

Calories Prediction from Food Images

Manal Chokr, Shady Elbassuoni

Department of Computer Science

American University of Beirut

Beirut, Lebanon

mmc35@mail.aub.edu - se58@aub.edu.lb

Abstract

Calculating the amount of calories in a given food item is now a common task. We propose a machine-learning-based approach to predict the amount of calories from food images. Our system does not require any input from the user, except from an image of the food item. We take a pragmatic approach to accurately predict the amount of calories in a food item and solve the problem in three phases. First, we identify the type of the food item in the image. Second, we estimate the size of the food item in grams. Finally, by taking into consideration the output of the first two phases, we predict the amount of calories in the photographed food item. All these three phases are based purely on supervised machine-learning. We show that this pipelined approach is very effective in predicting the amount of calories in a food item as compared to baseline approaches which directly predicts the amount of calories from the image.

1 Introduction

Maintaining a healthy diet is an important goal for many people. One way to achieve this is by tracking the amount of calories consumed. This tracking process, however, can be very tedious as it requires the user to keep a food journal and to do messy calculations to be able to estimate the amount of calories consumed in every food item. In fact, it has been also shown that people tend to underestimate the number of calories they are consuming most of the time (Elbel 2011)(Chandon and Wansink 2007).

Recently, automatic ways to calculate the amount of calories consumed in a food item have been surfacing. For instance, there are few mobile and Web apps nowadays that a user can use to do such calculations¹. Most of these tools, however, assume that the user will enter some information about the food item consumed. For instance, it might be expected that the user will enter the name of the food item or the ingredients, as well as the size of the food item. These tools typically take the user input and run it against a database of food items to be able to calculate the amount of calories in the user's consumed food item.

In this paper, we propose to alleviate the user from the burden of entering the above information in order to calcu-

late the number of calories consumed in a food item. This is particularly beneficial when such information is difficult to obtain. For instance, consider a food item eaten at a restaurant. The user might not be able to identify the exact size of the food item or all the ingredients and their portions.

Our proposed approach works as follows. The user submits an image of the food item to the system. Based on the image visual features and by means of a supervised machine-learning model, our system is able to predict what the type of the food item is. It also predicts the size of the food item (in grams) and then based on these two predicted values as well as the original features of the image, it predicts the amount of calories in the food item. We show that our machine-learning-based system and the pipelined approach we take can indeed accurately estimate the amount of calories in a given food item based solely on an image of the food item. Using a dataset of over 1,000 images of food items that belong to six different categories (burger, chicken, doughnut, pizza, salad and sandwich), we can accurately predict the amount of calories with a mean absolute error of 0.093.

While there have been some previous attempts to predict the amount of calories in a food item given its image, according to our knowledge, none of these previous attempts obtained such high accuracy as the one we obtained. Moreover, most previous attempts either restricted the types of food considered to single-ingredient food items containing basically one type of food (say a fruit or a vegetable) or relied on complementary information in addition to the images such as ingredients or cooking instructions.

Our main contributions can be summarized as follows:

- We build highly accurate machine-learning models to predict the type, size and calories of a food item based on its image.
- We build training datasets for these three supervised learning tasks mentioned above that will be made publicly available and can be further used by the research community.
- We run comprehensive experiments to study the effectiveness of machine learning in predicting the amount of calories in a food item given its image.

Copyright © 2017, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

¹www.myfitnesspal.com - www.loseit.com

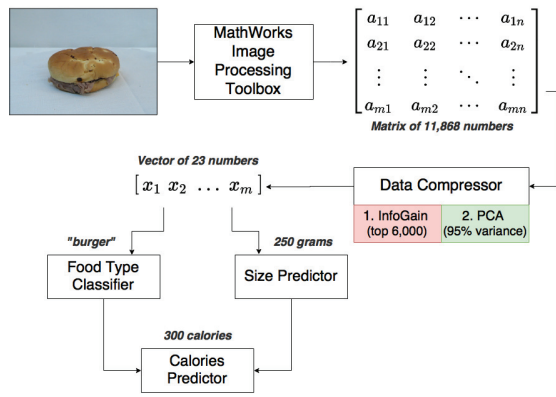


Figure 1: System Architecture

2 Our Approach

2.1 Overview

Our system architecture is depicted in Figure 1. Our system takes as input an image of a food item and outputs the amount of calories in this food item. To be able to do this, the image is passed through Mathworks Image Processing Toolbox. The toolbox extracts raw visual features from the image. This is explained in more details in Section 2.3.

Next, the image, represented by the extracted visual features is passed through a compression phase to reduce the number of features and scale the subsequent learning phases. The compression approach is based on various feature reduction techniques which are described in detail in Section 2.4. The compressed image is then passed through a classifier that predicts the type of the food item in the image. In addition, the compressed image is also passed to a regressor that estimates the size of the food item (in grams). The details of these two learning procedures are explained in Section 2.5 and Section 2.6, respectively.

Finally, the compressed image along with the predicted type and size of the food item are passed to another regressor that predicts the amount of calories in the food item. This is again based on supervised learning and the learning model used to achieve this is described in detail in Section 2.7.

2.2 Dataset

Our dataset is based on the Pittsburgh fast-food image dataset (Chen et al. 2009), which consists of images of food items belonging to over 13 categories. In each image, there is one food item, placed on a white surface with a white background. In our dataset, we sampled 1,132 images from the Pittsburgh dataset, which were evenly distributed among five food types, namely: burger, chicken, doughnut, pizza, salad and sandwich. We restricted the number of images used and the food types since annotating the dataset with size and calories information was a very time-consuming process, which had to be done very carefully in order to produce high-quality ground truth data. We will explain the annotation process of the dataset in subsequent sections.

2.3 Feature Extraction

We use Mathworks Image Processing Toolbox to extract raw features from food images. To reduce the training time in subsequent steps and improve the quality of the learnt models, each image is first cropped and brightened, and then re-sized to 4% of its initial size. The image is then passed through the toolbox which returns three different matrices (RGB matrices). The first matrix contains a cell for each pixel in the image that represents the red intensity of that pixel. Similarly, the second and third matrices contain a cell for each pixel that represent the green and blue intensity of that pixel, respectively. We consider each cell in each of these three matrices as one feature, which results in a total of 11,868 features representing one given image.

We also experimented with other ways of representing images. For instance, instead of the standard RGB representation, we use another representation of the image by averaging the red, green and blue components of each pixel. In addition, we also use a grayscale representation of the image, as well as, a black and white (BW) representation of an image with a level of 0.5, and another with a level of 0.7, where level is a threshold that determines whether a pixel is represented as 1 (black) or 0 (white). That is, we represent a given pixel as 0 if it has a luminance greater than level, and 1 otherwise. In these four last representations (averaged RGB, grayscale, BW-0.5 and BW-0.7), each image is represented using 3,956 features.

2.4 Data Compression

As we explained in the previous section, each image will be represented using a number of visual features, which in some cases range up to 11,868 features. This is a relatively large number of features, and as these features will be the basis of the learning models we build later, it is inevitable to do feature reduction in order to guarantee good generalization of the learnt models and reduce training time (Hall 1999).

To do feature reduction, we use the Principal Component Analysis (PCA) method. PCA is among the most principled and commonly used methods for dimensionality reduction in machine learning applications (Dunteman 1989). To perform PCA, we first split our dataset into two subsets: a training and validation set consisting of 80% of the images and a test set consisting of the remaining 20% of images.

PCA is then applied on the training and validation set *only* and the first m eigen vectors that preserve 95% of the data variance are retrieved. Since PCA is cubic in the number of original dimensions, it did not scale well in the case of RGB representation of images, where we originally had over 11,000 dimensions per image. To overcome this, we first perform feature selection by using the Information Gain (InfoGain) criteria for ranking the features (Hall 1999) with respect to the accuracy of the different prediction tasks we have. That is, after applying InfoGain, we got a ranked list of features in order of their influence on the prediction accuracy. We experimented with different cutoff points and determined that using the first 6,000 features as ranked by InfoGain performs the best when combined later with PCA.

Hence, we use these 6000 features to represent an image hence forth instead of the original 11, 868 features. Once each image is represented using these 6,000 features, PCA is applied to further reduce dimensionality, ending up with 23 features to represent each image. Note that both feature selection and PCA in this case were also applied on the training and validation set only. The learnt 23 eigen vectors resulting from applying this two subsequent steps are then used to transform each test image into the new 23-dimensional space.

Also note that in cases of the averaged RGB, grayscale and black and white representations of the images, we directly apply PCA for feature reduction without resorting to InfoGain first since the original number of features using these three methods of representation was less than 4000 and hence small enough to run PCA efficiently. Running PCA on these other representation modes resulted in 255 features for BW-0.5, 122 features for BW-0.7, 57 features for grayscale, and 45 features for the average RGB, as compared to 23 features in the case of the full RGB representation.

2.5 Food Type Classification

Given an image of a food item, our goal here is to predict the type of the food item. After extracting the visual features from the image as described in Section 2.3 and after performing feature reduction as we explained in the previous section, each image is represented by a small number of features. We then pass this feature vector to a classifier that outputs one of the six classes we are concerned with here, namely: burger, chicken, doughnut, pizza, salad or sandwich. This becomes an additional feature that is fed to the calories predictor as we describe in Section 2.7. We experiment with different types of classification models and report the results in the experiments section (Section 3).

2.6 Food Size Prediction

In this phase, our goal is to predict the size of a given food item in grams from its image. Similar to the case of the food type prediction, the predicted size of a food item is also fed to the calories predictor as an additional feature. However, unlike the food type classifier where the labels for the data instances (i.e., the type of each item) were present in the dataset, we do not have the sizes of the different food items in the dataset. We thus needed to generate these labels by annotating the dataset ourselves.

The annotation process we deployed to create the ground truth for the size prediction task was as follows. For each image, we first identified the restaurant the food item in the image belongs to, which is readily available information in the dataset. Then given the food item and its restaurant, we looked up all nutritional facts about that particular item available online. For example, for Mcdonald's food items, there is a tool available online that provides the exact size of each item and the total amount of calories in the food item. For other restaurants, where no such tools were available, we resorted to other online nutritional resources such as the fast food nutrition website ², which consists of a list of the

²FastFoodNutrition.org

most popular fast food restaurants and many nutritional facts about them including the size of each food item provided by those restaurants and their calories. In the cases where we still could not find the size of a given food item using the two above-mentioned approaches, we looked up the approximate size of each of the ingredients in the item taken from recipes found online as well ³, and summed these up to calculate the size of the whole food item.

Once each image in the dataset was associated with the size of the food item pictured, it was used to train a model that predicts the size of a given food item. We report the results of this in Section 3.

2.7 Calories Prediction

Finally, we describe our main task, which is calories prediction of a food item. Given the item's image and after feature extraction and reduction, the reduced features along with the two predicted features, namely: food type and size are passed to a machine-learning-based predictor which outputs a predicted amount of calories in the food item.

Our calories predictor was trained as follows. First, the dataset was annotated where each image was associated with the amount of calories the food item in the image contains. This was done in a very similar fashion to the case where we annotated the food items with size information. For each image, given the food item and its restaurant, we looked up all nutritional facts about that particular item available online which typically included the exact number of calories the item contains. In the cases where we still could not find the amount of calories in a given food item, we looked up the sizes and the amounts of calories of each of the ingredients in the item taken from online recipes, and used these to calculate the amount of calories in the whole food item.

Once the dataset was annotated with the calories information, it was used to train a model to predict the amount of calories in a food item. We report the results of our calories predictor and compare it to various baselines in Section 3.

3 Experimental Results

In this section, we present the experimental results for our three leaning tasks, namely: food type classification, size prediction and calories prediction. We then present some qualitative results of our approach and discuss its limitations and how to overcome them. All our experiments were run on an Intel(R) Xeon(R) computer with a 2.00 GHz CPU and a 32.0 GB RAM memory.

3.1 Food Type Classification

Recall that the goal of our food type classifier is to classify a food item based on its image into one of six classes, namely: burger, chicken, doughnut, pizza, salad and sandwich. In order to do so, we split our dataset consisting of 1,132 images into three folds: training (60% of the data), validation (20%) and test (20%).

Each image was passed through Mathworks Image Processing Tool to extract the raw visual features of the image (11,868 features) based on its RGB representation. We

³Mercola.com - ndb.nal.usda.gov

	Accuracy	F-measure
RGB - InfoGain - PCA	0.991	0.991
Averaged RGB - PCA	0.904	0.904
Grayscale - PCA	0.903	0.904
B&W 0.7 - PCA	0.898	0.895
B&W 0.5 - PCA	0.838	0.838

Table 1: Food Type Classification Results using SMO

	Mean Absolute Error
RGB - InfoGain - PCA	2.750
Averaged RGB - PCA	28.025
Grayscale - PCA	32.646
B&W 0.7 - PCA	43.052
B&W 0.5 - PCA	47.619

Table 2: Size Prediction Results using Random Forests

then applied InfoGain and selected the top 6,000 features followed by PCA and ended up with 23 features per image.

Based on the training data, we trained a set of different classifiers such as Naive Bayes, Regularized Linear Regression, Logistic Regression, Multilayer Perceptron (Egmont-Petersen, de Ridder, and Handels 2002) (i.e., neural networks), Random Forests and Support Vector Machines (SMO version) (Platt and others 1998). The validation set was used to set the different hyperparameters of the classification models such as the value of the regularization parameter in case of the Regularized Linear Regression or the number of layers in the Multilayer Perceptron. It was also used to select among the different models. Based on the validation set, SMO outperformed all other models.

We thus finally train an SMO classifier on the training and validation data combined and then use the test data to estimate the generalization error of the learnt SMO classifier. The learnt classifier had both an accuracy and f-measure of 0.991 on the test data (first row in Table 1).

As mentioned in Section 2.3, we also experimented with other methods of representing images. Instead of the RGB representation, we also used four other representation methods, namely: the averaged RGB representation, a grayscale representation, and black and white representation with a level of 0.5 (BW-0.5) and another with level 0.7 (BW-0.7). For each one of these four presentations, PCA was applied to reduce the number of features from 3,956 features to 45, 57, 122, and 255, respectively.

The same training and validation procedure was then applied as in the case of the RGB representation to train a food type classifier based on these other four representations. Table 1 shows the results of the SMO classifier on the test data for each one of these alternative image representations. As can be seen from the table, the RGB representation of the image followed by InfoGain and then PCA clearly outperforms all other representation methods and nearly reaches a 1.0 accuracy.

	Baseline	Our Approach
RGB - InfoGain - PCA	1.4547	0.0933
Averaged RGB - PCA	123.5702	113.8353
Grayscale - PCA	127.0762	113.9573
B&W 0.7 - PCA	133.7141	123.6653
B&W 0.5 - PCA	148.6885	148.7318

Table 3: Calories Prediction using Multilayer Perceptron

3.2 Size Prediction

To predict the size of a food item in grams based on its image, we divided our dataset into three sets: training (60%), validation (20%) and test (20%). Again, feature extraction was done based on the RGB representation of the image followed by feature reduction by means of InfoGain and PCA. As in the food type classification task, we trained various models (Regularized Linear Regression, SMO, Multilayer Perceptron and Random Forests) using the training data and used the validation set for setting the hyperparameters of the models and for model selection. The best model in terms of the accuracy of the size prediction task was the Random Forests with a mean absolute error of 2.75 grams.

We also compared the RGB representation of images to the other four methods of image representation. Again, the RGB representation clearly outperformed all other representation methods as can be seen in Table 2.

3.3 Calories Prediction

Finally, we show the experimental results of our main task, namely calories prediction of food items. As in the previous two learning tasks, we split our dataset again into three sets: training (60%), validation (20%) and test (20%). We then extracted raw features from the image based on the RGB representation and followed this by feature reduction by means of InfoGain and PCA. The resulting 23 features after performing feature reduction were then fed to our food type classifier which output a predicted food type for the image. The predicted food type becomes an additional feature for the image. Similarly, the reduced features were passed to our learnt size predictor which output an estimated size of the food item. This also learnt feature was appended to the image features for the calories prediction task. Thus, to summarize, each image was represented using 23 visual features plus an estimated food type and size.

For the calories prediction task, we again trained multiple classifiers based on the training data such as the Multilayer Perceptron, Support Vector Machines and Random Forests. Based on the validation set, the hyperparameters were set and the best model was selected, which happened to be the Multilayer Perceptron in this case, with a mean absolute error of just 0.0933 when tested on the test data.

Similar to the previous two learning tasks, we also compared the different representation methods and again the RGB representation method followed by InfoGain and PCA outperforms all other representation methods (see Table 3). In addition, we also compared our pipelined approach of predicting calories by taking into consideration the type and size of the food item to a plain baseline that only takes into



		
	<i>turkey breast</i>	<i>roast beef</i>
Actual	310 grams 450 calories	310 grams 530 calories
Predicted	309.966 grams 450.219 calories	310.08 grams 528.5218 calories

Table 4: Sample results for two different sandwiches

consideration the visual features of the image without the two additional learnt features, the type and size of the food item. As shown in Table 3, our pipelined approach with the two additional learnt features clearly outperforms the baseline approach with over 93% error reduction in the case of RGB representation. In fact, the mean absolute error is consistently reduced in most other representation methods, when the type and size of the food item is taken into consideration as can be seen in Table 3.

Note that we could not use any other baseline approaches to compare to, since most related work are either closed-source software tools or rely on additional information about the food items beyond its image, which is not applicable to our case where we want to predict the amount of calories in a food item based solely on its image.

3.4 Qualitative Results

Table 4 shows two examples of food items of type sandwich, their ground truth information and the output by our learnt models. Although these two food items have a lot of similarities in terms of type, color, size, and shape, our model was able to distinguish between the two and to accurately predict the amount of calories in each.

Table 5 shows another example of four different food items, along with their ground truth (type, size and calories). It also shows the output by our different machine learning models. As can be seen from the table, our models can accurately predict the type, size and amount of calories for each of these different food items.

Finally, note that all of the shown images as well the rest of the images in the dataset are similar in settings, that is they are all taken against a white background and on a white surface. This restriction might be unrealistic in real-world scenarios where we expect the user to take a snapshot of her food item and then use our system to predict the number of calories in the item. In that case, we do not expect either the background or the surface on which the item is placed to be as plain as in our dataset. This might have some negative effect on the accuracy of our calories prediction and more experiments need to be done with respect to this. However, we point out that there are techniques that can extract the main object in the image, in our case a food item, which we can use to eliminate the background and the surface in case any exist in the image of the food item.

Another limitation is that we only predict the amount of calories for one food item only. This might be very un-

realistic in real world-scenarios as well. Typically, a user would provide an image of a whole meal which would consist of multiple food items (for instance a burger, fries and say a salad). In that case, we need to extend our models to be able to first identify all the food items in a given meal based on its image. Once that is done, the rest of our system can be used seamlessly. We defer this task as well to future work.

4 Related Work

Automatically predicting the amount of calories in food items based on their images has received some attention in the computer vision domain. For example, Pouladzadeh et al. (2012) proposed an approach to do this by dividing an image of a food item into multiple segments, such that all the pixels represented by one segment have the same characteristics in terms of color, texture, size and shape. After segmenting the image, an SVM classifier was used to predict the amount of calories in the food item. However, their experiments were done only on images of single-ingredient food items, which limits the applicability of their approach for the type of food items we are concerned with here such as sandwiches or salads that typically would consist of multiple ingredients. In addition, their experimental results showed inconsistent performance ranging between 58.13% and 98.34% in accuracy of the prediction based on the type of the food item.

Sudo et al. (2014) addressed the problem of predicting calories and other nutritional values, by doing some segmentation and some regression analysis directly on image features. The dataset they used contained 2,500 recipes where each recipe is represented by an image, an ingredient list and cooking instructions. The first step proposed is to extract a label histogram where they divide the picture into regions, called segments, each corresponding supposedly to an ingredient and then they try to tag it, and based on different tags obtained in an image, they assign labels. They also computed the color histograms in order to compare the results for different sets of features. Finally, they used a Support Vector Regression model and reported an average error of 33.6% and 31.8% for calories prediction, using color histograms and label histograms, respectively.

Meyers et al. (2015) recently published a paper tackling the problem of calories prediction as well. They used previous results from (Beijbom et al. 2015) and (Bettadapura et al. 2015) in this field and they built a deep-learning approach to predict the amount of calories in a food item based on its image using convolutional neural networks. Their dataset was based on 23 restaurants with 2,517 food items in total. Similar to our approach, they also proposed predicting the size of the food items and other labels. Moreover, they used a GPS service to identify the geographical location from which the image was taken and hence map it to a certain restaurant. They obtained an error in prediction of about 20% on average.

According to our knowledge, these are the only related works in the literature that have addressed the problem of calories prediction based solely on food images. As can be seen from the above summary, none of these approaches achieve high accuracies in the prediction task as we report





				
	<i>breaded chicken breast</i>	<i>marble frosted doughnut</i>	<i>pizza green pepper and meat</i>	<i>chicken salad</i>
Actual	179 grams 460 calories	75 grams 240 calories	100 grams 310 calories	190 grams 180 calories
Predicted	chicken 181.57 grams 459.42 calories	doughnut 75.06 grams 243.11 calories	pizza 99.76 grams 309.02 calories	salad 190.89 grams 176.92 calories

Table 5: Sample results for four different food items

here. Moreover, most of these approaches rely on additional information about the food items such as the restaurant that the food item belongs to, or the list of ingredients and cooking instructions of the food item. This all limits the applicability of these approaches. On the other hand, our approach does not require any additional information apart from an image of the food item and can handle a wide range of complex food types consisting of multiple ingredients.

5 Conclusion

In this paper, we tackled the problem of predicting the amount of calories in food items based solely on their images. To achieve this, we adapted a pipelined approach that first predicts the type and size of the food item in the image, then uses this information in addition to the visual features of the image to predict the amount of calories in the food item. All our prediction tasks were performed using supervised machine learning, which was based on a carefully annotated dataset of fast food images. We compared our pipelined approach to a baseline approach that directly predicts the amount of calories based only on the image, and showed a reduction of over 93% in mean absolute error.

In future work, we will extend our dataset to include more food types other than the six we experimented with here. This will involve additional annotation tasks which can be outsourced to crowdsourcing platforms to scale this annotation process. Moreover, we will extend our dataset to include more diverse images with different settings such as the backgrounds or serving surfaces and study the effect of such factors on the prediction performance. Finally, we will extend our system to handle the more realistic scenario where the user provides an image of a meal rather than just one individual food item as we assumed here.

Acknowledgment

We would like to thank the American University of Beirut research board (URB) for the funding of this project.

References

Beijbom, O.; Joshi, N.; Morris, D.; Saponas, S.; and Khullar, S. 2015. Menu-match: restaurant-specific food logging from images. In *2015 IEEE Winter Conference on Applications of Computer Vision*, 844–851. IEEE.

Bettadapura, V.; Thomaz, E.; Parnami, A.; Abowd, G. D.; and Essa, I. 2015. Leveraging context to support automated food recognition in restaurants. In *2015 IEEE Winter Conference on Applications of Computer Vision*, 580–587. IEEE.

Chandon, P., and Wansink, B. 2007. The biasing health halos of fast-food restaurant health claims: lower calorie estimates and higher side-dish consumption intentions. *Journal of Consumer Research* 34(3):301–314.

Chen, M.; Dhingra, K.; Wu, W.; Yang, L.; Sukthankar, R.; and Yang, J. 2009. Pfid: Pittsburgh fast-food image dataset. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, 289–292. IEEE.

Dunteman, G. H. 1989. *Principal components analysis*. Number 69. Sage.

Egmont-Petersen, M.; de Ridder, D.; and Handels, H. 2002. Image processing with neural networks a review. *Pattern recognition* 35(10):2279–2301.

Elbel, B. 2011. Consumer estimation of recommended and actual calories at fast food restaurants. *Obesity* 19(10):1971–1978.

Hall, M. A. 1999. *Correlation-based feature selection for machine learning*. Ph.D. Dissertation, The University of Waikato.

Meyers, A.; Johnston, N.; Rathod, V.; Korattikara, A.; Gorbani, A.; Silberman, N.; Guadarrama, S.; Papandreou, G.; Huang, J.; and Murphy, K. P. 2015. Im2calories: towards an automated mobile vision food diary. In *Proceedings of the IEEE International Conference on Computer Vision*, 1233–1241.

Platt, J., et al. 1998. Sequential minimal optimization: A fast algorithm for training support vector machines.

Pouladzadeh, P.; Villalobos, G.; Almaghrabi, R.; and Shirmohammadi, S. 2012. A novel svm based food recognition method for calorie measurement applications. In *ICME Workshops*, 495–498.

Sudo, K.; Murasaki, K.; Shimamura, J.; and Taniguchi, Y. 2014. Estimating nutritional value from food images based on semantic segmentation. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, 571–576. ACM.