Identifying the Cuisine of a Plate of Food

Mabel Mengzi Zhang
University of California San Diego
mezhang@ucsd.edu

Abstract

Ingredients are the core components that make a dish what it is, besides the preparation process. Using the idea of attribute-based classification, we seek to classify plates of food to the correct cuisine by the country, using the ingredients as attributes for a plate of food. Because of the important role that ingredients have in any plate of food, this method can be generalized to any type of dishes with any type of ingredients, and can learn new dishes not seen in the training, as long as the ingredients can be specified to fit an attribute descriptor. Our dataset came from online sources and includes three cuisines, each with two dishes represented by 76 images. Even though our dataset is limited, reasonable results and a mean accuracy of 82.9% show that the method could be generalized to more categories.

1. Introduction

1.1. Context

Food is an indispensible part of our lives. In today's globalized market, food from different geographical regions show remarkable variations in the choice of ingredients and the ways to prepare them. Some cuisines have ingredient choices unimaginable to customers habituated to other cuisines, but still present surprisingly tasty dishes. Besides unique ingredients, even the same ingredients, depending on the preparation process, may end up preserving different fractions of the nutritional value and very different calories.

If restaurant networks provide a mobile app that can aim at a photograph of a plate of food and report its cuisine, the composing ingredients, and list other similar dishes, it could be a good marketing strategy. In addition, restaurants of a specific cuisine can also refer to other cuisines' use of ingredients to sparkle ideas for more creative dishes that utilize similar ingredients in new ways.

One core component of such an application is to recognize the cuisine and ingredients on a plate of food. Then the information may later be used to compare cuisines and find a network of common ingredients in the dishes.

1.2. Approach

An intuitive way that humans use to identify geographical cuisines, or even individual dishes, is by the types of ingredients that form the dish and how they appear in amount, shape, and arrangement. We focus on the type of ingredients and their amount on a plate to approach the classification, as they are the most fundamental components that any entry-level cook would have to control in order to make a dish what it is. Thus, our approach first tries to identify the ingredients on the plate, then using the ingredients found, it tries to classify which cuisine category the plate belongs to, according to the type of the ingredients and the amount present.

This ingredient-cuisine approach is a two-layer process, and we take advantage of this structure by using the attribute-based classifier [8]. We represent the ingredients as attributes in the first layer, and the cuisine as the categories in the second layer. We will describe the approach in further details in the subsequent sections.

In section 2, we discuss some of the existing works related to food recognition or classification. In sections 3 and 4, we explain the method and the implementation and dataset we used. Then we present and analyze the results in section 5. Finally, possibilities for future work are suggested in section 6.

2. Related Work

As far as we know, there are no previous work aiming to identify the cuisine on a plate of food. However, there are some relevant food classification problems tha have been explored and had similar technical challenges.

2.1. Identify ingredients on the dish

Color and texture are the most obvious properties that human use to identify food. Bolle *et al*. have successfully classified produce using color, texture, and other features [3]. A recently published food recognition work by Yang *et al*. also identifies each ingredient on the plate before deter-

mining the category of the food, such as sandwich or salad [12]. Therefore we also approach the problem by first identifying the ingredients on the dish before classifying the cuisine that the plate belongs to.

2.2. Feature descriptor for the cuisine

Yang *et al.*'s method first identified the ingredients, gave each ingredient a probability label, and then used pairwise local features among the ingredients to determine the food category, by calculating the distance, orientation, and other properties between each pair of ingredients [12]. The features were used across 61 categories of food, where 7 categories were analyzed in more detail [12].

Identifying cuisines is different from identifying the specific food, because different cuisines may have different typical arrangements on the plate that is recognizable without necessarily knowing all the ingredients composing the dish. For example, one does not have to know the composites of a piece of sushi to be able to tell that a dish is Japanese. Some dishes have unique ingredients, such as red ginger, that are indicative of the cuisine; however, some other dishes use commonly-seen ingredients, and it is the particular preparation process that brings forth a characteristic appearance. For example, rice may be compressed into tight round and triangular shapes in Japanese cuisine, but eaten loose in most Indian and Chinese dishes, dyed in yellow in Indian dishes, and appear orange in Mexican dishes.

Given that each cuisine can have a variety of plate layouts, we will assume a typical appearance known to the mass. Since not very much literature focuses on food recognition, there are possibly unexplored feature descriptors that can yield promising results specifically to cuisines.

Wu has approached a similar problem with classifying dishes by ingredients, and the results presented suggest that color textons yields a better result than color histograms and texture combined. [11]. Thus we took advantage of the idea and directly used color textons as clustering. The nature of our problem is different from Wu's in that we classify the type of cuisines from ingredients, whereas Wu classifies for ingredients. We also found a clustering program that yields more suitable results for us than the JSEG used in Wu's experiments.

Differing from all the work listed above, we use a two-level approach, where instead of classifying the cuisine directly from the image, we introduce an intermediate high-level description of ingredients, which are used instead of the raw image features to identify the cuisine. The advantage of our high-level description is that it is more intuitive and can learn new categories very quickly without having to rerun the training, as long as the user can specify the high-level feature fairly quickly by knowing roughly how much of each ingredient is commonly present in a new dish.

3. Method

3.1. High Level Image Information

We used color to cluster a plate of food into sections of different ingredients, and we used the amount of each type of ingredients on a plate to help determine the cuisine.

Color is one of the most important pieces of information that humans use to identify ingredients. For example, while whole zuchini and Chinese eggplant look very similar, we know that the green one is the zuchini, and the purple one is the eggplant. It is even more true for food on a plate, where the food is cut up into pieces, often similar small cubes or round slices, making it even more confusing to identify, if color information is not available. Therefore we preserve the color information from the image and use methods that can deal with color. Features such as Scale-Invariant Feature Transform (SIFT) and SIFT-like descriptors discard color information, and since we are currently only using one type of low-level image feature, we did not look at features that would discard color [9]. However, shape desciptors like SIFT are potentially helpful if the method is expanded to somehow include more than one low-level features, such as a combination of SIFT and RGB histograms. Details on how we used color to cluster a plate of food into ingredients is discussed in section 4.1.

The amount of each ingredient present on a plate is another important factor. It makes the difference between a plate of beef-stuffed large green peppers and a plate of bulgogi, or Korean stir-fried beef, with peppers on the garnish, as the amount of green pepper is more than or less than the amount of beef, respectively. We measure the amount of an ingredient by taking the ratio of that ingredient's area to the entire plate's food area, where the area is simply the number of pixels. During the training process, an ingredient's area ratio is used to train the attribute classifier for that ingredient. More details follow.

3.2. Attribute-Based Classification

As mentioned earlier, the attribute-based classifier is a two-layer method that uses high-level features called "attributes" as an intermediate layer between the raw image and the categorization results [8]. An attribute is a piece of factual information that can be described by high-level natural language and can generalize to all the categories, such as "green" and "lives in water." In our case, we use each ingredient as an attribute. After a certain number of specific attributes are chosen, an attribute vector is assigned to each category. For example, in a hypothetical two-attribute case of "green" and "lives in water", the binary attribute vector for salmon would be [0, 1], for zuchini would be [1, 0], and for seaweed would be [1, 1]. Then for each attribute, a binary classifier is trained with all the positive and negative image samples; this is the layer 1 classifier. Once all the

attribute classifiers are trained, they are used to do a second layer classification for the final category. For a typical image, the process goes from the image to extracting low-level feature descriptor such as RGB histograms, to assigning it an attribute vector, training the layer 1 attribute classifier, and training the layer 2 final classifier, in that order. See Figure 1 for a graphical representation of the two-layer classification hierarchy.

In Figure 1, the hierarchy represents the process from raw images to attributes and then to the final category. At the top, the green circles represents the raw images' low level feature, e.g. EMD, RGB histogram, SIFT, etc. At the green arrows stage, each image's attribute vector is used to form the training matrices for the training of m ingredient (attribute) classifiers. If an ingredient does not exist, the value for this image in the training matrix is 0. Otherwise the value is the ratio of the number of pixels of that ingredient to the total number of pixels of the food area. The yellow circles represent the m attribute classifiers, one for each ingredient. In a simple case, these would be binary classifiers indicating whether an image has an ingredient. In our case, we use the ratio of the ingredient area to the total food area, and then use regression to predict the ratio in test images to obtain predictions for the attribute classifier layer. Ground truth for test samples at this level is the image's area ratio for the specific ingredient classifier, as extracted from the image's attribute vector. At the stage of the yellow arrows, attribute vectors are used as training samples for the final classifier. Each attribute vector is 1 by m, where m is the number of ingredients. Each image has an attribute vector, indicating how much of each ingredient the image contains. Ground truth is the cuisine ID labeled for the image. Finally, the blue circle represent the final prediction of one cuisine out of k cuisines, each assigned an ID.

Ideally, all images in a category would share the same binary attribute vector description. However, the world is not ideal, and in the case of dishes, the ingredients in one dish may vary drastically depending on the chef, the season, and many other factors. This means that attribute vectors for different images in the same category are not necessarily the same, and that they cannot simply be binary. To this end, we have chosen to represent each attribute, or ingredient, using the area of that ingredient present in the image. The area is represented by the ratio of the number of pixels in an ingredient to the number of pixels of the entire food region in an image.

In determining the attributes, we could represent each ingredient as one attribute, and train as many layer 1 attribute classifiers as there are ingredients. However, the number of ingredients is very large across cuisines. To reduce the number of classifiers we have to train, we divided the ingredients into 16 types of ingredients, listed in Table 1, and we train the same number of attribute classifiers, one for each attribute.

Ingredient	*Description*
0	pasta
1	tomato
2	greens
3	red fish, as in tuna
4	seaweed, as in sushi
5	carrots, orange or pink veggies
6	meat, brown bread
7	orange fish, shrimp, fish eggs, fried food
8	white veggies, rice
9	dark veggies, eel-colored ingredients
10	egg yolk, yellow green veggies, cooked onions, light white fish
11	chili sauce, kimchi, red peppers, red veggies
12	whitefish, pinkish raw fish
13	cheese
14	flour (roasted, yellow), burnt cheese
15	pepperoni, sausage

Table 1. Ingredient categories used for the attribute vectors. Each column of an attribute vector is one ingredient category. The numerical value in each column represents the ratio of the corresponding ingredient category to the total food area in a specific image. Each image has exactly one attribute vector in each attribute that is present in the image.

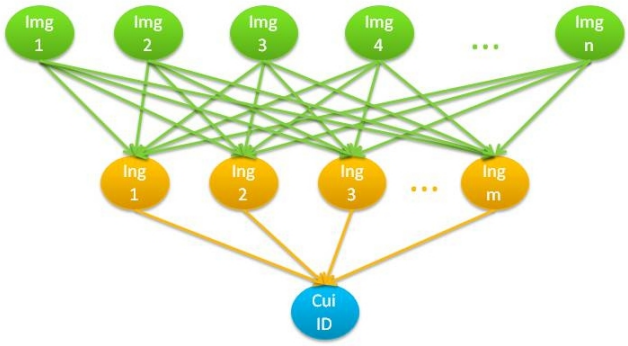

Figure 1. A visualization of the attribute-based classification, using the dataset as a concrete example. Img is short for sample image; Ing for ingredient, which we used as the attribute; and Cui for cuisine. From the top layer, low-level feature descriptors are extracted from the raw image, then used to train a number of attribute classifiers, which acts as an intermediate layer before classification. The attribute vectors are then used in a second layer of training to predict the category from the attributes solely, completely neglecting the raw image data. The advantage of this method is that any new types of dishes can be learned once the classifiers are trained, even if the new dish's attribute vector has never been seen before. This makes it possible to generalize to any type of cuisines and any dishes, as long as it can be described by an attribute vector, which in our case is a vector that indicates how much of an ingredient is present on a plate.

In the second layer, predicting the cuisine category from the attributes, we use the attribute vectors to train the final classifier, with the cuisine category ID as ground truth. Hence the final categorization of the cuisines is separate from the raw images, as it uses the intermediate high-level attributes layer to predict the results, as shown in the hierarchy in [8]. During testing, predictions of the area ratios are extracted from the individual layer 1 attribute classifiers and reshaped into attribute vector predictions of each image. These attribute vector predictions are then used as test sample data.

3.3. Low-Level Feature Descriptors

To train the layer 1 attribute classifiers, we used the Earth Mover's Distance [10]. Other low-level feature descriptors may be used, including RGB histogram, which we plan to experiment and compare with in the future. The results are presented in Section 5. Ground truth for the attribute classifier training is the area ratio from the attribute vector.

The Earth Mover's Distance is calcualted between each pair of clusters in an image, where each cluster is one ingredient, ideally. An image is first clustered into groups using the Texton-izer, an algorithm that clusters an image based on the color textons information [2]. The algorithm takes a number that specifies the number of clusters to divide each image into, and we chose 4, because that seems to be able to cover for all types of dishes, although we do not have emperical data to support the choice. Then each cluster is labeled with the ingredient ID as shown in Table 1, and clusters that contain no food items, such as the background or the container, are labeled with -1. (These labels are used later for the ingredient attribute classifier training, where the clusters that are labeled with a certain ingredient are used as positive data for that ingredient classifier, and clusters that do not belong to this ingredient are used as negative data. The setup is the same for all attribute classifiers.) Once the clusters are labeled with the ingredient ID, the image is loaded back into the program, and each cluster's RGB data is used to calculate the EMD against every other cluster.

We used the EMD implementation provided by the authors of [10]. The algorithm takes an adjustable maximum number of feature points from each of the two clusters in interest and a distance function, and then calculates the distance between them. We kept the number of feature points at the default of 100, and we select a random set of 100 points from the cluster for calculation, because the entire cluster contains tens of thousands of points, and it is timewise impractical to use all of them for calculation. Because of the random selection, the distance may be slightly different at each run, and the distance between two clusters may be different. For instance, EMD(cluster1, cluster2) may be different from EMD(cluster2, cluster1).

For each cluster, its EMD with the other clusters is put into a 1D vector, where each column is one ingredient. So for a cluster that has ingredient ID 1, its EMD vector looks like $[0, d_2, d_3, ...d_{15}]$, where d_i is the EMD between ingredient 1 and ingredient i, and d_1 is 0 because the EMD of a set of points and itself is 0. Similar for other clusters. Clusters that do not have any ingredients present are discarded and not entered into a EMD vector. The resulting matrix consists of n rows by m columns, where n is the number of ingredients present in this image and m is the number of ingredients in the system. This matrix is saved for later use in attribute classifier training, during which each ingredient's EMD vector, from all the images where the ingredient is present, are loaded as positive data to train that ingredient's attribute classifier.

4. Implementation

The main classification system is written in C++ with the OpenCV library [4]. MATLAB and Python were used in the data-acquiring stage to execute the code from [7] to query and download images from Flickr. A complete illustration of the process from raw image to final classification is shown in Figure 2.

4.1. Clustering an Image into Ingredient Regions

Segmentation is the first step that we needed to separate the image into regions of ingredients. We used the Texton-izer to segment, and in addition, cluster the image using color and texton information [13, 2]. It returns quite accurate clusters for the most part, where each cluster is one type of ingredient, or several similar ingredients, which is still acceptable. It also works when the clusters are not continuous in the image.

We have also tried JSEG, which was used in Wu's experiments, but the Texton-izer gave cleaner boundaries on our dataset, and the clustering in addition to segmentation is more useful for our purpose [11, 6]. See Figure 3 for an example of a clustered image.

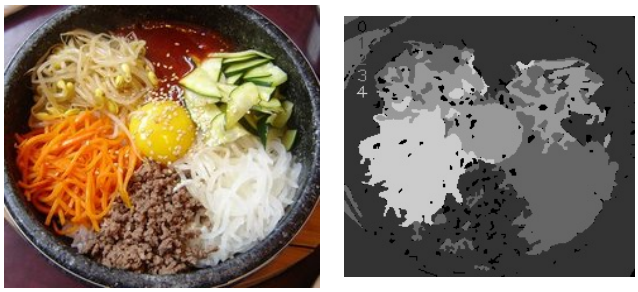

Figure 3. An image is clustered into regions by ingredient using the Texton-izer, which segments and clusters the image based on color textons.

4.2. Attribute-Based Classification

In the two-layer stages of the attribute-based classification, we used a SVM regression with RBF kernel for layer 1, and SVM classification with a third degree polynomial kernel for layer 2. SVM was chosen because it was the classifiers that the [8] used, and SVM is a good baseline regardless. For the implementation, we used the one in the OpenCV library [4].

4.3. Dataset

Flickr has a good variety of photographs of food items [1]. They are easily and freely downloadable with automated scripts, and the one we used to query and download the data is [7]. A subset of the data was obtained from Google Image Search. An existing dataset that we considered but did not use, the Pittsburgh Fast-Food Image Dataset (PFID), is readily available with many categories, but it is limited to fast-food items [5].

We selected photographs that are rich in color to avoid bad lighting or lighting that is too different from the majority of the images in the dataset. Images are cropped to the food region to avoid as much background and non-food areas as possible.

5. Results and Discussion

Accuracies for the several low-level feature descriptors we used for training the layer 1 attribute classifiers are listed in Table 2. Confusion matrix of the three classes is in Table 3.

Fold	1	2	3	4	5	Mean	Stdev
Acc	60.0	86.7	95.6	86.7	85.4	82.9	12.0

Table 2. Accuracy percentages for each of five cross validation folds and the average accuracy. Each of the first four folds contains 45 images, the last one contains 48, for a total of 228 images, with 76 images per category.

	1	2	3
1	100	0	0
2	11	86	4
3	12	25	63

Table 3. Confusion matrix of the final classification results. Row / column 1: Italian; row / column 2: Japanese; row / column 3: Korean. Correct classifications are on the diagonal. Cell (1, 1) means Italian classified as Italian, (1, 2) means Italian misclassified as Japanese, (1, 3) means Italian misclassified as Korean, etc.

Cross validation results show that fold 0 has noticeably lower accuracy than the other folds. We saw in the misclassified images that almost all of bibimbap, a Korean dish with many colorful ingredients, are misclassified. The cause of this may be that the training images are off-balanced and that most (15 out of 19) of the bibimbap images are in fold 0, whereas folds 1-4 contain all of the bulgogi images. Therefore, when fold 0 is used as test data, there were only 4 images that trained for bibimbap, which was insufficient to learn about bibimbap properly.

The dish that was the most overwhelmingly misclassified was bulgogi, the Korean roast beef. This dish tends to be consisted entirely of beef and few other ingredients, meaning that it sometimes could only have one existing ingredient, the beef. All of the misclassified bulgogi images turned out to be this case. When this happens, the beef ingredient's EMD would only be calculated against itself, which has a distance of 0, making the EMD data matrix a single-row matrix with all -1 except for one 0, which is probably problematic for training. Other images of bulgogi that are classified correctly have at least two rows of EMD data. To solve this problem, we have to come up with a way in constructing the EMD matrix to make single-ingredient dishes useful and not just contain -1 and 0 but something else in relation, so that the matrix can be meaningful for training.

A possible downside to the attribute-based classifier is that the cuisine level depends on the attribute level, so if the attribute level makes mistakes, then the mistakes would be carried through to the cuisine level.

6. Conclusion and Future Work

We have investigated the attribute-based classification in the domain of food recognition, more specifically, classifying dishes into their cuisines, using ingredient area ratios as attributes and Earth Mover's Distance as the low-level feature. EMD turns out to be problematic for dishes where only one ingredient is present, so the the image's only EMD is the ingredient's distance with itself, which is not meaningful and caused a majority of the misclassifications.

In the near future, we intend to examine more into every step of the process and expand the dataset to more categories and dishes. Experiments we may explore include different SVM kernels, other low-level image features such as RGB histograms, varying number of pixels to use for EMD, attempts to solve the one-ingredient problem with EMD, and revisions if the current algorithm is not as generalizable to a larger dataset as we expect.

Other possibilities for those interested may include using shape information, such as SIFT, in combination with the color texton clusters that we are already extracting, for example, by concatenating the shape descriptor to the attribute vector [9]. Currently, the spatial relation between the color texton clusters are not being used, so incorporating them into the feature vector would also be a potential improvement of accuracy.

Applicability of the algorithm is very wide. It may be used as the backbone of a mobile application that takes pho-

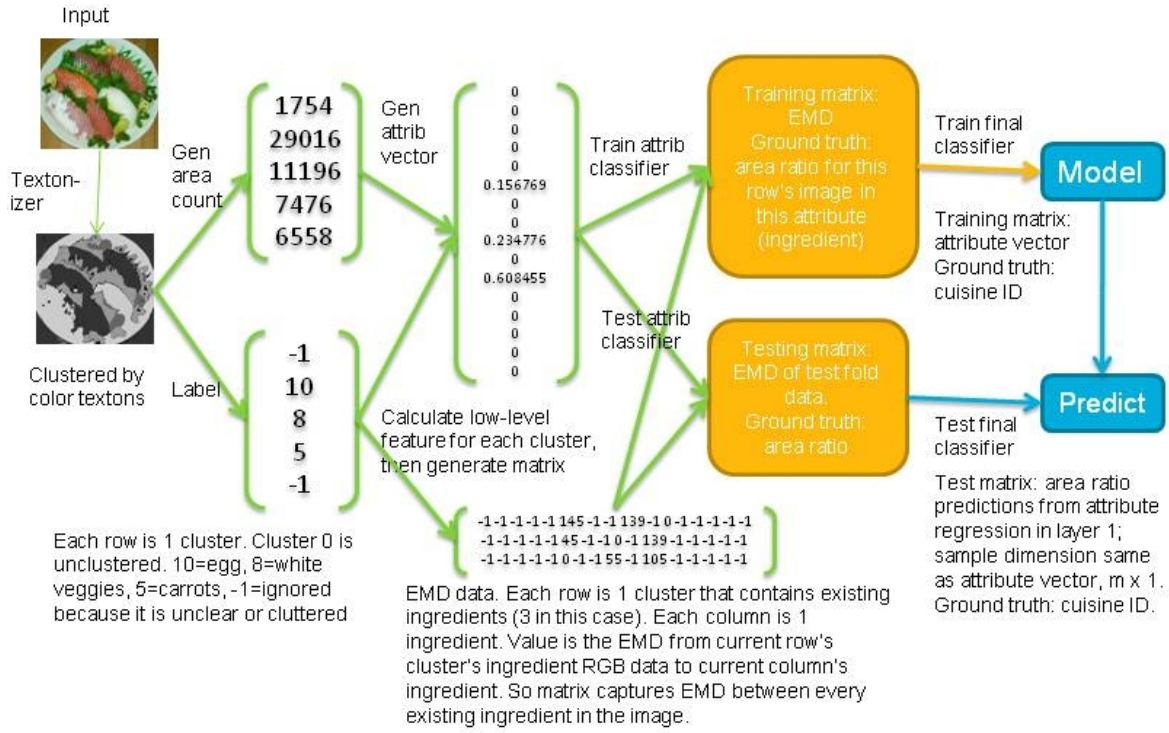

Figure 2. Flowchart of a sample image illustrating the process from raw image, to attribute, and to category prediction. Color coding corresponds to the layers in the attribute-based classification hierarchy in 1.

Figure 4. Samples of the classification results. Columns and rows correspond to those in the confusion matrix; read as [row] classified as [column], see Figure 3. Row / column 1: Italian; row / column 2: Japanese; row / column 3: Korean. Correct classifications are on the diagonal.

tos and send them into classification, which then shows the dish in a network in relation to other dishes and ingredients. This can help restaurant customers see what other dishes they are interested in, and which restaurants have them. It can also help restaurant owners quickly find out what ingredients others are using for the similar dishes and encourage market competition.

References

[1] Flickr http://www.flickr.com/. 5

[2] Texton-izer : An irregular textonizer http://code.google.com/p/texton-izer/. 4

[3] R. M. Bolle, J. H. Connell, N. Haas, R. Mohan, and G. Taubin. Veggievision: A produce recognition system. *WACV 1996*. 1

[4] G. Bradski. The opencv library http://opencv.willowgarage.com/wiki/. *Dr. Dobbs Journal of Software Tools*, 25(11):120, 122–125, Nov. 2000. 4, 5

[5] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and J. Yang. Pfid: Pittsburgh fast-food image dataset. *ICIP 2009*. 5

[6] Y. Deng and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(8):800–810. 4

[7] J. Hays and T. Berg. Code for finding and downloading images on flickr `http://graphics.cs.cmu.edu/projects/im2gps/flickr_code.html`. 4, 5

[8] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. *CVPR, 2009*. 1, 2, 4, 5

[9] D. Lowe. Distinctive image features from scale invariant keypoints. *IJCV*, 60(2):91–110, 2004. 2, 5

[10] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. *Proceedings of the 1998 IEEE International Conference on Computer Vision, Bombay, India*, pages 59–66, Jan. 1998. 4

[11] T.-F. Wu. Cafeteria vision: Identification and amount measurement of foods in a plate. 2, 4

[12] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar. Food recognition using statistics of pairwise local features. *CVPR 2010*. 2

[13] S.-C. Zhu, C.-E. Guo, Y. Wang, and Z. Xu. What are textons? *IJCV 2005*. 4