Grab, Pay and Eat: Semantic Food Detection for Smart Restaurants

Eduardo Aguilar, Beatriz Remeseiro, Marc Bolaños, and Petia Radeva, *Fellow, IAPR*).

***Abstract*—The increase in awareness of people towards their nutritional habits has drawn considerable attention to the field of automatic food analysis. Focusing on self-service restaurants environment, automatic food analysis is not only useful for extracting nutritional information from foods selected by customers, it is also of high interest to speed up the service solving the bottleneck produced at the cashiers in times of high demand. In this paper, we address the problem of automatic food tray analysis in canteens and restaurants environment, which consists in predicting multiple foods placed on a tray image. We propose a new approach for food analysis based on convolutional neural networks, we name Semantic Food Detection, which integrates in the same framework food localization, recognition and segmentation. We demonstrate that our method improves the state-of-art food detection by a considerable margin on the public dataset UNIMIB2016, achieving about 90% in terms of F-measure, and thus provides a significant technological advance towards the automatic billing in restaurant environments.**

***Index Terms*—food tray analysis, food recognition, semantic segmentation, convolutional neural networks**

I. Introduction

HAVING a poor routine of physical exercises and poor nutritional habits are two of the main possible causes of people's health-related issues like obesity or diabetes, among others. For these reasons, nowadays people are more concerned about these aspects of their daily life. Therefore, the need for applications that allow to keep track of both physical activities and nutrition habits are rapidly increasing, a field in which the automatic analysis of food images plays an important role. Focusing on self-service restaurants, food recognition algorithms could enable both monitoring of food consumption and the automatic billing of the meal grabbed by the customer. The latter is quite relevant because remove the need for a manual selection of the chosen dishes, allowing to speed-up the service offered by these restaurants.

Fig. 1: Example of images used in traditional approaches to food analysis (left) and food tray analysis (right).

From the computer vision side, several approaches have been proposed to tackle the problem, most of them using Convolutional Neural Networks (CNNs) [1], [2], [3], [4]. Several of the published work consider the development of methods for food recognition, i.e. being able to recognize the dish depicted in a picture in which a single plate is shown. An important consideration to take into account when modeling visual food-related information is its fine-grained nature, meaning that specially in the problem of food analysis the intra and inter-class similarity are hardly making difficult the problem of obtaining robust food recognition methods.

Several works in the literature have proposed methods for food intake self-monitoring [5], [6], in which the user should take pictures of each meal and the system would consequently track any nutritional information associated. Other approaches related to the problem of food intake include food portion estimation by using two images acquired by mobile devices [7]; food ingredients recognition from recipes using CNNs as multi-label predictors [8], [9]; multimodal multitask deep belief networks for learning both visual information and image-ingredient representation [10]; bayesian models for analyzing similarities between cuisines [11]; or cross-modal learning for multi-attribute recognition and recipe retrieval [12].

Instead of applying personalized tracking, there are several contexts where social monitoring or recognition is required. A clear example is food tray detection in public spaces [13], [14], where the sample consists of a tray picture that includes all the food that a user is about to consume (see Fig. 1) and the model is intended to process all pictures from any possible users taking food at the same restaurant. The development of a system able to apply food tray detection in a controlled, but social and public environment could enable several applications. The most straightforward context of applicability would be automatic billing in self-service restaurants, where the system could solve the need for a person selecting what the customer

This work was partially funded by TIN2015-66951-C2-1-R, SGR 1219, and CERCA Programme / Generalitat de Catalunya. E. Aguilar acknowledges the financial support of CONICYT Becas Chile, B. Remeseiro acknowledges the support of the Spanish Ministry of Economy and Competitiveness under *Juan de la Cierva* Program (ref. FJCI-2014-21194), and M. Bolaños acknowledges the support of an FPU fellowship (ref. FPU15/01347). P. Radeva is partially supported by ICREA Academia 2014. The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU.

E. Aguilar is with the Departamento de Ingeniería de Sistemas y Computación, Universidad Católica del Norte, Antofagasta 0610, Chile (email: eaguilar02@ucn.cl).

B. Remeseiro is with the Departament de Matemàtiques i Informàtica, Universitat de Barcelona, Barcelona 08007, and also with the Department of Computer Science, Universidad de Oviedo, Gijón 33203, Spain (email: bremeseiro@uniovi.es).

M. Bolaños and P. Radeva are with the Departament de Matemàtiques i Informàtica, Universitat de Barcelona, Barcelona 08007, and Computer Vision Center, Cerdanyola 08007 (Barcelona), Spain (emails: marc.bolanos@ub.edu, petia.ivanova@ub.edu)

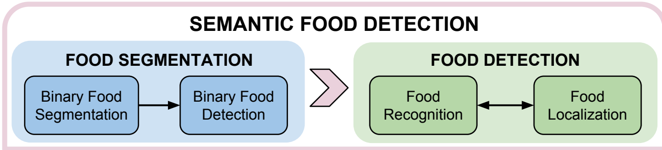


Fig. 2: Main tasks of our Semantic Food Detection framework.

grabbed before paying. A different application could consider the design of smart trays [15], which could provide food recommendations depending on what the customer is selecting. The provided recommendations could be based on calorie counting, healthy food, specific nutritional composition, etc. In addition, if we also consider a system able to log the food consumed by every individual along time, it could provide health-related recommendations in a long-term way.

There are several aspects that make the food tray analysis a challenging problem [14]: 1) multiple foods placed on the same placemat, 2) different foods served in the same dish, 3) visual distortions and illumination changes due to shadows, and 4) objects placed on a tray that do not correspond to any type of food. On the other hand, unlike traditional approaches to food analysis, difficulties due to intra-class variability have less influence on the problem of food tray detection.

In this work, we propose a novel method that unifies the problems of food detection, localization, recognition and segmentation into a new framework that we call Semantic Food Detection. As Fig. 2 shows, we integrate the information extracted by two main approaches: a) food segmentation and b) object detection trained for food detection, by taking advantage of the benefits provided by both algorithms in a CNN framework. The first one allows us to determine where the food is in terms of pixel and bounding boxes. The second one allows us to locate and recognize the foods present in the images. The Semantic Food Detection framework combines the information that both algorithms provide in order to prevent false food detections and thus provide a better performance.

Our main contributions are: 1) a novel framework that integrates the problems of food detection, localization, recognition and segmentation; and 2) a novel approach to address the problem of food tray analysis, that integrates a fully convolutional network for semantic segmentation and a convolutional neural network for object detection through a probabilistic approach and a custom non-maximum suppression. Our method achieves about 90% in terms of F2-score, and it is able to outperform the state-of-art methods by more than 10% and 20% with respect to recall and mean average accuracy.

The remainder of this paper is organized as follows: Section II includes an overview of the related work, Section III presents the proposed Semantic Food Detection approach, Section IV shows the experimental results and discussion, and Section V closes with the conclusions and future research.

II. Related Work

Nowadays, there is a great interest in conducting research for visual food analysis, mainly in its applicability for diet monitoring based on the intrinsic nutritional information contained in food images. In this field, researchers have focused on different several aspects related to automatic food analysis.

The most basic aspect tackled in the literature is the *binary food detection* problem that determines the presence or absence of food in an image. This problem is also called food/non-food classification or food detection [16]. The first approximation was proposed by Kitamura et al. [17], who combine a BoF model and a SVM achieving a high accuracy on a tiny dataset of 600 images. An improvement of about 4% is achieved in terms of overall accuracy using a CNN-based method [16]. From this, numerous researchers have proposed CNN-based models either for feature extraction [2], [3] or for the whole recognition process [1], [4]. The best results obtained on public datasets with more than 15,000 images [1], [2] have been reported in [3] through the combination of CNN GoogLeNet for feature extraction, PCA for dimension reduction and SVM for classification. As for its applicability, this problem has commonly been investigated for indexing WEB images [17] or as a pre-processing method for an automatic food recognition system [1], [4]. It has been also used to detect bounding boxes in an food images [18], and to automate the process of image cleaning required when gathering images of a food dataset [19].

In food analysis, once images containing food are identified, *food recognition* is usually the next step to apply. Again, CNN-based models have been able to progressively improve the results of food recognition models reaching an accuracy of about 90% in datasets with around 100 different food classes [20]. In general, the best proposals are based on the winning models of the ILSVRC challenge [21], and a fine-tuning process is usually applied either making some architectural model changes (e.g. addition or removal of layers) [22], [23] or not [24]. Several datasets have been proposed to tackle this problem: a) datasets including fine-grained classes (e.g. apple pie, pork chop, pizza), like UECFOOD-256 [25] or Food-101 [26]; and b) datasets based on high-level categories (e.g. dessert, meat, soup), like Food-11 [4]. The best result when using fine-grained classes was achieved by the WISeR model [20], which combines the food traits and the vertical structure of some food, extracted by the standard squared convolutional kernel and the proposed slice convolutional kernel, respectively. Regarding the high-level categories, the best results were obtained by [27] through a novel approach that fuses several CNN models, achieving a 10% improvement in terms of accuracy with respect to the baseline method.

Most of the approaches focused on food recognition only exploit the visual content, but they ignore the context. However, geolocation and other information have also been explored in the literature for restaurant-oriented food recognition: on-line restaurant information is used in [28], similarly to [29] in which nutritional information is also retrieved; whilst the menu, the location and user images of dished are used in [30]. On the other hand, Herranz et al. [31] go a step further since their target is not only to improve both classification performance and efficiency, but also to better model contextual data and its relation with the other elements.

To date, most food recognition algorithms and datasets focus on classifying images that include only one dish [20], [23],

[24]. However, in some cases, there may be more than one dish in the image and, in some cases, the dish can contain several kinds of food. *Food localization* and *food segmentation* are two tasks intended to cope with these problems. The former consists in extracting the regions of the images where the food is located. Up to our knowledge, the only available approach that does not require segmenting the food before extracting the bounding boxes is the one proposed by [18]. The task of food segmentation consists in classifying each pixel of the images representing a food. The latest research for food segmentation proposes an automatic weakly supervised methods [32], [33], which are based on Deep Convolutional Neural Networks and Distinct Class-specific Saliency Maps, respectively.

Regarding image segmentation for general purposes, fully convolutional networks (FCNs) [34] are the state-of-art in semantic segmentation. They are composed of convolutional layers only, i.e they do not have any fully-connected layer. They consist of a down-sampling path and an up-sampling path, which allow to take input images of arbitrary size and produce outputs of equivalent size, by means of an efficient inference and learning process. Several FCN models can be found in the literature applied to semantic segmentation. SegNet [35] is a deep FCN that consists of a VGG16-based encoder, a decoder and a final pixel-wise classification layer. DeepLab [36] uses *atrous convolutions* in the up-sampling path, allowing to incorporate larger context with no increase in parameters. RefineNet [37] is a multi-path refinement network that allows to obtain high-resolution predictions by using residual connections. PSPNet [38] is a pixel-level prediction framework that includes a pyramid pooling module to exploit the capability of global context information. Tiramisu [39] is an extension of Densely Connected Convolutional Neural Networks (DenseNets) for semantic segmentation, based on the idea of connecting each layer to every other layer in a feed-forward fashion. Its main benefits include a more accurate and easier training, with much less parameters.

According to the experimentation presented in the respective manuscripts, PSPNet [38] and Tiramisu [39] are the most competitive models. PSPNet is based on Residual Networks (ResNets), whilst Tiramisu is based on DenseNets. DenseNets can be seen as an extension of ResNets, with some characteristics that make them very appropriate for semantic segmentation problems: parameter efficiency, implicit deep supervision, and feature reuse. For all these reasons, Tiramisu will be the model of reference in our research.

In this manuscript, we deal with the identification of different foods placed on a food tray, by integrating the four food analysis problems mentioned above. To the best of our knowledge, only one approach with this purpose has been evidenced in the literature [14]. The authors introduced an additional food dataset composed of images taken in a canteen environment named UNIMIB2016. In addition, they proposed a pipeline for food recognition that performs classification based on the candidate regions obtained by combining two separate images segmentation processes, through saturation and color texture (JSEG). The best result was achieved by combining global and local (patch-based) classification approaches. Regarding the classification, for each region, they are carried out both in a sub-image (global strategy) and in several image patches (local strategy), the feature extraction using a CNN model based in AlexNet, and then the classification by an SVM. Furthermore, in the local strategy, an additional post-processing phase is needed to merge the labels of all image patches of the respective region. Then, the classification obtained by both approaches is combined by exploiting the sum of posterior probabilities to judge the final classification decision. Our approach differs mainly in three aspects: 1) we perform semantic segmentation by learning the best discriminant features between different foods from the dataset instead of using a segmentation approach based on generic image processing methods; 2) we locate and classify simultaneously all the foods placed in the tray by considering the context instead of performing the classification for each region individually, which implies a significant improvement in both result and processing time; and 3) we integrate the outputs of both methods to avoid false detections and thus make better decisions, instead of performing the classification directly based on the segmentation results. Additionally, our method is able to perform the food segmentation and detection processes in parallel, allowing to speed up the processing time.

III. Semantic Food Detection

This work proposes a method for food tray semantic detection that integrates food vs non-food semantic segmentation with food localization and recognition. Fig. 3 depicts the pipeline of our approach, subsequently explained in detail.

A. Food Segmentation

Food segmentation deals here with the problem of separating the food and food-related items, from the tray and other background elements, thus obtaining a binary image. For this purpose, we apply semantic segmentation techniques that work in a supervised learning framework, unlike the most segmentation methods that focus on image properties (e.g. color or texture). Notice that semantic segmentation could be used to directly segment the input image into the different food categories. However, the most recent methods in this field provide great results with datasets that contain a relatively low number of classes, such as CamVid with 11 semantic classes or Gatech with 8 [39]. The number of categories used in food analysis is much higher, thus increasing the difficulty of the task and providing not so satisfactory results [34].

Among the FCN models found in the literature applied to semantic segmentation, the Tiramisu model was considered [39], as mentioned in Section II. Its down-sampling and up-sampling paths are connected by skip connections, and its architecture is composed of dense blocks, each one of them containing a set of concatenated layers for a better training.

After training our FCN model with food tray images, the binary images predicted by it are used in the next step, which aims at tracing the exterior boundaries of the food regions, avoiding the holes inside them. In this manner, small holes that may appear inside regions are discarded and thus the regions are homogenized. For this task, we use the Moore-Neighbor tracing algorithm modified by Jacob's stopping criteria [40].

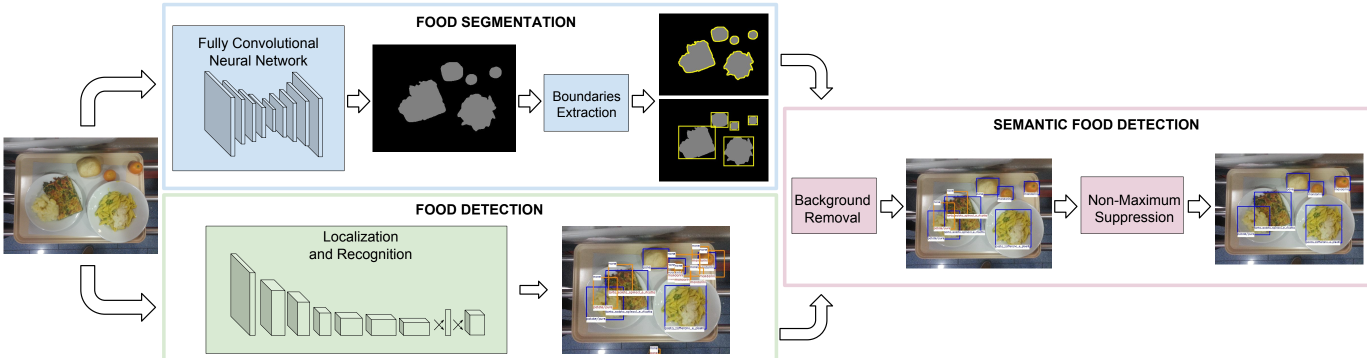


Fig. 3: Detailed workflow of the proposed Semantic Food Detection method: food segmentation and food detection methods are applied in parallel, before combining them for a final detection on food tray images.

Once the boundaries are traced, the bounding boxes that contain the regions are determined, thus obtaining a binary food detection. As small regions may also appear in the predicted images, and they usually correspond to false positives, this step also includes their elimination by considering a threshold criterion. Figure 4 illustrates an example of the outputs obtained in the food segmentation procedure, including the binary image provided by the FCN model, the boundaries extracted and the bounding boxes generated.

Fig. 4: Food segmentation output (left to right): binary prediction by FCN, regions boundaries and food bounding boxes.

B. Food Detection

In this work, following the definition of the object detection problem [21], we consider as Food Detection the localization and recognition of food. For this purpose, we propose retraining an object detection algorithm to apply food detection instead. In particular, we chose one of the best object detection approaches in the state-of-art, YOLOv2 [41], [42]. As for the model, the authors propose a new FCN called Darknet-19, composed by 19 convolutional layers and 5 max pooling layers to tackle the recognition task. They modified this network for object detection by removing the last convolutional layer and adding four convolutional layers for producing 13x13 feature maps. At each cell on the output feature maps, the network predicts B bounding boxes with five coordinates for each, among them is the confidence score t_o, and $c = 1, \ldots, C$ conditional class probabilities, $Pr(Class_c|Object)$. Predictions are obtained from the last convolutional layer having a size equal to 1×1 and F filters, where the number of filters is calculated as: $F = (B \times (5 + C))$. From this, it is possible to determine the class-specific confidence score, CS_c for each bounding box as follows:

$$CS_c = Pr(Class_c|Object) * \sigma(t_o) \tag{1}$$

where $\sigma(.)$ stands for a logistic activation to constrain the predictions to fall in the range between 0 and 1. Note that, in the experiment, we use the original setting of B (equal to 5).

C. Semantic Food Detection

In object detection, one of the most common errors are false positives, which can be classified based on the type of error: localization error, confusion with similar objects, confusion with dissimilar objects, and confusion with background [43]. Our Semantic Food Detection proposal focuses on reducing two of the most common errors of object detectors [41]: localization errors, specifically those corresponding to duplicate detections; and errors produced by the confusion with the background. For this purpose, we propose the following procedure that integrates the detection and segmentation algorithms:

1) Background Removal: The first step involves the application of both boundaries extracted (contour and bounding box) from the Food Segmentation procedure in order to remove the background detections. Let $Y = \{b_1^Y, \ldots, b_N^Y\}$ be the set of bounding boxes obtained with the detection method, $S_1 = \{b_1^S, \ldots, b_L^S\}$ and $S_2 = \{c_1^S, \ldots, c_L^S\}$ the set of bounding boxes and contours extracted by the Food Segmentation method, respectively. Considering each element belonging to the sets named above as a set of points (x, y) that defines a polygon, we calculate the probability of a bounding box, b_i^Y to belong to the background Bkg as follows:

$$Pr(Bkg|b_i^Y) = \min(CS_c(\overline{b_i^Y}), \max(Pr(\overline{S_1|b_i^Y}), Pr(\overline{S_2|b_i^Y})))$$

where $CS_c(\overline{b_i^Y})$ is the complement of the confidence score, $1 - CS_c(b_i^Y)$ for the i-th detection, $Pr(\overline{S_1|b_i^Y})$ is the probability that b_i^Y is a false detection on the extracted boxes, S_1:

$$Pr(\overline{S_1|b_i^Y}) = 1 - \max_{j=1,\ldots,L} \frac{|b_i^Y \cap b_j^S|}{|b_i^Y|}$$

where $|.|$ stands for the cardinality of a set of pixels corresponding to an image region, and $Pr(\overline{S_2|b_i^Y})$ the probability that b_i^Y does not intersect with any contour in S_2:

$$Pr(\overline{S_2|b_i^Y}) = \min_{j=1,\ldots,L} Ind(b_i^Y \cap c_j^S = \emptyset)$$

where $Ind(*)$ is an indicator function with value 1 if the condition is true, and 0 otherwise.

Bounding boxes with a probability higher than 50% to be background ($Pr(Bkg|b_i^Y) > T, T = 0.5$) are considered to be false detections, and are therefore removed.

2) Non-Maximum Suppression: The second step involves the application of a greedy procedure to eliminate duplicate detections by non-maximum suppression [44]. Once the Background Removal is applied, the remaining detections $Y' \subseteq Y$ are sorted in descending order by the confidence score $CS_c(b_j^Y)$ and grouped into C sets $Y^1, \ldots, Y^C \subset Y'$, where C is the number of classes. Then, for each $Y^c, c = 1, \ldots C$, we greedily select the highest scoring bounding boxes while removing detections that are lower in the ranking and their maximum intersection ratio (MIR) with respect to the i-th previously selected bounding boxes is more than 50%, where MIR score for the j-th bounding box is calculated as:

$$MIR_j = \max_{\forall i, i<j} \frac{|b_i^Y \cap b_j^Y|}{\min(|b_i^Y|, |b_j^Y|)}$$

Notice that the chosen food detection method already incorporates a non-maximum suppression procedure. In our framework, we propose an additional personalized non-maximum suppression that differs mainly in two aspects: 1) we consider the predicted classes for the bounding boxes, and 2) we propose a MIR score instead of the traditional IoU. The last one was applied because in some cases the overlapped predictions for the same class could have a completely different dimension and proportion, and then, the IoU score will be very small even if one bounding box is completely inside the other.

IV. Experimental Results

In this section, we first describe the dataset used to evaluate the proposed approach, which is composed of images taken in self-service restaurants. Then, we describe the evaluation measures used and present the results obtained with the different methods and model configurations.

A. Dataset

UNIMIB2016 [14] is a food dataset that has been collected in a self-service canteen. Each image includes a tray with some food placed both on plates and placemats. The acquisition process was performed on a semi-controlled environment using a Samsung Galaxy S3 smartphone. As a result, images acquired have a resolution of 3264×2448 in RGB, and present visual distortions and variable illuminations, making them challenging for any task of automatic food analysis.

The dataset is composed of 1,027 images that include a total of 73 food categories. Among them, only 1,010 images and 65 categories were used for experimentation, as suggested in [14] due to the low number of samples of the categories not considered. For experimental purposes, the dataset has been split in training and test sets: the former contains 650 images ($\approx 64\%$), whilst the latter contains 360 ($\approx 36\%$).

The annotations included in the dataset contain, for each food item: the polygon defining its boundaries, the bounding box and the food label. Figure 5 illustrates an image of the UNIMIB2016 dataset with its corresponding annotations.

Fig. 5: A representative sample of the UNIMIB dataset [14]: original image (left) and food annotations (right).

B. Food Segmentation

Metrics. In order to evaluate the different food segmentation approaches, several performance measures have been used. First, two pixel-wise metrics commonly used in semantic segmentation problems have been considered [34]:

- *Global pixel accuracy (GA).* The pixel-wise accuracy computed over all the pixels of the dataset.
- *Intersection over Union (IoU).* Also known as Jaccard index, it is defined as:

$$IoU(c) = \frac{\sum_i t_i == c \wedge p_i == c}{\sum_i t_i == c \vee p_i == c} \quad (2)$$

  where c is a class, i represents all the pixels of the dataset, t_i are the target labels, and p_i are the predicted labels. Note that this metric is calculated for each single class c, and then the mean across the classes is computed.

To perform a fair comparison with [14], three region-based metrics have been also considered [45]:

- *Covering (CO).* The covering of the ground truth (GT) by the segmented (S) images measures the level of overlapping between each pair of regions (R and R'):

$$C(S \rightarrow GT) = \frac{1}{N} \sum_{R \in GT} |R| \cdot \max_{R' \in S} \frac{|R \cap R'|}{|R \cup R'|} \quad (3)$$

  where N is the number of pixels of the image.
- *Rank index (RI).* It compares the compatibility of assignments between pairs of elements in the ground truth (GT) and the segmented (S) images:

$$RI(S, GT) = \frac{1}{\binom{N}{2}} \sum_{i<j} [\mathbb{I}(t_i == t_j \wedge p_i == p_j) + \mathbb{I}(t_i \neq t_j \wedge p_i \neq p_j)] \quad (4)$$

  where $\binom{N}{2}$ is the number of possible unique pairs among the N pixels of each image, and $\mathbb{I}$ is the identity function.
- *Variation of information (VI).* It measures the distance between the ground truth (GT) and the segmented (S) images in terms of their average conditional entropy:

$$VI(S, GT) = H(S) + H(GT) - 2 \cdot MI(S, GT) \quad (5)$$

  where H and MI are, respectively, the entropy and the mutual information. In this case, the lower the better.

Notice that these three metrics are calculated for each single image, and then the mean across images is computed.

Experimental setup. Regarding the methods used for semantic segmentation, we trained three networks based on

Tiramisu [39]: 1) *Tiramisu56*: 56 layers, with 4 layers per dense block and a growth rate of 12; 2) *Tiramisu67*: 67 layers, with 5 layers per dense block and a growth rate of 16; and 3) *Tiramisu103*: 103 layers, with a variable number of layers per dense block (from 12 to 4 in the downsampling path, and from 4 to 12 in the upsampling) and a growth rate of 16. Additionally, the *Classic Upsampling*, which uses standard convolutions in the upsampling path instead of dense blocks [46], has been also considered for comparative purposes.

All the FCN models were trained with the UNIMIB2016 dataset [14] (images resized to 360×480), and two-target labels: food vs non-food. The models were initialized with HeUniform and trained with RMSprop [39]. The training process consists of two steps: first, the models were trained with cropped images (224×224) for data augmentation and batch size 3, with an initial learning rate of $1e-3$ and an exponential decay of 0.995 per epoch; and second, their parameters were fine-tuned with full size images (360×480) and batch size 1, using a learning rate of $1e-4$. The outputs were monitored using the global accuracy and the IoU, with a patience of 100 during pre-training and 50 during fine-tuning.

Table I includes the results achieved with the four networks for semantic segmentation, as well as with the two segmentation methods from [14]: the JSEG algorithm [47], and the segmentation pipeline proposed in [14]. With respect to the pixel-wise measures, all the networks produced competitive results (over 0.96). The Tiramisu models outperformed the Classic Upsampling, thanks to the dense blocks, despite a lower number of parameters used. In general, the Tiramisu model benefits from having more parameters and depth. However, in this binary problem the Tiramisu103 produced overfitting whilst the Tiramisu67 achieved the best results, with a good trade-off between depth and performance. Regarding the region-based measures, all the FCNs provided better results than the two approaches from [14], which demonstrated the adequacy of the proposed methods for our problem.

TABLE I: Results obtained with our Food Segmentation approach in test set.

	No.	Pixel-wise		Region-based		
	param	GA	IoU	CO	RI	VI
JSEG [47]	-	-	-	0.385	0.389	3.106
Ciocca et al. [14]	-	-	-	0.916	0.931	0.429
Classic Upsam.	12.7M	0.991	0.962	0.984	0.982	0.125
Tiramisu56	1.4M	0.992	0.967	0.986	0.984	0.112
Tiramisu67	3.5M	**0.993**	**0.971**	**0.987**	**0.986**	**0.105**
Tiramisu103	9.4M	0.992	0.968	0.986	0.984	0.111

C. Semantic Food Detection Performance

Metrics. In order to evaluate food recognition and localization, we chose three standard measures commonly used in multi-class object recognition problems:

- *Recall* (Rec). The proportion of true positives detected.
- *Precision* (Pre). The proportion of the true positives against all the positive results.
- F_β*-measure*. A weighted average of precision and recall. We use $\beta = 2$ (F_2) to place more emphasis on wrong classified or undetected foods.

For comparative purposes, the measures used by Ciocca et al. [14] were also considered:

- *Standard Accuracy* (SA). It is equivalent to the recall.
- *Macro Average Accuracy* (MAA). The proportion of correctly classified foods, but taking into account the class imbalance of the dataset:

$$MAA = \frac{1}{C}\sum_{c=1}^{C}\frac{TP_c}{NF_c}, \tag{6}$$

  where C is the number of classes, TP_c is the number of correctly classified foods of class c, and NF_c is the total number of foods of class c.
- *Tray Accuracy* (TA). The percentage of trays for which all the foods contained are correctly recognized:

$$TA = \frac{1}{T}\sum_{t=1}^{T} Ind(\frac{TP_t}{NF_t} = 1), \tag{7}$$

  where T is the number of food tray images, TP_t is the number of correctly classified foods on the tray t, and NF_t is the total number of foods on the tray t.

Experimental setup. YOLOv2 was first pre-trained on the ILSVRC dataset. Following, we adapted it by changing the output of the model to 65 classes and applied a fine-tuning using UNIMIB2016 images (resized to 416×416). For training the model, we used the framework Darknet [48]. The models were trained during 4000 iterations with a batch size of 32, and a learning rate of $1e-3$. In addition, we applied a decay of 0.9 to the iterations 3000 and 3500. To avoid overfitting, we use standard data augmentation procedures with random crops and distortions in the HSV color space [42].

Once YOLOv2 training is completed, the next step is to determine the confidence threshold to be used during localization and recognition of the food. A low confidence threshold implies a greater number of detections, which maximizes the likelihood that all the foods present in the image will be detected. At the same time, it also increases the chances of obtaining false detections. Taking into account that the confidence defined by the detection method considers two factors (the fit of the bounding box to the object and the predicted class), we chose the minimum threshold according to the number of classes. Given that the target dataset has 65 classes, the minimum threshold chosen is $\frac{1}{65}$. With this value, it can be interpreted that the bounding boxes extracted will have a recognition probability greater than a random value when the detected bounding box fits the object perfectly. Following the interpretation given, we chose $\frac{1}{2}$ as maximum threshold, which implies a high probability, at least 50%, that the localized object is correctly classified.

Table II shows the results obtained in the training set using different confidence thresholds. The tested thresholds range from the minimum and maximum values mentioned above. As can be observed, when the threshold increases, the precision also increases considerably, whilst the rest of the indicators are hardly affected. When comparing the results

TABLE II: Results obtained by YOLOv2 and the proposed approach in training set using different confidence thresholds.

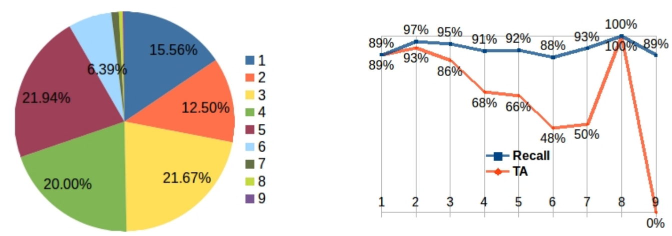
		1/65	1/32	1/16	1/8	1/4	1/2
YOLOv2 [42]	Pre	**0.511**	0.687	0.832	0.926	0.968	0.994
	Rec	0.999	0.998	0.997	0.995	0.988	0.966
	MAA	0.999	0.997	0.995	0.992	0.981	0.952
	TA	0.997	0.992	0.988	0.982	0.960	0.895
Proposed	Pre	**0.918**	0.952	0.973	0.984	0.991	0.996
	Rec	0.998	0.997	0.996	0.994	0.987	0.965
	MAA	0.999	0.999	0.996	0.994	0.981	0.951
	TA	0.995	0.992	0.988	0.982	0.957	0.894

TABLE III: Tray Food Analysis results, from top to bottom: food detection method 1) without segmentation, 2) with segmentation, and 3) with ground-truth segmentation to perform the recognition. The best results per block are in boldface.

	F_2	Pre	Rec	MAA	TA
YOLOv2 [42]	0.786	0.489	**0.927**	**0.850**	**0.769**
YOLOv2 + III-C2	**0.856**	**0.659**	**0.925**	**0.849**	0.772
Ciocca et al. [14]	-	-	0.798	0.636	**0.789**
YOLOv2 + III-C1	0.844	0.628	**0.923**	**0.846**	0.761
Proposed	**0.905**	**0.841**	**0.922**	**0.845**	0.764
Mezgec et al. [19]	-	-	0.864	-	-
Ciocca et al. [14]	-	-	0.891	0.684	**0.871**
YOLOv2 + III-C1	0.854	0.651	**0.926**	**0.850**	0.769
Proposed	**0.911**	**0.856**	**0.926**	**0.850**	0.775

obtained between YOLOv2 and the proposed method, for the minimum threshold, it can be observed that a significant improvement in precision is obtained (≈40%) with only a slight decrease in the other indicators (0.1%-0.2%). Another interesting aspect to highlight is the comparison of results when using the maximum threshold, since they are practically identical for both methods. This means that, for a threshold of $\frac{1}{2}$, there are almost no false detections that can be reduced with our procedure. For the remaining experiments, the minimum threshold was chosen for two main reasons: 1) it obtains the best results for the $Recall$, MAA and TA indicators; and 2) it allows us to discard the false positives that appear when combining the results with the food segmentation procedure.

The Semantic Food Detection results on the test set are shown in Table III. In order to see the performance of the different parts of our pipeline, we group the results of this table in three rows: the first one corresponds to the results obtained with YOLOv2 retrained for food detection, and our proposed framework without considering the information extracted from the segmentation method to perform the classification (YOLOv2 + III-C2); the second one corresponds to the results of the baseline method [14], our framework without considering the personalized non-maximum suppression procedure (YOLOv2 + III-C1), and our proposed framework; and the third row is similar to the second one, but replacing the segmentation method by the ground truth segmentation. As for the results achieved, it should be highlighted that our proposal outperforms the food recognition, with respect to the state-of-art method (Ciocca et al. [14]) in a 12.4% for *Recall* and 20.9% for *MAA*. Regarding *TA*, a decrease of 2.5% is observed. However, we consider that this measure does not reflect how well the recognition works mainly due to the imbalance in the quantity of food in the trays, which varies between 1 and 9 (see Fig. 6 (a)), as well as because *TA* measures the amount of food trays in which all positive samples have been correctly predicted, but does not penalize when there are false positives.



(a) (b)

Fig. 6: a) Distribution of the trays according to the number of foods that are placed in them. b) Results in terms of *Recall* (blue) and *TA* (orange), for each item of the distribution.

In order to apply a complete comparison, we also replicated the evaluation proposed by [14], in which the authors considered a perfect segmentation using the ground truth (GT) and applied their detection method (bottom section of Table III). In our case, there is no significant improvement with respect to the use of the proposed semantic segmentation, because our proposal considers the integration of the extracted information with the segmentation to refine the predictions already obtained by the object detection method. In contrast, Ciocca et al. [14] performed the recognition directly on the segmented objects. Comparing to the results obtained in [14], we can see that their method improves significantly in terms of *Recall* using the GT for segmentation, achieving to match our results. However in terms of *MAA*, despite improving its performance, our results are still about 16% better. A low *MAA* with a high *Recall* implies that the classifier has a strong bias towards the classes that have a greater amount of instances. Therefore, even if we consider a perfect segmentation to contrast the results, our proposal keeps a better performance for recognition and a lower bias towards the dominant classes.

The results obtained with the proposed approach based on the number of objects to be classified per food tray is shown in Fig. 6 (b). As expected, the *TA* measure tends to decrease as the number of objects increases, however there is no clear trend for the *Recall*. One of the lowest results in both measures is obtained in trays containing 6 foods, whereby we can determine that the errors correspond to 17 misclassified objects along 12 trays, that is, an average error of 1.42 objects per incorrectly classified tray. Despite having a low *TA* (0.478), the results are good considering the *Recall* obtained, since it

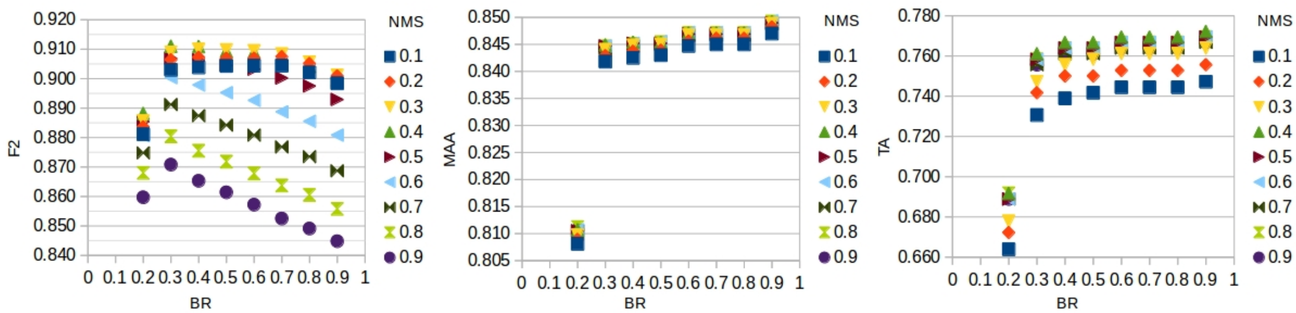


Fig. 7: Results of our proposal when varying the background removal (BR) and non-maximum-suppression (NMS) thresholds.

is preferable to minimize the number of errors per tray if we think of a semi-automatic food billing system, in which the operator would make minor corrections if necessary.

When reviewing the overall mean of errors by misclassified trays, we can see that our classifier has an average of 1.09 errors along 85 trays classified incorrectly, compared to [14] that has an average of 3.33 errors along 76 trays classified incorrectly. That said, even though the baseline method achieves to completely classify 9 trays more than our proposal, due to its overall performance, the misclassified trays have about three times as many objects wrongly classified per tray.

The results achieved with our approach consider a value of 0.5 for the thresholds used in both procedures, Background Removal (BR) and Non-Maximum Suppression (NMS). However, our approach achieves a good performance not only with a unique combination of values, but also with a wide range of them. Specifically, for the problem at hand, we can obtain results close to the ones described with any value in the ranges [0.3-0.6] for BR and [0.3-0.5] for NMS thresholds (see Figure 7). The flexibility of choosing threshold values in a wide range suggests that our approach is robust with respect to its parameters. Furthermore, considering the F2 score, any value of the parameters for our proposed method produces better results than YOLOv2 + III-C1.

Finally, some examples of the results obtained by means of our proposed Semantic Food Detection method are shown in Figure 8. In general terms, the classifier achieves a good performance in a variety of food items, where the main difficulties encountered are due to the following issues: 1) unlabeled food items, because they are not part of the 65 classes (eg. fresh cheese) or because they are not belonging to the same tray and that have been recognized by our algorithm; 2) the same food items placed very close (eg. mandarine); 3) foods ignored because they are not clearly distinguishable whether correspond to a meal or not (eg. pudding); and 4) confusions with classes corresponding to different kinds of cakes (eg. torta_cream), meats, pastas, among others.

V. Conclusions

We present a novel system that performs Semantic Food Detection applied to the problem of food tray analysis in self-service restaurants. More precisely, we integrate both techniques, food/non-food semantic segmentation with food detection, through the application of two procedures: a probabilistic procedure that allow us remove the background detections, and a custom non-maximum suppression procedure to avoid the occurrence of duplicate detections.

Regarding the architecture, we deal with the problem at hand using two pathways in parallel for food detection and semantic segmentation. The purpose of applying this separate computation is to take advantage of the benefits of each method separately to later combine them. In this manner, they do not condition each other, but reinforce themselves. In particular, if we propose an end-to-end architecture which directly feeds the segmentation output into the detection, the segmentation errors could not be recovered and, therefore, they could negatively influence the detection performance.

As for the results, our proposal significantly outperforms the state-of-art in terms of recall and mean average accuracy. Furthermore, our model is less sensitive to class imbalance and the mean of errors per foods placed on a tray is about 1, when the classifier is not able to recognize the whole tray well. The latter is quite relevant if our approach is applied in a semi-automatic billing system, in which the cashier would have to make only small changes to generate the final bill, and in this way to streamline the process involved in a self-service restaurant of *grab a meal, pay, and eat*. Furthermore, our proposed approach takes less than 0.5 seconds to predict all foods present in a image, considering the use of a personal computer with a low performance GPU (GeForce 940MX).

Our future research is focused on semantic detection of food ingredients and completely automating the self-service billing by integrating the restaurant menu by geolocalization.

References

[1] H. Kagaya and K. Aizawa, “Highly accurate food/non-food image classification based on a deep convolutional neural network,” in *Int. Conf. on Image Analysis and Processing*, 2015, pp. 350–357.

[2] F. Ragusa, V. Tomaselli, A. Furnari, S. Battiato, and G. M. Farinella, “Food vs Non-Food Classification,” in *Int. Workshop on Multimedia Assisted Dietary Management*, 2016, pp. 77–81.

[3] E. Aguilar, M. Bolaños, and P. Radeva, “Exploring food detection using cnns,” in *EUROCAST 2017*, 2018, pp. 339–347.

[4] A. Singla, L. Yuan *et al.*, “Food/Non-food Image Classification and Food Categorization using Pre-Trained GoogLeNet Model,” in *Int. Workshop on Multimedia Assisted Dietary Management*, 2016, pp. 3–11.

[5] K. Aizawa, Y. Maruyama, H. Li, and C. Morikawa, “Food balance estimation by using personal dietary tendencies in a multimedia food log,” *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 2176–2185, 2013.

[6] G. Waltner, M. Schwarz, S. Ladstätter, A. Weber, P. Luley, M. Lindschinger, I. Schmid, W. Scheitz, H. Bischof, and L. Paletta, “Personalized Dietary Self-Management using Mobile Vision-based Assistance,” in *Int. Workshop on Multimedia Assisted Dietary Management*, 2017.

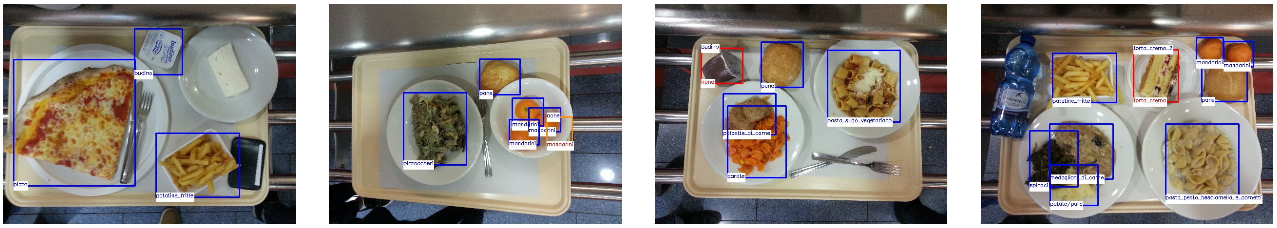


Fig. 8: Some samples of the results obtained using the proposed approach, from left to right: food tray with all the objects correctly detected (blue), false detection sample (orange), and two samples with one misclassified object (red).

[7] J. Dehais, M. Anthimopoulos, S. Shevchik, and S. Mougiakakou, "Two-view 3D Reconstruction for Food Volume Estimation," *IEEE Trans. Multimedia*, vol. 19, no. 5, pp. 1090–1099, 2017.

[8] M. Bolaños, A. Ferrà, and P. Radeva, "Food ingredients recognition through multi-label learning," in *Int. Conf. on Image Analysis and Processing*, 2017, pp. 394–402.

[9] J. Chen and C.-W. Ngo, "Deep-based ingredient recognition for cooking recipe retrieval," in *ACM Multimedia Conf.*, 2016, pp. 32–41.

[10] W. Min, S. Jiang, J. Sang, H. Wang, X. Liu, and L. Herranz, "Being a Supercook: Joint Food Attributes and Multimodal Content Modeling for Recipe Retrieval and Exploration," *IEEE Trans. Multimedia*, vol. 19, no. 5, pp. 1100–1113, 2017.

[11] W. Min, B. Bao, S. Mei, Y. Zhu, Y. Rui, and S. Jiang, "You are what you eat: Exploring rich recipe information for cross-region food analysis," *IEEE Trans. Multimedia*, 2017.

[12] J. Chen, C. Ngo, and T. Chua, "Cross-modal recipe retrieval with rich food attributes," in *ACM Multimedia Conf.*, 2017, pp. 1771–1779.

[13] G. Ciocca, P. Napoletano, and R. Schettini, "Food recognition and leftover estimation for daily diet monitoring," in *Int. Conf. on Image Analysis and Processing*, 2015, pp. 334–341.

[14] ——, "Food Recognition: A New Dataset, Experiments, and Results," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 3, pp. 588–598, 2017.

[15] G. Raimato, "The Design of a Smart Tray with Its Canteen Users: A Formative Study," in *Int. Conf. in Methodologies and Intelligent Systems for Technology Enhanced Learning*, vol. 617, 2017, p. 36.

[16] H. Kagaya, K. Aizawa, and M. Ogawa, "Food detection and recognition using convolutional neural network," in *ACM Int. Conf. on Multimedia*, 2014, pp. 1085–1088.

[17] K. Kitamura, T. Yamasaki, and K. Aizawa, "FoodLog: capture, analysis and retrieval of personal food images via web," in *Workshop on Multimedia for Cooking and Eating Activities*, 2009, pp. 23–30.

[18] M. Bolanos and P. Radeva, "Simultaneous food localization and recognition," in *Int. Conf. on Pattern Recognition*, 2016, pp. 3140–3145.

[19] S. Mezgec and B. Koroušić Seljak, "Nutrinet: A deep learning food and drink image recognition system for dietary assessment," *Nutrients*, vol. 9, no. 7, p. 657, 2017.

[20] N. Martinel, G. L. Foresti, and C. Micheloni, "Wide-Slice Residual Networks for Food Recognition," *preprint arXiv:1612.06543*, 2016.

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. of Comput. Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[22] K. Yanai and Y. Kawano, "Food image recognition using deep convolutional network with pre-training and fine-tuning," in *IEEE Int. Conf. on Multimedia & Expo Workshops*, 2015, pp. 1–6.

[23] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, and Y. Ma, "Deepfood: Deep learning-based food image recognition for computer-aided dietary assessment," in *Int. Conf. on Smart Homes and Health Telematics*, 2016, pp. 37–48.

[24] H. Hassannejad, G. Matrella, P. Ciampolini, I. De Munari, M. Mordonini, and S. Cagnoni, "Food image recognition using very deep convolutional networks," in *Int. Workshop on Multimedia Assisted Dietary Management*, 2016, pp. 41–49.

[25] Y. Kawano and K. Yanai, "Automatic Expansion of a Food Image Dataset Leveraging Existing Categories with Domain Adaptation," in *ECCV Workshops*, 2014, pp. 3–17.

[26] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101–mining discriminative components with random forests," in *Eur. Conf. on Comput. Vision*, 2014, pp. 446–461.

[27] E. Aguilar, M. Bolaños, and P. Radeva, "Food Recognition Using Fusion of Classifiers Based on CNNs," in *Int. Conf. on Image Analysis and Processing*, 2017, pp. 213–224.

[28] V. Bettadapura, E. Thomaz, A. Parnami, G. D. Abowd, and I. Essa, "Leveraging context to support automated food recognition in restaurants," in *IEEE Winter Conf. on Appl. of Comput. Vision*, 2015, pp. 580–587.

[29] O. Beijbom, N. Joshi, D. Morris, S. Saponas, and S. Khullar, "Menu-match: restaurant-specific food logging from images," in *IEEE Winter Conf. on Appl. of Comput. Vision*, 2015, pp. 844–851.

[30] R. Xu, L. Herranz, S. Jiang, S. Wang, X. Song, and R. Jain, "Geolocalized modeling for dish recognition," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1187–1199, 2015.

[31] L. Herranz, S. Jiang, and R. Xu, "Modeling Restaurant Context for Food Recognition," *IEEE Trans. Multimedia*, vol. 19, no. 2, pp. 430–440, 2017.

[32] W. Shimoda and K. Yanai, "CNN-based food image segmentation without pixel-wise annotation," in *Int. Conf. on Image Analysis and Processing*, 2015, pp. 449–457.

[33] ——, "Foodness Proposal for Multiple Food Detection by Training of Single Food Images," in *Int. Workshop on Multimedia Assisted Dietary Management*, 2016, pp. 13–21.

[34] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, 2017.

[35] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017.

[36] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2018.

[37] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *IEEE Conf. on Comput. Vision and Pattern Recognition*, 2017, pp. 1925–1934.

[38] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conf. on Comput. Vision and Pattern Recognition*, 2017, pp. 2881–2890.

[39] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The One Hundred Layers Tiramisu: Fully convolutional DenseNets for Semantic Segmentation," in *CVPR Workshops*, 2017, pp. 1175–1183.

[40] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, "Digital Image Processing Using MATLAB," *Pearson Prentice Hall*, 2004.

[41] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE Conf. on Comput. Vision and Pattern Recognition*, 2016, pp. 779–788.

[42] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *IEEE Conf. on Comput. Vision and Pattern Recognition*, 2017, pp. 6517–6525.

[43] D. Hoiem, Y. Chodpathumwan, and Q. Dai, "Diagnosing error in object detectors," *Eur. Conf. on Comput. Vision*, pp. 340–353, 2012.

[44] P. F. Felzenszwalb, R. B. Girshick, D. Mcallester, and D. Ramanan, "Object Detection with Discriminatively Trained Part Based Models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1–20, 2009.

[45] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011.

[46] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241.

[47] Y. Deng and B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 8, pp. 800–810, 2001.

[48] J. Redmon, "Darknet: Open Source Neural Networks in C," http://pjreddie.com/darknet/, 2016.