Recognition and Classification of Fast Food Images

Shaoyu Lu, Sina Lin, Beibei Wang
shaoyu88@stanford.edu , sinalin@stanford.edu , beibeiw@stanford.edu

Abstract

We aim to utilize learnt machine learning algorithms to do fast food recognition. Our goal is to find a computational efficient algorithm with high accuracy. We use Bag of SIFT, color histogram and a combination of them as the features. k-NN and SVM method (with and without kernel) are used to classify fast food images to eight classes.

1. Introduction

Food recognition is of great importance nowadays for multiple purposes. On one hand, for people who want to get a better understanding of the food that they are not familiar of or they haven't even seen before, they can simply took a picture and get to know more details about it. On the other hand, the increasing demand of dietary assessment tools to record the calorie and nutrition has also been a driving force of the development of food recognition technique. Therefore, automatic food recognition is very important and has great application potential.

However, food varies greatly in appearance (e.g., shape, colors) with tons of different ingredients and assembling methods. This makes food recognition a difficult task for current state-of-the-art classification methods, and hence an important challenge for Computer Vision researchers. Yoshiyuki Kawano and Keiji Yanai [Kawano and Yanai, 2013] proposed a real-time food recognition system which adopted a liner SVM with a fast χ^2 kernel, bounding box adjustment and estimation of the expected direction of a food region. Lukas Bossard et al. [Bossard et al., 2014] presented a novel method based on Random Forests to mine discriminative visual components and efficient classification.

Figure 1. Examples from the food image dataset. Left: "Original" dataset. Middle: "ColorCorrected" dataset. Right: "ColorCorrected+Segment" dataset.

In this effort, we intent to utilize learnt machine learning algorithms to do fast food recognition and classification. Our goal is to develop a computational efficient algorithm with high accuracy. Different features and models have been implemented and compared.

2. Dataset

The dataset we used in this study was based on the Pittsburgh Fast-food Image Dataset (PFID) images. This dataset was proposed by Chen et al. [Chen et al., 2009] to properly evaluate the performances of food recognition. This dataset is composed by 1359 food images with RGB-color of fast-food dishes mainly acquired in laboratory.

3. Pre-processing

Because the original dataset contains different lighting for the same food, we used white balance to do color correction in order to minimize the within class variance. Besides, the background is not related to the food, thus background segmentation was used to enable only food features being extracted. Figure.1 shows one set of example from the three datasets (original, color corrected, color corrected plus background segmentation). We labeled the food images into eight categories: Bread Sandwich (breakfast

sandwich), Burger, Chicken, Donut, Pancake, Pizza, Salad and Sandwich. Classification results comparison among these three datasets will be discussed later.

4. Features and Models

4.1. Features

In this study, two popular features in terms of image processing including Bag of SIFT (Scale-Invariant Feature Transform) [Lowe, 1999] and color histogram were chosen to capture the image content in our fast food images.

We chose SIFT to extract food image textures and used bag of features since it's invariant to spatial translation and rotation, and it can provide fixed length feature vectors.We chose color histogram to extract color distributions. To ensure both efficiency and accuracy, we choose 16 bins per RGB color. Later on, we explored combining these two features together and obtained a better result.

4.2. Models

k-Nearest Neighbors algorithm (k-NN) [Cover and Hart, 1967] and Support Vector Machine (SVM) [Suykens and Vandewalle, 1999] were used as learning methods in our study. They are two popular discriminative classifiers with no distribution requirement. k-NN is simple to implement with usually good result. And the accuracy of this method can be highly influenced by parameter k (Figure 3). SVM was chosen since it's very robust and well developed for classification. [Kim et al., 2012] It was originally designed for two-class problems. Therefore, we utilized "one-versus-the-rest" method in our multi-class problem.

Moreover, χ^2 Kernel was utilized for both methods as some study shows that it's good at image texture recognition. [Zhang et al., 2007] χ^2 Kernel is often used with bag of feature since it is a more natural distance measure between histograms than the euclidean distance. [Yang et al., 2009] The χ^2 Kernel comes from the χ^2 distribution:

$$\sum_{i=1}^{k} \left(\frac{X_i - \mu_i}{\sigma_i} \right)^2 \quad (1)$$

where $X_i \sim N(\mu_i, \sigma_i^2), i = 1, \ldots, k$.

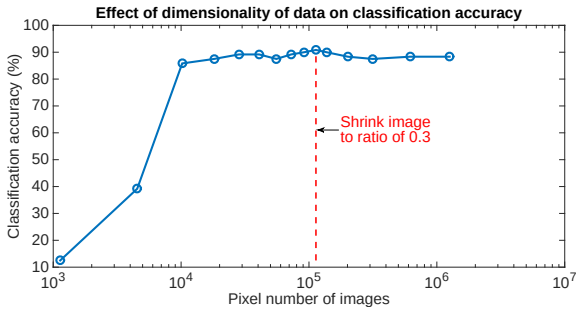

Figure 2. Effects of dimensionality on classification. The data was "ColorCorrected". Modeling used bag of SIFT and SVM with χ^2 kernel.

The bag of SIFT and χ^2 Kernel were employed using code from VLFeat [Vedaldi and Fulkerson, 2012]. SVM code is from LIBLINEAR [Fan et al., 2008] and k-NN is from Matlab Machine Leaning toolbox. We developed our own code to compute the color histogram without counting the background color.

5. Performance Analysis

5.1. Effects of data dimensionality

Data dimensionality can affect the classification accuracy a lot. Thus, we need to find an optimal dimension. From Figure.2, at the beginning, the accuracy increases very fast as the dimension increases. However, when the dimension reaches 10^4, the accuracy increases very slowly, and even decreases a little when dimension reaches 10^5. This is because, when the dimension is too high, the volume of the data space concentrates on the surface, therefore available feature dimension won't increase. Thus increasing dimensionality won't help classification at this point. In this study, the optimal dimension is 113375, which corresponds to shrink image to 30% along both X- and Y-axis.

5.2. Effects of "K" in k-NN

From Figure.3, increasing number of neighbors improves the accuracy because it helps reduce noise effects on k-NN classification. But including too much neighbors would ruin the classification, as neighbors from incorrect class would be involved. From the classification accuracy on test data, we choose the number of nearest neighbors to be 4.

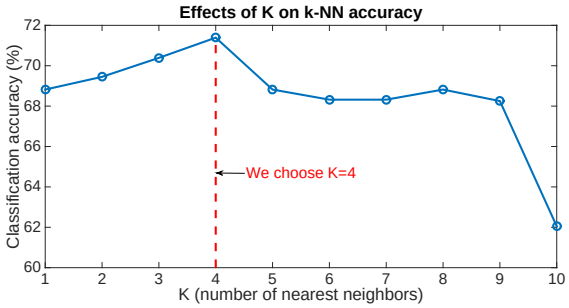

Figure 3. Effects of number of nearest neighbors on k-NN. The data here is the "ColorCorrected", χ^2 kernel is used.

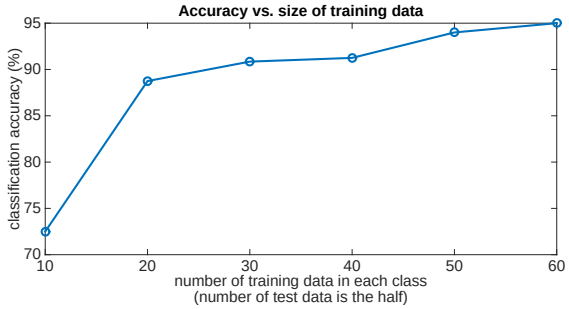

Figure 4. Effects of data size on classification. The data here is the "ColorCorrected", using bag of SIFT and SVM with χ^2 kernel.

5.3. Effects of training Data Size

In this test, we kept the ratio of training data size to test data size to be 2 to 1 and increased the training data size to see the effects. From Figure.4, the more training data we have, the better classification accuracy we can obtain. Notice that the accuracy increases slowly when the training size reaches 30. Since the time complexity would increase linearly with data matrix size, to balance the accuracy with speed, we choose to randomly pick 50 images from each class as training data and 25 as test data.

6. Results and Discussion

6.1. The importance of kernel

As shown in Figure.5, χ^2 kernel helps SVM gain much better accuracy. This is because SVM is a linear classifier who suffers from under-fitting. Kernel enables SVM to work on high-dimensional non-linear problem, which greatly benefits SVM. Thus we chose to utilize kernel in following experiments.

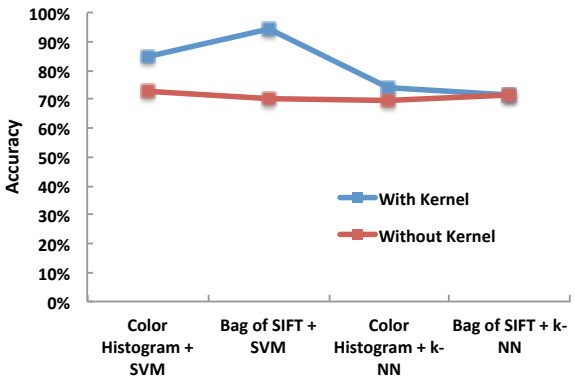

Figure 5. Kernel effects on SVM and k-NN classification accuracy

6.2. Comparison of features and models

Confusion matrix was used to capture the classification accuracy on the test dataset. Figure.6 shows examples of confusion matrix of classification results on "ColorCorrected" dataset with Color Histogram (left) and bag of SIFT (right), using k-NN (top) and SVM (bottom). χ^2 kernel was implemented in all cases. Comparing top two with bottom two figures, we can see SVM achieves much better results than k-NN.

Moreover, from the top two figures, color histogram performs better on images with more distinct color features such as salad (The accuracy by color histogram is 100% on salad while the accuracy by Bag of SIFT is 84%). On the other hand, Bag of SIFT performs better for images with distinct textures, such as donut (The accuracy by color histogram is 72% on donut while the accuracy by Bag of SIFT is 96%). The bottom two matrixes indicate the same conclusion.

In order to take advantages of both color and texture characters, we linearly combined Color histogram and bag of SIFT to a new feature (named "ColorSIFT"):

$$\text{ColorSIFT} = \alpha \cdot \text{ColorHist} + (1-\alpha) \cdot \text{BagOfSIFT} \quad (2)$$

where α is the weight parameter for color histogram and $1-\alpha$ is the weight parameter for bag of SIFT. From experiments, we find the optimal of α is 0.4. As shown in Figure.7, the accuracy increased to 97.5% compared with the formal best (94%).

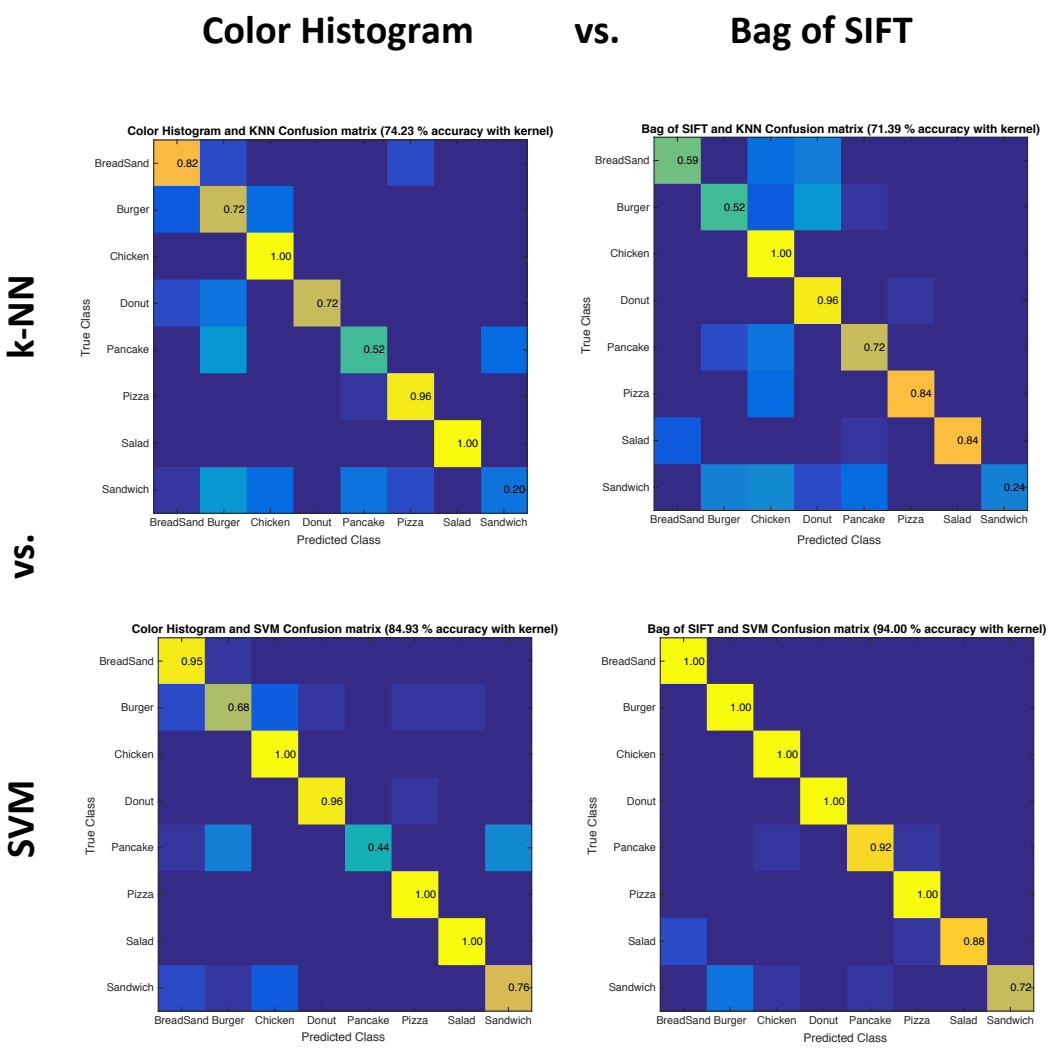

Figure 6. Confusion Matrix on "ColorCorrected" dataset with Color Histogram (left) and bag of SIFT (right), using models of k-NN (top) and SVM (bottom). χ^2 kernel was utilized in all cases.

Model	Feature	Original Dataset	Segmented Dataset	Color Corrected Dataset
SVM (with kernel)	Bag of SIFT	93.00%	92.00%	94.00%
	Color Histogram	85.93%	78.73%	84.93%
SVM (without kernel)	Bag of SIFT	55.84%	58.70%	70.05%
	Color Histogram	68.43%	65.50%	73.00%
k-NN (with kernel)	Bag of SIFT	66.82%	62.82%	71.39%
	Color Histogram	78.93%	68.86%	74.23%
k-NN (without kernel)	Bag of SIFT	68.93%	60.43%	71.43%
	Color Histogram	73.50%	65.00%	69.50%

Table 1. Classification accuracy comparison among the Original, "ColorCorrected" and "ColorCorrected+Segment" dataset.

6.3. Comparison between Different Dataset

From Table.1, we can see three datasets achieve almost the same accuracy via different algorithms.

7. Conclusion

To recognize different kinds of food with various appearance, we extracted color and texture features

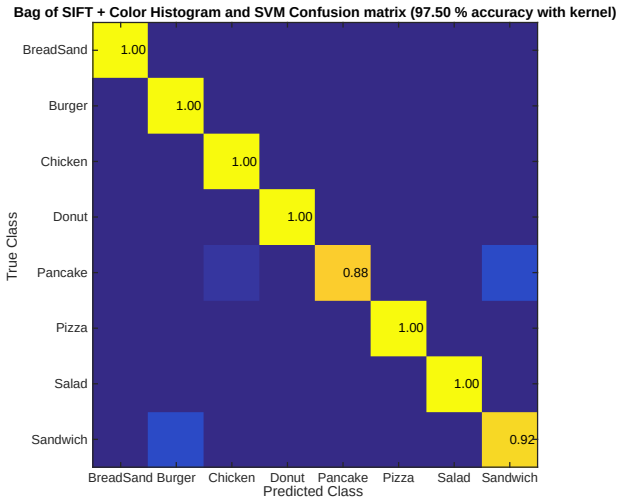

Figure 7. Confusion Matrix on "ColorCorrected" dataset with combined feature of Color Histogram (left) and bag of SIFT (right), using model of SVM with kernel.

from food images, and utilized k-NN and SVM to do the classification. To achieve a higher accuracy, we linearly combined the two kind of features and adopted the χ^2 kernel. With performance analysis like dimensionality and data size, we accomplished a high-accuracy result with great computational efficiency.

8. Future Work

We plan to study for better background segmentation and color correction in image processing. We are also going to explore more machine learning algorithms and technique details. Moreover, we will test algorithms with food images with real-life environment.

References

[Bossard et al., 2014] Bossard, L., M. Guillaumin, and L. Van Gool, 2014, Food-101–mining discriminative components with random forests, *in* Computer Vision–ECCV 2014: Springer, 446–461.

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

[Cover and Hart, 1967] Cover, T., and P. Hart, 1967, Nearest neighbor pattern classification: Information Theory, IEEE Transactions on, **13**, 21–27.

[Fan et al., 2008] Fan, R.-E., K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, 2008, Liblinear: A library for large linear classification: The Journal of Machine Learning Research, **9**, 1871–1874.

[Kawano and Yanai, 2013] Kawano, Y., and K. Yanai, 2013, Real-time mobile food recognition system: Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on, IEEE, 1–7.

[Kim et al., 2012] Kim, J., B.-S. Kim, and S. Savarese, 2012, Comparing image classification methods: K-nearest-neighbor and support-vector-machines: Proceedings of the 6th WSEAS international conference on Computer Engineering and Applications, and Proceedings of the 2012 American conference on Applied Mathematics, World Scientific and Engineering Academy and Society (WSEAS), 133–138.

[Lowe, 1999] Lowe, D. G., 1999, Object recognition from local scale-invariant features: Computer vision, 1999. The proceedings of the seventh IEEE international conference on, Ieee, 1150–1157.

[Suykens and Vandewalle, 1999] Suykens, J. A., and J. Vandewalle, 1999, Least squares support vector machine classifiers: Neural processing letters, **9**, 293–300.

[Vedaldi and Fulkerson, 2012] Vedaldi, A., and B. Fulkerson, 2012, Vlfeat: An open and portable library of computer vision algorithms (2008).

[Yang et al., 2009] Yang, J., K. Yu, Y. Gong, and T. Huang, 2009, Linear spatial pyramid matching using sparse coding for image classification: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 1794–1801.

[Zhang et al., 2007] Zhang, J., M. Marszałek, S. Lazebnik, and C. Schmid, 2007, Local features and kernels for classification of texture and object categories: A comprehensive study: International journal of computer vision, **73**, 213–238.