Food Images Classification using Bag of Visual Words

Joseph Garcia
Quezon City, Philippines
jrenangarcia@gmail.com

Dylan Valerio
Quezon City, Philippines
dylan_valerio@yahoo.com

1. INTRODUCTION

There is a general consensus on the fact that people love food. Understanding food in everyday life is a problem which has been considered in different research areas due its important impact under the medical, social and anthropological aspects [1]. For instance, a person's diet is strictly linked to one's overall health care. The impact of computer vision in recognizing food images (via mobile/wearable cameras), as well as their properties like calorie content, nutrition information can significantly help in establishing a wider comprehension of the relationship between people and their meals. Diet monitoring systems can automatically provide this useful information for nutritionists to assess and track the food intake of their patients.

On the other hand, the importance of food as a basic need for human life has clearly rippled through into the digital life. Food is nowadays considered to be one of the most photographed objects. This is evident in the abundance of food photography in social networks, dedicated photo sharing sites and mobile applications. Automatic food recognition would not only help users effortlessly organize their extensive photo collections but would also help online photo repositories make their content more accessible.

Food understanding from images however is considered to be a challenging computer vision task since food is intrinsically deformable and presents high variability in appearance. In this study, we address the problem by experimenting on different recognition methods and assessing their effectiveness in food classification.

2. OBJECTIVES

The objectives of the study are as follows:

a. Compare food image classification performance on Feature vector representation and Bag of Words representation using accuracy, precision and recall measures.
b. Explore different feature fusion techniques for Bag of Words model. Assess expressiveness of each food category with the type of feature being used to represent the image.
c. Construct models in doing classification on available food images. Evaluate their performance on overall-level and food category-level basis.
d. Obtain the most effective classification model and feature representation technique which yields the highest performance on food images.
e. Provide discussions on the effects of varying dictionary size, type of classifier variant and feature selection used for food images classification.

3. METHODOLOGY

3.1 Data Set

The Food-101 data set was obtained from the Computer Vision Laboratory in Zurich, Switzerland [2]. Compared with other food images benchmark data sets such as the Pittsburgh Fast-food Image Dataset (PFID) [3] and UNICT-FD889 Dataset [1], Food-101 is not a collection of standardized food images taken under controlled laboratory conditions. It is a collection of **real-world food images** downloaded from *foodspotting.com*, a site which allows users to take pictures on what they are eating, annotate with the type of food and upload this information online. It has a total of 101 food categories, with 1000 images each. Each class has 750 training images and 250 test images.

In this study, we will be using a subset of the Food-101 data set. We extract 20 very diverse but also visually and semantically similar food categories. The categories are as follows:

apple pie	baby back ribs	caesar salad	carrot cake	chicken curry
chocolate mousse	churros	dumplings	french fries	garlic bread
hamburger	hot dog	ice cream	pancakes	peking duck
pizza	spaghetti carbonara	spring rolls	steak	takoyaki

We extracted 150 images per category resulting to a total of 3,000 images. 2,000 food images (100 images per category) are used for training the classification model and the remaining 1,000 food images (50 images per category) are used for the training set in evaluating classification performance.

3.2 Challenges

The classification of food images is at the same time, an interesting and challenging problem since the high variability of the image content makes the task difficult even for current state-of the- art classification methods. The image representation to be employed in the classification process therefore plays an important role.

As stated in the previous section, images to be analyzed were taken in *"**real word**"* settings. Food images are observed to be noisy with irregular illumination, orientation and contrast. Furthermore, varying camera angles and overlapping food components can make classification difficult.

Figure 3.1 – Hamburger images with varying brightness and overlapping components

Figure 3.2 – Sample food images from the selected 20 different categories

Some classes also have great intra-class variability. Food, in general varies greatly in appearance (shape, colors) with different ingredient pairings and assembling methods.

In the figure below, all images are categorized as *apple pie*. Irregular orientation, illumination and the presence of other food component make the distinction of images within the same class difficult.

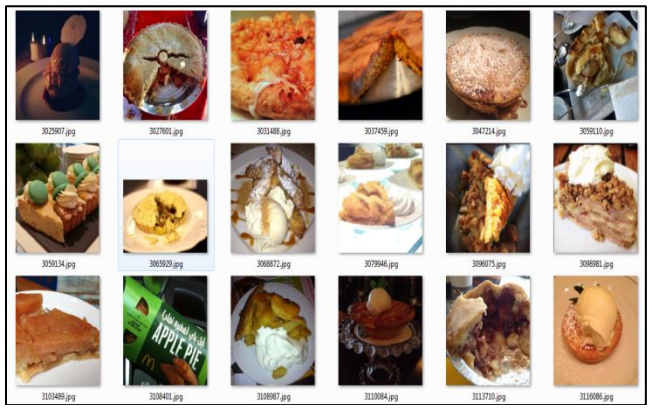


Figure 3.3 – Sample images belonging to *apple pie* category

Inter-class variability also becomes an issue in food classification. Images from **different** classes may be highly similar visually and pose a challenge even for humans to distinguish

Figure 3.4 – Visually similar food categories (*Row 1:* Baby back ribs; *Row 2:* Peking duck; *Row 3:* Steak)

Figure 3.4 contains food images from baby back ribs, peking duck and steak categories. We can observe that these images are visually similar in terms of their appearance, color and texture. The classification system needs to use the most discriminative features from the available data and weigh properly the training information coming from visually similar classes in the learning process to minimize category confusion.

3.3 Bag of Visual Words Model

The BoW model is commonly used in natural language processing and information retrieval for text documents [4]. In this model, a document is statistically modeled as an instance of a multinomial word distribution and is represented as a frequency of occurrence word histogram. The representation as a frequency vector of word occurrences does not take grammar rules or word order into account. It does, however preserve key information about the content of the document. Each visual word is assumed to be independent and each image is described by a set of order less local feature representation. A visual word can be considered to be a representative of several similar patches.

To represent an image using BoW model, the image must be represented as a document. The image must be broken down into a list of visual elements, and a way to discretize the visual element space, since the number of possible visual elements in an image is enormous. In the visual BoW model, the image feature extraction step takes place in a procedure involving detection of interest points, feature description and codebook generation. The model can thus take the form of a histogram representation of the image, based on a collection of its local features. Each bin in the histogram is a codeword index out of a finite vocabulary of visual words generated in an unsupervised manner of the data. Images are compared and classified based on this discrete and compact histogram representation.

Common points of interest detection approaches include using a regular sampling grid, a random selection of points, or selecting points with high information content using salient point detectors (SIFT or SURF descriptors).

The final step of the BoW model is to convert vector-represented patches into visual words and generate a representative *dictionary*. A visual word can be considered to be a representative of several similar patches. The vectors are clustered into K groups in the feature space. The resultant cluster centers serve as a vocabulary

of K visual words. Words are denser in areas with greater variability across images in the database. A given image can now be represented by a unique distribution over the generated dictionary of words.

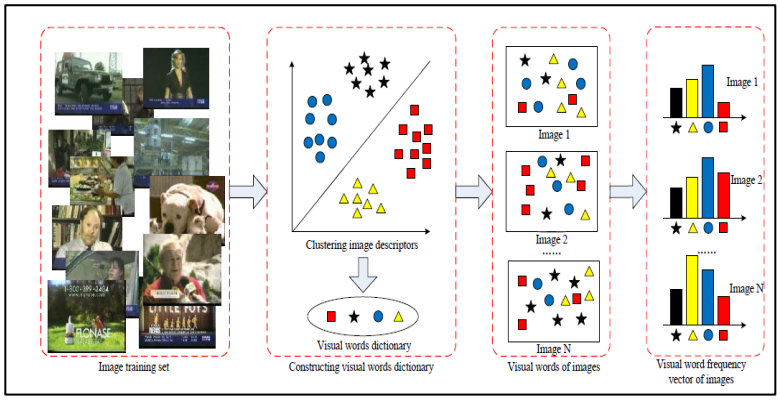


Figure 3.5 – Illustration of the BoW model

The BoW model includes four key steps**:**

(1) Local feature extraction– the essential aspect of the BoW concept is to *extract global image descriptors and represent images as a collection of local properties calculated from a set of sub-images called patches.*

(2) Code book representation - a code book or a visual dictionary is way that images can be represented as a set of local features. The idea is to group the feature descriptors of all patches and the representatives of all resulting clusters are then used as entries of the unified dictionary or codebook. The entries are called visual words.

(3) Feature quantization – after obtaining the codebook, each local feature is *quantized* or linked to one visual word using unsupervised learning algorithms such as nearest neighbor assignment or feature matching

(4) Image representation – after all the local features are mapped to the codebook, an image can be globally represented by the BoW frequency histogram of the visual words in the dictionary. Obtained histogram vector can be used as inputs for classification model like k-NN classifiers, SVM, Neural networks and other variants.

3.4 Features

The nature of food images is often defined by the different colors and textures of its different local components, such that humans can identify them reasonably well from a single image, regardless of the above variations. Hence, food recognition is a specific classification problem calling for models that can exploit local information.

Three major types of local features are generally used for image classification: color, shape and texture. Based on previous works, it was shown that shape or contour properties are less effective as compared with color and texture due to foods being intrinsically deformable and present high variability in their appearances. [5]

In this study, we will be utilizing scale-invariant key points (SIFT), color (RGB color histogram) and texture (Local Binary Patterns) as features for the classification of food images.

SIFT (Scale Invariant Feature Transform)

The scale-invariant feature transform has been proven to be one of the most robust among the local invariant feature descriptors with respect to different geometrical changes. It represents blurred image gradients in multiple orientation planes and at multiple scales. The SIFT algorithm has showed great success in object recognition and detection due to its **invariance in translation, scaling, rotation, and small distortions**. The basic idea is to identify the extreme points in the scale space, and filter these extreme points to find the stable feature points known as keypoints. The local attributes of orientation gradient and descriptors are computed, and the keypoints are described by 4x4x8 matrix **(128 dimension vector).**

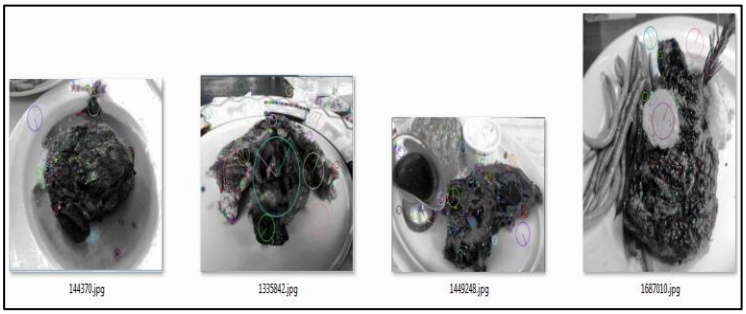


Figure 3.6 – Sample *steak* images with detected SIFT keypoints

Local Binary Pattern Operator (LBP)

The local binary pattern texture features are operators that describe the surroundings of a pixel by generating a bit-code from the binary derivatives of a pixel as a complementary measure for local image contrast. The LBP operator takes some of the neighboring pixels using the center gray value as a threshold and based on two parameters, i.e., P and R which represent the number of considered neighbor points and the radius of the considered circle respectively. It has been widely used in object recognition and achieved good results in face recognition problems. Its key advantages are its invariance to monotonic gray level changes, rotation and its computational simplicity.

The basic idea is to summarize the local structure in an image by comparing each pixel with its neighborhood. Each pixel becomes a center and its neighboring pixels are analyzed. If the intensity of the center pixel is greater-equal its neighbors, then it gains a value of 1 and otherwise. The result is a binary number for each pixel, such as *11001111*. With *8* surrounding pixels there are 2^8 possible combinations, which are called Local Binary Patterns or sometimes abbreviated as LBP codes.

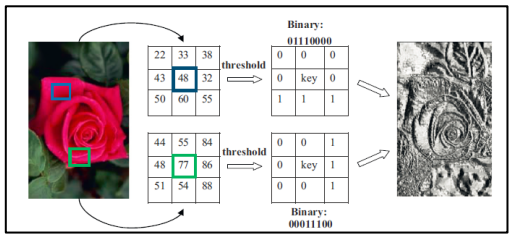


Figure 3.6– For each pixel, compare the pixel to each of its 8 neighbors. Where the center pixel's value is greater than the neighbor, write "1". Otherwise write "0".

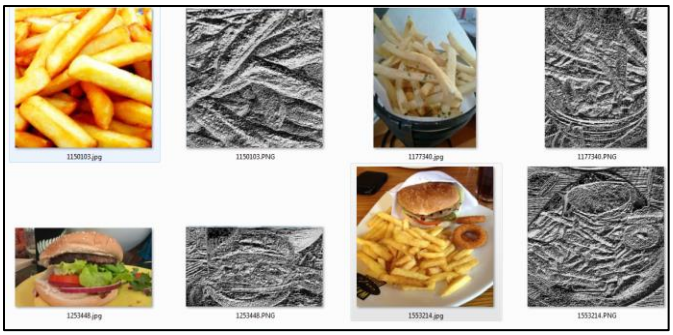


Figure 3.7 – Sample food images and corresponding Local Binary Patterns

RGB Color Histogram

A color histogram represents the number of pixels that have colors in each of a fixed list of color ranges that span the image's color space, the set of all possible colors. It is mostly used to show the statistical distribution of colors and the essential tone of an image.

Since we are working with B, G and R planes, we know that our color values will range in the interval [0,255]. For each color plane, we keep count of the number of pixels that fall in the range of each bin. 256 bins are used per color plane resulting to a total color vector of 256 x 3 (R, G, B) = 768-dimension color feature vector.

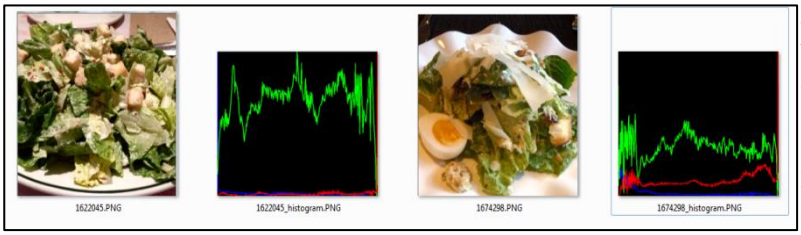


Figure 3.7 – Sample *Caesar salad* food images with color histogram

3.5 Image Feature Representation

Feature-Based Representation

In this method, we represent an input image globally by feature extraction from (1) color histogram and (2) texture histogram.

a) **Color (RGB) Histogram** – Given an input food image in RGB format, histogram equalization is first performed to attenuate image intensity and enhance contrast. We then calculate for the corresponding color histogram for the 3 channels and concatenate them to form an output 256 x 3 = ***768-dimension vector*.** The feature vector can now be used to represent the source image in either training and in testing.

b) **Texture (LBP) Histogram** – We used an existing implementation in OpenCV's `LBPHFaceRecognizer()` for recognizing faces using Local Binary Patterns. Before feeding the image into the model, we convert the RGB image into grayscale and do grayscale histogram equalization first. The model internally represents an image as a histogram of LBP textures calculated from the given training data.

Bag of Visual Words (BoW) Representation

a) SIFT- BoW

The basic idea of the SIFT-BoW representation is that a set of local image points is sampled by an interest point detector and visual descriptors are extracted by the Scale Invariant Feature Transform (SIFT) descriptor on each point. (*128-dimension vector*) Given a food image, we apply SIFT detection and produce an ***n x 128 dimension feature vector*** where n represents the number of keypoints detected,

b) SIFT-LBP BoW

On the recent works of Qin et.al [6], they have observed that SIFT descriptors perform poorly when the background is lacking of texture or is corrupted with noise due to the fact that SIFT fails to find stable keypoints in these cases. On the other hand, Local Binary Pattern has been proven to be a very robust texture filter to supplement the SIFT in filtering out the noises. It is expected that the characteristics of an object in an image can be better captured by integrating these two features. Thereby, the SIFT-LBP integration was introduced.

In extracting the local features for the Bag of Features model, a 128-dimensional SIFT descriptor is integrated with an 8x8 image patch of local binary patterns extracted on each SIFT keypoints. Local feature size is then calculated as 128 + 64 = ***n x 192*-dimension feature vector**.

c) Color SIFT-LBP BoW

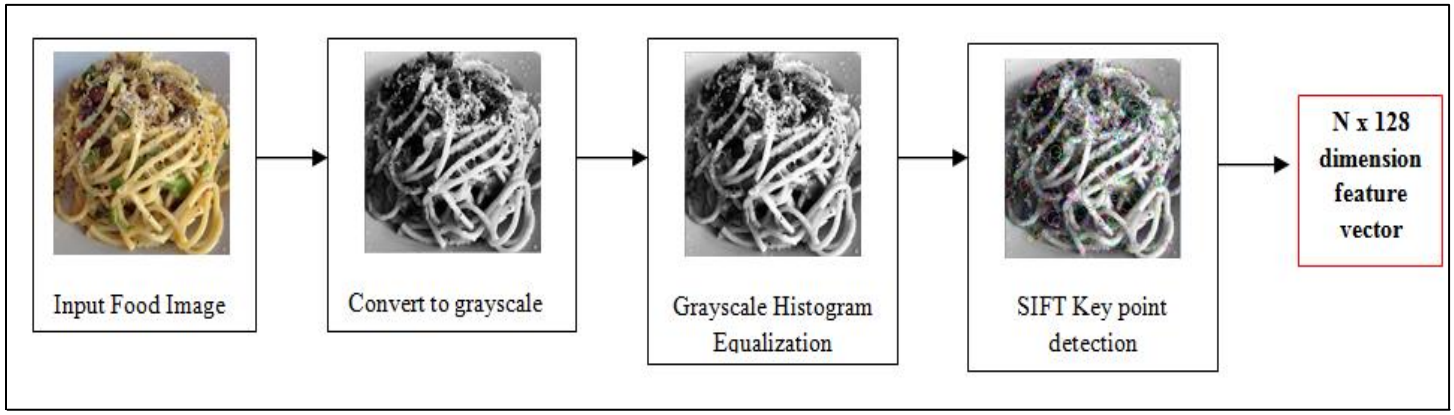
Traditional SIFT BoW model does not utilize any color information in image description. This may affect classification performance since most food categories are easily distinguishable to color as shown in Figure 3.7. We incorporate color information by integrating color histograms with texture and SIFT key points.

The approach is similar with SIFT-LBP BoW model. Aside from extracting an 8x8 image of local binary patterns from each SIFT key point, we also compute for the RGB color histogram of a 15x15 image patch centered on the key point. Feature dimension increases from the previous n x 192 feature vector to
n x 192 + (256 x 3) = ***n x 960 – dimension feature vector*.**

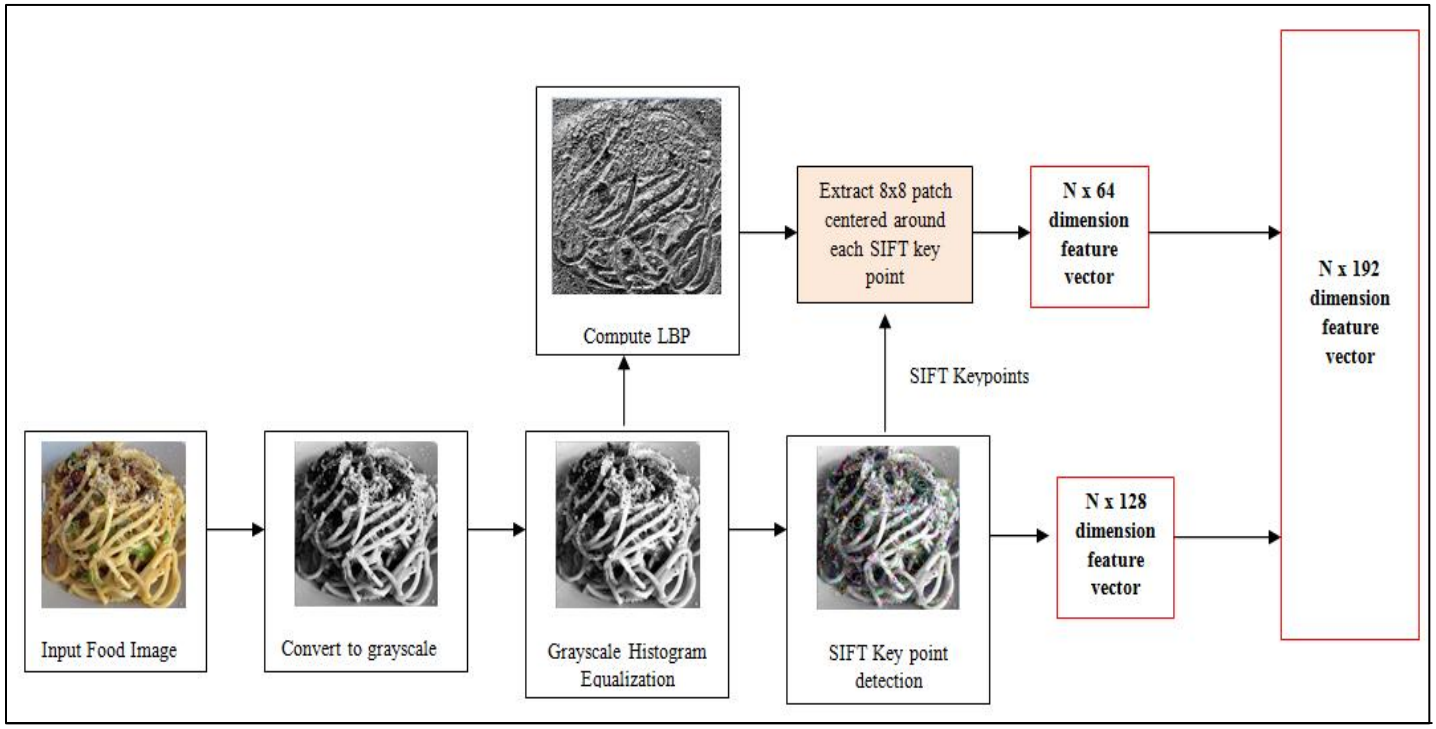
We present here a detailed step-by-step visualization on how the different techniques in representing images as bag of words work.

Input Image: spaghetti_carbonara/1482041.jpg

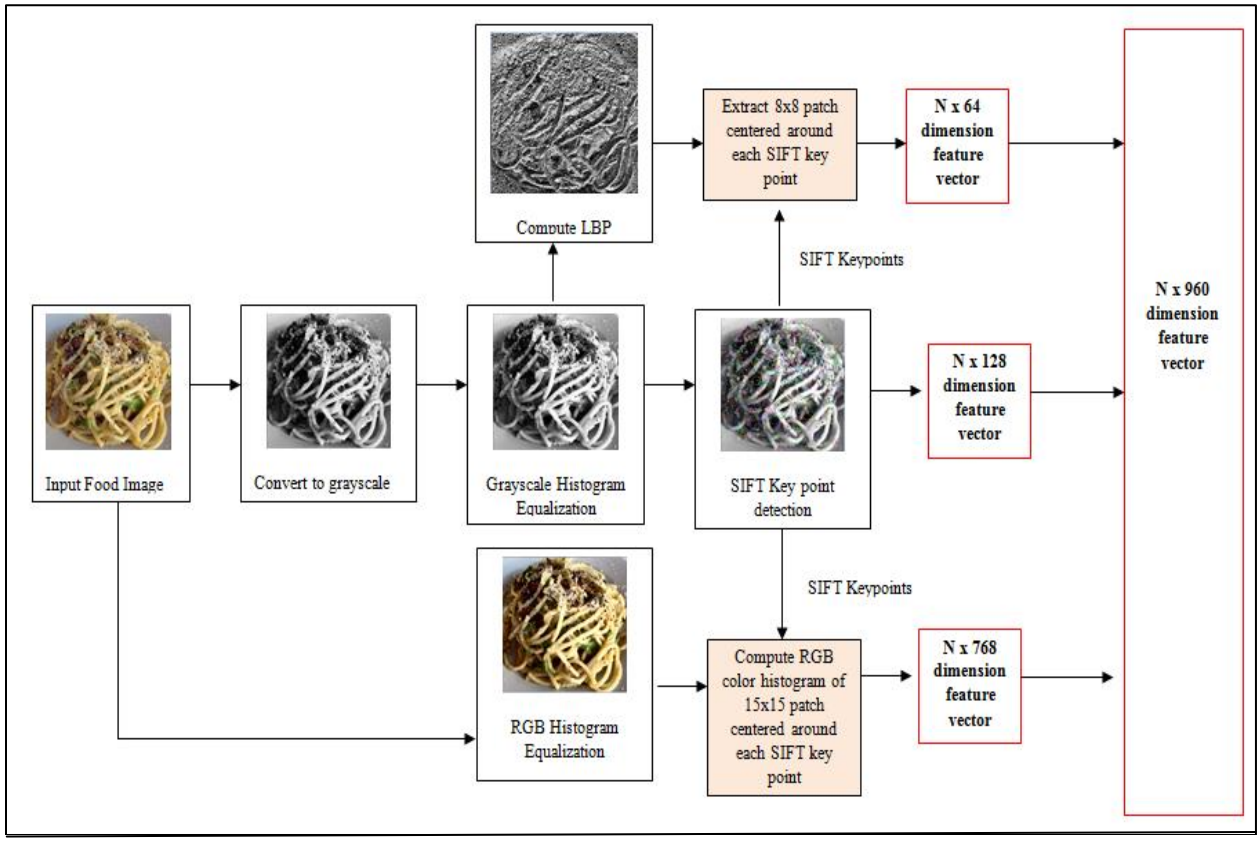
1) SIFT-BoW Method – output vector of *N x 128 features*



2) **SIFT- LBP BoW Method –** output vector of *N x 192 features*



3) **Color SIFT- LBP BoW Method –** output vector of *N x 960 features*



4. EXPERIMENTAL RESULTS

4.1 Overall Classification Performance

We used 2,000 food images (100 images per category) for training our food images classification models and 1,000 food images (50 images per category) as the validation set. Overall classification accuracy (number of correctly classified test images over the entire test set) is used as the overall performance measure. Support Vector Machines and k-Nearest neighbors are selected, for comparison as the classification models.

Experiment 1: RGB color and LBP texture histogram were used for the feature representation of an image in both training and testing.

Classifier	Overall Classification Accuracy
(1) Color – RGB Color Histogram	
K-NN(1)	8.40%
K-NN(5)	**9.70%**
K-NN(20)	8.20%
K-NN(50)	9.50%
Linear - SVM	8.40%
RBF - SVM	8.40%
(2) Texture – Local Binary Pattern Histogram	
OpenCV implemented object classifier model	**7.9%**

Table 4.1 – Experiment results on Feature-based representation

Experiment 2:
For each image in the training set, SIFT key point features are extracted. Then, these features are described by SIFT descriptors, LBP patches and color histograms depending on the feature representation variant used. To obtain the visual dictionary, K-means clustering is used with the specified number of visual words *k*. Finally, for each image in training and test set, a bag of word descriptor is generated according to the visual dictionary and is fed into the classifier models.

Classifier	Dictionary Size	Overall Classification Accuracy
(1) SIFT Bag of Words ***(128 dimension vector)***		
K-NN (20)	50	21%
K-NN (20)	100	20.7%
K-NN (50)	500	20%
K-NN (50)	1000	20%
K-NN (20)	2000	18.2%
Linear SVM	50	24.60%
Linear SVM	100	26.20%
Linear SVM	500	26.00%
Linear SVM	1000	28.80%
Linear SVM	2000	28.80%
RBF SVM	50	26.40%
RBF SVM	100	28.10%
RBF SVM	500	28.60%
RBF SVM	**1000**	**30.60%**
RBF SVM	2000	28.40%

Table 4.2 – Overall accuracy results on **SIFT-BoW**

Classifier	Dictionary Size	Overall Classification Accuracy
(2) SIFT-LBP Bag of Words ***(192 dimension vector)***		
K-NN (20)	50	18.60%
K-NN (20)	100	19.2%
K-NN (50)	500	17.9%
K-NN (20)	1000	17.9%
K-NN (20)	2000	14.8%
Linear SVM	50	24.60%
Linear SVM	100	25.60%
Linear SVM	**500**	**27.40%**
Linear SVM	1000	26.50%
Linear SVM	2000	24.30%
RBF SVM	50	25.20%
RBF SVM	100	25.50%
RBF SVM	500	26%
RBF SVM	1000	25.70%
RBF SVM	2000	25.30%

Table 4.3 – Overall accuracy results on **SIFT-LBP BoW**

Classifier	Dictionary Size	Overall Classification Accuracy
(3) Color SIFT-LBP Bag of Words ***(960 dimension vector)***		
K-NN (50)	50	16.10%
K-NN (50)	100	19.40%
K-NN (50)	500	20.20%
K-NN (50)	1000	19%
Linear SVM	50	20.50%
Linear SVM	100	21.20%
Linear SVM	**500**	**23.20%**
Linear SVM	1000	22.90%
RBF SVM	50	19.30%
RBM SVM	100	21.10%
RBM SVM	500	24%
RBM SVM	1000	23.10%

Table 4.4 – Overall accuracy results on **Color SIFT-LBP BoW**

Results from the 1st experiment found in Table 4.1 show that global feature-based representation (color and texture) is not very effective in classification of food images. We achieved the highest accuracy of 9.70% and 7.90% classification accuracy for color and texture, respectively. Even considering SVM instead of k-NN as the classification model does not merit noticeable improvement in classification performance. This supports the idea that food classification performs better when local properties such as patches, key points are exploited instead of a global representation of an image based on its color or texture histogram features.

In Experiment 2, we run different variants of the bag of words feature representation on our test set and as shown in Table 4.2, SIFT BoW using 1,000 visual words obtained the highest accuracy at 30.60% and an RBF-kernel SVM. SIFT-LBP model has the next highest performance (27.40% in Table 4.3) but with half the required dictionary size. Color SIFT-LBP achieved lower accuracy compared with the rest (23.20% highest in Table 4.4).

This suggests that at 960-dimensions for color, sift and texture, the classification model may be experiencing the curse of dimensionality wherein too much specified features may affect the discriminative computing capability of the model. Using RBF as the kernel type produces the best output (30.60%). However, the addition of texture and color features resulted in lower accuracy performance. Linear kernel SVMs provided better results for higher feature dimensions (SIFT-LBP and Color SIFT-LBP models) as compared with results using RBF kernels.

Results from the experiments show that incorporating local information on a bag of words model improves classification on food images. Experiment 2 bag of words results are significantly better than feature histogram models in Experiment 1. However, at an obtained best classification rate at 30.60%, food image classification is proven to be a difficult task, even for state-of-the-art feature representation and classification methods.

4.2 Category-level Classification Performance

In this section, our objective is to have a more comprehensive understanding on how computer vision applies to ***each*** food category instead of looking on it at an overall classification level. The expressiveness of a food category to a feature representation can be better observed in this type of analysis.

Aside from overall accuracy measure, category retrieval-specific measures such as precision and recall were also used for assessment. In classification problems, recall pertains to the total classification accuracy per food category *(e.g. out of the 50 carbonara images in the test set, 31 are correctly classified as carbonara, therefore, recall is 31/50 = 62%*). Precision, on the other hand pertains of the "*purity*" of a category retrieved upon classification of all images. (e.g. *out of the 54 images which are classified as carbonara, only 31 images are correctly belonging to the carbonara category so precision of the carbonara category is – 31/54 = 57.40%*).
For the formal definition:
Precision (P) is a measure of the accuracy provided that a specific class has been predicted

$$\textbf{\textit{Precision(P)}} = \frac{true\ positives}{true\ positives + false\ positives}$$

Recall (R) is a measure of the ability of a prediction model to select instances of a certain class from the data set. It is computed using the below formula:

$$\textbf{\textit{Recall(R)}} = \frac{true\ positives}{true\ positives + false\ negatives}$$

In general, we want to get a good value of recall while tolerating only a certain percentage of false positives. A single measure that trades off precision versus recall is the F_1-Measure which is the weighted harmonic mean of precision and recall computed as:

$$\textbf{F}_1\ \textbf{Score} = \frac{2 \times precision \times recall}{precision + recall}$$

F_1 Scores were computed from the results of the **best** performing experiment setup for SIFT, SIFT-LBP and Color SIFT-LBP obtained from Experiment 2 (Tables 4.2, 4.3, 4.4).

Food Category	F-SCORE		
	SIFT	SIFT-LBP	Color SIFT-LBP
apple pie	**0.17**	0.11	0.16
baby back ribs	**0.35**	0.28	0.32
caesar salad	0.33	0.33	**0.60**
carrot cake	**0.21**	0.20	0.17
chicken curry	0.18	0.08	**0.25**
chocolate mousse	**0.24**	0.21	0.14
churros	**0.42**	0.32	0.06
dumplings	**0.52**	0.53	0.25
french fries	**0.57**	0.43	0.19
garlic bread	**0.37**	0.27	0.28
hamburger	**0.22**	0.20	0.07
hot dog	0.20	0.23	**0.24**
ice cream	0.23	**0.33**	0.23
pancakes	0.24	0.22	**0.27**
peking duck	0.20	**0.23**	0.14
pizza	0.29	0.37	**0.38**
spaghetti carbonara	**0.64**	0.51	0.32
spring rolls	0.23	**0.10**	0.09
steak	0.18	0.22	**0.28**
takoyaki	0.11	**0.20**	0.18

Table 4.5 – F_1 score performance for the best classifier model with the highest values per category, highlighted in yellow

In Table 4.5 above, 10 out of the 20 categories had the highest F-score when only SIFT descriptors were used. With an F-Score of 0.64, spaghetti carbonara was the category which SIFT descriptors can best represent. This was followed by french fries (0.57), dumplings (0.52) and churros (0.42). Food categories in which SIFT descriptors were shown to be least effective are hot dog (0.20), peking duck (0.20), steak (0.18), chicken curry (0.18), apple pie (0.17) and takoyaki (0.11).

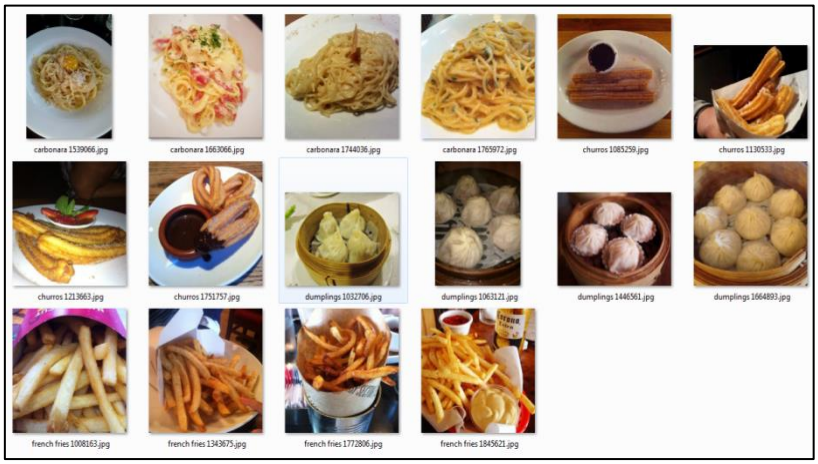


Figure 4.1 Sample images from carbonara, churros, dumplings and French fries food categories

Incorporating local binary patterns as texture descriptors improves classification on ice cream images (F-score increases from 0.23 to 0.33) and pizza (Increased from 0.29 to 0.37). There are also observed improvements on steak, takoyaki and hotdog category F-Scores when using SIFT-LBP model. However, a more evident result is that the task difficulty is increasing as we also increase the number of features from 128 to 196 dimensions. Most categories experience a significant F-score drop when texture is incorporated (e.g. carbonara F-score drops from 0.64 to 0.51).

The integration of texture and SIFT key points on food images, in general does not provide an improvement on ***recall*** performance (per-class classification). What it does is improve ***precision*** performance seen in *pizza* and *ice cream* food images *(resulting to higher F-scores than the SIFT-only model counterpart*). Having an equivalent recall but a higher precision means that the model produces the same number of images correctly classified in a category but with fewer numbers of images misclassified into that category. Integrating texture feature information makes the model "*more conservative*" in assigning images to texture-intensive categories (e.g. ice cream, pizza) resulting in categories with fewer false positives.

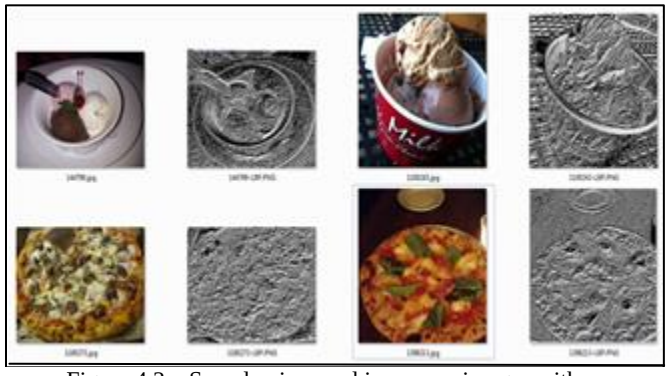


Figure 4.2 – Sample pizza and ice cream images with corresponding Local Binary Patterns

In the case of Color SIFT-LBP model, a lower F-score compared with the other two methods is observed in most food categories. F-score for churros images drops to 0.06 from 0.42 (SIFT) and 0.32 (SIFT-LBP). Both Caesar salad and chicken curry's F-scores, however improves when color feature is included. In the below figure, Caesar salad and chicken curry images are more expressive to color due to their intrinsic single color dominant feature. For example, we are expecting a more concentrated green channel histogram when processing Caesar salad images.

Figure 4.3 - Chicken curry and Caesar salad sample images

Top Food Categories	Mean F-Score	Bottom Food Categories	Mean F-Score
spaghetti carbonara	0.49	takoyaki	0.17
dumplings	0.43	chicken curry	0.17
caesar salad	0.42	hamburger	0.16
french fries	0.40	apple pie	0.15
pizza	0.35	spring rolls	0.14

Table 4.6 – Top and bottom food categories based on their Mean F-score

Figure 4.4 – Sample images belonging to the bottom categories

5. DISCUSSION OF RESULTS

5.1 Effect of Number of *K* Visual Words in Dictionary

Increasing the number of K words to be used in constructing the dictionary does not guarantee a better classification performance on food images. Convergence in unsupervised learning using K-means is strongly linked to the input dictionary size and the number of feature dimensions (*Constructing a 1,000 word dictionary for Color SIFT-LBP model -196 dimensions took more than 15 hours to complete*). The figure below shows that the best accuracy (30.6%) was achieved using 1,000 visual words on SIFT-BOW model.

For both SIFT-LBP and Color SIFT-LBP models, accuracy gradually improves as dictionary size is being increased from 50 to 500. However, an accuracy drop is encountered when dictionary size approaches 1,000. For the SIFT model, the best accuracy is achieved at word size 1,000 and goes down at size 2,000.

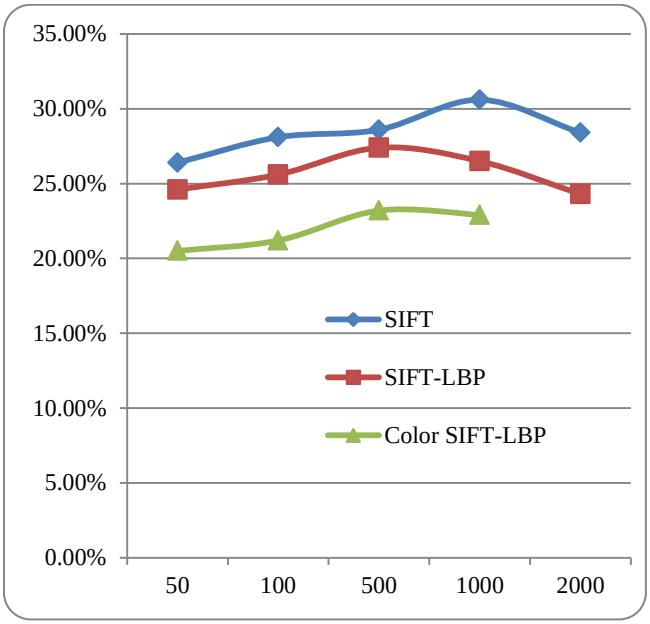


Figure 5.1 – Increasing dictionary size vs overall accuracy

5.2 Feature Selection by Information Gain Ratio

Feature selection is a technique to reduce the dimensions of the data to its most significant elements. As dimensionality can reduce the performance of a classifier, we produce experiments to reduce the feature size of the code book. The setup is to take the information gain ratio of each feature, and take progressively smaller percentages of the total number of features. Information gain is a measure of deciding the relevance of an attribute. It can be described as the reduction of entropy of a distribution, by learning the state of a random variable. Information gain ratio is an extension to reduce the bias towards variables with a large number of possible states.

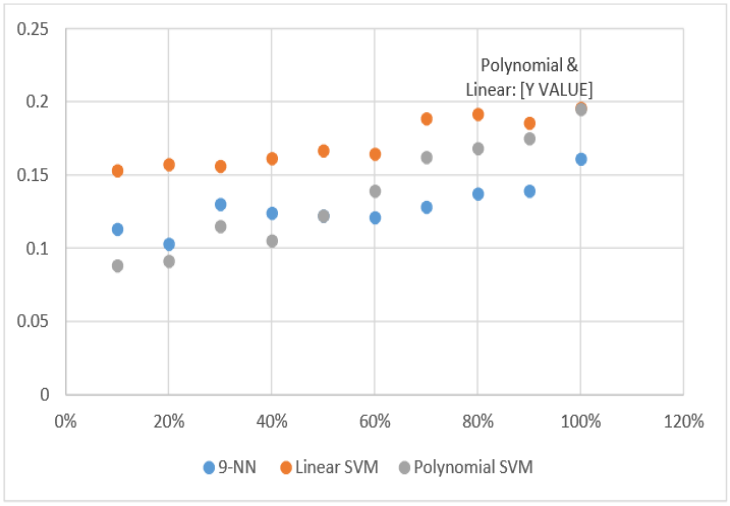


Figure 5.2 - Feature Selection on SIFT-BOW

The results are described at Figure 5.2. It is clear from the experiments that this does not boost the performance of our classifiers. Even k-NN, which is particularly vulnerable to the curse of dimensionality, has its performance reduced. This is explained by the quantization procedure from the SIFT feature descriptors. The bag-of-words representation creates a dictionary with a fixed word size and word length. More precisely, k-means have been used to extract cluster prototypes equal to the specified word length that serve well to represent unique feature descriptors for each image in the dataset. Reducing this initial number significantly worsens the predictive capability of our classifiers due to the fact that the cluster prototypes represent more poorly the feature descriptors of our dataset.

5.3 Nonlinear Support Vector Machines

We have carried out experiments with a support vector machine with a polynomial kernel of degree 2 and complexity constant set to 1. Our dataset has multiple categories, so we set our training scheme to one against all. The results are in Table 4.7. The results are of the same performance as our linear support vector machines.

FEATURE SET	ACCURACY
SIFT	21.5%
SIFT-LBP	19.51%
SIFT-LBP-COLOR	19.5%

Table 5.1. Classification performance for our polynomial degree SVM's with dictionary size of 1000 words

5.4 Confusion Matrix

A confusion matrix is a specific table layout which allows the visualization of the precision-recall trade-off performance of the food classification model. It contains information about the actual and predicted classifications predicted by the model. The below confusion matrix is created from the classification results of the 1,000 visual word SIFT-BoW model *(30.60% overall accuracy)*.

	apple pie	baby back ribs	caesar salad	carrot cake	chicken curry	chocolate mousse	churros	dumplings	french fries	garlic bread	hamburger	hot dog	ice cream	pancakes	peking duck	pizza	spaghetti carbonara	spring rolls	steak	takoyaki	**Recall**
apple_pie	9	2	3	3	4	5	0	2	2	4	0	2	0	3	3	4	1	3	0	0	18%
baby_back_ribs	1	18	0	1	4	1	1	0	1	2	3	3	3	2	2	1	0	0	5	2	36%
caesar_salad	1	2	20	3	0	1	1	0	2	2	3	1	1	1	1	3	3	0	1	4	40%
carrot_cake	1	3	4	10	2	5	3	2	2	4	0	2	1	2	1	2	1	2	0	3	20%
chicken_curry	1	3	3	5	7	0	7	1	0	6	3	0	2	3	3	1	2	2	0	1	14%
chocolate_mousse	2	1	0	2	1	12	6	5	1	2	3	0	4	1	4	1	1	3	1	0	24%
churros	5	0	0	0	1	3	26	1	0	3	2	0	2	1	3	0	0	2	1	0	52%
dumplings	0	0	0	0	0	3	4	27	6	0	1	3	2	0	2	0	1	1	0	0	**54%**
french_fries	1	0	0	1	0	3	3	3	33	0	1	2	2	0	0	0	0	1	0	0	**66%**
garlic_bread	3	0	2	6	0	1	2	0	1	22	1	2	1	2	1	4	0	1	1	0	44%
hamburger	2	0	3	0	0	2	2	0	4	3	12	6	3	0	5	1	3	0	2	2	24%
hot_dog	3	0	3	0	1	4	2	2	6	0	6	10	3	1	3	2	0	4	0	0	20%
ice_cream	5	0	4	1	2	3	5	5	1	0	2	3	11	2	0	0	2	2	0	2	22%
pancakes	4	7	0	2	1	3	2	2	0	5	3	1	0	10	5	1	0	1	2	1	20%
peking_duck	3	6	0	2	0	1	2	0	1	1	4	3	0	3	11	7	0	2	2	2	22%
pizza	2	3	9	0	0	2	1	0	1	6	3	0	2	1	3	14	2	0	1	0	28%
spaghetti_carbonara	0	0	3	1	0	0	1	0	0	2	1	1	2	0	1	1	33	1	2	1	**66%**
spring_rolls	0	0	1	3	0	3	2	2	4	2	2	8	3	1	8	0	1	10	0	0	20%
steak	5	7	1	4	4	0	1	1	0	3	2	0	3	2	2	4	2	0	7	2	14%
takoyaki	5	2	15	3	2	0	2	1	0	2	5	2	1	0	1	1	1	2	1	4	8%
Sum	53	54	71	47	29	52	73	54	65	69	57	49	46	35	59	47	53	37	26	24	
Precision	16.98%	33.33%	28.17%	21.28%	24.14%	23.08%	35.62%	**50.00%**	**50.77%**	31.88%	21.05%	20.41%	23.91%	28.57%	18.64%	29.79%	**62.26%**	27.03%	26.92%	16.67%	

Figure 5.3 – Confusion matrix using 1,000 visual-word SIFT BoW model

The below points contain the summary of findings upon observation and analysis of the computed confusion matrix.

- We obtained the highest recall values for *dumplings*, *french fries* and *spaghetti carbonara* at recall values of 54%, 66% and 66%, respectively. The three food categories also posted the highest precision values (46%, 48% and 57%). This suggests that the model performs well on these set of images and at the same time contain a tolerable percentage of false positives.
- High intra-class variability in preparation and presentation can be observed thru an almost uniform distribution among misclassified images on the entire category set. For example, apple pie images are more likely to be misclassified as chicken curry, chocolate mousse, garlic bread and caesar salad. As reference to Figure 3.3, the pose variations in apple pie images (pie slice, whole pie, close up shot) and presence of other food components (ice cream, whipping cream, and fruit toppings) make classification more challenging.
- We can also observe the model's confusion to categories with high inter-class variability. Baby back ribs images are most likely to be confused with steak images and vice versa. Similarly, Peking duck images are most likely to be confused with baby back ribs.
- Confusion between takoyaki and caesar salad are also evident due to the presence of garnishes and greens which cause occlusion or visibility reduction to the primary food component. Misclassified takoyaki to caesar salad images (15 images) far exceed the number of correctly classified takoyaki images (4).

Figure 5.4 – Sample *caesar salad images*

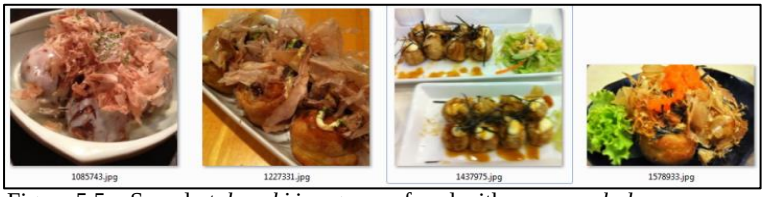


Figure 5.5 – Sample *takoyaki* images confused with *caesar salad*

- By looking at the confusion matrix, we can observe that 40% (20 images) of the test spring roll images are belonging to the following categories: french fries (4 images), hot dog (8 images) and peking duck (8 images). Similarities between caesar salad and takoyaki food images are presented in the below figure.

Sample spring roll images confused with **french fries**: Similarity with golden brown color and cylindrical-like shape.

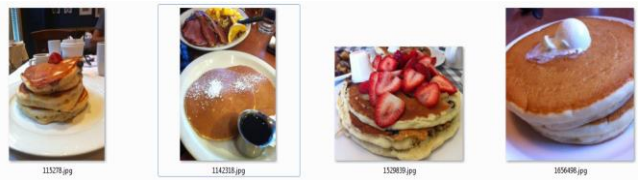
Sample spring roll images confused with **hot dog**: Similarity with hot dog-like shape and reddish parts evident in the spring roll wrapper

Sample spring roll images confused with **peking duck**: Similar brownish color with peking duck images. Garnishes and sauce which are usually present to peking duck images can be seen in spring roll images

Figure 5.4 – Sample images of spring roll and visually similar categories

- Pancake food images are most likely to be misclassified to peking duck and apple pie food categories. Simple food variations such as extra decorations, camera close up shots and plate positioning can greatly increase the likelihood of food image misclassifications.

Sample pancake images confused with apple pie



Sample pancake images confused with peking duck

Figure 4.5 – Sample pancake images misclassified to apple pie and peking duck categories

5. CONCLUSION

In this study, we applied different image classification techniques for the analysis and understanding of food images. Our experiment results show that using Bag of Visual Words representation can achieve better classification results as compared to the traditional feature-based methods. Three variants of Bag of Word representation were implemented: SIFT descriptors, SIFT-LBP descriptors and Color SIFT-LBP descriptors. SIFT descriptors achieved the highest overall accuracy rate of 30.6%. Category-level analysis was performed to understand the effectiveness of feature representation methods (SIFT, texture, color) on the classification accuracy of each food category.

Towards the end of the study, we discuss classification effectiveness when increasing the number of visual words in the dictionary, using nonlinear SVM classifiers and performing feature selection. Indeed, processing food images captured in a real word environment is shown to be a difficult computer vision problem. More sophisticated methods such as food boundary segmentation and component–level analysis may also be considered to address the issues encountered in food images analysis and further improve food classification performance.

REFERENCES

[1] L. Bossard, M. Guillaumin and L. Van Gool, "Food-101 -- Mining Discriminative Components with Random Forests," in European Conference on Computer Vision, 2014.

[2] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar and J. Yang, "PFID: pittsburgh fast-food image dataset," in 16th IEEE international conference on Image processing (ICIP'09), Piscataway, NJ, USA, 2009.

[3] G. M. Farinella, D. Allegra and F. Stanco, "A Benchmark Dataset to Study the Representation of Food Images," in International Workshop on Assistive Computer Vision and Robotics, 2015.

[4] U. Avni, H. Greenspan, E. Konen, M. Sharon and J. & Goldberger, "X-ray categorization and retrieval on the organ and pathology level, using patch-based visual words," in Medical Imaging, IEEE Transactions on, 30(3), 2011.

[5] G. M. Farinella, M. Moltisanti and S. Battiato, "Classifying food images represented as Bag of Textons," in 2014 IEEE International Conference on Image Processing, Paris, 2014.

[6] Yu, J., Qin, Z., Wan, T., & Zhang, X. (2013). Feature integration analysis of bag-of-features model for image retrieval. *Neurocomputing*, *120*, 355-364.

6. APPENDIX

We have used the computer vision library, OpenCV in our study. Our code is in C/C++, with the software architecture in Figure 6.1. We have divided our project to two main methods, which can serve to separate the time-consuming feature extraction stage FeatureExtractor class) and the repeatedly used evaluation stage (Evaluator class). We have a number of extractor classes, which reflect the features that we used for this study. We also have utility code separated from our main components

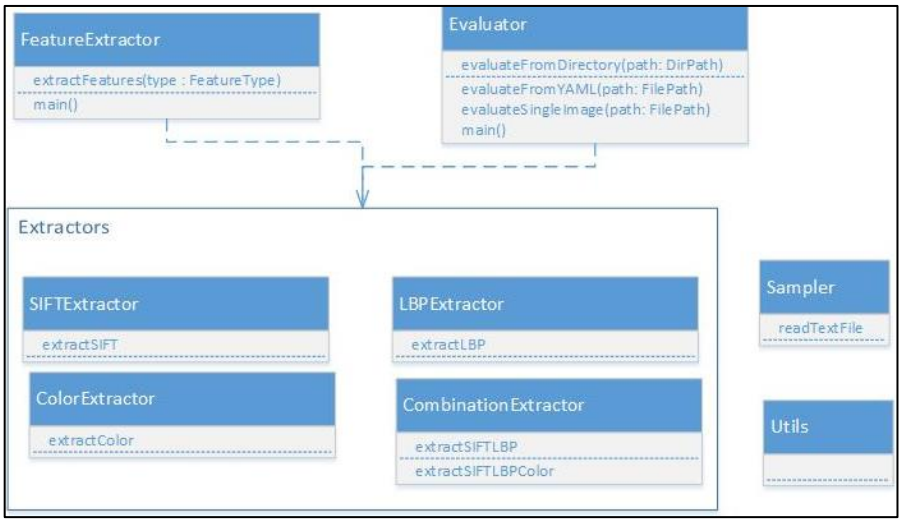


Figure 6.1 – Software Architecture

7. Review of Related Literature

Food recognition is a topic of research with many promising results. The Bag of Words (BoW) model has been used in conjunction with feature extraction techniques such as SIFT, SURF and Textons.

Battiato et al illustrated how the BoW model can be with Textons [5]. Using the PFID dataset, they extracted Textons, which attempts to capture an image's texture. More precisely, they used a set of filters, from the Maximum Response filter bank, which is composed by filters computer at multiple orientation and scales. A 4-dimensional vector for each color channel is associated to every pixel of the food images. They also included in their dictionary a class-based quantization. Instead of clustering words from the entire dataset, they generated words from each class. They argue that each class-based vocabulary is more expressive than the global-based vocabulary, which is related to other classes in the dataset. They have achieved results significantly better than BoW generated from SIFT.

Farinella et al also used Textons and compares it with Local Binary Patterns (LBP), and SIFT-based BoW models in a near duplicate information retrieval problem [3]. They used a variant of LBP, Pairwise Rotation Invariant Co-occurrence Local Binary Patterns (PRICoLBP) which preseves the relative angles between the orientations of LBP feature pairs. They used the $\chi 2$ metric to measure similarity between two different samples with the PRICoLBP descriptor. They used the UNICT-FD889 Dataset which is a collection of 889 distinct plates of food with 4 samples each. For evaluation, they used information retrieval metrics such as the probability of a successful query with relation to a ranking of search results. Their Texton features achieved the best results.