Food Recognition Using Statistics of Pairwise Local Features

Shulin (Lynn) Yang[1] yang@cs.washington.edu
Mei Chen[2] mei.chen@intel.com
Dean Pomerleau[2,3] dean@pomerleaus.com
Rahul Sukthankar[2,3] rahuls@cs.cmu.edu

[1]University of Washington [2]Intel Labs Pittsburgh [3]Robotics Institute, Carnegie Mellon

Abstract

Food recognition is difficult because food items are deformable objects that exhibit significant variations in appearance. We believe the key to recognizing food is to exploit the spatial relationships between different ingredients (such as meat and bread in a sandwich). We propose a new representation for food items that calculates pairwise statistics between local features computed over a soft pixel-level segmentation of the image into eight ingredient types. We accumulate these statistics in a multi-dimensional histogram, which is then used as a feature vector for a discriminative classifier. Our experiments show that the proposed representation is significantly more accurate at identifying food than existing methods.

1. Introduction

Automatic food recognition is emerging as an important research topic in object recognition because of the demand for better dietary assessment tools to combat obesity. The goals of such systems are to enable people to better understand the nutritional content of their dietary choices and to provide medical professionals with objective measures of their patients' food intake [18]. People are not very accurate when reporting the food that they consume. As an alternative to manual logging, we investigate methods for automatically recognizing foods based on their appearance. Unfortunately, the standard object recognition approaches based on aggregating statistics of descriptive local features perform poorly on this task [5] because food items are deformable and exhibit significant intra-class variations in appearance; the latter is the case even for relatively standardized items from fast food menus.

Our research is motivated by the observation that a food item can largely be characterized by its ingredients and their relative spatial relationships. For instance, sandwiches are often composed of a layer of meat surrounded on either side by slices of bread, while salads consist of assorted greens

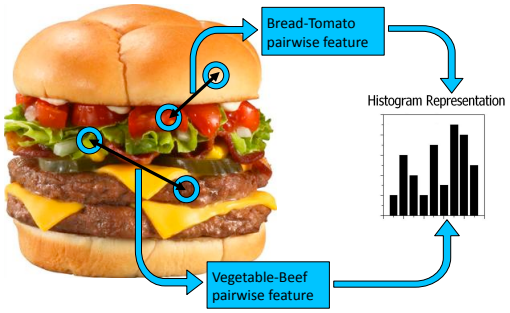


Figure 1. Exploiting spatial relationships between ingredients using pairwise feature statistics improves food recognition accuracy.

whose spatial layout can vary widely. Our hypothesis is that although detectors for any given ingredient are likely to be unreliable, one can still derive sufficient information by aggregating pairwise statistics about ingredient types and their spatial arrangement to reliably identify food items, both at the coarse (e.g., sandwich vs. salad) and fine (e.g., Big Mac vs. Baconator) levels. Fig. 1 illustrates how we can exploit the spatial relationship between ingredients to identify this item as a Double Bacon Burger from Burger King.

Although this hypothesis has intuitive appeal, applying it to food recognition is challenging for several reasons. First, even a simple food item can contain numerous visually distinct ingredients, not all of which may be visible in a given image due to occlusion or variations in assembly. In particular, we cannot rely on the presence of distinctive low-level features nor on reliable edges between ingredients. Second, although perfect accuracy is not required at the ingredient level, the proposed method does require the pixel-level segmentation to generate distributions over ingredients from very small local patches. Finally, there can be significant intra-class variation in the observed sizes and spatial relationships between ingredients, requiring us to build representations that can cope with this appearance variability in a principled manner.

Our proposed approach is illustrated in Fig. 2 and can be

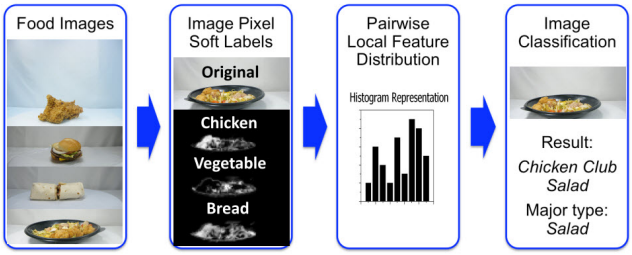


Figure 2. Framework of proposed approach: (1) Each pixel in the food image is assigned a vector representing the probability with which the pixel belongs to each of nine food ingredient categories, using STF [16]. (2) The image pixels and their soft labels are used to extract statistics of pairwise local features, to form a multi-dimensional histogram. (3) This histogram is passed into a multi-class SVM to classify the given image.

summarized as follows. First, we assign a soft label (distribution over ingredients) to each pixel in the image using a semantic texton forest [16]. For instance, the distribution corresponding to a pixel in a red patch could have a high likelihood for "tomato" and low for "cheese" while another pixel might have high values for "bread" and "cheese" but not "tomato". Next, we construct a multi-dimensional histogram feature where each bin corresponds to a pair of ingredient labels and discretized geometric relationships between two pixels. For example, the likelihood of observing a pair of "bread" and "tomato" labels that are 20–30 pixels apart at an angle of 40–50 degrees. Since we use soft labels, each observation from a pair of pixels contributes probability mass to a number of bins. Thus, the histogram aggregated from many samples of pixel pairs in the image serves as a representation of the spatial distribution of ingredients for that image. Finally, we treat this histogram as a feature vector in a discriminative classifier to recognize the food item. Our representation is not specific to food and could also be suitable for recognizing objects or scenes that are composed of visually distinctive "ingredients" arranged in predictable spatial configurations.

The remainder of this paper is organized as follows. Section 2 discusses the related work. Section 3 details our proposed approach. Section 4 describes our experimental methodology. Section 5 presents food recognition results on the PFID [5] dataset. Finally, Section 6 presents our conclusions and proposes directions for future work in this area.

2. Related Work

There has been relatively little work on the problem of food recognition. Shroff *et al*. [17] proposed a wearable computing system to recognize food for calorie monitoring. Bolle *et al*. [2] developed an automatic produce ID system called VeggieVision to help with the produce checkout process. Yang *et al*. [5] introduced the PFID dataset for food recognition along with benchmarks using two baseline algorithms: color histogram and bag of SIFT [11] features. Russo *et al*. [14] monitored the production of fast food using video. Wu and Yang [20] analyzed eating videos to recognize food items and estimate caloric intake.

Object category recognition has been a popular research area in computer vision for many years (see [6] for a comprehensive survey). These include approaches based on local features such as the SIFT [11] descriptor or global features such as color histograms or GIST [12] features; images are typically represented as "bags" of these features without explicit spatial information. Other approaches, such as the "constellation model" [3] and its numerous extensions, model objects as a collection of parts with a predictable spatial arrangement.

Of particular interest to food recognition are methods that explicitly address deformable objects. Shape context [1] picks n pixels from the contours of a shape, obtains $n-1$ vectors by connecting a pixel to the others, and uses these vectors as a description of shape at the pixel. Felzenszwalb proposes techniques [7] to represent a deformable shape using triangulated polygons. Leordeanu *et al*. [10] proposed an approach that recognizes category using pairwise interaction of simple features. A recent work [8] learns a mean shape of the object class based on the thin plate spline parametrization.

These approaches work well for many problems in object recognition, but two issues make them hard to apply to foods. First, these approaches all require detecting meaningful feature points, such as edges, contours, key points or landmarks. But precise features like these are typically not available in images of food. Second, shape similarity in food images is hard to exploit with these approaches, since the shape of real foods is often quite amorphous.

To overcome these challenges, our approach does not rely on the detection of features like edge or keypoints. Instead, we use local, statistical features defined over randomly selected pairs of pixels. The statistical distribution of pairwise local features effectively captures important shape characteristics and spatial relationships between food ingredients, thereby facilitating more accurate recognition.

3. Pairwise local feature distribution (PFD)

In this section, we give the definition of our image representation — the statistics of pairwise local features, namely pairwise feature distribution (PFD). Using this representation, each image can be represented by a multi-dimensional histogram.

3.1. Soft labeling of pixels

Before obtaining the statistics of local features, we classify all image pixels into several categories based on the appearance of the image patch around the pixels. The local features will later be gathered separately based on different pixels categories. We use nine handpicked categories representing eight common fast food ingredients: beef, chicken, pork, bread, vegetable, tomato/tomato sauce, cheese/butter, egg/other, plus a category for the background.

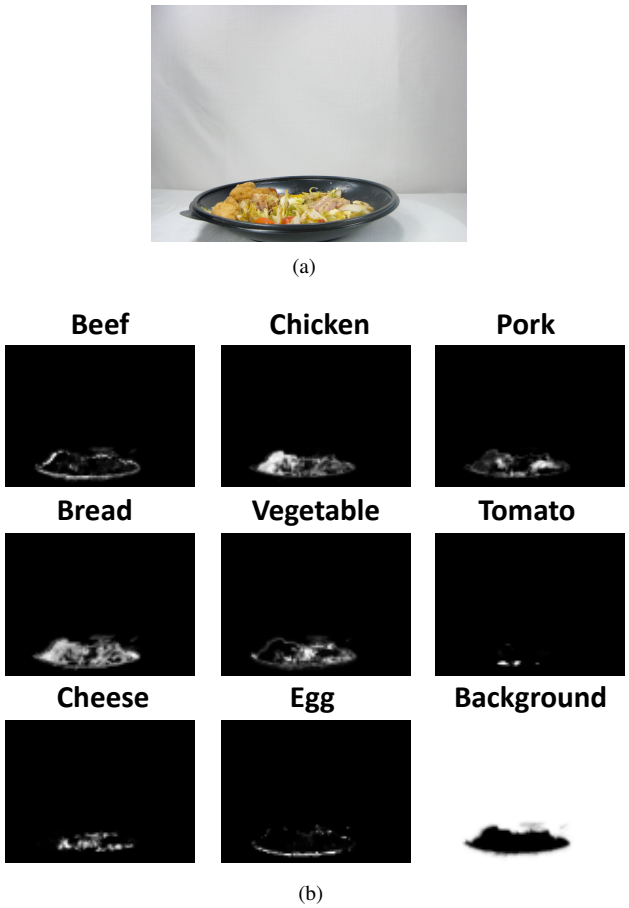
Rather than labeling every pixel as a member of a single category, we use soft labeling to create a vector of probabilities corresponding to the likelihood that the given pixel belongs to each of the nine categories. Soft labeling defers commitment about the identity of a pixel, allowing for uncertainty about the true label for a pixel, and for the fuzzy distinction between ingredient categories.

Many methods could be used for pixel classification. We chose to employ the Semantic Texton Forest (STF) [16]. STF is a method for image categorization and segmentation that generates soft labels for a pixel based on local, low-level characteristics (such as the colors of nearby pixels). This is achieved using ensembles of decision trees that are each trained on subsets of manually-segmented images. A pixel is classified by descending each tree from root to leaf based on low-level features computed around the given pixel. The leaves contain probabilistic class distributions (over ingredients) that our algorithm uses to compute its soft labels. We chose to use STF because it incorporates human knowledge in the form of manually labeled training images, and because it has proven effective for soft labeling in other challenging domains like the PASCAL Visual Object Classes (VOC) task.

With STF, each pixel is assigned a vector of K probabilities, where K is the number of pixel categories. $K = 9$ in our case. Fig. 3 is a visualization of the soft labeling of an image.

3.2. Global Ingredient Representation (GIR)

The simplest way to use the soft pixel labels produced by STF to represent a food image is to create a single one-dimensional histogram with 8 bins representing how frequently each of the 8 food ingredient categories appear in the image. To create an overall histogram representing the distribution of ingredients for the image, we sum up the soft labels for the 8 food ingredients for all the non-background pixels in an image. Then we normalize the histogram by the number of non-background pixels in the image. This global ingredient representation (GIR) is intuitive and easy to compute, but it does not capture the spatial relationships between ingredients that are important for distinguishing one food from another. It will serve as a useful benchmark for comparison with the more sophisticated, pairwise local features described below.



(a)

(b)

Figure 3. Visualization of soft labeling: (a) original food image. In (b), each of the nine images represents the probability that a given pixel belongs to that category. The brighter the pixel is, the larger its probability.

3.3. Pairwise Features

We attempt to capture the spatial relationship between pixels of different food ingredients using pairwise local features. Fig. 4 is an explanation of the four pairwise features we have explored.

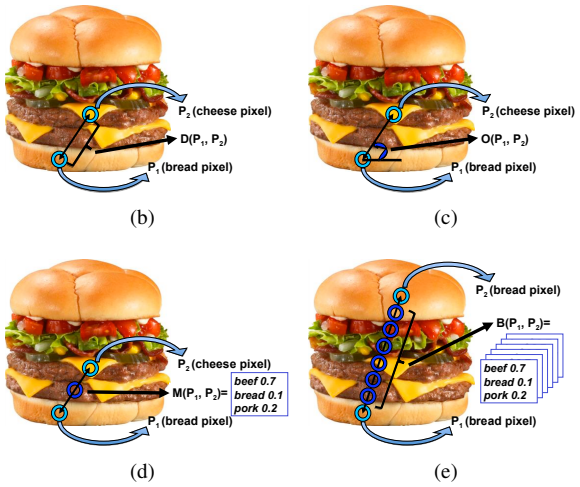
Pairwise distance reflects the distance between two pixels in an image. Similar to the feature descriptors in the Shape Context method [1], the pairwise distance feature is defined to be greater when the two pixels in the pair are close to each other. We employ Sturges' formula [19], $k = \log_2\lceil n + 1\rceil$ and use the log of the distance between the pair of pixels P_1 and P_2.

$$D(P_1, P_2) = \log\lceil |P_1, P_2| + 1\rceil \quad (1)$$

Pairwise orientation, $O(P_1, P_2)$, is defined as the angle of the line connecting the pixel pair. It ranges in $[0^o, 360^o)$,

Pairwise Feature	Feature Description	Expression	Fig.
Distance	The distance between two pixels P_1 and P_2	$D(P_1, P_2)$	(b)
Orientation	The orientation of the line between two pixels P_1 and P_2	$O(P_1, P_2)$	(c)
Midpoint category	The type of the pixel in the middle of two pixels P_1 and P_2	$M(P_1, P_2)$	(d)
Between-pair category	The type of all pixels between two pixels P_1 and P_2	$B(P_1, P_2)$	(e)

(a)



(b) (c) (d) (e)

Figure 4. Four pairwise local features. In the table in (a), column 1 is the name of the feature; column 2 is a description of the feature; column 3 shows the expression of the feature in the following text; column 4 refers to their illustration figures. (b) (c) (d) (e) are illustrations of these pairwise local features.

and positive is counter clockwise.

Pairwise midpoint, $M(P_1, P_2)$, is a discrete feature defined as the soft label of the pixel at the midpoint of the line connecting the two pixels. When this feature is added to the multi-dimensional histogram, the midpoint pixel's soft label (distribution over ingredient types) determines the weight of its contribution to the relevant bins.

Like the pairwise midpoint feature, the between-pair feature captures information about the pixel labels lying on a the line connecting the two pixels in the pair. But rather than picking only the soft label of the midpoint, the between-pair feature incorporates all pixel soft labels along the line connecting the pair of pixels. So the feature for each pixel pair has t discrete values, t being the number of pixels exist along the line between a pair of pixels. We use $B(P_1, P_2) = \{T_i | i = 1, 2, ..., t\}$ to represent the feature set for pixels P_1 and P_2.

In addition to these individual features, we also explore joint features that are composites of the above features, such as a joint feature of distance and orientation, $DO(P_1, P_2)$, a joint feature of orientation and midpoint, $OM(P_1, P_2)$, as shown in Fig. 5.

Joint Pairwise Feature	Feature Description	Expression
Distance and Orientation	The conjunction of distance and orientation feature between pixels P_1 and P_2	$DO(P_1, P_2)$
Orientation and Midpoint category	The conjunction of orientation and midpoint category feature between pixels P_1 and P_2	$OM(P_1, P_2)$

Figure 5. Joint pairwise features

3.4. Histogram representation for pairwise feature distribution

Computing the complete set of pairwise features in an image can be computationally expensive since, for an image with M pixels, we would need to consider $\binom{M}{2}$) pairs. It suffices to consider $\binom{M}{2}$ rather than M^2 pairs because our pairwise relationships are symmetric; we convert directed orientation to a symmetric form by averaging counts in both directions. Nonetheless, $\binom{M}{2}$ can still be prohibitively large for high-resolution images.

Thus, for computational efficiency, we randomly sample N ($N = 1000$) pixels from the non-background portion of the image and estimate pairwise statistics using these $\binom{N}{2}$ pixel pairs. We use a set $\mathcal{P}$ to represent the N pixels we randomly pick from an image: $\mathcal{P} = \{P_1, P_2, ..., P_N\}$. The soft labels of the N pixels are represented as $\mathcal{S} = \{S_{ik} | i = 1, 2, ..., N;\ k = 1, 2, ..., 8\}$. We calculate pairwise local features ($D(P_i, P_j)$, $O(P_i, P_j)$, $M(P_i, P_j)$ and $B(P_i, P_j)$) for all pixels, and accumulate their values into a distribution. The distribution is weighed by the soft labels of the two pixels. Specifically, this weight is the product of the probabilities that a pixel is assigned to the ingredient label corresponding to the given bin.

We use a multi-dimensional histogram to represent the distribution of pairwise local features, as detailed in Fig. 6. The first two dimensions of the histogram are the ingredient labels. The other dimensions are the weighted pairwise features calculated for the given pair of pixels.

The first and second dimensions of the above multi-dimension histograms have 8 bins, representing the eight pixel categories (excluding the background). The other dimensions have either 12 bins for pairwise distance or orientation, or 8 bins for midpoint or between-pair category.

3.5. Histogram normalization

We normalize the distribution of certain pairwise features to achieve invariance to transformations such as ro-

Histogram Type	Histogram Description	Size
Distance	label × label × distance	8×8×12
Orientation	label × label × orientation	8×8×12
Midpoint category	label × label × midpoint	8×8×8
Between-pair category	label × label × between-pair	8×8×8
Distance and Orientation	label × label × distance × orientation	8×8×12×12
Orientation and Midpoint category	label × label × orientation × midpoint	8×8×12×8

Figure 6. Histogram representation of PFD

tation and scaling. We compute the mode of pairwise distance or orientation, and shift the corresponding dimension of the multi-dimension histogram so that it is centered on the mode. Since we apply a logarithm to the distance feature, changing the scale of an image becomes equivalent to shifting its histogram in the distance dimension, resulting in scale invariance. Similarly, centering the orientation histogram at its mode value results in rotation invariance. Since we only measure relative distances between pixels, our representation is also translation invariant.

For joint features, we normalize the joint histogram by centering it at the mode value of the marginal distributions of each of its continuous dimensions.

3.6. Classification with local feature distributions

Using PFD or Global Ingredient Representation, each image is represented as a multi-dimension histogram. We employ a Support Vector Machine (SVM) for classification using a χ^2 kernel [15].

The symmetrized approximation of χ^2 is defined as:

$$d_{\chi^2}(x,y) = \sum_i \frac{(x_i - y_i)^2}{x_i + y_i}, \tag{2}$$

where x and y represent the PFD histograms of two images, and x_i and y_i are two bins of the histograms.

The χ^2 kernel is defined as:

$$k_{\chi^2}(x,y) = e^{-d_{\chi^2}(x,y)}. \tag{3}$$

With this pre-computed kernel matrix, we apply the SVM to classify images into multiple food categories. In our implementation, we use the libSVM package [4].

4. Experimental Methodology

In this section, we briefly review our dataset, baseline approaches, and detail the implementation of the pixel-level segmentation using which we generate our ingredient soft labels. Experimental results are given in Section 5.

4.1. Dataset

We evaluate our work on the recently-released Pittsburgh Food Image Dataset (PFID) [5] and compare our proposed approach against their two baseline methods. The PFID dataset is a collection of fast food images and videos from 13 chain restaurants acquired under lab and realistic settings. Our experiments focus on the set of 61 categories of specific food items (e.g., McDonald's Big Mac) with masked background. Each food category contains three different instances of the food (bought on different days from different branches of the restaurant chain), and six images from six viewpoints (60 degrees apart) of each food instance.

We follow the experimental protocol proposed by Chen *et al*. [5] and perform 3-fold cross-validation for our experiments, using the 12 images from two instances for training and the 6 images from the third for testing. We repeat this procedure three times, with a different instance serving as the test set and average the results. The protocol ensures that no image of any given food item ever appears in both the training and test sets, and guarantees that food items were acquired from different restaurants on different days.

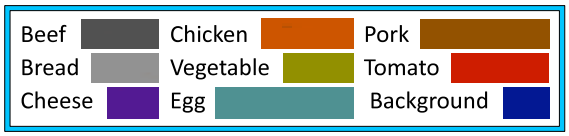
4.2. Baseline approaches

We use the two standard baseline algorithms specified by PFID: color histogram + SVM and bag of SIFT [11] features + SVM, as briefly described below.

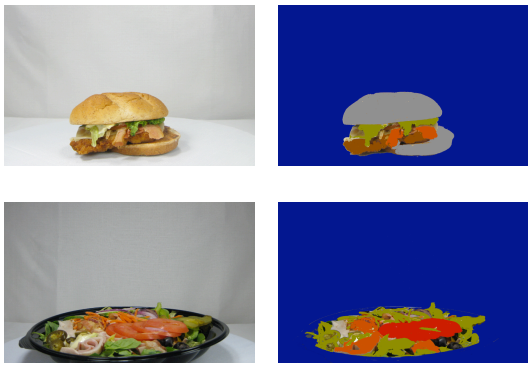
Due in part to its simplicity, the color histogram method has been popular in object recognition for more than a decade. We employ a standard RGB 3-dimensional histogram with four quantization levels per color band. Each pixel in the image is mapped to its closest cell in the histogram to generate a $4^3 = 64$ dimensional representation for each image. Then we use a multi-class support vector machine (SVM) implementation [4] for classification.

Several studies (e.g., [9]) have demonstrated the merits of using a bag of SIFT features. The basic idea of this approach is to represent each image as a histogram of occurrence frequencies defined over a discrete vocabulary of features (pre-generated using k-means clustering) and then to use the histogram as a multi-dimensional vector in an SVM.

To compare against these baseline approaches, we conduct experiments using the six pairwise local features and the GIR feature proposed in Section 3. We describe the experimental details below.



(a)



(b)

Figure 7. (a) shows nine colors that are used to label training images for STF. Each color represents one of the food ingredient types or the background. (b) shows two samples of manually labeled images for STF: use the nine labels in (a) to color all pixels of the 16 training images for STF. (Best viewed in color.)

4.3. Pre-processing with STF

The initial step in generating the six pairwise local features is to obtain pixel-wise soft labels for the images. We use Semantic Texton Forests (STF) [16] to generate these as follows. First, we train STF using 16 manually-segmented food images, two examples of which are shown in Fig. 7(b). We found 16 training images to be sufficient to cover the appearance of the food ingredients in the PFID dataset. While more data could potentially generate better STF ingredient labels, using a small training set shows that our algorithm can operate with a small amount of relatively noisy training data.

After training, we employ STF to soft label all $61 \times 6 \times 3$ food images in the dataset, creating a 9-element probability vector for each pixel, representing the likelihood of it belonging to each of the eight food ingredient types or the background.

The output of STF is far from a precise parsing of an image. Fortunately, our method does not require a perfect segmentation of food ingredients nor precise soft labeling results.

5. Results

This section summarizes our results, both at the individual food item level (61 categories) and at the major food type level (7 broad categories).

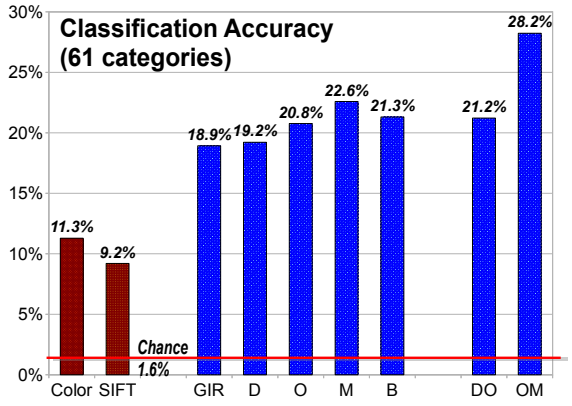


Figure 8. Classification accuracy for 61 categories

5.1. Classification accuracy on the 61 categories

Fig. 8 summarizes the classification accuracy results on the 61 categories for the two baselines (color histogram, bag of SIFT), the global ingredient representation (GIR), and the six pairwise local features.

The chance recognition rate for each of the 61 categories is below 2% $(1/61)$. The two baselines (color histogram and bag of SIFT features) reach only about 10%. The GIR global ingredient histogram method achieves 19%, and the local pairwise features methods range from 19% to 28%. The latter, achieved with the OM feature, is more than twice the accuracy of the PFID algorithms [5] and nearly $20\times$ chance.

Fig. 9 shows the confusion matrices for four approaches: color histogram, bag of SIFT features, GIR, and pairwise feature OM. The darker the diagonal line, the more effective an approach, because it has a higher probability of classifying food of a given category as itself. Clearly, there is a more salient diagonal line in the matrix of the OM feature than the matrices for the other three approaches, which indicates that a greater number of food items are correctly classified as themselves using joint feature OM. In both of the baseline approaches, we can see straight vertical lines in their confusion matrices, showing that certain food categories are frequently (and incorrectly) chosen as the classification result; GIR and OM do not exhibit this problem.

There is a straightforward explanation for why the combination of orientation and midpoint is the higher-order feature that gives the best accuracy. This pair of features is able to leverage the vertically-layered structure of many fast foods. This vertical layering is particularly characteristic of burgers and sandwiches where meat and vegetables are sit-

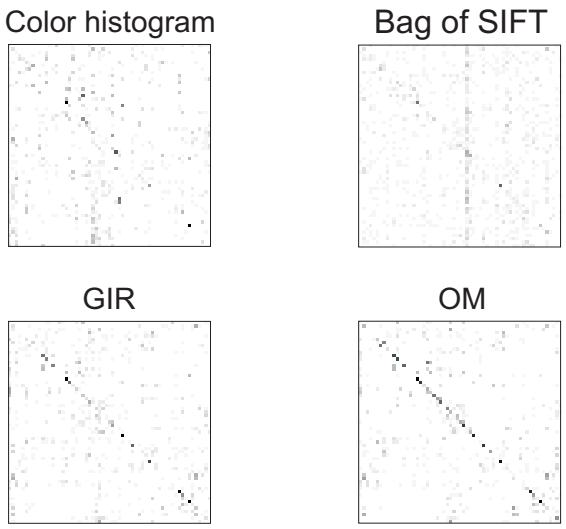


Figure 9. Confusion matrix comparison for 61 categories. Rows represent the 61 categories of food, and columns represent the ground truth categories.

uated between two (or more) horizontally oriented slices of bread. The orientation feature relates only two endpoints and therefore by itself cannot capture the three-way relationships like bread-meat-bread or bread-vegetable-bread that are characteristic of burgers and sandwiches. The midpoint feature captures three-way relationships, but without the orientation constraint, the statistics of the midpoint feature will be dominated by uniform triples like bread-bread-bread and meat-meat-meat created from horizontal samples, which form the majority. By combining the orientation and midpoint features, the algorithm is able to discover and leverage the vertical triplets that are key to accurately discriminating burgers and sandwiches from other food classes (e.g., pizza), as well as distinguishing between individual burgers and sandwiches (e.g., a Big Mac vs. a Quarter Pounder).

5.2. Classification accuracy into 7 major food types

The relatively low accuracy of even the best method described above is due in part to the fact that many foods with similar appearances and similar ingredients are assigned to different categories in the PFID food database, as shown in Fig. 10. Such cases are challenging, even for humans, to distinguish.

In order to match more closely to the way people categorize foods, we organize the 61 PFID food categories into seven major groups — sandwiches, salads/sides, chicken, breads/pastries, donuts, bagels, and tacos. Then, we use the baseline methods and our approach to classify images in the dataset into these categories. Fig. 11 shows the classification results.

Figure 10. Similar food items in PFID: these two foods are semantically and visually similar, but are distinct food items and thus appear in different PFID categories. Such subtle distinctions make fast food recognition inherently challenging for vision algorithms. (left) Arby's Roast Turkey and Swiss; (right) Arby's Roast Turkey Ranch Sandwich.

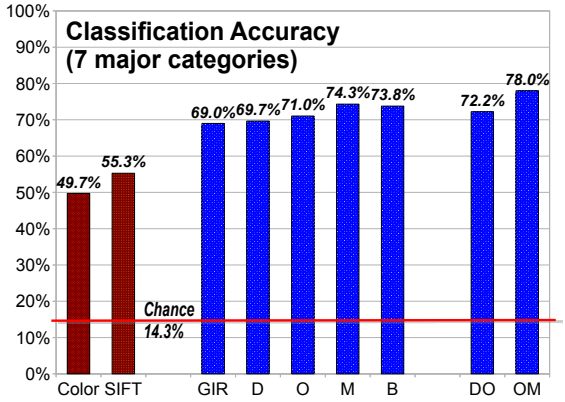


Figure 11. Classification accuracy for 7 major types

The rankings between algorithms observed in the previous experiment continue to hold true in this broader categorization task, with OM achieving nearly 80% accuracy. More interestingly, the category-level confusion matrices (Fig. 12) provide some insights into the classification errors. We see that sandwiches are often confused with foods whose major ingredients include bread, such as donuts and bagels.

6. Conclusion

Food recognition is a new but growing area of exploration. Although many of the standard techniques developed for object recognition are ill-suited to this problem, we argue that exploiting the spatial characteristics of food, in combination with statistical methods for pixel-level image labeling will enable us to develop practical systems for food recognition.

Our experiments on a recent publicly-released dataset of fast food images demonstrate that our proposed method significantly outperforms the baseline bag-of-features models

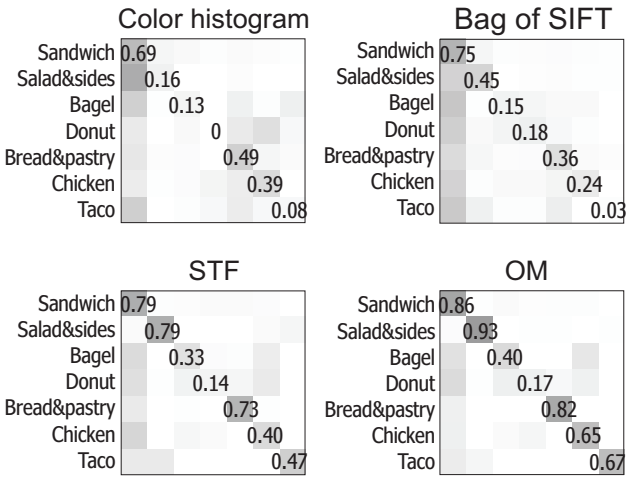


Figure 12. Confusion matrix comparison for 7 major categories. Rows represent the major 7 food categories, and columns represent the ground truth major categories.

based on SIFT or color histograms, particularly when we augment pixel-level features with shape statistics computed on pairwise features.

In future work, we plan to extend our method to: (1) perform food detection and spatial localization in addition to whole-image recognition, (2) handle cluttered images containing several foods and non-food items, (3) develop practical food recognition applications, and (4) explore how the proposed method generalizes to other recognition domains. Preliminary experiments indicate that pairwise statistics of STF features in conjunction with Gist [13] perform significantly better than Gist+STF+SVM alone on scene recognition tasks.

Acknowledgments

We would like to thank Lei Yang for providing SIFT+SVM baseline code and the PFID project for help with the food recognition dataset.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape context: a new descriptor for shape matching and object recognition. In *NIPS*, 2000. 2, 3

[2] R. Bolle, J. Connell, N. Haas, R. Mohan, and G. Taubin. VeggieVision: A produce recognition system. In *WACV*, 1996. 2

[3] M. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *ECCV*, 1998. 2

[4] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. 5

[5] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and J. Yang. PFID: Pittsburgh fast-food image dataset. *Proceedings of International Conference on Image Processing*, 2009. 1, 2, 5, 6

[6] S. Dickinson. The evolution of object categorization and the challenge of image abstraction. In *Object Categorization: Computer and Human Vision Perspectives*. Cambridge University Press, 2009. 2

[7] P. F. Felzenszwalb. Representation and detection of deformable shapes. *PAMI*, 27(2), 2005. 2

[8] T. Jiang, F. Jurie, and C. Schmid. Learning shape prior models for object matching. In *CVPR*, 2009. 2

[9] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. *Proc. Workshop on Statistical Learning Computer Vision*, 2004. 5

[10] M. Leordeanu, M. Hebert, and R. Sukthankar. Beyond local appearance: Category recognition from pairwise interactions of simple features. In *CVPR*, 2007. 2

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2), 2004. 2, 5

[12] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42(3), 2001. 2

[13] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42(3), 2001. 8

[14] R. Russo, N. da Vitoria Lobo, and M. Shah. A computer vision system for monitoring production of fast food. In *Proceedings of Asian Conference on Computer Vision*, 2002. 2

[15] B. Schiele and J. L. Crowley. Object recognition using multidimensional receptive field histograms. In *ECCV*, 1996. 5

[16] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, 2008. 2, 3, 6

[17] G. Shroff, A. Smailagic, and D. Siewiorek. Wearable context-aware food recognition for calorie monitoring. In *Proceedings of International Symposium on Wearable Computing*, 2008. 2

[18] R. Spector. Science and pseudoscience in adult nutrition research and practice. *Skeptical Inquirer*, 33, 2009. 1

[19] H. A. Sturges. The choice of a class interval. *J. American Statistical Association*, 1926. 3

[20] W. Wu and J. Yang. Fast food recognition from videos of eating for calorie estimation. In *Proceedings of International Conference on Multimedia and Expo*, 2009. 2