Epicure: A Meal Recognition System

Team Tarang Chugh Farhan Hormasji Max Ellison

Objective

To develop a comprehensive and robust meal-recognition system (Epicure), consisting of different modules of a computer vision system. And to understand the interdependence and working of various sub-modules for preprocessing, segmentation, feature extraction, classifier training, testing and cross-validation.

Epicure is able to recognize 10 different classes of food: Apple, Banana, Broccoli, Cookie, Egg, French Fries, Hot Dog, Rice, Strawberry and Tomato.

Literature Review

Some of the published work which we referenced and found interesting is as follows:

Yang et. al. [2] proposed a food recognition system specialized for hamburgers, pizza and tacos where they trained models for the raw food materials like bread, cheese, etc. They classified the food class based on the detected materials and their physical relations achieving an accuracy of 28.2%.

Zong et al. [3] developed a food recognition system based on the SIFT (Scale Invariant Feature Transform) and LBP (Local Binary Pattern) and achieved a better accuracy that Yang et. al.

There are various other approaches based on SIFT based Bag-of-features, Gabor, Color Histograms, Multiple Kernel Learning, etc. and have achieved accuracies of up to 61% on popular American Fast Food Database [4]

Dataset

Epicure is designed to robustly classify 10 different classes of food namely: Apple, Banana, Broccoli, Cookie, Egg, French Fries, Hot Dog, Rice, Strawberry, Tomato.

Into the Wild Dataset

Training Dataset: The training dataset consists of 500 food images. There are 50 food images for each of 10 different classes. These images are collected from internet [5,6] and captured by cellphone camera. In all the training images, food is present in natural setting. To make the system robust and ready for real world application, there are images in training dataset with different view angles and illumination conditions.

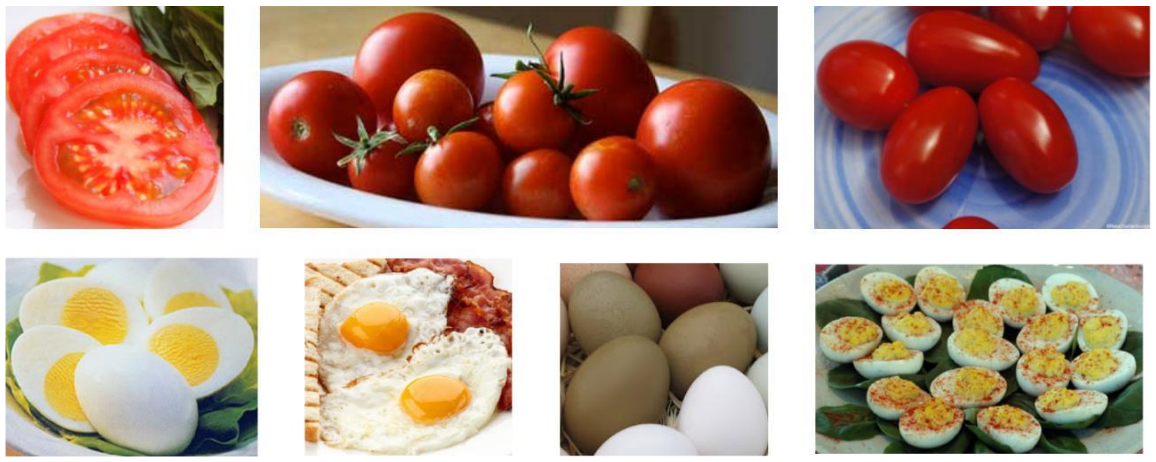


Fig 1. Sample images from training dataset of tomato and egg class illustrating varied view angles, shape forms and illumination

An additional set of 40 images belonging to Utensils class is also present in the dataset. This class of images is present to train a model of commonly occurring background/noise to filter out the features we extract to distinguish food classes.

Fig 2. Sample images from utensil class.

Test Dataset: The test dataset consists of many food images with some images having multiple class labels.

Fig. 3 Sample images from testing dataset

Conducive Dataset

We also generated a much conducive secondary dataset with comparatively constrained background settings, in order to analyze the impact of background noise on accuracy.

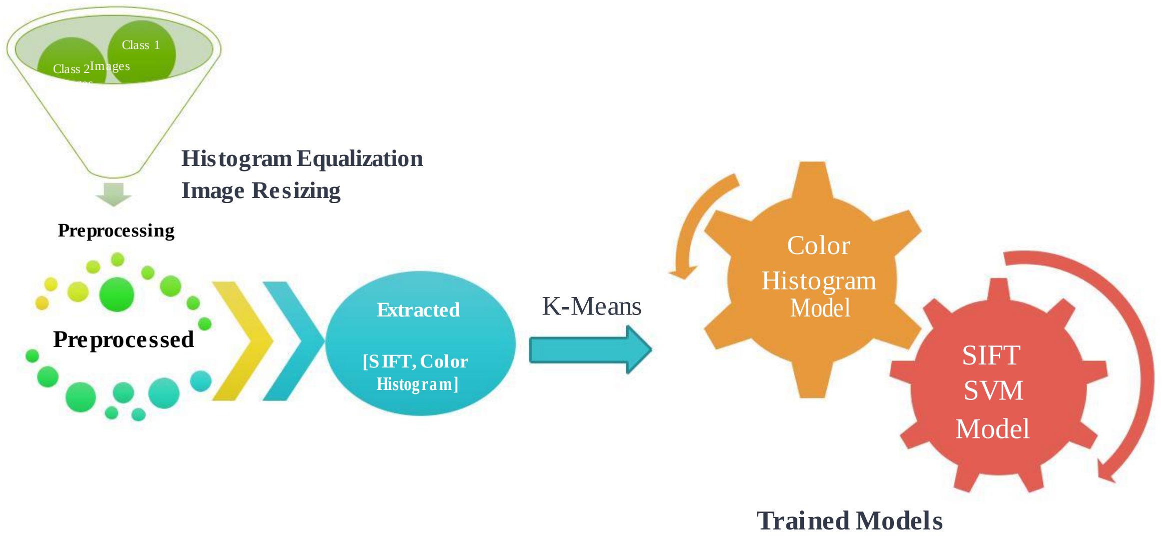
This dataset consists of 7 classes: Apple, Banana, Broccoli, Cookie, Egg, Rice, Strawberry, Tomato.

Some of the images from the training dataset (715 images in 7 classes) are as follows:

Some of the images from the testing dataset (70 images for 7 classes) are as follows:

Algorithm

Flow Diagram of the proposed training algorithm



Preprocessing

Histogram Equalization

Each image in training and testing dataset is preprocessed for contrast enhancement by performing histogram equalization on intensity values.

We first convert the RGB image to HSV image and perform the histogram equalization on V (intensity channel). This approach is much more robust and computationally efficient than performing the histogram equalization on each of RGB channels.

Image Resizing

To save computation power, we ran an iterative approach to identify the resolution above which it does not make much of a difference in accuracy. In our experiments we found that if we put the max(dimension of image) = 1000 pixels, the results are comparable to original large images.
Note: We only reduce the size of the larger images.

Feature Extraction

Scale Invariant Feature Transform (SIFT)

SIFT [7] is widely used in content-based image-retrieval applications, because of its high computational efficiency and resiliency to scale variations. It uses gradient orientation histograms to identify scale and orientation invariant key points (feature descriptors).

Exploring the image on different scales, it localizes and filters the key points which are stable over various affine transformations. Common examples of such points are corners and those with good contrast.

The SIFT descriptor is computed over a region of interest, around a detected key point. It samples the orientation and magnitude of image gradient, resulting in a 128 dimensional feature vector.

For a given image, SIFT key points are extracted and for each key point a 128 dimensional feature vector is obtained. SIFT Models are computed for each class by combining the key points from each image in that particular class.

K-Means

Inherently, individual SIFT descriptors are local key points and do not represent the global shape of the object. In order to utilize these key points to form a global signature of a food class, it is required to identify the spatial relation between them. We use the K-means clustering algorithm [8] to obtain clusters from the set of extracted key points from each image. Using an iterative approach, we found that K=10 clusters, works best for the problem at hand.

Since we have 50 images for each class, and 10 SIFT feature clusters for each image, the size of each SIFT Model is 500 x 128.

Color Histogram

To form a color histogram for an image, regardless of whether it is training or testing, first we segment the image using k-means. We used 2 for k, as this worked best at separating the food from the background for most training images. In order to separate the foreground segment from the background, we took the area of the bounding box for both segments. The segment with the larger area was deemed the background.

Once the foreground image was identified, the image was converted from RGB to HSV. RGB histograms were tested, but resulted in too many false positives and did not reduce the search space enough. HSV is typically chosen for computer vision applications to help reduce noise in the color information and separate hue from intensity. As will be discussed later, hue was most useful in matching objects by color. The intensity histogram was equalized, and all histograms were normalized.

Training

Support Vector Machines

We learnt the binary class Support Vector Machine [9] Models (one vs. rest model) on SIFT Feature Vector for each of the 10 food classes. Each model was trained on 1000, 128-dimensional feature vectors.

To avoid any bias, these 1000 feature vectors consisted equal number of genuine and impostor class features. Each class already had 500 clustered feature vectors and for rest, we randomly selected 500 out of the remaining classes.

The SVM was trained on both dot-product (default) kernel function and RBF kernel function. Since we achieved much higher accuracy with RBF as compared to dot-product kernel function, we report only RBF kernel accuracies.

Color Histogram Models

To determine if which class a test image belongs to using color, we compare each of their histograms. H, S, and V are combined into a 36 bin histogram. Then the Manhattan distance between the test image and every image in every class is found. The minimum Manhattan distance between the test image and each class is ranked and stored in an array, and this acts as the first classifier.

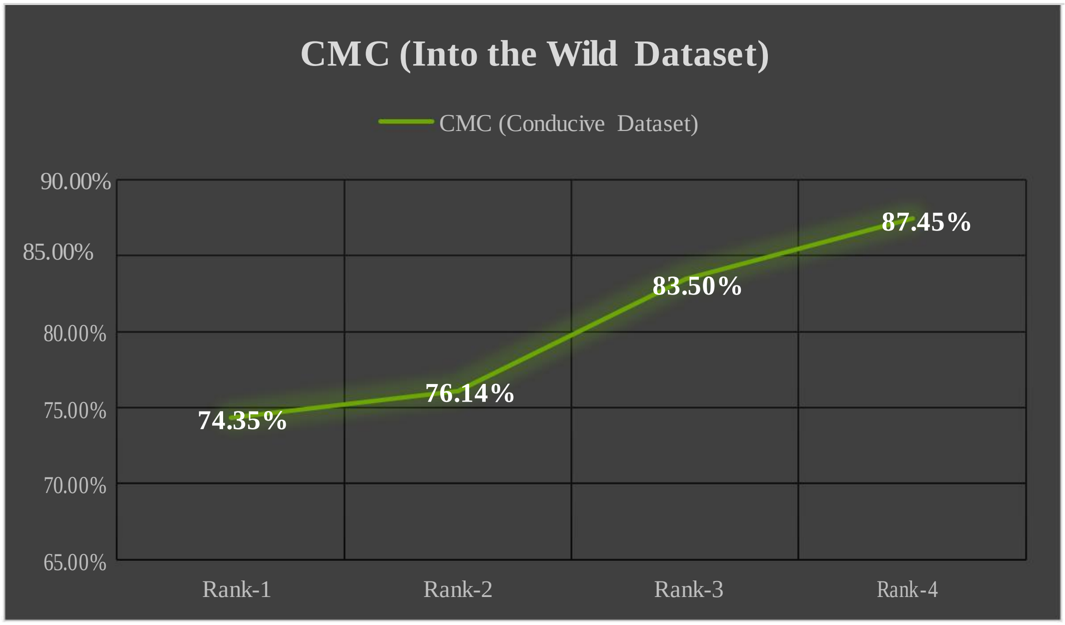
Testing

To test our image, first we read in the RGB image. We use k-means clustering, with k=2, so separate the background form the foreground. Background is identified by the area of the bounding box. We then convert the image to HSV and create a histogram of H,S, and V. This histogram is compared to the histogram of all the training data. By taking the minimum distance of the test image to each class, we can create a ranking of similarities. The top 3 classes in this ranking are then passed to the SIFT classifier. A sift model of the test image is created, and svmclassify is called on this model and each of the top 3 classes. The class with the most features matched is the class the test image is classified as.

Results

Into the Wild Dataset

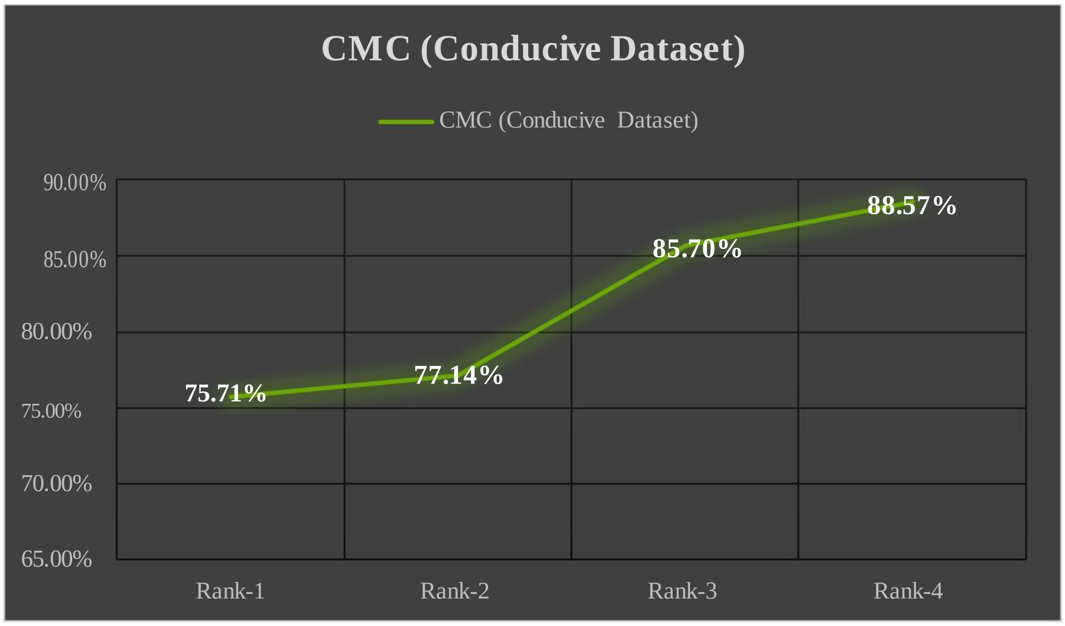
In the Into the Wild dataset, our system is able to achieve an accuracy of 74.35% at Rank-1 accuracy, which increases to 87.45% at Rank-4.



Time Efficiency: 89.40 seconds

Conducive Dataset

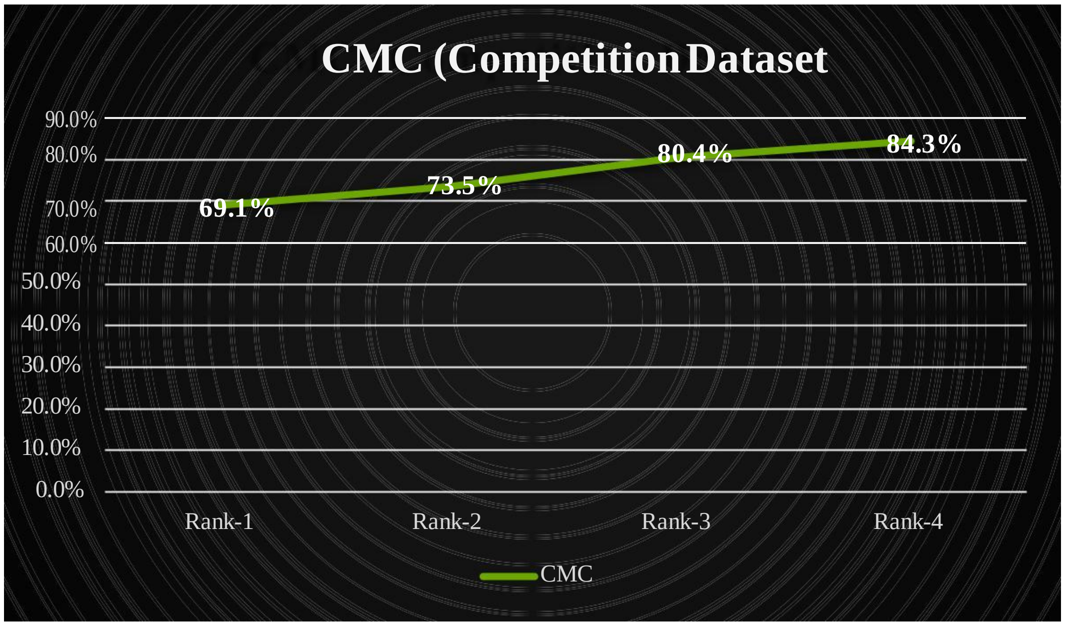
In the conducive dataset, our system is able to achieve an accuracy of 75.71% at Rank-1 accuracy, which increases to 88.57% at Rank-4.



Time Efficiency: 88.40 seconds

Competition Dataset

Our system is able to achieve 69.1% of Rank-1 Accuracy, which increases to 84.3% at Rank-4.



Time Efficiency: 57.772 seconds.

Future Work

Although our proposed algorithm achieves an accuracy less than the state of the art algorithms, but we have learnt many lessons while working on this project. Some of them are listed as follows:

1) It gets very hard to balance the trade off between accuracy and time requirement / computational efficiency of the system.
2) Similarly, there exists a thin line between generalization and over-fitting of the model.
3) Sometimes feature extractors don't behave as we intuitively think and want them to behave.
4) Segmentation is a big challenge.
5) More the training dataset, better it is. The accuracy drastically improved when we increased the dataset from 20 training images to 50 training images for each class.
6) Need to think intuitively about the relations/ patterns that exist and suitably optimize the parameters of various computer vision modules (feature extraction, training classifier, etc.)

Lessons Learnt

While exploring the different dimensions of how a computer vision system works, we realized that there is a vast scope of improvement in our application. Following are some of the reachable goals we aim for:

1. To explore the Deep Neural Networks to identify the hidden relations between the images of each class.
2. Use of gradient features (eg. HOG) in conjunction to color histograms.
3. Building an android app for this task.
4. Estimating the quantity of the food by predicting the size of any known objects in the background. Use of 3D alignment and matching to predict the distance of food from camera.

References

[1] http://www.cse.msu.edu/~cse803

[2] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar, "Food recognition using statistics of pairwise local features," in Proc. of IEEE Computer Vision and Pattern Recognition, 2010.

[3] Z. Zong, D.T. Nguyen, P. Ogunbona, and W. Li, "On the combination of local texture and global structure for food classification," in Proc. of International Symposium on Multimedia, pp. 204–211, 2010.

[4] M. Chen, K. Dhingra, W. Wu, L. Yang, R. Sukthankar, and J. Yang, "PFID: Pittsburgh fast-food image dataset," in Proc. of IEEE International Conference on Image Processing, pp. 289–292, 2009.

[5] www.google.com

[6] www.bing.com

[7] D. G. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, 60(2):91–110, 2004.

[8] http://www.mathworks.com/help/stats/kmeans.html

[9] CORTES, C. AND VAPNIK, V. 1995. Support-vector network. Mach. Learn. 20, 273–297.