CSE 803 Project Report

Team Member:

Kai Wu , Rundong Zhao, Ze Zhang

1. Introduction

Content-based image detection is a complex problem in Computer Vision. In this project, we tried to implement a food image recognition system. We used multiple techniques, like Bag-of-Words model, SIFT, K-means Clustering and Neural Network. The average detection rate for our training dataset is 84%, and the average detection rate for the test dataset is 80%.

2. Dataset

To recognize image, we need to have images each class for training and testing. For the training, it might be better to have as many as we can. At the beginning, we thought it was okay to have around 10 images per class, but later we realized that we need more than that. After we collected data ourselves from our life, we started downloading from the Internet. There were many images in the Internet, but we tried to use clear pictures for the training. We tried to find pictures with white background or black background to detect the object easily. For the training, we do not use any images, which contain several classes' object. Some of images contain only one object in the image (Figure2.1 Left). Others contain more than one objects (Figure2.1 Right).

Figure 2.1 (left) one object of class (Right) several objects of the class

In our project, we choose 10 classes to recognize. Shown in table below:

Apple	banana	broccoli	burger	fries	hotdog	pizza	rice	salad	strawberry

Table 2.1 target recognize object type

Each type we have around 60 training images.

We have a limitation on each class for training. First of all, it should be clear image and have white or black background. We used mostly white background. For the apple class, we consider original apple (Figure2.2 Left and Middle) and sliced apple (Figure 2.2 Right)

Figure 2.2 (Left and Middle) Original apples (Right) sliced apple with non-sliced apple

For the banana class, we consider original banana (Figure2.3 Left) and peeled off banana (Figure2.3 Right).

Figure 2.3 (Left) Ordinal banana (Right) Banana with peeled off banana

For the broccoli class, we consider whole, cut, and cooked broccoli (Figure2.4).

Figure 2.4 (Left) Whole broccoli (Middle) Small size of the broccoli (Right) Cooked broccoli

For the burger class, we do not have any limitation on burger. We downloaded images from the Internet. The image should have two pieces of bread and meat, vegetable or some fries between two piece of breaad (Figure2.5).

Figure2.5 Burgers with meat, cheese, vegetables and fries between two breads

For the french-fry class, we consider only French fry and without ketchup or any other sauce (Figure2.6).

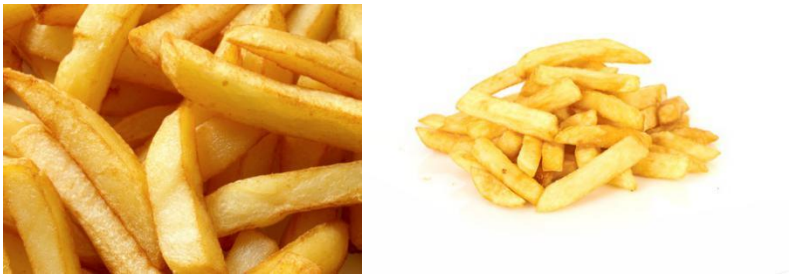

Figure2.6 French fry without ketchup

For the hotdog class, we consider it as a bun with sausage and with or without some sauce (Figure2.7). It could have some pickles, onions, or vegetable also. The most important this is the bun and sausage.

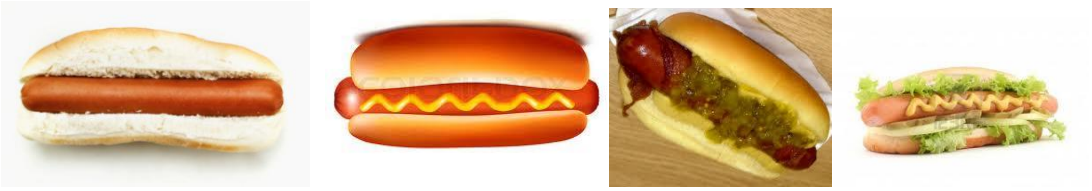

Figure2.7 (Left) Hotdog without sauce (Middle Left) Hotdog with sauce (Middle right) Hotdog with pickles and sauce (Right) Hotdog with vegetable

For the pizza class, we consider whole and slice pizza (Figure2.8). We collected

many different kinds of pizzas, such as pepperoni, cheese, combination, and so on.

Figure2.8 (Left) Whole pepperoni pizza (Right) Sliced cheese pizza

For the rice class, we consider cooked and uncooked rice as class of rice (Figure2.9).

Figure2.9 (Left) Cooked rice (Right) Uncooked rice

For the salad class, we do not think of pasta salad in this case. We collected all images with many vegetables, fruit and meat in it (Figure2.10).

Figure2.10 (Left) Fruit and vegetables salad (Right) Meat mixed salad

For the strawberry class, we consider a red strawberry with the stem, which is green (Figure2.11). Some images are sliced strawberry.

Figure2.10 (Left) Strawberry with stem (Right) Sliced strawberry with the whole strawberry

After training, we used several image to test how it works well. We used little bit complicated images than the training image. Some images have several classes' objects in one image. Other images have clear image. The other images have zoomed in and out.

3. Algorithm design

In our project, we use Scale Invariant Feature Transform (SIFT) [2] to collect a lot of features from training images. Then we use BoW(bag of words) [1] to process these feature vectors. In BoW, we use k-means[3] to cluster those features into a visual vocabulary. For each of training image we build a histogram of word frequency (assigning each feature found in the training image to the nearest word in the vocabulary). Then input these histograms to a Neural Network [4] to train. Finally, build a histogram for test images and classify them with the Neural Network just trained.

The idea of food recognition system is described as follow chart:

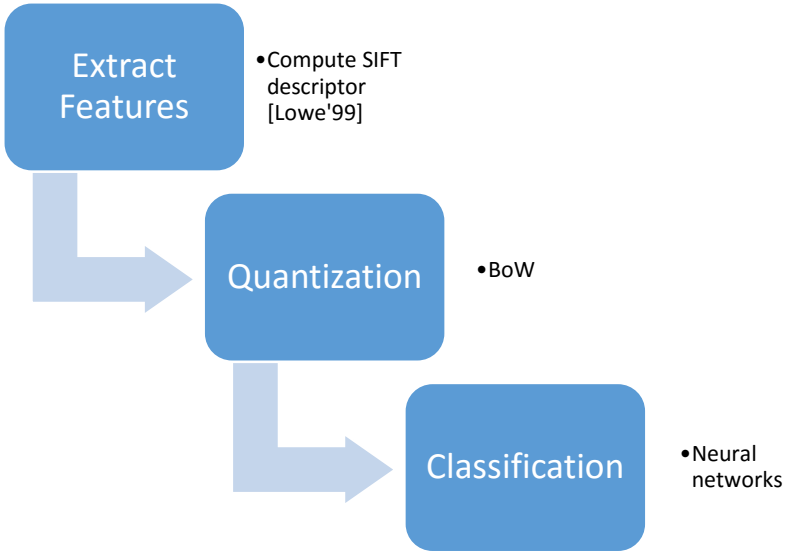

Figure 3.1 System Procedure

3.1 Feature extraction

In this project, we used SIFT (Scale-invariant feature transform) to collect local features of images. SIFT is very robust image descriptor which represents a collection of feature vectors.

Figure 3.2 Feature extraction

3.1.1 Scale Invariant Feature Transform (SIFT)

As known, SIFT is a method to detect and describe local features in images, as descried in Lowe's paper, Sift is invariant to image transformations as rotation , scaling

and translation. Because the selected key points in images are invariant to image transformation, they could be used as local feature descriptors. Also, by normalizing image descriptors, these features could be invariant to different illumination.

In our project, we used Sift descriptors that has been computed in Lowe's method. For all of our training images, we compute their feature vectors, so we can get a lot of 128-dimensional feature vectors in the end.

3.2 Quantization

3.2.1 BoW model

The main idea of BoW model is to quantize local descriptors into "visual words" which decreases the descriptors' amount dramatically. In this way, each image can be viewed as a long and sparse vector of words. Then we can apply scalable indexing and fast search on this vector space.

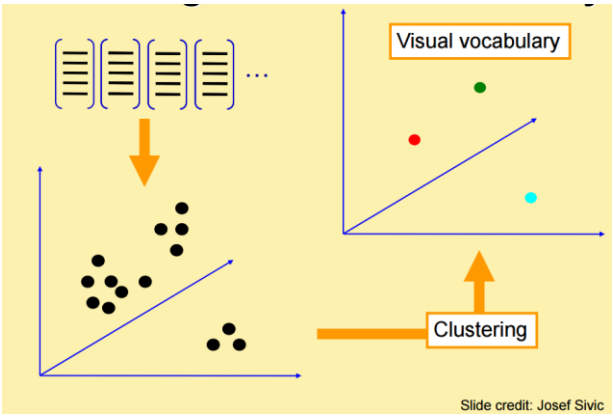

Figure 3.1 Bag of words model (from Josef Sivic)

After we collected a large sample of features from each food image, we using K-means Clustering to quantize the feature space according to their statistics. In that case, different types of food are "visual words" which are the K cluster centroids. Once each cluster centroid do not change any more, the vocabulary is established, the corpus of sampled features can be discarded. A novel image's features can be translated into words by determining which visual word they are nearest to in the feature space.

3.2.2 K-means Clustering

K-means Clustering is a method of vector quantization, originally from signal processing. It aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells.

Algorithm:

- Randomly initialize K cluster centroids
- Iterate until convergence:

  Assign each data point to the nearest centroid

  Re-compute each cluster centroid as the mean of all points assigned to it.

3.2.3 Histogram

Since we now have a clustered bag of features, we are ready to convert all the training images into the histogram representation. To calculate the similarity between training images and bag of image features, we need to calculate Euclidean distance between them. We will eventually have corresponding frequencies by doing this.

3.3 Neural Network Model

artificial neural networks (ANNs) are a family of models inspired by biological neural networks (the central nervous systems of animals, in particular the brain) and are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. Artificial neural networks are generally presented as systems of interconnected "neurons" which exchange messages between each other. The connections have numeric weights that can be tuned based on experience, making neural nets adaptive to inputs and capable of learning.

An ANN is typically defined by three types of parameters: The interconnection pattern between the different layers of neurons. The learning process for updating the weights of the interconnections. The activation function that converts a neuron's weighted input to its output activation.

Training a neural network model essentially means selecting one model from the set of allowed models (or, in a Bayesian framework, determining a distribution over the set of allowed models) that minimizes the cost criterion. Most of the algorithms used in training artificial neural networks employ some form of gradient descent, using backpropagation to compute the actual gradients. This is done by simply taking the derivative of the cost function with respect to the network parameters and then changing those parameters in a gradient-related direction

We use the neural network toolbox in matlab to complete our algorithm here. We input several classes of processed images and their labels to train the neural network, neural network will learn from training images and update weights for every interconnection. Finally, we can get a trained neural network which can be used to classify different kinds of images.

4. Result

The required training time is approximately 4 hours. Applying test images to the neural network recognition system to get results is very fast, usually within 1 second.

4.1 Training data accuracy

Apple	Banana	Broccoli	Burger	Fries	hotdog	pizza	Rice	Salad	Strawberry
0.79	0.86	0.95	0.74	0.87	0.74	0.78	0.87	0.9	0.88

Figure 4.2 Training result

The overall accuracy is 0.84.

4.2 Test data accuracy

Apple	Banana	Broccoli	Burger	Fries	hotdog	pizza	Rice	Salad	Strawberry
0.76	0.74	0.91	0.72	0.69	0.73	0.82	0.9	0.9	0.8

Figure 4.3 Test result

The overall accuracy is 0.80.

5. Analysis

The result is fairly good. But there is still space to improve the classification results. SIFT does not consider the influence the background, but our training images all contain background. So in order to improve the accuracy, we can apply some pre-computation on the images to extract the foreground from the background. This is a hard task if we want to automatically do that, where need much more research.

Reference

[1] Josef Sivic and Andrew Zisserman. Video google: Efficient visual search of videos. In Toward Category-Level Object Recognition, pages 127–144, 2006.

[2] David G. Lowe. Object recognition from local scale-invariant features. In ICCV, pages 1150– 1157, 1999.

[3] MacQueen, J. B. Some Methods for classification and Analysis of Multivariate Observations

[4] Howard Demuth, Mark Beale. Neural Network Toolbox Guideline