
Source Image
Predicted Food Bounding Box: [pasta with pesto]
Ground Truth Food Bounding Box: [pasta with pesto]
Predicted Food Mask: [pasta with pesto]
Ground Thruth Food Mask: [pasta with pesto]
Source Image
Predicted Food Bounding Box: [grilled pork cutlet]
Ground Truth Food Bounding Box: [grilled pork cutlet]
Predicted Food Mask: [grilled pork cutlet]
Ground Thruth Food Mask: [grilled pork cutlet]
Source Image
Predicted Food Bounding Box: [beans]
Ground Truth Food Bounding Box: [beans]
Predicted Food Mask: [beans]
Ground Thruth Food Mask: [beans]
Source Image
Predicted Food Bounding Box: [bread]
Ground Truth Food Bounding Box: [bread]
Predicted Food Mask: [bread]
Ground Thruth Food Mask: [bread]

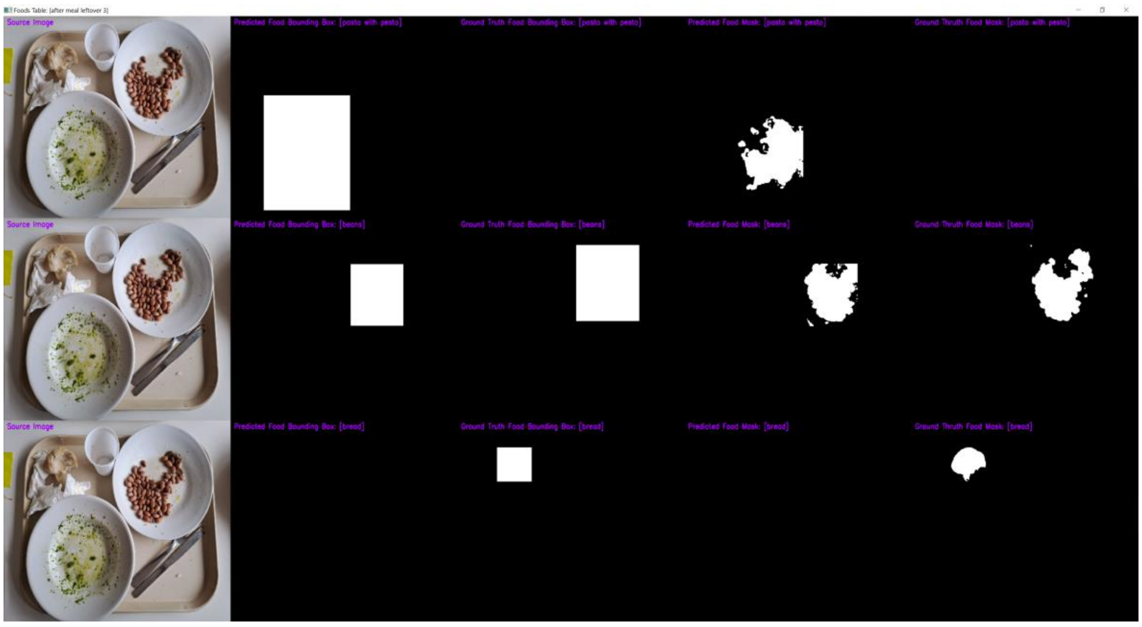
foods_table after meal leftover 1


Foods Table: [after meal leftover 1]
Source Image
Predicted Food Bounding Box: [pasta with pesto]
Ground Truth Food Bounding Box: [pasta with pesto]
Predicted Food Mask: [pasta with pesto]
Ground Thruth Food Mask: [pasta with pesto]
Predicted Food Bounding Box: [grilled pork cutlet]
Ground Truth Food Bounding Box: [grilled pork cutlet]
Predicted Food Mask: [grilled pork cutlet]
Ground Thruth Food Mask: [grilled pork cutlet]
Predicted Food Bounding Box: [seafood salad]
Ground Truth Food Bounding Box: [seafood salad]
Predicted Food Mask: [seafood salad]
Ground Thruth Food Mask: [seafood salad]
Predicted Food Bounding Box: [beans]
Ground Truth Food Bounding Box: [beans]
Predicted Food Mask: [beans]
Ground Thruth Food Mask: [beans]
Predicted Food Bounding Box: [bread]
Ground Truth Food Bounding Box: [bread]
Predicted Food Mask: [bread]
Ground Thruth Food Mask: [bread]

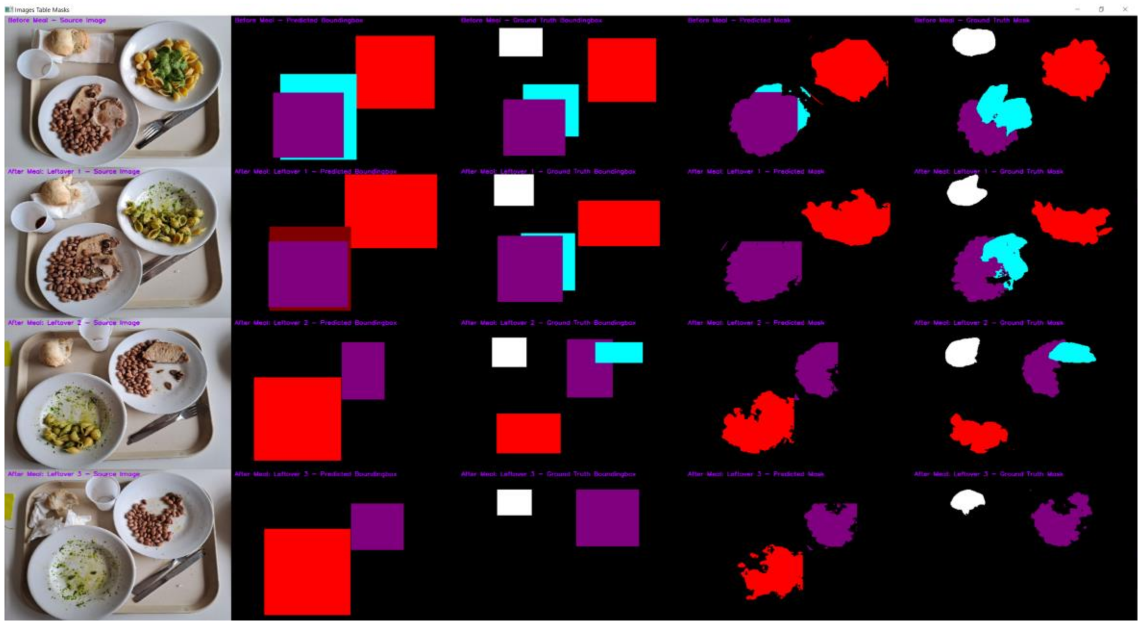
foods_table after meal leftover 2

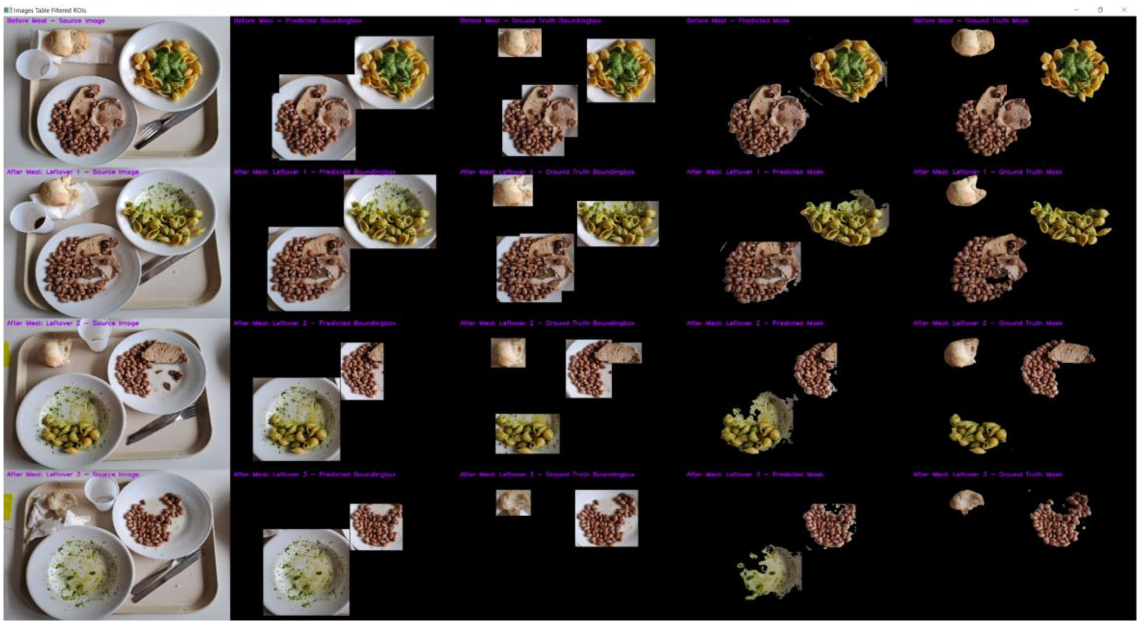

Foods Table: [after meal leftover 2]
Source Image
Predicted Food Bounding Box: [pasta with pesto]
Ground Truth Food Bounding Box: [pasta with pesto]
Predicted Food Mask: [pasta with pesto]
Ground Thruth Food Mask: [pasta with pesto]
Predicted Food Bounding Box: [grilled pork cutlet]
Ground Truth Food Bounding Box: [grilled pork cutlet]
Predicted Food Mask: [grilled pork cutlet]
Ground Thruth Food Mask: [grilled pork cutlet]
Predicted Food Bounding Box: [beans]
Ground Truth Food Bounding Box: [beans]
Predicted Food Mask: [beans]
Ground Thruth Food Mask: [beans]
Predicted Food Bounding Box: [bread]
Ground Truth Food Bounding Box: [bread]
Predicted Food Mask: [bread]
Ground Thruth Food Mask: [bread]

foods_table after meal leftover 3


Foods Table: [after meal leftover 3]
Source Image
Predicted Food Bounding Box: [pasta with pesto]
Ground Truth Food Bounding Box: [pasta with pesto]
Predicted Food Mask: [pasta with pesto]
Ground Thruth Food Mask: [pasta with pesto]
Source Image
Predicted Food Bounding Box: [beans]
Ground Truth Food Bounding Box: [beans]
Predicted Food Mask: [beans]
Ground Thruth Food Mask: [beans]
Source Image
Predicted Food Bounding Box: [bread]
Ground Truth Food Bounding Box: [bread]
Predicted Food Mask: [bread]
Ground Thruth Food Mask: [bread]

images_table


Images Table Masks
Before Meal – Source Image
Before Meal – Predicted Boundingbox
Before Meal – Ground Truth Boundingbox
Before Meal – Predicted Mask
Before Meal – Ground Truth Mask
After Meal: Leftover 1 – Source Image
After Meal: Leftover 1 – Predicted Boundingbox
After Meal: Leftover 1 – Ground Truth Boundingbox
After Meal: Leftover 1 – Predicted Mask
After Meal: Leftover 1 – Ground Truth Mask
After Meal: Leftover 2 – Source Image
After Meal: Leftover 2 – Predicted Boundingbox
After Meal: Leftover 2 – Ground Truth Boundingbox
After Meal: Leftover 2 – Predicted Mask
After Meal: Leftover 2 – Ground Truth Mask
After Meal: Leftover 3 – Source Image
After Meal: Leftover 3 – Predicted Boundingbox
After Meal: Leftover 3 – Ground Truth Boundingbox
After Meal: Leftover 3 – Predicted Mask
After Meal: Leftover 3 – Ground Truth Mask


Images Table Filtered ROIs
Before Meal – Source Image
Before Meal – Predicted Boundingbox
Before Meal – Ground Truth Boundingbox
Before Meal – Predicted Mask
Before Meal – Ground Truth Mask
After Meal: Leftover 1 – Source Image
After Meal: Leftover 1 – Predicted Boundingbox
After Meal: Leftover 1 – Ground Truth Boundingbox
After Meal: Leftover 1 – Predicted Mask
After Meal: Leftover 1 – Ground Truth Mask
After Meal: Leftover 2 – Source Image
After Meal: Leftover 2 – Predicted Boundingbox
After Meal: Leftover 2 – Ground Truth Boundingbox
After Meal: Leftover 2 – Predicted Mask
After Meal: Leftover 2 – Ground Truth Mask
After Meal: Leftover 3 – Source Image
After Meal: Leftover 3 – Predicted Boundingbox
After Meal: Leftover 3 – Ground Truth Boundingbox
After Meal: Leftover 3 – Predicted Mask
After Meal: Leftover 3 – Ground Truth Mask

Conclusions and considerations

In this final report we have described the approach proposed by our group for solving the problem of food recognition and leftover estimation. By the results collected, we can conclude that our approach does not currently work really well, however the performance is more than acceptable, since it cannot obviously be compared to the performance of a full-deep learning approach.

However, the performance of our approach could potentially significantly be improved as we are going to describe in the next.

The detection task is not perfect, especially for what concerns the classification of the leftover foods detected in the after-meal images. Moreover, a lot of food samples are not even detected. This problem is due to the dataset we are using, that was populated by only images of foods before meal, making the leftovers classification and detection more difficult for our classifiers. This problem cannot currently be solved since no datasets of (Italian) leftover foods is present in the online literature. Hence, the only way to enhance the performance of our model is to build a second dataset from scratch, containing foods images after meal (i.e., of leftovers), and then upgrading with them the current dataset.

As we have previously described all the miss-matching in the classification among foods detected before and after meal lead to bad performance, since they are identified as different foods w.r.t. the ground truth ones, thus they are identified as non-detected foods and consequentially they cannot be segmented.

In few cases our segmentation approach is not really precise, taking into consideration part of the dishes in the border of the food (such as the case of the salad), but this fact is not so worrying, since it is hard to set the right parameters performing well for every sample, as discussed previously. Overall, the segmentation performance is good, and the result comes from the fact that we adopted an ensemble of methods. If desired, this performance can further be improved by adding other weak segmentation algorithms to the ensemble.

Therefore, the low performance results derive from the fact that we keep in the mean of AP and IoU the 0 scores of foods not detected by the classification and detection process. So, there is a right but consistent decrease of mIoU and mAP performances, due to the weak detection.

In conclusion, despite the difficulty of the work faced, we can consider ourselves more than satisfied, since the structure of the program is extremely modular and scalable for future improvements, and even if the performances currently are not brilliant, they can be easily significantly improved by collecting a better dataset. Hence, with a well-made and specific dataset and classification model we would have achieved higher performances.

Members contribution to the project

- **Osti Simone:**
  - **Estimated hours:** 120+ hours (hours of training tens of model non included)
  - **Contribute:** build dataset, write code for detection and classification, train and test tens of models (all in one or 3 models divided by course or 2 models), tune threshold for discard false positive.
- **Christian Francesco Russo:**
  - **Estimated hours:** 160+ hours
  - **Contribute:**
    - Defined, implemented and wrote report about the overall structure of the project (for the C++ code).
    - Projected, implemented and wrote report about the classes: *Main.cpp*, *FoodRecognizer_and_LeftoverEstimation.h/.cpp* (except performance evaluation), *ROIsDetection.h/.cpp*, *FoodDetection.h/.cpp* (only structure and integration with the rest of the code, not the core of the program), FoodSegmentation.h/.cpp, PerformanceEvaluation.h/.cpp, *DatasetCategories.h/.cpp*, *Util.h/.cpp, CMakeList.txt*
    - Invented and implemented the method for finding the tray, the adaptive method for finding the dishes, the 2 novel methods and the ensemble for the segmentation.
- **Spinato Matteo:**
  - **Estimated hours:** 120+ hours
  - **Contribute:** tested some segmentation methods, build dataset, data augmentation, tune ROI and segmentation parameters, performance evaluations understanding and implementation.