Food Image Segmentation for Dietary Assessment

Joachim Dehais[1], Marios Anthimopoulos[1,2], Stavroula Mougiakakou[1,3]

[1]ARTORG Center for Biomedical Engineering Research, University of Bern, Switzerland

[2]Department of Emergency Medicine, Bern University Hospital "Inselspital", Bern, Switzerland

[3]Department of Endocrinology, Diabetes and Clinical Nutrition, Bern University Hospital "Inselspital", Bern, Switzerland

{firstname.lastname}@artorg.unibe.ch

ABSTRACT

The prevalence of diet-related chronic diseases strongly impacts global health and health services. Currently, it takes training and strong personal involvement to manage or treat these diseases. One way to assist with dietary assessment is through computer vision systems that can recognize foods and their portion sizes from images and output the corresponding nutritional information. When multiple food items may exist, a food segmentation stage should also be applied before recognition. In this study, we propose a method to detect and segment the food of already detected dishes in an image. The method combines region growing/merging techniques with a deep CNN-based food border detection. A semi-automatic version of the method is also presented that improves the result with minimal user input. The proposed methods are trained and tested on non-overlapping subsets of a food image database including 821 images, taken under challenging conditions and annotated manually. The automatic and semi-automatic dish segmentation methods reached average accuracies of 88% and 92%, respectively, in roughly 0.5 seconds per image.

CCS Concepts

• Applied computing → Health informatics • Computing methodologies → Computer vision

Keywords

Diet assessment; dish segmentation; food recognition; diabetes; obesity; computer vision; smartphone.

1. INTRODUCTION

The increasing prevalence of diet-related chronic diseases such as obesity and diabetes has raised major concerns over the last decades. A key factor for preventing or treating such diseases is diet management. However, traditional methods are often ineffective due to the inability of patients to accurately assess their food intake. Hence, there is an apparent need for innovative solutions and services to automatically and accurately assess meals. To this end, several smartphone-based systems have been recently proposed that use as input a number of meal images and output its nutritional content, based on computer vision techniques. Such a system has to first detect the food in the image, recognize it and estimate its size before calculating the corresponding nutritional profile from the available databases. In cases where multiple foods may exist, an additional segmentation module is employed.

In this study, we propose methods for segmenting multiple food items in an already detected dish. An automatic segmentation method is presented able to detect and segment an arbitrary number of different food items in a dish. This method uses a convolutional neural network (CNN) to automatically detect food borders that guide a region growing/merging technique. A semi-automatic version of the method is also proposed that improves the results by using minimal input from the user. The output of the more reliable semi-automatic segmentation could be potentially used for retraining the CNN border detector and gradually improve the performance of the system. All methods were trained and tested on the same food image datasets containing real-word meal images, and yielded promising results.

MADiMa'16, October 16 2016, Amsterdam, Netherlands

DOI: http://dx.doi.org/10.1145/2986035.2986047

2. PREVIOUS WORK

To simplify the problem of food detection/segmentation, the proposed solutions make different assumptions on the content of the image. These assumptions consider the number, color, and shape of dishes in the image, the possible number of food items in each dish and the visual properties of the background.

Many of the well-known image segmentation algorithms have been applied to food. Shroff et al. [1] considered a simplified case where the plate is white and all foods are clearly separate. Under these conditions, a simple adaptive thresholding method performed adequately. A comparison of three segmentation methods was presented in [2] involving active contours [3], normalized cuts [4] and local variation [5], which concluded that the latter performs best. In the experiments multiple dishes were considered and the background was defined as all the pixels of the most frequent color in the image. Furthermore, Bettadapura et al. applied the contour detection method of [6] for food images in [7], combined with location and segmentation heuristics. In another study, Anthimopoulos et al. [8] employed mean-shift filtering in the CIELab color space to segment the food inside a given dish. In [9], the proposed method first detects the plates and for each one calculates a food saliency map that guides an active contour approach [3] to segment the food from the plate. However, segmentation between different foods is not considered.

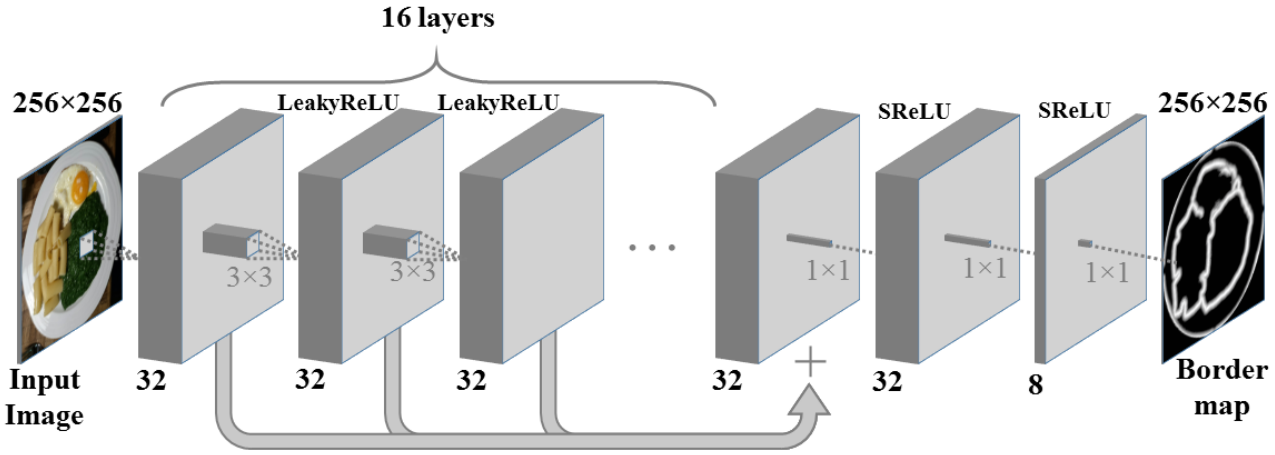


Figure 1. Architecture of the proposed CNN for food border detection

Other approaches utilized classification schemes trained on food images to improve segmentation. In [10], normalized cuts were employed with a dynamic choice of segment number guided by the food classification results. Matsuda et al. [11], combined multiple segmentation methods and kept those with the highest classification confidence. Four inputs are considered: (i) the whole image, (ii) a deformable part model segmentation [12], (iii) the detected plates via a Hough transform circle detector [13], and (iv) the JSEG segmentation [14]. In [15], greedy, binary region merging was used with color and texture features, to build a hierarchical representation of the image. This representation was then used to classify all possible segmentations and select the one with the highest confidence. Puri et al. [16] proposed a sliding window classification scheme to produce a food recognition map which was also used for segmentation. In [17] a back propagation based saliency map was created by using a CNN which guided grab-cut to perform the final segmentation. However, multi-food dishes were not considered. Similarly in [18] a CNN with global average pooling was used to produce a food probability map and also classify the detected foods.

It has been shown that food segmentation may benefit from machine learning in some cases, however relying too much on recognition could also be dangerous. Food recognition is generally a more challenging problem than segmentation, while its difficulty may change when adding new classes in a dynamically evolving system. Recently, CNNs have been used to detect edges and object boundaries/contours with great success [19]-[21]. Here, motivated by this success, we propose a method to segment multiple foods in a dish by using a CNN that detects food borders. Although recognizing foods among hundreds of classes is a very complex problem, learning to detect their borders is often easier and less dependent on the considered classes. In this way, our method can exploit the exceptional performance of CNNs and constantly improve through additional training without relying on the food recognition result.

3. METHODS

In our previous work [22], we presented methods for food detection and segmentation on mobile devices. Round dishes were first detected using robust model fitting on the image edges. Then, the foods in the dish were detected and segmented based on a region growing/merging technique while a semi-automatic version of the method was also proposed. In this study, we improve the previously proposed methods by optimizing the region growing/merging framework and incorporating a food border map. The border map is created by using a deep CNN trained on the manually annotated borders of the training set. The map is used to guide the region growing/merging algorithm so the created regions do not overlap with the predicted borders.

3.1 Border Map Generation

The architecture of the proposed CNN is shown in Figure 1. Input of the network is the original food image after being cropped according to the detected dish, and rescaled to 256×256 (Figure 2a). The target output is a map with the same size where: (i) border pixels have the value one, (ii) pixels which are more than 8 pixels away from any border are zero, and (iii) the rest of the pixels have values between one and zero, inversely proportional to their distance from the closest border (Figure 2b). To generate the target maps, we used the manual annotations of the images that consist in a polygon on the border of each existing food item. The distance transform [23] was used to calculate the distance from every pixel to the closest border pixel and the resulting values were truncated over 8 and normalized to [0, 1]. The resulting map was inverted and served as the target.

The proposed network has 16 convolutional layers with each layer having 32 kernels of size 3×3. After each convolutional layer, batch normalization is performed followed by a leaky ReLU [24] activation with its parameter set to 0.01. The output of each activation layer is used as input for the next convolutional layer. Finally, the activations from all 16 layers are summed and passed through three convolutional layers with 32, 8 and 1 kernels of size 1×1, respectively. Batch normalization is still done after each layer but here an s-shaped rectifier - SReLU [25] follows as activation,

(a) (b) (c)

Figure 2. Food border generation: (a) original image cropped on the detected dish and rescaled to 256×256; (b) the border's desired target; (c) the resulting CNN based border map.

which involves learned parameters and showed faster convergence. These last layers resemble the dense layers used in many of the popular CNNs, however they only perform densely on a single pixel level. Hence, they densely combine the values across feature maps and produce for each pixel an output in the range [0, 1] that represents the networks confidence of being on a border. By adding the output from every convolutional layer and passing it to the last dense-like layers, we prevent the problem of vanishing gradients and manage to easily train very deep networks with each layer contributing to the resulting map. This technique is inspired by the shortcut connections used in residual CNNs [26], networks with hundreds of layers that have achieved the best results so far, in many image classification/detection benchmarks. Finally, the training of the network is done by minimizing the mean absolute error (MAE) with the Adam optimizer [27]. Figure 2c shows the output of the network for the image in Figure 2a. The resulting border map looks quite similar to the desired output, however the produced borders do not always define closed segments, which is a requirement in this application. Therefore, we use a region growing/merging algorithm to extract the final segmentation.

3.2 Region Growing/Merging

The proposed segmentation algorithm relies on the Seeded Region Growing (SRG) method [28] and the Statistical Region Merging (SRM) paradigm [29]. SRG partitions images in a fast, nonparametric way: it iteratively expands image regions by adding in each iteration, the pixel with the lowest distance. At the beginning, each region is assigned just one pixel and all the neighboring pixels are candidates for expansion. Every time a pixel is added to a region, we calculate the distance of the region to this pixel's direct neighbors and add them to the list of candidates if they do not already belong to a region. When all pixels have been added, SRG terminates and the resulting regions are iteratively merged together using the SRM. The SRM is also a nonparametric aggregation method: at each iteration, the two regions with the smallest merging cost between them are joined until a stopping criterion is met. Each of the aforementioned methods involve critical choices that may substantially affect the result. For the SRG, these choices are the way the initial seeds are generated and the distance used between a region and a neighboring pixel, while for the SRM, we must choose the merging cost and the stopping criteria.

To generate the seeds for the automatic method, we create a fixed number of points on the plate, following a hexagonal grid pattern (Figure 3a). To avoid putting seeds on image edges, each of these vertices is moved to the point with the smallest gradient in a 3×3 neighborhood, and added as a seed. A seed is also created for the dish by selecting a small band of pixels on the inside of the dish border. Pixels outside the dish are not considered for segmentation. The seeds are then given to SRG to be grown to regions (Figure 3b). The distance used by SRG is a linear combination of three factors: (i) the CIELab based color distance proposed in [22], (ii) the border magnitudes of the point to be added and its closest point in the region as defined in the CNN-based map and (iii) the geometric distance from the region's seed. The first factor is a requirement for color similarity between the region and the candidate pixel. The second assures that there is no border between the two. In this way, we delay the addition of border pixels to a region and thus avoid growing regions over the borders. The last factor enforces compactness on the shape of the created areas similar to [30], which has been shown to be beneficial. Hence, the distance between a region R and its neighboring pixel p is defined as:

(a) (b) (c)

Figure 3. Region growing/merging processes: (a) initial seeds; (b) grown regions; (c) regions after merging

$$dist(R,p) = \alpha * dist_{color}(R,p) + \beta * border(R,p) + \gamma * dist_{seed}(R,p) \quad (1)$$

with $$\alpha, \beta, \gamma \geq 0 \; and \; \alpha + \beta + \gamma = 1$$

In this, $border(R,p)$ is the average border magnitude of pixel p and its closest pixel in R, $dist_{seed}(r,p)$ is the geometric distance between p and the seed point of R, and

$$dist_{color}(R,p) = \sqrt{|L^R - L^p| + (a^R - a^p)^2 + (b^R - b^p)^2} \quad (2)$$

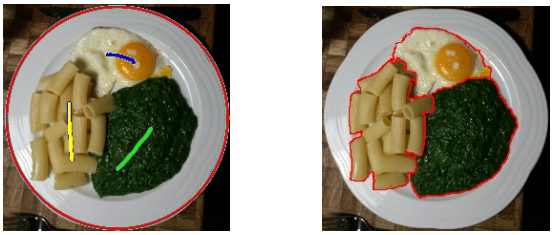
where (L^R, a^R, b^R) is the average color of the region and (L^p, a^p, b^p) is the color of the pixel. The proposed color distance puts emphasis on the color components, thus reducing the effect of intensity changes often caused by shadows. The α, β, and γ are estimated by experimenting on the training dataset using a grid of values.

In the merging step of [22], the same color distance (eq.2) was applied between the average colors of two regions. Here, instead of using just the difference between the average channel values in eq. 2, use the Earth Mover's Distance (EMD) [31] between the single-channel histograms. The EMD quantifies the cost of moving samples among the bins of a histogram to obtain another: it is the product of the distance between bins by the number of samples to move between those bins. This makes the unit of the EMD equal to number of samples by color distance, which we normalize by the number of samples to obtain a color distance. In addition, this color distance is divided by the edge length between two regions, and multiplied by the edge's border magnitude, to form the final merging cost (eq. 3). In this way, we obtain a more accurate color difference while we also avoid the merging of regions separated by short but strong edges.

$$Cost_{merge}(r_i, r_j) = EMD_{color}(r_i, r_j) * \frac{border(r_i, r_j)}{edge\_length(r_i, r_j)} \quad (3)$$

where $border(r_i, r_j)$ is the border magnitude between the regions r_i and r_j, $edge\_length(r_i, r_j)$ is the length of the edge they share and

$$EMD_{color}(r_i, r_j) = \sqrt{EMD_L(r_i, r_j) + EMD_a(r_i, r_j)^2 + EMD_b(r_i, r_j)^2}$$



(a) (b)

Figure 4. Semi-automatic segmentation processes: (a) user-given seeds; (b) grown regions

where $EMD_c(r_i, r_j)$ is the EMD between the color distributions of regions r_i and r_j in channel c. As shown, the combination of the three channels in a single color distance puts again emphasis on the color components to avoid the effect of random intensity changes. The merging iterations of SRM stop when all the remaining regions are large enough and sufficiently different to each other. To this end, the size of each region is multiplied by the cost to merge it with each of its neighbors. When all the resulting products are greater than a fixed threshold, the SRM stops.

Finally, a semi-automatic version of the segmentation algorithm is proposed where the seeds are not generated randomly but they are given by the user (Figure 4a). In this case, region merging is not performed since only one seed is given per food item. The results (Figure 4b) are equal or better to the previous method, so the method can be used for correction when the automatic results are not satisfactory.

4. DATA AND EXPERIMENTAL SETUP

The dataset used for the experiments consists of 821 meal images, each containing just one round dish with either one or multiple food items. Out of these, 246 images were acquired by the authors, in the restaurants of the Bern University hospital, "Inselspital". The rest were taken by 20 individuals with Type 1 Diabetes that participated in a pilot clinical study on the usage of the carbohydrate intake assessment system GoCARB [32]. The dataset contains a large variety of foods photographed under a wide range of shooting conditions. All images were manually annotated by drawing polygons around each food item while the ellipse of the dish is also indicated. In addition, the polygons and ellipses are converted to segmentation maps where each segment is a single connected component.

The dataset was randomly split into two main subsets the training set that consists of 70% of the images and the test set with the rest 30%. All methods were trained or tuned on the training set and tested on the test set. The result was evaluated after comparing to the ground truth by using region-based metrics similar to the Huang and Dom Index (HDI) [33]. Let $S = \{S_i\}_{i=1}^{m}$ and $T = \{T_i\}_{i=1}^{n}$ be two segmentations, where S_i (resp.T_i) is region i from segmentation S (resp. T) and m, n are the number of segments in S and T. We define two normalized directional indices based on worst and average segmentation performance:

$$NI_{min}(T=>S)=Min_i\left(\frac{Max_j(|S_i \cap T_j|)}{|S_i|}\right) \quad (4)$$

$$NI_{sum}(T=>S)=\frac{\sum_i Max_j(|S_i \cap T_j|)}{\sum_i |S_i|} \quad (5)$$

For each index, the two reverse directions are combined in a harmonic mean to give the final two indices for the evaluation:

$$F_x=\frac{2*NI_x(T=>S)*NI_x(S=>T)}{NI_x(T=>S)+NI_x(S=>T)}, \qquad x = min \text{ or } sum \quad (6)$$

Finally, both measures of equation 6 are averaged on the entire test dataset and denoted as $\overline{F_{min}}$ and $\overline{F_{sum}}$. The background segment is excluded from the computation of the measures to make the results independent from the size of the dish. To test the semi-automatic segmentation, we sampled two points inside each food segment and joined them with a straight line to imitate a user stroke.

The experiments were conducted on a machine with an Intel i7-3770 CPU and a GPU NVIDIA GeForce Titan X under a Linux OS. The CNN was implemented using the Keras [34] framework with a Theano [35] back-end and the experiments were performed on the GPU.

5. RESULTS

5.1 Border Map Generation

In order to train the proposed CNN and tune its hyper-parameters, we created an additional validation set by splitting the original training set. Thus, the dataset used for the CNN training contains 60% (=492) of the images while the rest 10% (=84) was randomly chosen to be used for validation. Moreover, to increase the size of the training set and avoid overfitting we employed a data augmentation approach. For each image, seven more images were created by applying flip and rotation operations, as well as their combinations leading to 492*8=3936 images. During augmentation, the corresponding target map for each image was also transformed accordingly. The Adam optimizer [27] was used to minimize the mean absolute error in a single-sample gradient descent training. Using batch training accelerated the training epoch but not the overall convergence of the network.

Figure 5 shows the curves of the training and validation error during training. As shown, the validation error begins converging to about .058 from nearly the first 50 epochs. The best validation error was .0577 and it was achieved in the 94th epoch. The duration of each epoch on the GPU was 721 seconds. The corresponding test error was .0545. The various algorithmic choices and the tuning of the involved hyper-parameters were based on a series of experiments. Table 1 presents the result of some of these experiments. As shown increasing the layers to 32 or the kernels per layer to 64 did not

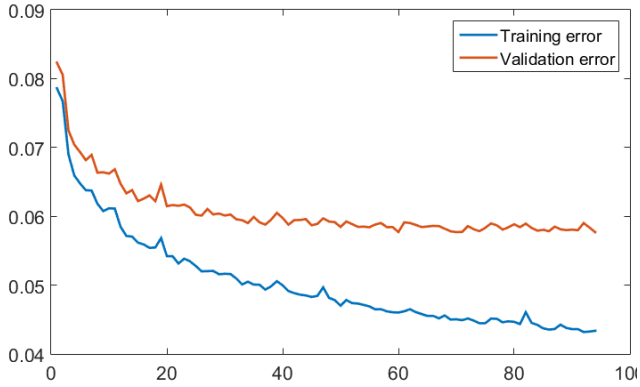


Figure 5. Curves of the training (blue) and validation (orange) mean absolute error over the training epochs of the proposed CNN

yield a significant improvement in performance. On the contrary, reducing them to 8 and 16, respectively, deteriorated the result substantially. Furthermore, 3×3 was found to be the best choice for the kernel size. A higher kernel size resulted in a larger network with wider receptive field, without increasing the performance of the network. Kernel sizes of 3×3 have been a common choice for many successful CNNs over the last few years. Finally, the batch normalization resulted in a much faster convergence although it almost doubled the duration of each training epoch.

Table 1 - Comparison of different CNN architectures

Convolutional layers	Kernels per layer	Kernel size	Validation error
8	32	3×3	.0632
32	32	3×3	.0575
16	16	3×3	.0594
16	64	3×3	.0572
16	32	5×5	.0576
16	**32**	3×3	**.0577**

5.2 Region Growing/Merging

Table 2 shows the results of the proposed food segmentation method and compares it with methods from the related literature. For all methods, the true location of the dish was used to remove the background and dish segments. Flood fill corresponds to a version of region growing with a threshold on the maximal distance between a pixel and a region, similar to [34], but using the proposed color distance (eq.2). The proposed automatic segmentation outperformed our previous works [22] and [8] by an absolute 2% and 5% in $\overline{F_{sum}}$. The improvement in $\overline{F_{min}}$ was much higher showing the stability of the proposed method. On the other hand, local variation proved to be very unstable. In the semi-automatic case, the proposed method was still better than our previous work and significantly better than flood fill. This is probably because the latter relies on sensitive thresholds to stop.

Table 2 - Comparison of segmentation methods

Automatic	$\overline{F_{min}}$ **(%)**	$\overline{F_{sum}}$ **(%)**	**Time (s/image)**
Proposed	76.2	87.6	0.5
Region growing/merging [22]	66.8	85.6	0.42
Mean-shift [8]	65.2	82.6	1.46
Local Variation [5]	36.3	72.0	1.91
Semi-Automatic	$\overline{F_{min}}$ **(%)**	$\overline{F_{sum}}$ **(%)**	**Time (s/image)**
Proposed	85.9	92.2	0.43
Region growing/merging [22]	84.5	91.3	0.41
Flood fill	77.1	85.7	0.6

Finally, we tested the proposed method on global photometric changes which were found to have only minor effects on the result. We applied changes of contrast of -20%/+25%, and gamma correction powers of .8/1.25. The $\overline{F_{min}}$and $\overline{F_{sum}}$scores remain within 1% of the normal value for the semi automatic methods. For the automatic method, the lowest $\overline{F_{min}}$ score was 2% below normal, for a gamma factor of 1.25. $\overline{F_{sum}}$ scores remained within .6% of the normal value. The stability of the results to these changes illustrates the robustness of the software to global color variation.

6. CONCLUSION

In this paper, we presented methods for the segmentation of food in a meal image with an already detected dish. The proposed method is an improved version of our previous work and uses a CNN-based food border map to guide a region growing/merging technique. The results show an absolute improvement in the segmentation performance by at least 2%. Moreover, a semi-automatic version of the method is also proposed that improves the results with minimal user input and was designed to help the user improve the result when the automatic segmentation fails. An additional benefit of using the trained border map is the fact that the system can utilize the semi-automatic results to retrain the CNN and improve its performance. This procedure could help the automatic algorithm reach accuracies very close to the semi-automatic after gathering enough data. Future work includes the extension of the methods to images with multiple dishes and the combination with 3D shape of the scene to improve the segmentation.

7. REFERENCES

[1] Shroff G, Smailagic A., Siewiorek D. P. Wearable context-aware food recognition for calorie monitoring, In 12th IEEE International Symposium on Wearable Computers (Pittsburgh, USA, September 28-October 1, 2008), 119–120

[2] He, Y., Khanna, N., Boushey, C.J., Delp, E.J. Image segmentation for image-based dietary assessment: A comparative study, in *International Symposium on Signals, Circuits and Systems*(Iasi, Romania, July 11-12, 2013), 1-4

[3] Kass M., Witkin A., Terzopoulos D. Snakes: active contour models. In *Int. J. Comput. Vis.* 1(1998), 321–331

[4] Shi J. and Malik J. Normalized cuts and image segmentation, in *IEEE Tran. Pattern Anal. Mach. Intell.* 22, 8(2000), 888–905

[5] Felzenszwalb, P. F., Huttenlocher, D. P. Image segmentation using local variation. In: *IEEE Conference on Computer Vision and Pattern Recognition*(Santa Barbara, June 23-25, 1998), 98-104

[6] Arbelaez P., Maire M., Fowlkes C., and Malik J.: Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 5 (2011), 898–916

[7] Bettadapura V., Thomaz E., Parnami A., Abowd G. D., Essa I. A.: Leveraging context to support automated food recognition in restaurants. In *IEEE Workshop on the Applications of Computer Vision* (Waikoloa Beach, USA, January 6-9, 2015), 580-587.

[8] Anthimopoulos, M.; Dehais, J.; Diem, P.; Mougiakakou, S., Segmentation and recognition of multi-food meal images for carbohydrate counting, In *IEEE 13th International Conference on Bioinformatics and Bioengineering* (Chania, Greece, November 10-13, 2013)

[9] Chen H.C. et al. Saliency-aware food image segmentation for personal dietary assessment using a wearable computer. In *Meas Sci Technol.* 26, 2(February 2015)

[10] Zhu F., Bosch M., Khanna N., Boushey C.J., Delp E.J. Multiple hypotheses image segmentation and classification with application to dietary assessment. In IEEE J. Biomed. Health. Inform. 19, 1(2015), 377-88

[11] Matsuda Y., Hoashi H., Yanai K.: Recognition of multiple-food images by detecting candidate regions, in *IEEE International Conference on Multimedia and Expo* (Melbourne, Australia, July 9-13, 2012), 25-30

[12] Felzenszwalb P. F., Girshick R. B., McAllester D., Ramanan D.: Object detection with discriminatively trained part-based models. In *IEEE Trans. Pattern Anal. Mach. Intell.*, 32, 9(2010), 1627–1645

[13] Duda, R. O., Hart P. E.: Use of the Hough transformation to detect lines and curves in pictures. In *Comm. ACM*, 15 (1972), 11–15

[14] Deng Y., Manjunath B. S., Unsupervised segmentation of color-texture regions in images and video. In *IEEE Trans. Pattern Anal. Mach. Intell.*, 23, 8, 800–810

[15] Zhang W, Yu Q, Siddiquie B, Divakaran A, Sawhney H. "Snap-n-Eat": Food recognition and nutrition estimation on a smartphone. In *J. Diabetes Sci. Technol.* 9, 3 (May 2015), 525-533

[16] Puri M., Zhu Z., Yu Q., Divakaran A., Sawhney H.: Recognition and volume estimation of food intake using a mobile device. In *IEEE Winter Conference on the Applications of Computer Vision* (Snowbird, USA, December 7-8, 2009), 1–8.

[17] Shimoda W., Yanai K. CNN-based food image segmentation without pixel-wise annotation. In *Lecture Notes in Comp. Sci.* 9281 (August 2015).

[18] Bolaños, M. and Radeva, P. (2016). Simultaneous food localization and recognition". In *arXiv:1604.07953*

[19] S. Xie and Z. Tu, "Holistically-Nested Edge Detection," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 1395-1403.

[20] Wei Shen, Xinggang Wang, Yan Wang, Xiang Bai and Z. Zhang, "DeepContour: A deep convolutional feature learned by positive-sharing loss for contour detection," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 3982-3991.

[21] G. Bertasius, J. Shi and L. Torresani, "DeepEdge: A multi-scale bifurcated deep network for top-down contour detection," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 4380-4389.

[22] J. Dehais, M. Anthimopoulos, and S. G. Mougiakakou, "Dish Detection and Segmentation for Dietary Assessment on Smartphones," in the 8th International Conference on Image Analysis and Processing (ICIAP2015), Genoa, Italy, September 7-8, 2015, vol. 9281, pp. 433–440.

[23] Felzenszwalb, Pedro F. and Huttenlocher, Daniel P. Distance Transforms of Sampled Functions, TR2004-1963, TR2004-1963 (2004)

[24] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in ICML Workshop on Deep Learning for Audio, Speech, and Language Processing (WDLASL 2013), 2013, vol. 28.

[25] Jin, X., Xu, C., Feng, J., Wei, Y., Xiong, J., Yan, S.: Deep learning with s-shaped rectified linear activation units. CoRR abs/1512.07030 (2015)

[26] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv:1512.03385, 2015.

[27] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in Int. Conf. Learn. Represent., 2015, pp. 1–13.

[28] Adams, R., Bischof, L.: Seeded region growing. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(6), 641–647 (1994)

[29] Nock, R., Nielsen, F.: Statistical region merging. IEEE Trans. Pattern Analysis and Machine Intelligence 26(11), 1452–1458 (2004)

[30] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, SLIC Superpixels Compared to State-of-the-art Superpixel Methods, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, num. 11, p. 2274 - 2282, May 2012.

[31] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In IEEE International Conference on Computer Vision, pages 59-66, January 1998.

[32] L. Bally, J. Dehais, M. Anthimopoulos, M. Lainer, D. Rhyner, G. Rosenberg, T. Zueger, , S. Mougiakakou, C. Stettler, Effects of an automated carbohydrate-assessment tool (GoCARB) on glycaemic profile in individuals with type 1 diabetes: a clinical pilot study, EASD virtual meeting 2016.

[33] Huang, Q., Dom, B.: Quantitative methods of evaluating image segmentation. In: International Conference on Image Processing, Vol. 3, pp. 53-56 (1995)

[34] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.

[35] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," arXiv eprints, vol. abs/1605.02688, May 2016. [Online]. Available: http://arxiv.org/abs/1605.02688

[36] Morikawa, C., Sugiyama, H., Aizawa, K.: Food region segmentation in meal images using touch points. In: ACM Workshop on Multimedia for Cooking And Eating Activities (2012)