Comparison of Feature Extraction Methods for Image Analysis and its Application for Classification of Food Items in Computer Vision

Rohit Mahajan
Master of Technology, Student
Department of Electronics and Telecommunication
K.J. Somaiya College of Engineering
University of Mumbai
rohit.am@somaiya.edu

Dr. Ameya K. Naik
Associate Professor
Department of Electronics and Telecommunication
K.J. Somaiya College of Engineering
University of Mumbai
ameyanaik@somaiya.edu

Abstract- This Paper discusses food image classification methods using Image Analysis and Computer Vision. It explains why food image classification and identification is important and which methods have been proposed to solve the same problem. In this report Different methods for feature extraction and classification are described. It also compares work done in past using features like SIFT, SURF, HARISS, BRISK, Bag of Words Model with its advantages and disadvantages. Database of Food-101 obtained from Caltech-101 is used to generate results discussed in this Paper.

Index Terms - *Image Analysis, Computer Vision, Food-101.*

*****

I. INTRODUCTION

There is a growing concern about chronic diseases and other health problems related to diet including hypertension, obesity, heart disease and cancer. The need of accurate methods and tools to measure food and nutrient intake becomes imperative for epidemiological and clinical research linking diet and disease. Food recognition came into existence in the computer vision society few years back. Scientific Community aim to develop an image analysis system to automatically identify and quantify foods consumed from images of foods.

Martin et al. [1] used a color histogram based method to classify food items. Wu et al. [2] discussed a feature points based(SIFT) method to recognize food items. The accuracy obtained is under 70%.Yang et al. [3] discussed the texture based histogram of food items to the texture distributions for food recognition and the accuracy improved as much as 78%. Zhu et al. [4] detected food textures from the image first and then utilize the histogram of textures to classify food items.

Puri.M et al [5] proposed a method which uses speech input from user for improving recognition method. It used color neighbourhood and maximum response features together in texton model of histogram. Texton histograms and feature selected using Adaboost are classified using SVM In the self-made dataset, the author obtained accuracies from 95% to 80%, from 2 to 20increase in food classes.

Yang.J [6] used a self-made database of video and images for food classification problem. Color histograms and SIFT Features are used to classify seven different classes of food items.For SIFT approach 56% accuracy and for color histogram 47% classification accuracy was obtained using SVM.

Yanai.K [7] discussed use of Gabor filter, SIFT Features and color histograms together in MKL for recognition of food items. Accuracy of 61% was obtained for 50 food class example.Delp.E.J [3] suggested use of color and texture feature together with SVM classifier. Accuracy was 58% and 94% for original food and food replica respectively.

Tan and Kong used SIFT with Bag of visual word model and as classifier he used Bayesian probabilistic classifier. They obtained accuracy of 92% for six food class problem and for each food item, database of 50 was used.

Kan.J [8] used color, size, texture, shape and context features and classified using Artificial Neural Network. For recognition of hamburgers, fries, chicken nuggets, and apple pies classification accuracy of 95%, 80%, 90%, and 90% was obtained respectively.

KeijiYanai [9] developed an app for real time recognition of food items. The SVM along with Chi square technique gave a recognition accuracy of 81.55%. For each food item the dataset size of 50 was chosen.

Image datasets are a prerequisite to visual object recognition research such as object modeling, detection, classification, and recognition. In fact, publicly available data collection and evaluation play a vital role in the development of automated object recognition technologies. The research community has developed both general- and specific-purpose datasets. The former contains a variety of objects and is primarily designed to support category level object recognition research (e.g., Caltech FOOD101 [10]). By providing standardized data on which researchers can train and test their algorithms, such datasets have made it possible to compare different approaches for object category recognition.

II. IMAGE ANALYSIS ALGORITHMS AND SVM

A. Scale Invariant Feature Transform(SIFT) :

SIFT is the most sought after method for Feature Extraction developed by David Lowe[11]. SIFT is rotation and scale invariant. The following steps are involved in Construction of SIFT Descriptors.

Construction of scale space is the first step in algorithm. We have to create 4 octaves and 5 blur levels as suggested by David Lowe[11]. Next step involves LoG Approximation. The Laplacian of Gaussian is great for finding interesting points (or key points) in an image. But it's computationally expensive. For finding keypoints, weuse the super-fast approximation. These are maxima and minima in the Difference of Gaussian image we calculate in second step.

$$D(x, y, \sigma) = (G(x, y, k\,\sigma) - G(x, y, \sigma) \times I(x, y) \quad (1)$$

Where G is Gaussian Operator where as I is a image. Now the difference of these Gaussian images is termed as DoG(Difference of Gaussian). Getting rid of bad key points is the next step. Edges and low contrast regions are bad keypoints. Eliminating these makes the algorithm efficient and robust. Assigning an orientation to the keypoints is the next step.An orientation is calculated for each key point. Any further calculations are done relative to this orientation. This effectively cancels out the effect of orientation, making it rotation invariant. Finally, with scale and rotation invariance in place, one more representation is generated. This helps uniquely identify features. Let's say it has 50,000 features. With this representation, it can easily identify the feature it is looking for (say, a particular eye, or a sign board).

The figure shows SIFT Descriptors in8directionsforeachorientationhistogram, with the length of each arrowis equivalent to the magnitude of that histogramentry. Thus to create a 128 element descriptor we generate a 4x4 array grid with 8 orientation bins in each sample.

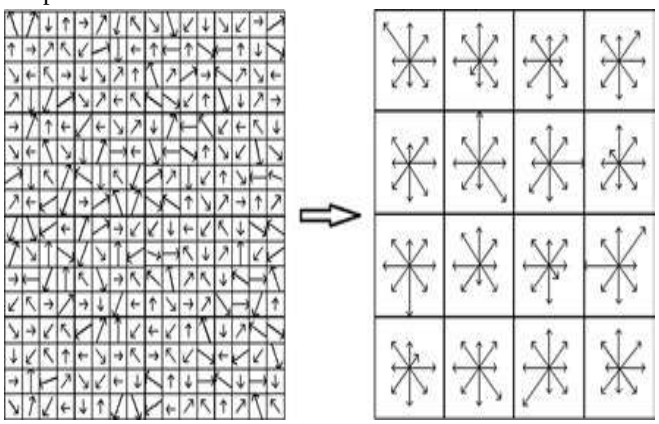

Figure 1: SIFT Descriptor Generation

B. SpeedUpRobustFeatures (SURF):

SURF algorithm ,is based on multi-scalespace theory and the feature detector is based on Hessian matrix. Since Hessian matrix has good performance and accuracy. In image I,x=(x,y) is the given point, the Hessian matrix H(x,σ) in x at scale σ, It can be define as

$$H(x, \sigma) = \begin{bmatrix} Lxx(x, \sigma) & Lxy(x, \sigma) \\ Lyx(x, \sigma) & Lyy(x, \sigma) \end{bmatrix}$$

WhereL_{XX}(x,σ)istheconvolutionresultofthesecondorderderivative ofGaussianfilter $\frac{\partial^2}{\partial x^2}(\sigma)$with the image I in point x.

SURFcreatesa"stack"without2:1downsamplingforhigherlevelsinthepyramidresultinginimagesofthesameresolution .Duetotheuseofintegralimages,SURFfiltersthestackusingaboxfilterapproximation ofsecond-order Gaussianpartial derivatives.Since integralimagesallowthecomputationof rectangularboxfiltersinnerconstant time.InFigure 2ShowtheGaussiansecondorderspartialderivativesiny-directionandxy-direction.

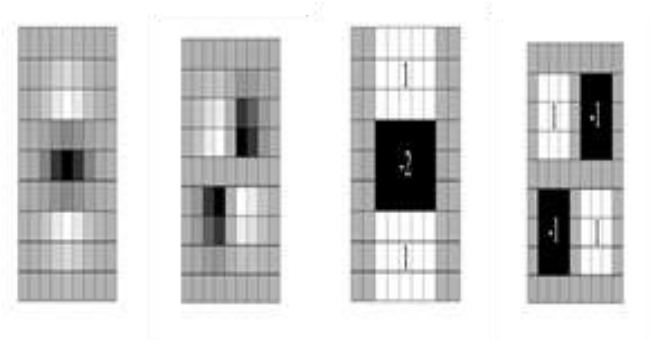

Figure 2: The Gaussian second orders partial derivatives in y-direction and xy-direction.

In descriptors, SIFT is good performance compare to other descriptors. The SURF descriptor is based on similar properties. The first step consists of fixing a reproducible orientation based on information from a circular region around the interest point. And second construct a square region aligned to the selected orientation, and extract the SURF descriptor from it. In order to be invariant to rotation, it calculate the Haar-wavelet responses in x and y direction shown in Figure 3.

Figure 3: Haar wavelet types used for SURF

C. Binary Robust Invariant Scalable Key Points(BRISK):

BRISK is composed out of three parts. The very first part is a sampling pattern of sample points in the region around the descriptor. Second step is Orientation compensation that is to use some mechanism to measure the orientation of the keypoint and rotate it to compensate for rotation changes. Third part include finding Sampling pairs, These pairs are used to compare while building the final descriptor.

The BRISK descriptor has a hand-drawn sampling Pattern, which is different than other descriptors. It consist of concentric rings.

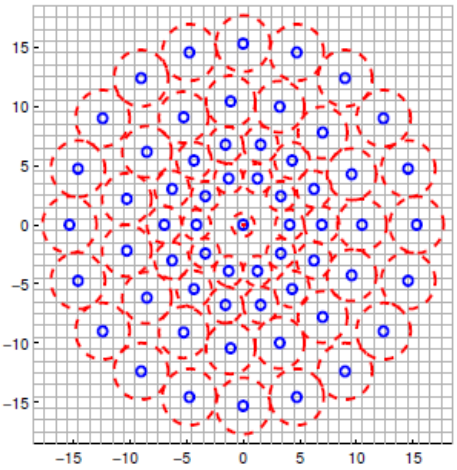

Figure 4: BRISK Descriptor

When considering each sampling point, we take a small patch around it and apply Gaussian smoothing. The red circle in the Figure 4 above illustrations the size of the standard deviation of the Gaussian filter applied to each sampling point.

a) Long and Short Pairs:

When using this sampling pattern, we distinguish between short pairs and long pairs. Short pair are pairs of sampling points that their distance is below a certain threshold d_max and long pairs are pairs of sampling points that their distance is above a certain different threshold d_min, where d_min>d_max, so there are no short pairs that are also long pairs.

Total Area Covered:

$$\mathcal{A} = \{(\mathbf{p}i, \mathbf{p}j) \in \mathrm{R2} \times \mathrm{R2} \mid i, j \in \mathrm{N}, j<i< \}$$

Short Pairs:

$$\mathcal{S} = \{(\mathbf{p}i, \mathbf{p}j) \in \mathcal{A} \mid \|\mathbf{p}j - \mathbf{p}i\| < \delta max\} \subseteq \mathcal{A}$$

Long Pairs:

$$\mathcal{L} = \{(\mathbf{p}i, \mathbf{p}j) \in \mathcal{A} \mid \|\mathbf{p}j - \mathbf{p}i\| > \delta min\} \subseteq \mathcal{A}$$

Long pairs are used in BRISK to determine orientation and short pairs are used for the intensity comparisons that build the descriptor.

b) Orientation Computation:

- The Long distance pairs are used.
- Calculate local gradient between the Long pairs.
- Sum up all the local gradients.

$$g(P_i, P_j) = (P_j - P_i).\frac{I(P_j, \sigma_j) - I(P_i, \sigma_i)}{||P_j - P_i||^2}$$

$$g = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{L}.\sum_{(P_i,P_j)\in L} g(P_i, P_j)$$

$$\theta = \arctan2(g_y, g_x)$$

c) Building the descriptor and descriptor distance:

As with all binary descriptors, building the descriptor is done by performing intensity comparisons. BRISK takes the set of short pairs, rotate the pairs by the orientation computed earlier and makes comparisons of the form:

$$b = \begin{cases} 1, & I(P_j^{\alpha}, \sigma_j) > I(P_i^{\alpha}, \sigma_i) \\ 0, & \text{Otherwise} \end{cases}$$

D. Bag of Visual Words Model:

To represent an image using BOW model, an image can be treated as a document. Similarly, "words" in images need to be defined too. To achieve this, it usually includes following three steps: feature detection, feature description, and codebook generation. A definition of the BOW model can be the "histogram representation based on independent features".

a) Feature representation:

After feature detection, each image is abstracted by several local patches. Feature representation methods deal with how to represent the patches as numerical vectors. These vectors are called feature descriptors. A good descriptor should have the ability to handle intensity, rotation, scale and affine variations to some extent. One of the most famous descriptors is Scale-invariant feature transform (SIFT). SIFT converts each patch to 128-dimensional vector. After this step, each image is a collection of vectors of the same dimension (128 for SIFT), where the order of different vectors is of no importance.

Figure 5: Collection of visual words

b) Codebook generation:

The final step for the BOW model is to convert vector-represented patches to "Code words", which also produces a "codebook". One simple method is performing k-means clustering over all the vectors. Code words are then defined as the centers of the learned clusters. The number of the clusters is the codebook size.Thus, each patch in an Figure 5 is mapped to a certain codeword through the clustering process and the image can be represented by the histogram of the Code words.

Train an Image Classifier With Bag of Visual Words:

Step1: Detects and extract features from the image and then uses the approximate nearest neighbor algorithm to construct a feature histogram for each image. Then increment histogram bins based on the proximity of the descriptor to a particular cluster center. The histogram length corresponds to the number of visual words that are constructed. The histogram becomes a feature vector for the image.

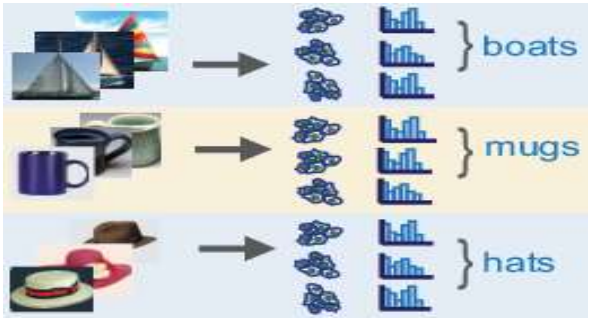

Figure 6: BOW Feature Vector Generation

Step2: Repeat step 1 for each image in the training set to create the training data.

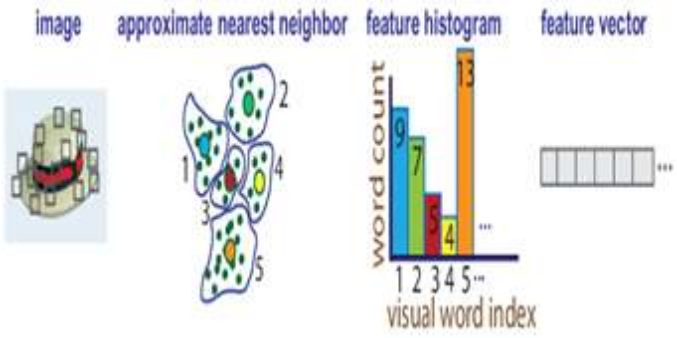

Figure 7: BOW Histogram Generation

Step3: Evaluate method to test the classifier against the validation image set. The output confusion matrix represents the analysis of the prediction. A perfect classification results in a normalized matrix containing 1s on the diagonal. An incorrect classification results fractional values.

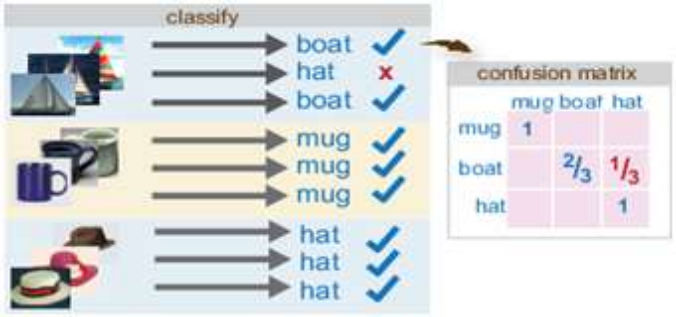

Figure 8: BOW Classification Results

E. SVM

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyper plane. It is given labeled training data (supervised learning) and the algorithm outputs an optimal hyper plane which categorizes new examples. SVM classifies data by finding the best hyper plane that separates all data points of one class from those of the other class. The best hyper plane for an SVM means the one with the largest margin between the two classes.

a) Hyper plane:

A hyper plane is a linear decision surface that splits the space into two parts. It is a subspace of one dimension less than its ambient space. If the space is 2-dimensional, its hyper planes are the 1-dimensional lines. If a space is 3-dimensional then its hyper planes are the 2-dimensional planes.

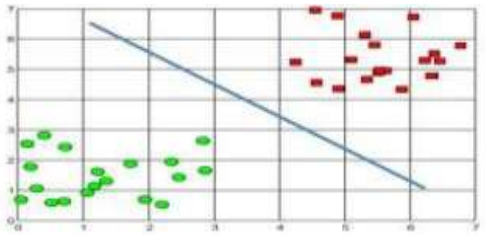

Figure 9: Hyperplane in 2D

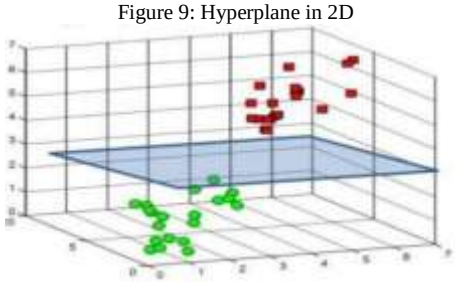

Figure 10: Hyperplane in 3D

b) Kernel trick:

Most of the applications in which SVMs are used require a more powerful tool than a simple linear classifier. This stems from the fact that in these tasks the training data can be rarely separated using an hyper plane. So now the idea is to gain linearly separation by mapping the data to a higher dimensional space.

Kernel methods owe their name to the use of kernel functions, which enable them to operate in a high-dimensional, implicit feature space without ever computing the coordinates of the data in that space, but rather by simply computing the inner products between the images of all pairs of data in the feature space. This operation is often computationally cheaper than the explicit computation of the coordinates. This approach is called the kernel trick.

Types of Kernel Transformations:

1. Linear
2. Quadratic
3. Polynomial (Degree more than 3)
4. RBF (Radial basis function i.e. Gaussian)
5. MLP (Multi-Layer Perceptron)

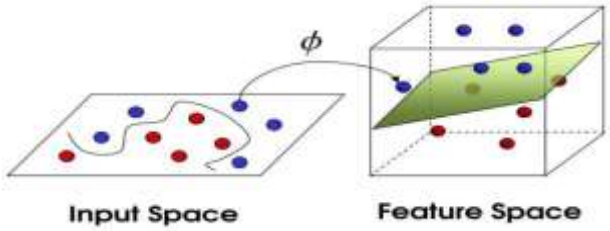

Figure 11: Kernel Transformation

III. IMPLEMENTATION DETAILS

A. SURF Features with SVM Implementation:

While implementing SURF algorithm along with SVM, We choose number of food classes to be 2. If there is any need to increase number of classes either we go for Multi-SVM or we can classify the different food classes by

comparing them on many to one basis. Now we will enlist the information about existing database.

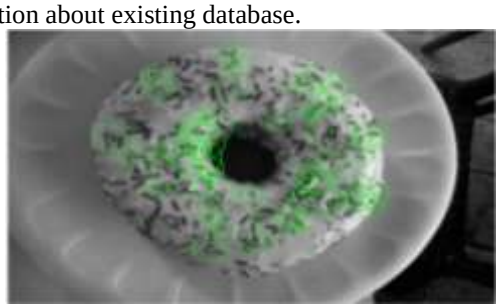

Figure 12: Extracted SURF Features

B. BRISK Features with SVM Implementation:

While implementing BRISK algorithm along with SVM, We choose number of food classes to be 2. If there is any need to increase number of classes either we go for Multi-SVM or we can classify the different food classes by comparing them on many to one basis. Now we will enlist the information about existing database.

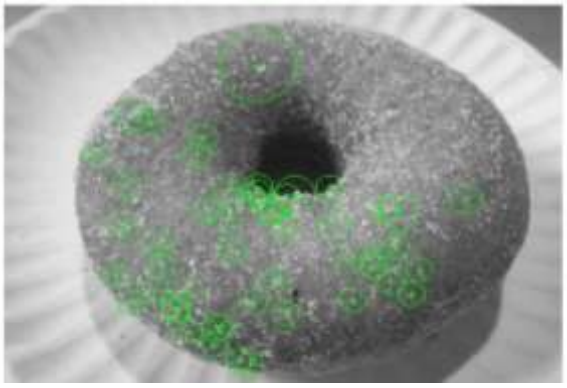

Figure 13: Extracted BRISK Features

C. SIFT Feature and BOW Model with SVM Implementation:

While implementing BOW model we extract SIFT Features from it and cluster them then use it along with SVM, We choose number of food classes to be 2. We can increase the number of food classes as we want but to train them will take a large amount of time and will also require Computer with high RAM (At least 16GB). The time taken to process these three food data classes were about 3 min with a Laptop of 4GB RAM. Now we will enlist the information about existing database.

The program first selects the Base number of images to be processed, which is selected as least number of images of all food classes i.e 80 in this case. Then it will randomly divide it into Training and Test set in the 30:70 ratio and then train the BOW model and obtain the results.

The Figure 13 shows number of visual words and frequency of their occurrence. The Confusion Matrix given below shows the accuracy with which the Food items were identified. Fries had highest accuracy as compared to other classes.

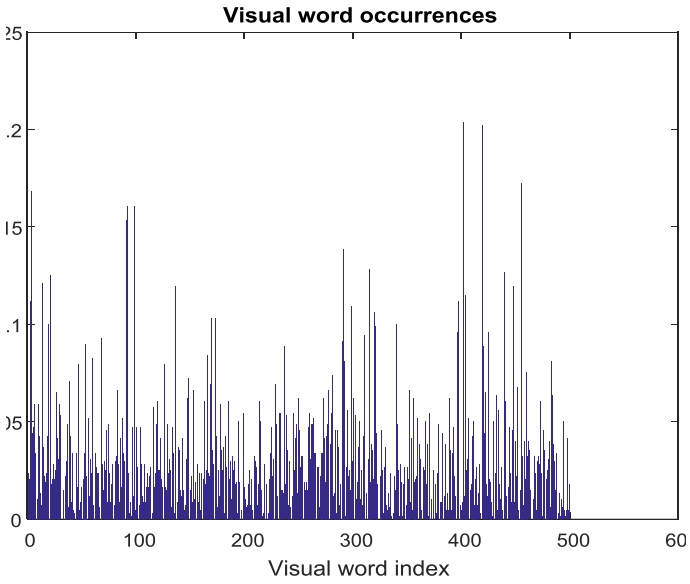

Figure 14: Visual Word Occurrence Graph

Table 1: SIFT Features and BOW Model with SVM Result

Feature Extraction Algorithm	Images in Training Set	Images in Testing Set	Accuracy
SURF	22	50	86%
BRISK	22	50	80%
SIFT and BOW	22	50	93%

From the results obtained, It has been observed that SIFT combined with Bag Of Visual Words gives us the best result but also time taken for computation of result was as high as 3 min. SIFT is rotation and scale invariant. In case of SURF and BRISK result are obtained rather quickly that is in about 30 sec but Accuracy is less compared to SIFT.

IV. CONCLUSION

In this Paper, we described an image classification system for identifying food items in images of eating occasions. Acceptable food identification accuracy has been achieved. Automatic identification of food items in an image is not an easy problem. It is understood that we will not be able to recognize every kind of food. Recent researches in continuously refining and developing the system to increase its accuracy and usability is exhaustively going on, which is done by exploring contextual information in addition to visual characteristics.

REFERENCES

[1] C Martin, S Kaya, and B Gunturk. Quantification of food intake using food image analysis.Engineering in Medicine and Biology Society. Annual International Conference of the IEEE, pages 6869 – 6872, 2009.

[2] W Wu and J Yang. Fast food recognition from videos of eating for calorie estimation.Multimedia and Expo, IEEE international Conference on, pages 1210 – 1213, Jan 2009.

[3] S Yang, Mei Chen, Dean Pomerleau, and Rahul Sukthankar. Food recognition using statistics of pairwise local features.Computer Vision and Pattern Recognition, IEEE Conference on, pages 2249 – 2256, 2010.

[4] F Zhu, M Bosch, I Woo, S Kim, C Boushey, D Ebert, and E Delp. The use of mobile devices in aiding dietary assessment and evaluation.Selected Topics in Signal Processing, IEEE Journal of, 4(4):756 – 766, 2010.

[5] Puri.M, Zhu.Z, Yu.Q and Sawhney.H, "Recognition and volume estimation of food intake using a mobile device" in proc, workshop Appl.2009.

[6] Chen.M, Dhingra.K, Wu.W, Sukthankar.R and Yang.J, "PFID: Pittsburgh fast-food image dataset" in proc, 16th IEEE int. conf. image process. 2009.

[7] Joutou.T and Yanai.K, "A food image recognition system with multiple kernel learning" in proc, 16th IEEE int. conf. image process. 2009.

[8] Kong.F and Kan,J" Dietcam: Automatic assessment and evaluation" Pervasive comput.,voI.8 ,Feb 2012 , pp. dietary mobile

[9] Yoshiyuki Kawano and KeijiYanai, "Real time mobile food recognition" IEEE conf. pattern recognition, pp-356 -36 3, Jan 2013.

[10] Food 101 database download link: https://www.vision.ee.ethz.ch/datasets_extra/food-101

[11] D. G. Lowe, “Distinctive image features from scale invariant keypoints,” International journal of computer vision, vol. 60, no. 2, pp. 91–110, October 2004.