COMP 991: Research & Reading

Fall 2013

Under the supervision of

Prof. Jan-Michael Frahm

University of North Carolina, Chapel Hill

“Food Detection in Images”

Submitted by:

Sahil Narang

Contents

Introduction

The goal of this project is to develop a robust individual specific travel prediction model which can be used to augment image geolocation. Estimating geographic information from an image is an excellent, difficult high-level computer vision problem with applicability in a wide array of disciplines. Most work [3][4] in this area has focused on a purely data driven approach i.e. matching the given scene against a huge geotagged image database. On the other hand, there has been very little research in incorporating individual travel patterns and temporal cues. Hence, the goal of this project is to develop a robust travel model which can serve to provide more accurate geolocation. Recently, Guerzhoy et. al. [2] put forth the concept of learning latent factor models of human travel. Inspired by this approach, I wish to explore latent factors which could potentially improve travel prediction.

Intuitively, one can imagine that the socioeconomic status of an individual would play a key role in determining the individual's travel propensity. However, due to the unavailability of such a dataset, determining the socioeconomic status of an individual is a non-trivial task. For example, Kalogerakis et. al. [4] and Guerzhoy et. al. [2] used the publicly-available Flickr.com image streams of individuals to build their travel models. In other words, my task is reduced to learning the socioeconomic status of an individual from his/her uploaded photographs. Given the mammoth scope of this project, I have broken down the goals into two categories: Long Term & Short term. These are:

Long term Goals

1. Determine the socioeconomic status of an individual on the basis of the type of restaurants he or she frequents. For example, an individual who regularly frequents expensive restaurants is likely to be in a higher socioeconomic bracket than one who eats at fast food joints.
2. Determine whether or not the individual has children. An individual who has children is more likely to travel to family oriented destinations.
3. Learn season specific latent factors. For example, destinations in the south are likely to be more popular in the winters than ones in the north.
4. Formulate and investigate other latent factors that may potentially contribute to an individual's travel habits.
5. Build a robust travel model based on the above mentioned latent factors and integrate it with data driven image geolocation.

Short term Goals

1. In order to determine an individual's socioeconomic status on the basis of his dining history, one first needs to determine if the individual actually 'dined' at a restaurant. Since our dataset comprises of images, this implies the need for accurate food detection in images.
2. In order to explore additional latent factors, we require a basic travel model which can be augmented with the factors mentioned above. Hence, one of the short term goals of this project is to develop a basic travel model [2] which incorporates two latent factors: distance between source location & destination, and the desirability of a destination.

For the rest of the report, I will focus on food detection. Food detection, by itself, is a non-trivial computer vision task. It is worth noting the distinction between food recognition and food detection. Food recognition refers to the problem of detecting and determining the type of food i.e. it is a multi-class classification problem. On the other hand, food detection refers to the problem of simply detecting food i.e. it is a binary classification problem.

Food Detection

In recent years, the number of camera users has increased because digital cameras are very popular consumer products. Since a human face is the most popular subject captured with the consumer digital

cameras, techniques for face detection have been extensively developed. As well as human faces, food is also targeted when recording food logs or memorizing special moments of travel. Additionally, there are some applications which make food appearance more delicious in a photo. For these reasons, food is the subject frequently taken in everyday life.

Food detection and recognition is an extremely difficult task since food items are deformable and exhibit significant intra-class variations in appearance. This might be why there has been relatively very little work in this domain. Most of the work has focused on automatic food image recognition with significant constraints. Such systems aim to classify unknown food images and estimate volume, nutrition and so on. Using these systems, people who often take a photo of food can easily analyze their photos, and people can easily record everyday food from photos. However, these methods impose severe constraints. In these researches, food is located in the entire screen or food region is estimated based on circle detection. Images such as those containing several plates need to be preprocessed wherein the regions of food are manually cut out. Moreover, if the plate does not have circle shape, plate detection based on circle detection will fail.

In case of food recognition, Joutou et al. [6] proposed a food image recognition system for 50 kinds of food images. They extract Bag-of-Features (BOF), color histogram and Gabor texture feature, and then apply Multiple Kernel Learning (MKL) method to those features in order to classify query images into 50 categories. This work was extended by Hoashi et. al. [13] who incorporated more categories and increased the number of features to 17. Chen et. al. [11] introduced the first visual dataset of fast foods with the aim of being used dietary assessment. The data comprises of 101 food types obtained from 11 popular fast food chains. They further benchmark the data using color histogram and bag of SIFT features in conjunction with a discriminative classifier. Yang et al. [8] proposed a fast food image recognition system. They use pairwise statistics which represents geometric relationship such as distance and orientation between many pairs of local features. Puri et al. [7] proposed a food intake assessment system which recognizes food types and estimates volumes and nutrition information. They recognize food types by using color feature and texture feature, and estimate volumes by 3D reconstruction.

In case of food detection, Miyano et. al. [1] proposed a relatively simple but effective discriminative BoF based model which uses SURF features and Color Histogram. Furthermore, they incorporate a sliding window approach to detect regions of food and then combine their results using a pixel-by-pixel voting mechanism. Their dataset, however, is fairly small and is restricted to few food categories. Nie et. al. [9] proposed an ellipse detection method by splitting, filtering and grouping edges which can be used to automatically recognize dining plates. However, their work focuses on chronically recorded videos acquired by a wearable device which implies that the image is focused on the dining plate.

In this project, I chose to adopt the same approach as the one put forth by Miyano et. al. [1]. The reasoning behind this is the fact that the abovementioned complex recognition models are not necessary for detection. Furthermore, the hard constraints imposed by these models makes the objective of this project infeasible since we require accurate food detection in a variety of scenes.

Bag of Features Model (BoF)

The past decade has seen the rise of the Bag of Features [14] approach in computer vision. Bag of Features (BoF) methods have been applied to image classification, object detection, image retrieval, and even visual localization for robots. BoF approaches are characterized by the use of an orderless collection of image features. Lacking any structure or spatial information, it is perhaps surprising that

this choice of image representation would be powerful enough to match or exceed state-of-the-art performance in many of the applications to which it has been applied. Due to its simplicity and performance, the Bag of Features approach has become well-established in the field. At a high level, the procedure for generating a Bag of Features image representation is summarized as follows:

1. Build Vocabulary: Extract features from all images in a training set. Vector quantize, or cluster, these features into a "visual vocabulary," where each cluster represents a "visual word" or "term." In some works, the vocabulary is called the "visual codebook." Terms in the vocabulary are the codes in the codebook.
2. Assign Terms: Extract features from a novel image. Use Nearest Neighbors or a related strategy to assign the features to the closest terms in the vocabulary.
3. Generate Term Vector: Record the counts of each term that appears in the image to create a normalized histogram representing a "term vector." This term vector is the Bag of Features representation of the image.

There are a number of design choices involved at each step in the BoF representation. The following section briefly covers some of these design choices:

1. **Feature Detection & Representation**

   Feature detection is a separate process from feature representation in BoF models. Feature detection is the process of finding keypoints in the image. Feature descriptors can then be used to encode information from the pixels in the neighborhood of these keypoints. Feature detectors may be categorized as interest point detectors, visually salient detectors and grid/randomized sampling. Interest point detectors typically detect keypoints using scale space representations of images. Popular interest point detectors include Lowe's keypoint detector [21], Scale Saliency [16], Harris-Affine detector [17], MSER [18]. In the biometric computer vision literature, the interest point operator is based on computational models of visual attention. Examples of such detectors include Itti and Koch's salience model [19] and Bruce and Tsotsos's salience model [20]. The most popular feature descriptor is SIFT [15] which is essentially a histogram of responses to oriented oriented gradient filters. Other popular descriptors include SURF [22], OpponentSIFT [23], Gabor filter banks, image moments etc.

2. **Size of Vocabulary**

   A visual vocabulary is generated by clustering the detected keypoints in their feature space and treating each cluster as a unique visual word of the vocabulary. Different from text vocabulary in information retrieval, the size of visual vocabulary is determined by the number of keypoint clusters. A small vocabulary may lack the discriminative power since two keypoints may be assigned into the same cluster even if they are not similar to each other. A large vocabulary, on the other hand, is less generalizable, less for- giving to noises, and incurs extra processing overhead. The trade-off between discrimination and generalization motivates the studies of visual vocabulary size. A recent survey [14] shows that previous works used a wide range of vocabulary sizes, leading to difficulty in interpreting their findings. For instance, Lazebnik et al. [25] adopted 200-400 visual words, adopted 1000, Sivic et al. [24] adopted 6,000 -10,000, etc.

3. **Quantization and Distance Measures**

   Vector Quantization (Clustering) is used to build the visual vocabulary in Bag of Features algorithms. Nearest-neighbor assignments are used not only in the clustering of features but also in the comparison of term vectors for similarity ranking or classification. There are a great many clustering/vector quantization algorithms, and this report does not attempt to enumerate them. Most BoF implementations use Kmeans clustering algorithm [24,25,26].

   - *Term Weighting*

One of the earliest strategies for handling quantization issues at a gross level is to assign weights to the terms in the term vector. With term weights, one can penalize terms found to be too common to be discriminative and emphasize those that are more unique. This is the motivation behind the popular Term Frequency-Inverse Document Frequency (TF-IDF) weighting scheme used in text retrieval [27]. Term vectors can also be represented as binary strings. A 1 is assigned for any term that appears in the image, 0 otherwise.

- *Soft Weighting*
  A given feature may be nearly the same distance from two cluster centers, but with a typical "hard assignment" method, only the slightly nearer neighbor is selected to represent that feature in the term vector. Thus, the ambiguous features that lie near Voronoi boundaries are not well-represented by the visual vocabulary. To address this problem, researchers have explored multiple assignments and soft weighting strategies. Multiple assignment is where a single feature is matched to k nearest terms in the vocabulary. Soft weights are similar, but the k nearest terms are multiplied by a scaling function such that the nearest term gets more weight than the k'th nearest term. These strategies are designed to mitigate the negative impact when a large number of features in an image sit near a Voronoi boundary of two or more clusters [28][29].

4. **Kernels for BoF**
   Support Vector Machines (SVM) have been one of the most popular classifiers for BoF. The choice of a good kernel function is critical for statistical learning. Although there is a number of general purpose kernels off the shelf, it is unclear which one is the most effective for BoF in the context of visual classification. Jiang et. al. [30] compared the performance of the linear, RBF & histogram intersection kernel. They found that the generalized RBF kernels perform better than linear kernel and HI kernel with non-trivial margin. They attribute this to the fact that visual words are correlated to each other, and are not linearly separable. Among all the generalized RBF kernels, the X^2 RBF kernel, Laplace RBF kernel, and sub-linear RBF kernel consistently outperform the traditional Gaussian RBF kernel. This can be attributed to the responses of kernels to background variance. Ideally, a kernel should only emphasize regions containing the target concept, while tolerating the background variance without amplifying the effect.

Dataset

Generating a dataset is a tedious task, especially in case of food detection since there are innumerable categories. Chen et. al. [11] introduced the first visual dataset of fast foods with the aim of being used dietary assessment. Their data comprises of 101 food types obtained from 11 popular fast food chains. However, this dataset is not relevant to our work due to a number of reasons. Firstly, this project is not limited to detecting fast food, secondly we are more inclined towards detecting 'restaurant dishes' since that would be most helpful to assess the individual's socioeconomic background. Hence, I have only incorporated a very small subset of their dataset.

Presently, the dataset (Figure 1) comprises of 940 images downloaded from Flickr. In order to ensure their usability for training the model, only images that are centered on food and have a plain background are selected. For the sake of simplicity and as an initial test, most of these are images of pizza.

Figure 1: Images from the Positive Training set

Figure 2: Images from the Negative training set

Implementation

1. **Keypoint detector and descriptor**

   Given their robustness and proven superior performance, I used SIFT features. Essentially, the keypoints are found using DoG. In DoG, the input image is successively smoothed with a Gaussian kernel and sampled. The DoG representation is then obtained by subtracting two successive smoothed images. Thus, all the DoG levels are constructed by combined smoothing and sub-sampling. The DoG is an approximate but more efficient version of LoG. Once the keypoint is found, a 128 dimensional feature vector is generated that captures the spatial structure and the local orientation distribution of a region surrounding a keypoint. I used the VLFeat library [31] to generate these features. Their algorithm uses two parameters:

   - peak_thresh: The *peak threshold* filters peaks of the DoG scale space that are too small. This was set to 0.5.
   - edge_thresh: The *edge threshold* eliminates peaks of the DoG scale space whose curvature is too small. This was set to 2.

2. **Visual Vocabulary**

   196 images were selected to generate the codebook. This dataset comprised of 98 randomly selected food images and 98 randomly selected images from the im2gps dataset [5]. I experimented with various vocabulary sizes (100, 300, 500) and decided to settle with 500 as it provided the best performance. This seems to be in conjunction with the experiments performed by Jiang et. al. [25]. This size, however, may not be optimal for a larger dataset.

3. **Training**

   The 940 food images comprised the positive training set and 256 images (Figure 2) randomly selected from the im2gps dataset served as the negative training set. Each keypoint was 'hard' assigned to its 'nearest neighbor' cluster using Euclidian distance as a metric. I then applied the TF-IDF technique to normalize the training set.

   TF-IDF is defined as: $tf_i.\log(N/N_i)$, where tfi is the term frequency of the i'th word, N is the number of documents (images) in the database, and Ni is the number of documents in the database containing the i'th word. The log term is called the inverse document frequency and it serves to penalize the weights of common terms. A SVM classifer with a RBF kernel was then trained using the LibSVM library [10]. Cross validation was performed to determine the optimum gamma and cost parameters in order to generalize the classifer.

4. **Detection**

   In the detection phase, I used a sliding window approach. The window size was set to be 200x200 pixels with a step size of 20 pixels. For each window, a 500d feature vector was generated using the same method as in the training phase. The SVM classifer was then used to determine whether or not the window contains food, henceforth called 'food subregion'. Windows with food classification probability less than a certain threshold (0.3) were filtered out i.e. these windows are no longer considered to have food.

5. **Connection of Food Subregions**

   All pixels included in the food subregions vote as show in Fig. 3(a). If the voted count of each pixel is larger than a pre-defined threshold (5), that pixel is regarded as inside the food region, which is labeled as shown in Fig. 3(b). Finally, as shown in Fig. 3(c), a bounding box is drawn to encapsulate those pixels.

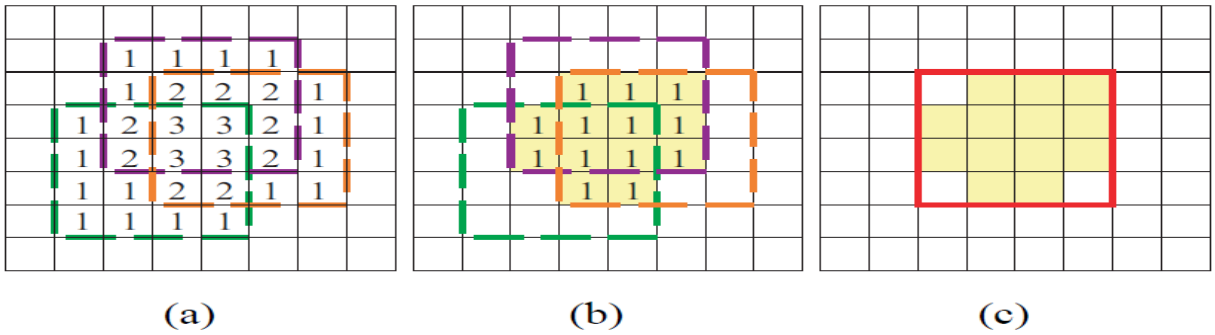

Figure 3: Connecting Food Subregions

Results

The test set comprised of 96 food images, predominantly restaurant dishes, and 109 random non-food images. The results are as follows:

Classification Results	Number of Images
True	109
False Positives	40
False Negatives	16

Hence, accuracy of the classifer can be computed as:

$$\text{Accuracy} = 149/205 = \mathbf{72.68\%}$$

Figure 4: Examples of incorrect classification

Figure 5: Examples of correct classification

Conclusion

The classifer performed surprisingly well on a wide array of food types even though the training data was heavily biased i.e. most of the positive examples were images of pizza. Hence, this approach seems quite promising. The BoF model requires a number of design decisions and given the time constraints, I was unable to comprehensively experiment with all of these factors. This, therefore, implies that there is a lot of scope for improvement.

Current & Future Work

As discussed in the introduction, food detection is just a first step towards the overall goal of the project. Therefore, I now plan on working on the aforementioned long term goals. However, with regards to food detection, I suggest the following avenues for future work:

1. Expand the database to include more food categories.
2. Experiment with different kernels, vocabulary sizes, feature detectors, weighting mechanism etc.
3. Include color feature alongside the texture based features. Recent works have shown the drastic improvement in classification accuracy due to the inclusion of color features.
4. Investigate the use of a one-class SVM instead of the two-class SVM. I did experiment with the one-class SVM but unsuccessfully so. The advantage of a one class SVM is that they don't require a negative training set. This makes the process easier since this problem requires a 'one-vs-all' classifer i.e. there is no definitive negative data.
5. Investigate the use of a part based model i.e. a hierarchy of classifiers. Some ingredients, such as rice, noodles, bread, have a distinct texture and can be used as common visual cues. Hence, one could train a classifier for different ingredients and then combine the results to hopefully, get a more accurate model.

References

[1] Miyano, R., Uematsu, Y., and Saito, H. Food Region Detection using Bag-of-features Representation and Color Feature. Ruiko Miyano, Yuko Uematsu, and Hideo Saito. VISAPP 1, page 709-713.

[2] Guerzhoy, M., and Hertzmann, A. (2012) Learning latent factor models of human travel. In NIPS Workshop on Social Network and Social Media Analysis: Methods, Models and Applications.

[3] Lin, T., Belongie, S., and Hays, J. (2013). Cross-View Image Geolocation. In proc. CVPR

[4] Kalogerakis, E., Vesselova, O., Hays, J., Efros, A. (2009). Image Sequence Geolocation with Human Travel Priors. In Proc. ICCV

[5] Hays., J., and Efros, A. (2008). IM2GPS: Estimating Geographic Information from a single image. In Proc. CVPR

[6] Joutou, T. and Yanai, K. (2009). A food image recognition system withmultiple kernel learning. In Proc. of ICIP.

[7] Puri,M., Zhu, Z., Yu, Q., Divakaran, A., and Sawhney, H. S. (2009). Recognition and volume estimation of food intake using a mobile device. In Proc. ofWACV, pages 1–8.

[8] Yang, S., Chen, M., Pomerleau, D., and Sukthankar, R.(2010). Food recognition using statistics of pairwise local features. In Proc. of CVPR, pages 2249–2256.

[9] J. Nie, Z. Wei, W. Jia, L. Li, J. D. Fernstrom, R. J. Sclabassi, M. Sun, "Automatic detection of dining plates for image-based dietary evaluation," In Processings of 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), , pp.4312-4315, Aug. 31 2010-Sept. 4, 2010, Buenos Aires, Argentina.

[10] C.-C. Chang and C.-J. Lin, LIBSVM: a library for support vector machines, 2001. http://www.csie.ntu.edu.tw /~cjlin/libsvm/

[11] Pfid: Pittsburgh fast-food image dataset - Mei Chen, Kapil Dhingra, Wen Wu, Lei Yang, Rahul Sukthankar, Jie Yang- 2010

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

[13] H. Hoashi, T. Joutou, and K. Yanai, "Image recognition of 85 food categories by feature fusion," in Proc. of International Symposium on Multimedia, pp. 296-301, 2010.

[14] S. O'Hara and B.A. Draper, "Introduction to the bag of features paradigm for image classification and retrieval," Arxiv preprint arXiv:1101.3354, 2011.

[15] Lowe DG (2004) Distinctive image features from Scale-Invariant keypoints. International Journal of Computer Vision 60(2):91–110.

[16] Kadir T, Brady M (2001) Saliency, scale and image description. International Journal of Computer Vision 45(2):83–105.

[17] Mikolajczyk K, Schmid C (2004) Scale & affine invariant interest point detectors. International Journal of Computer Vision 60(1):63–86.

[18] Matas J, Chum O, Urban M, Pajdla T (2004) Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing 22(10):761–767.

[19] Itti L, Koch C (2000) A saliency-based search mechanism for overt and covert shifts of visual attention. Vision Research 40(10-12):1489–150.

[20] Bruce NDB, Tsotsos JK (2009) Saliency, attention, and visual search: An information theoretic approach. Journal of Vision 9(3):1–24, URL http://journalofvision.org/9/3/5/

[21] Lowe DG (1999) Object recognition from local scale-invariant features. In: Proc. ICCV

[22] Bay H, Tuytelaars T, Gool LV (2006) Surf: Speeded up robust features. In: Proc. ECCV

[23] van de Sande KE, Gevers T, Snoek CG (2010) Evaluating color descriptors for object and scene recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on 32(9):1582–1596

[24] Sivic J, Zisserman A (2003) Video google: A text retrieval approach to object matching in videos. In: Proc. ICCV

[25] Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proc. CVPR, vol 2

[26] Jiang YG, Ngo CW, Yang J (2007) Towards optimal bag-of-features for object categorization and semantic video retrieval. In: Proc. CIVR

[27] Salton G, McGill MJ (1983) Introduction to Modern Information Retrieval. McGraw-Hill Computer Science Series, McGraw-Hill, New York, NY

[28] Philbin J, Chum O, Isard M, Sivic J, Zisserman A (2008) Lost in quantization: Improving particular object retrieval in large scale image databases. In: Proc. CVPR

[29] Jiang YG, Ngo CW, Yang J (2007) Towards optimal bag-of-features for object categorization and semantic video retrieval. In: Proc. CIVR

[30] Y.-G. Jiang, C.-W. Ngo and J. Yang, Towards Optimal Bag-of-Features for Object Categorization and Semantic Video Retrieval, Proc. ACM Int',l Conf. Image and Video Retrieval, pp. 494-501, 2007

[31] VLFeat Sift Implementation. URL: http://www.vlfeat.org/overview/sift.html