Epicure: A Meal Recognition System

Presentation by:

Farhan Hormasji

Max Ellison

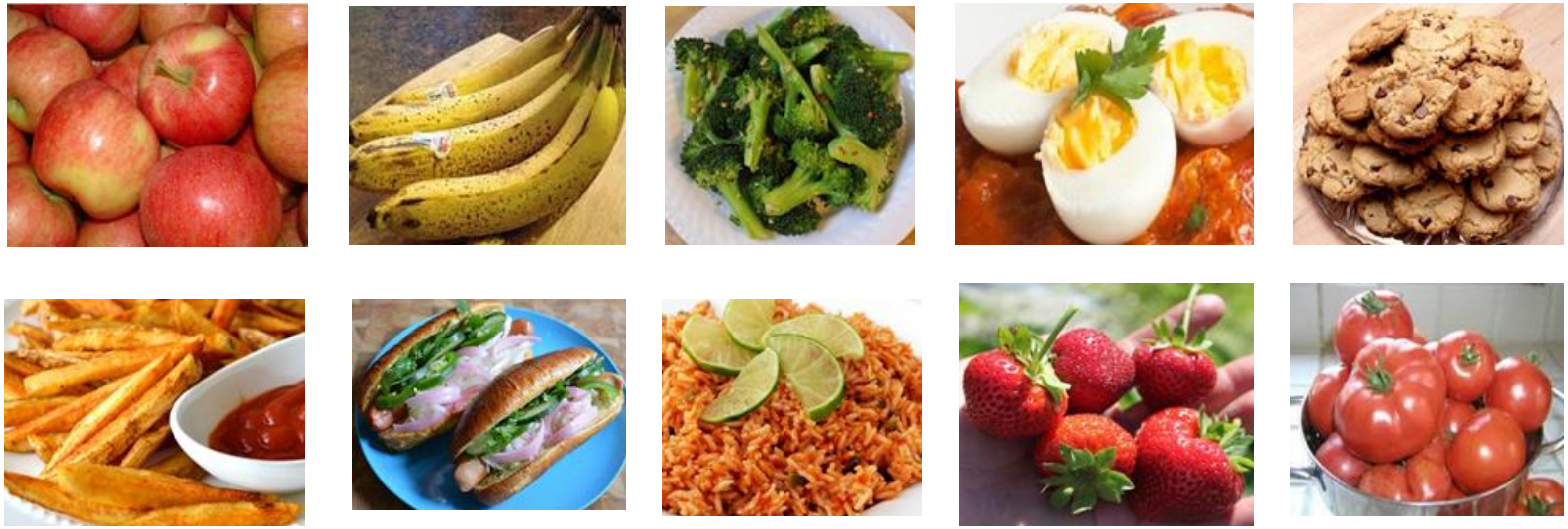
Tarang Chugh

Agenda

- Overview
  - Objective
  - Datasets
- Proposed Algorithm
  - Preprocessing
  - Feature Extraction
  - Classifier Training
- Results and Critique
  - Accuracy
  - Discussion

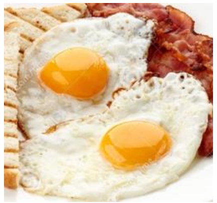
Objective

- To develop a comprehensive and robust meal recognition system, able to identify at least 10 different classes of food
- To understand the interdependence and working of various vision modules: preprocessing, segmentation, feature extraction, etc



Dataset – I (Into the wild)

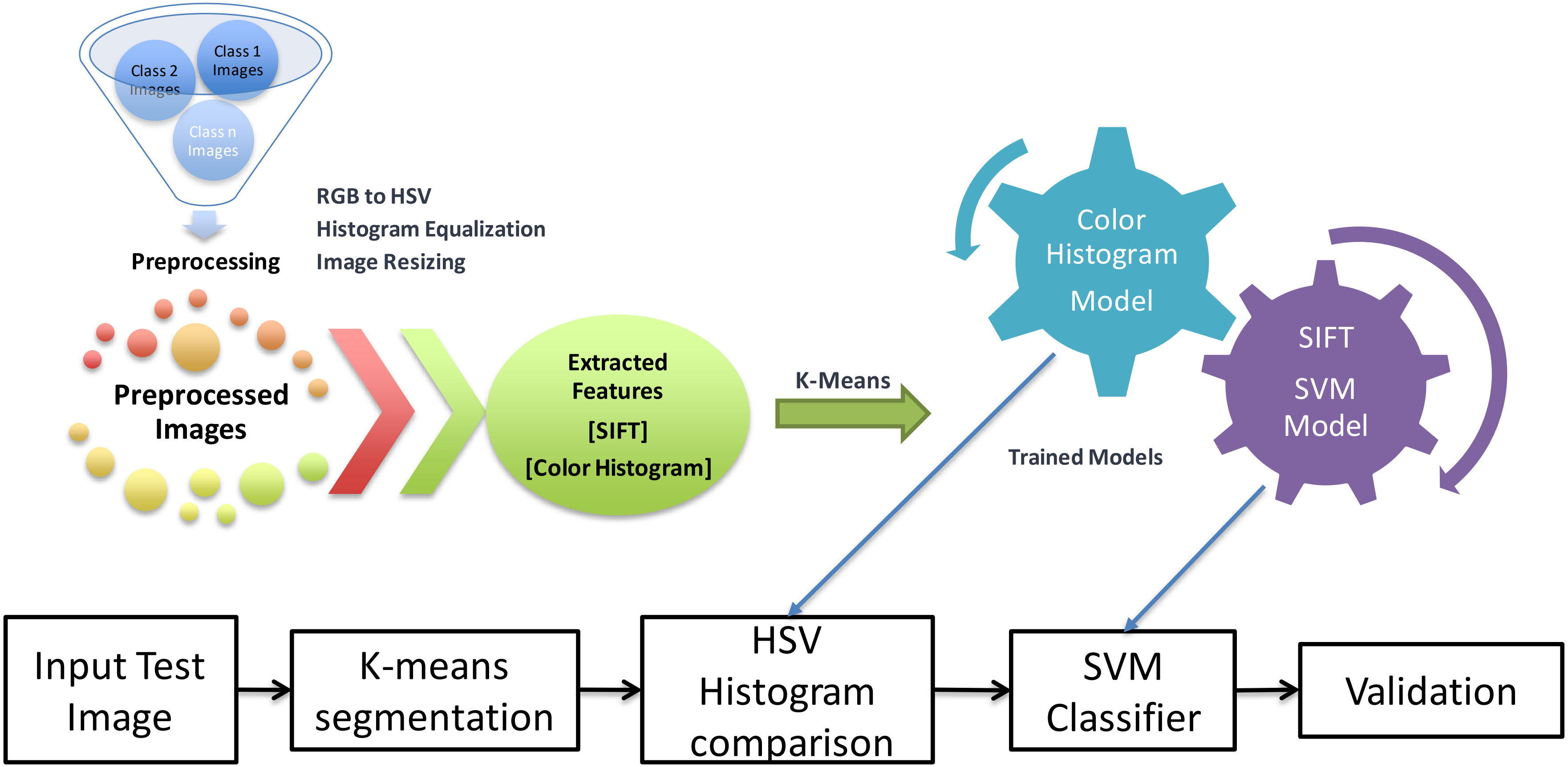
- **Training Set:** 50 images for each class
- Collected images from internet and few captured by us
- Manually filtered to ensure a mix bag of different view-angles and food-forms, illumination variations
- **Testing Set:** 100 images, at least 10 images for each class (25 images with multiple labels)



Dataset- II (Conducive Dataset)

- **Training Set:** 715 Images in 7 Classes
- Collected images from internet with easy to segment background
- To understand the impact of background noise on classification accuracy
- **Testing Set:** 70 images for 7 classes

Proposed Algorithm



Class 1 Images

Class 2 Images

Class n Images

Preprocessing

RGB to HSV

Histogram Equalization

Image Resizing

Preprocessed Images

Extracted Features

[SIFT]

[Color Histogram]

K-Means

Color Histogram Model

SIFT SVM Model

Trained Models

Input Test Image

K-means segmentation

HSV Histogram comparison

SVM Classifier

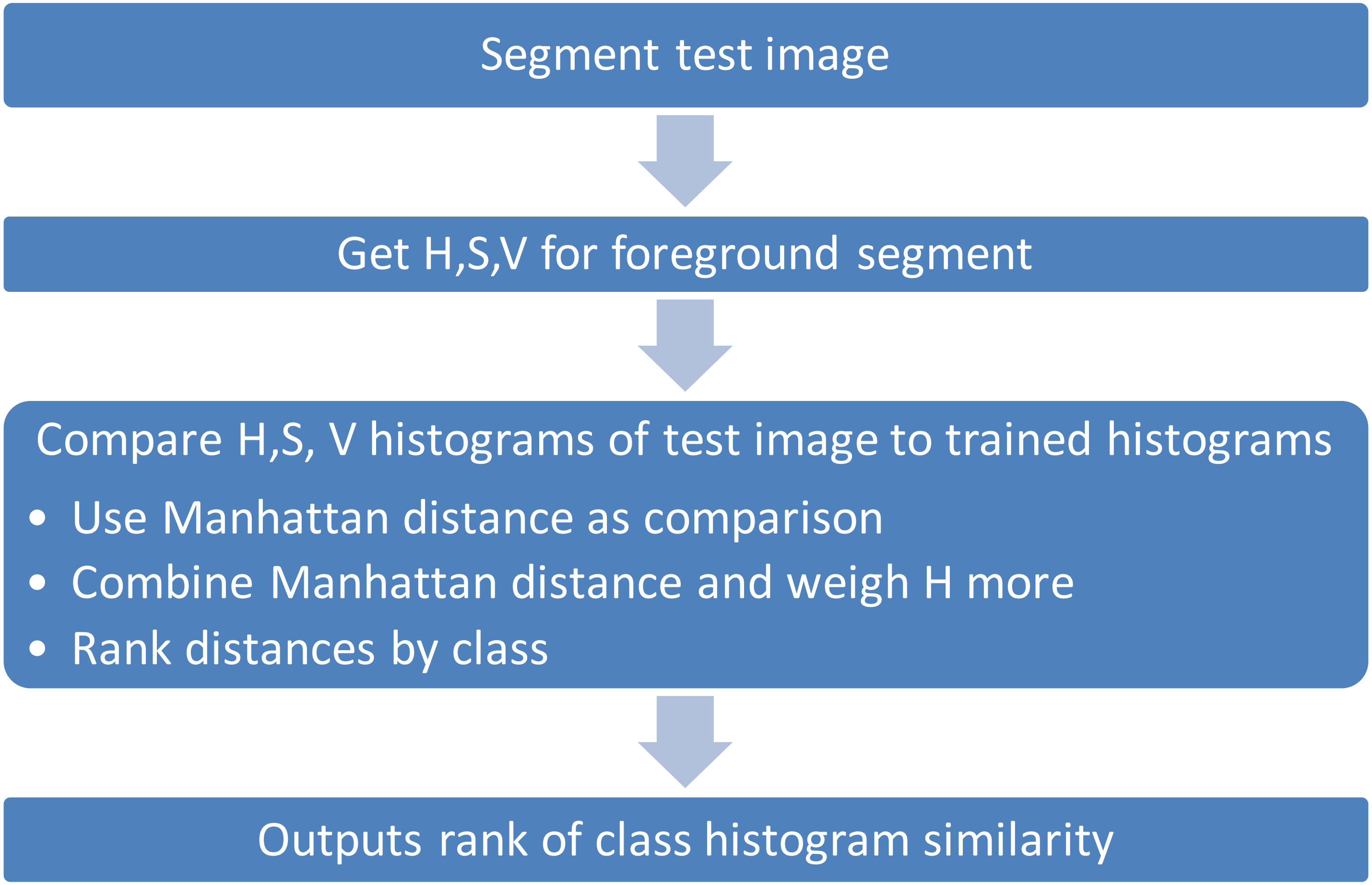
Validation

Pre-Processing

- Contrast Enhancement
  - Used HSV over RGB for color histogram
  - Performed histogram equalization on Intensity values
  - RGB to Grayscale for SIFT

- Improving Efficiency
  - Manual segmentation of training images to focus on regions of interest
  - Images resized such that max(row, col) = 1000 pixels

Color Histogram



Segment test image

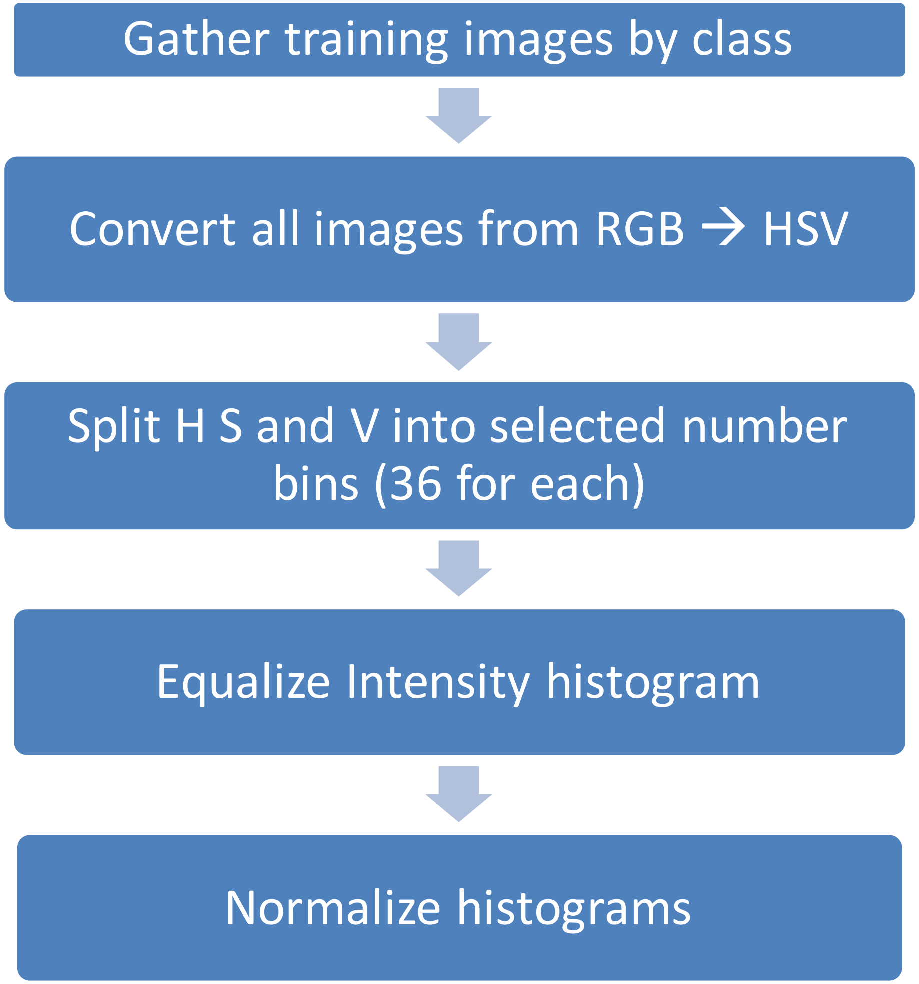
Get H,S,V for foreground segment

Compare H,S, V histograms of test image to trained histograms

- Use Manhattan distance as comparison
- Combine Manhattan distance and weigh H more
- Rank distances by class

Outputs rank of class histogram similarity

Training – Color Histogram



Gather training images by class

Convert all images from RGB → HSV

Split H S and V into selected number bins (36 for each)

Equalize Intensity histogram

Normalize histograms

Feature Extraction

SIFT : Scale Invariant Feature Transform

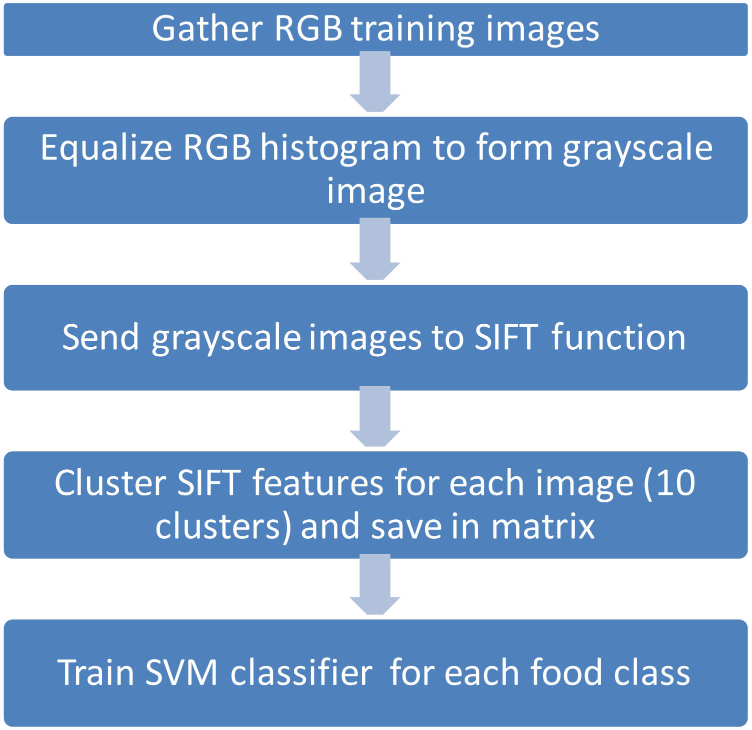
For each image,

- Extract SIFT: detects local key points (N) at different scale
- Returns N x 128 key point descriptors with their locations
    - How to make it a global descriptor?

Apply K-Means..

- Tried different values of K, worked best for K=10
- Total Clusters for each class = 50 x 10 = 500 key point clusters
- 10 Sift Models for each class with 500 x 128 feature vectors

Training – SIFT



Gather RGB training images

Equalize RGB histogram to form grayscale image

Send grayscale images to SIFT function

Cluster SIFT features for each image (10 clusters) and save in matrix

Train SVM classifier for each food class

Feature matrix passed to SVM is a 600 row matrix

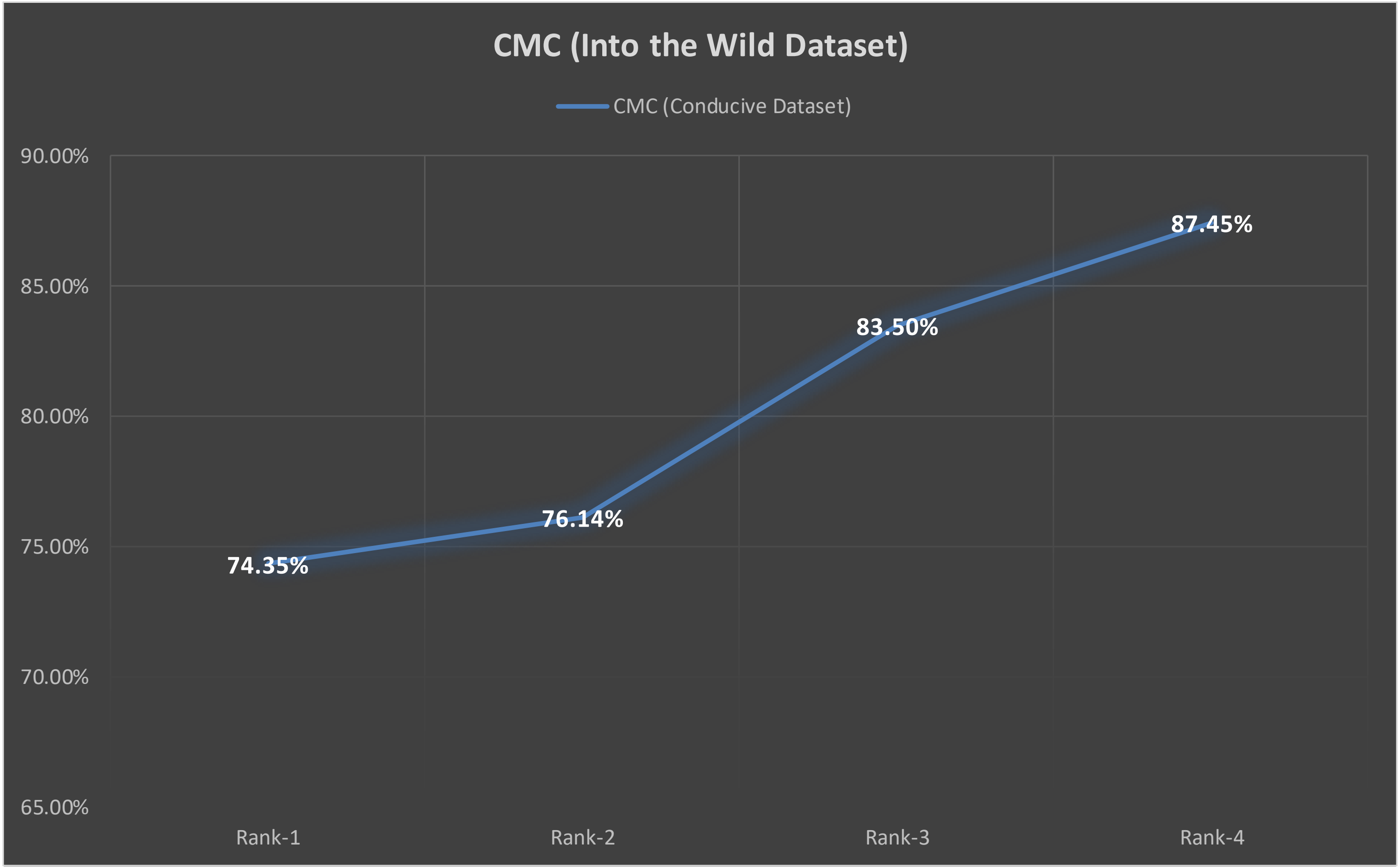
- First 300 rows are SIFT features belonging to food class being trained
- Last 300 rows are SIFT features belonging to other classes
- Features randomly chosen

Optimization By Filtering SIFT Features: Utensil Class

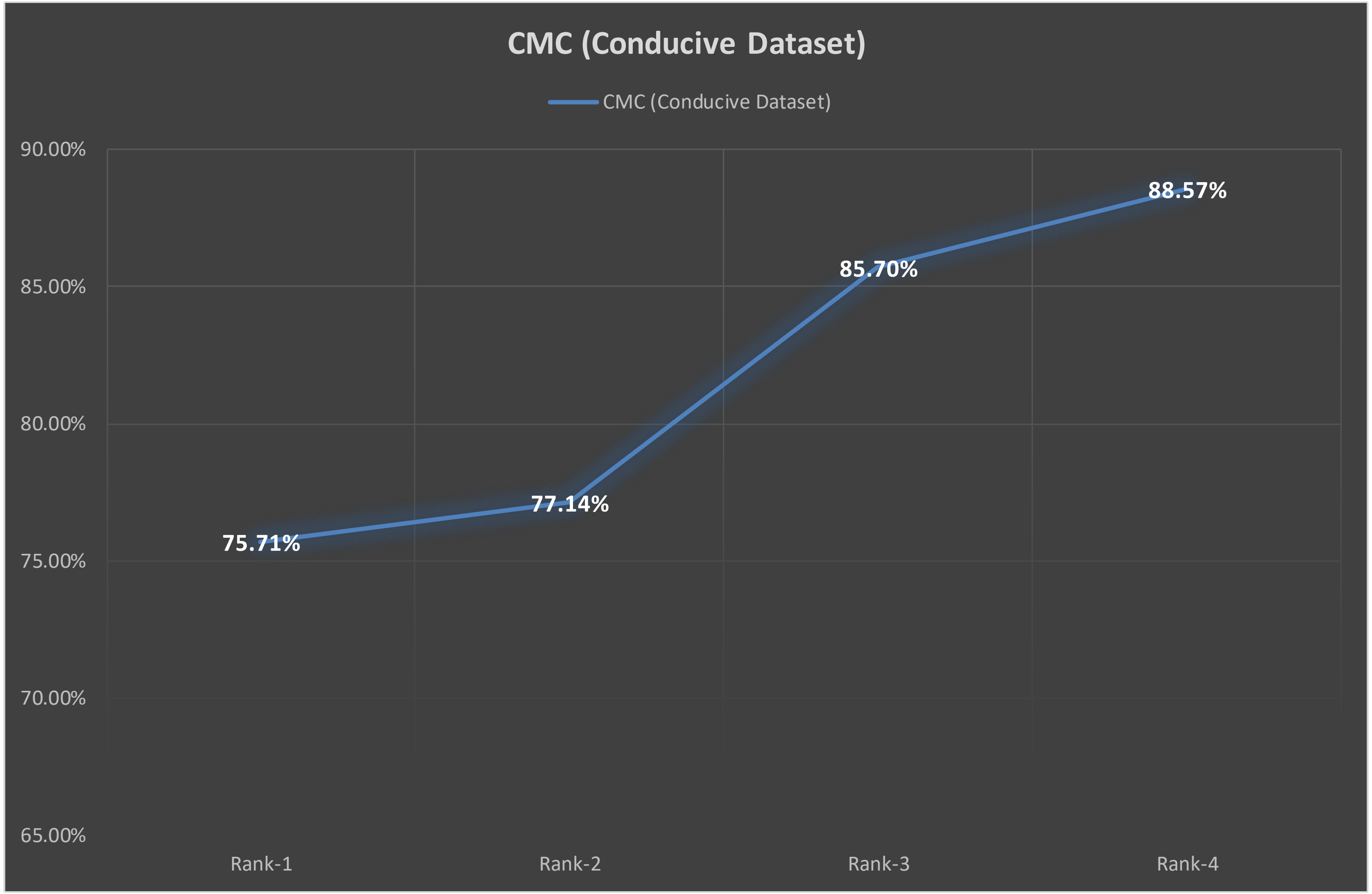
- Analyzed SIFT key points
- Use knowledge about most common backgrounds? Utensils?
- Extracted sift features for a new negative class “Utensils”
- 40 images consisting of bowls, plates, baskets in different view-angles
- Improvement of 3% in Rank-1 accuracy

Results and Critique

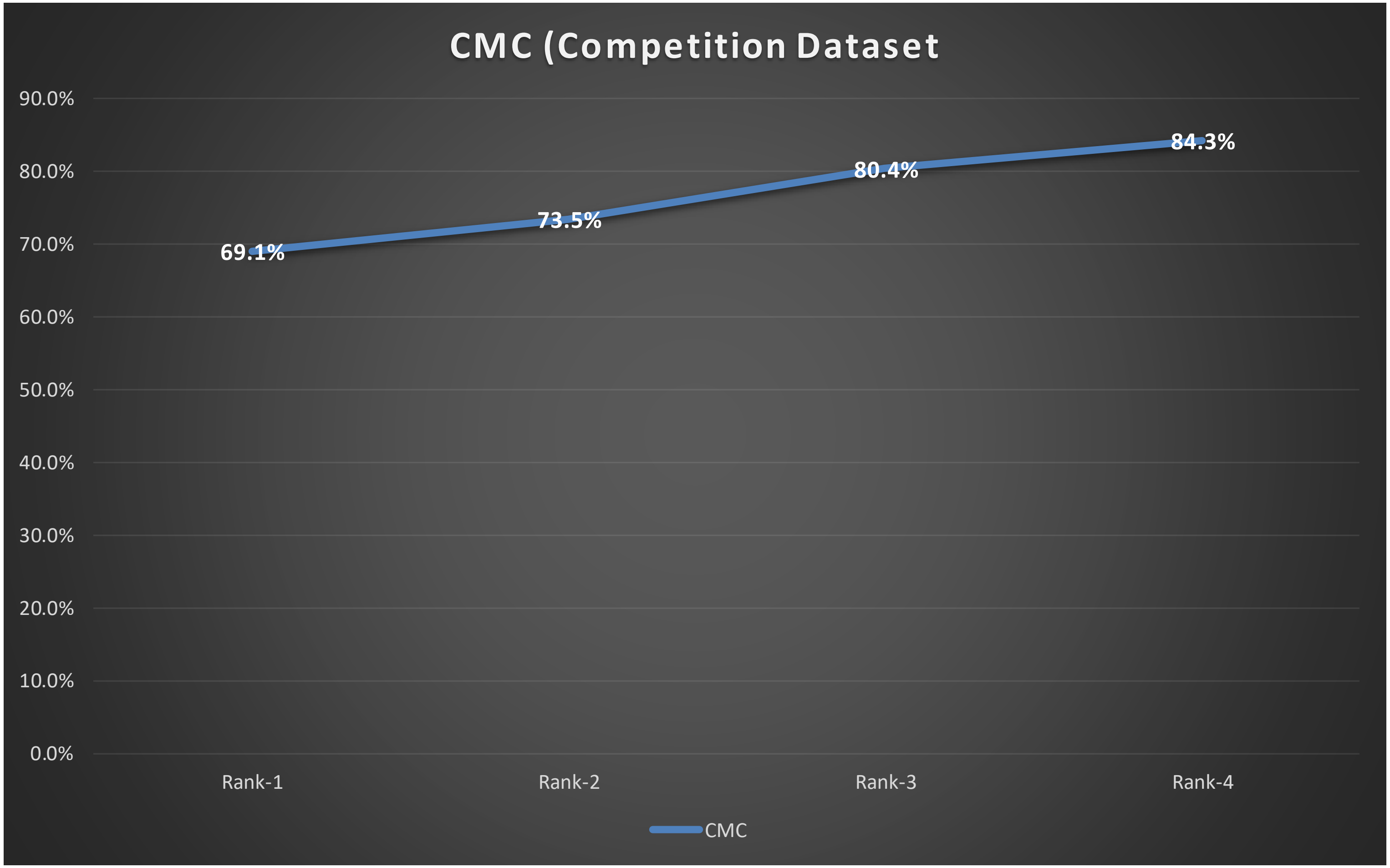
Results



Time Efficiency : 89.40 seconds



Time Efficiency : 88.40 seconds



Time Efficiency : 57.72 seconds

Discussion of Algorithm

- Color was by far best classifier

- Features attempted but not used:
  - Watershed segmentation
  - Histogram of Oriented Gradient

- Some classes are bound to be falsely classified
  - Apple, Tomato (color)
  - French Fries, Banana
  - Hotdog, Banana (shape)

Lessons Learnt

- Very hard to balance between accuracy and time requirement / computational efficiency of the system

- A very thin line between generalization and over-fitting of the model

- Segmentation is a big challenge

- More the training dataset, better it is

- 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.)

Future Work

- To explore the Deep Neural Networks to identify the hidden relations between the images of each class
- Use of gradient features (eg. HOG) in conjunction to color histograms
- Building an android app for this task
- 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.

Questions?