Food Recognition and Calorie estimation using Machine Learning

Abhishek Nilesh Shah
NYU Computer Science
ans556@nyu.edu

Anurag Dhaipule
NYU Computer Science
ad3531@nyu.edu

Bharathi Priyaa T
NYU Computer Science
bt978@nyu.edu

ABSTRACT

This project report presents our work on recognizing fast food images and estimating predetermined calories pertaining to the food. In this report we present food image recognition as classification task by classifying images containing a single food item into one of the food classes. We use fast food image dataset and present our analysis on using different classifiers for the six food classes using standard feature sets of computer vision and very popular, convolutional neural net.

General Terms

Computer Vision, Machine learning

Keywords

classification, image recognition

1. INTRODUCTION

Estimating calories from images have many real world applications. With fitness and weight watching becoming much more rampant than a decade back, people have become more conscious to know what they eat and how much calories a plate of meal contains. It would be convenient to take a picture of a meal and get an estimate of how much the meal contains. This application would have two parts: 1) To recognize the food item, 2) To estimate calories based on portion size, quantity, ingredients.

Thus our projects aims to tackle this task: Predict the calorie content of a meal given an image of the same. The actual scope of this task would involve 1) Identifying food from the image 2) Estimating the portion size, composition of ingredients and individual calories of the same. Owing to lack of calorie-specific data for different portion sizes of a same food item. We have tried to simplify the scope of our project by ignoring the quantity of food and composition of each dish and treat it like an image classification problem and calculate the calories based on a pre-determined list of calories of food item. This work can be extended for developing mobile applications that can estimate calories of meal by identifying food in real time clicked images.

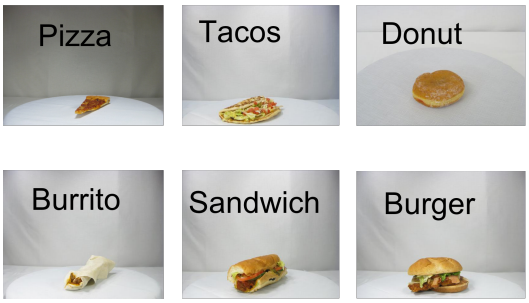


Fig. 1: Categories of food items

2. DATA SET

Pittsburgh Fast food Image Dataset (PFID)[1] was used for our experiments. In this dataset, food images are collected from different famous fast food joints like Subway, Taco Bells etc. of Pittsburgh and classified into 101 categories. This dataset contains images of different fast food items available in restaurants of Pittsburgh. The images were taken in lab for 3 different instances of same food item from different angles, orientations and different illumination conditions. Researchers have presented their work on subset of categories from the above mentioned 101 categories. We present our work on 6 food items.
We also crawl reddit from different subreddits of sandwiches, donuts, burgers etc. and collect total 147 images for our test dataset apart from using some of the available restaurant stills in the original pittsburgh fast food image dataset.
.

We chose a total 676 images of the categories mentioned in Figure 1. The distribution of the images in each of these categories has been described in Figure 2.

3. FEATURE ENGINEERING

We used the following features to capture the image content in our fast food images.

3.1 Color Histogram

Gives the distribution of colors in an image.

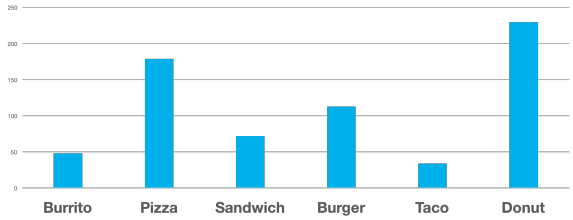


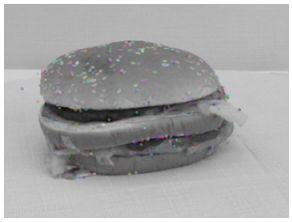
Fig. 2: Distribution of items

3.2 HOG - Histogram of Gradients

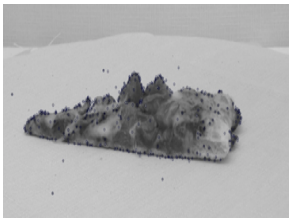
HOG[5] is a local feature descriptor which describes an image based on its local objects by calculating edge gradients. An image is divided into small regions called cells and for the pixels within each cell an intensify gradient is calculation . For each cell a histogram of gradients is computed.

3.3 SIFT- Scale Invariant feature transform

SIFT[6] feature calculates keypoints of images, like curvatures/corners etc. and preserves the relative positions of the keypoints regardless of rotation or scale. Bag of words is used to extract useful features for classification task.



(a) SIFT keypoints in burger



(b) SIFT keypoints in pizza

3.4 LBP - Local binary patterns

LBP[7] gives an 8 digits binary number by comparing each pixel cells value with its 8 neighbors. A histogram of frequencies of each higher pixel cell is returned.

3.5 VGG Net - Feature

Vggnet Features obtained using Deep Convolutional Neural Networks(19 layers). The VggNet[4] for identifying the visual content has been one of the state of art methods in ImageNet Large Scale Visual Recognition Challenge(ILSVRC). Apart from its impressive accuracy in the ILSVRC, one of the important reasons for using these features is the learned weights of their network are available on their site.

4. METHOD

4.1 Resizing Images

Some of the images were very large and hence extracting some of the features like SIFT for training data was taking a lot of time. So, as suggested in the paper [1] we resized our training images to 20% of it's original size.

4.2 Calorie map

This calorie map has been prepared by taking one type of each of burritos, pizzas, sandwiches, burgers, tacos and donuts. But with different contents calories may change.

Categories	KCal
Burrito	300
Pizza	250
Sandwich	400
burger	450
Tacos	156
Donuts	400

4.3 Stratified sampling

The data was divided into training and validation set using stratified sampling technique where different strata are formed according to the classes, then from each class, data is chosen proportionally to make the training and validation data maintaining the original distribution of each class in the data. We divided the data into training and validation in 80:20 ratio.
As this is a classification task, after extracting the above mentioned features we train our model using different classifiers like:

(1) LinearSVC
(2) Decision Tree Classifier
(3) Ensemble Methods like AdaBoost, Random Forest Classifier

We also extend our experiment by combining more than one features under different parameter settings of the classifiers. We present our results and analysis in the section below. There was a significant improvement in validation accuracy when we tried combining complementary features. The feature sets that were combined are as follows:

(1) Color Histogram + SIFT
(2) LBP + HOG
(3) Color Histogram + HOG

These combinations are complementary because they extract the geometrical patterns and colors in the image both. SIFT method extract the keypoints from the images so the feature sets of images are of different sizes. One of the very common methods used to employ SIFT features for experiments in Machine Learning is BOW(Bag of Words) model with SIFT features descriptors. The SIFT descriptors that are extracted after detection are clustered into pre-defined number of clusters called codewords. We use FLANN (Fast Approximate Nearest Neighbor Algorithm) for matching and forming codewords. The experiments are done using 50 such codewords.
It has been stated that the VGGNet has been shown to generalize well to other datasets as well[4]. The trick is to use the output of the 19 layers and fit a softmax regression layer to generate features for the food dataset on top of it. The optimizer used in minimizing softmax loss Adam Optimizer[2] for the better performance. This setting works on less training dataset and hence we use around 30 images for the experiment and similar settings were used to form training and validation set as mentioned above.
We use openCV[8] to extract the features like HOG, SIFT, LBP and Color Histograms. OpenCV was also used to extract Bag Of Words and employing matching algorithm. We use classifiers from

Scikit Learn [9] for our experiments. Lua and Torch[3] were used to extract VggNet Features and applying SoftMax Regression.

5. ANALYSIS

The analysis for several classifiers has been presented in this section. We evaluate our system with accuracy of the classifier and we present the confusion matrices for different classes for each type of classifiers where proportions of total number of items are indicated on 0-1.0 scale using different colors. So an ideal confusion matrices will have all red blocks on the diagonal and rest all blocks will be blue. Confusion matrices can be analyzed by observing the first diagonal elements having close values to red. Also y axis indicates number of true labels and x axis indicates predicted labels. The labels are as follows:
d-donuts
p-pizzas
br-burger
s-sandwich
t-tacos
bo-burrito

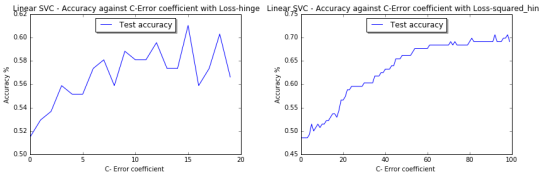
5.1 Linear SVC

We tried running Linear SVC over each of the individual features, by fine tuning its parameters. Here below, we have listed the optimal parameters obtained with SIFT features [Figure 3 on page 3]

5.1.1 Parameter tuning

C Error coefficient - We noticed that as we increase C from 0-100, there was some seasonality observed when we took the loss=hinge. The optimal range of C for loss=hinge was [0,10].

Loss hinge/squared hinge - When we took loss=squared hinge, the range of C's was from [0,100]. We observed higher accuracy when we took loss=squared hinge. The fine tunes parameters are [hinge:C-15, squared hinge:C-60]



(a) Changing C with loss=hinge (b) Changing C with loss=squared hinge

Fig. 3: Linear SVC with SIFT features

5.2 Decision tree classifier

The following shows the results of running Sklearn.Decision tree classifier over each individual features, by fine tuning its parameters. Here below, we have listed the optimal parameters obtained for SIFT features.

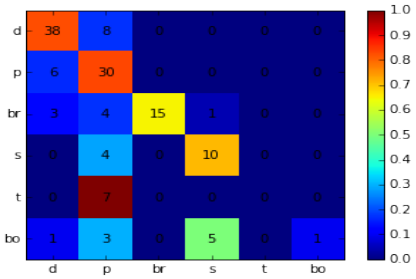


Fig. 4: Confusion Matrix for Linear SVC with loss=squared hinge

5.2.1 Parameter tuning

(max tree depth) - The relevant range for max tree depth was [1-30]. Optimal value for tree depth=15. [Figure 6]

(max leaf nodes) - The optimal range for max tree depth was [1-30]. Optimal value is 7. [Figure 7].

(max features) - Number of max-features differed based on the feature we used. [Figure 8]

(criterion) - We tried running Decision tree with the optimal value of parameters, with two different criterion to measure quality of a split - Gini and Entropy. Using entropy as criterion gave us better results [Figure 5 on page 3]

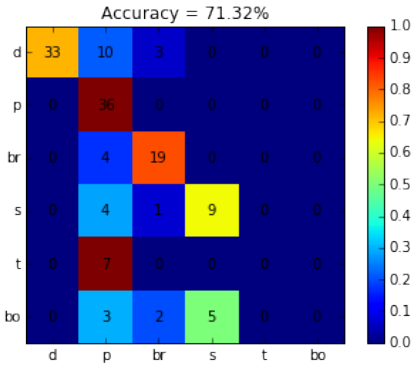


Fig. 5: Decision Tree(SIFT) with max-Depth=15, criterion=Entropy,max-leaf nodes=7

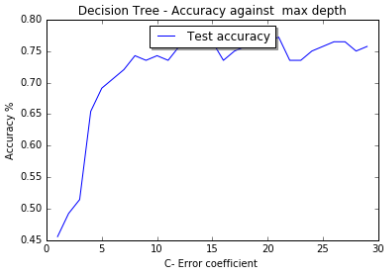


Fig. 6: Decision Tree with Depth[0-30]

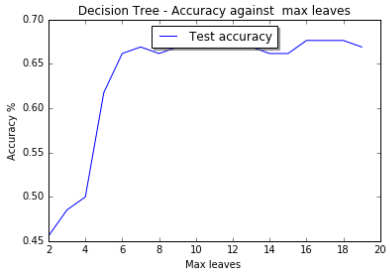


Fig. 7: Decision Tree with Max Leaves

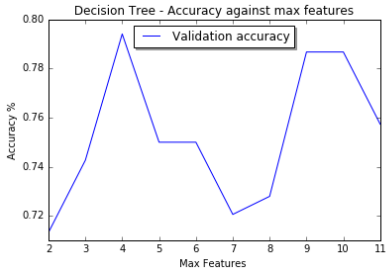


Fig. 8: Decision Tree with Max Features[0-10]

5.3 Combination of Features

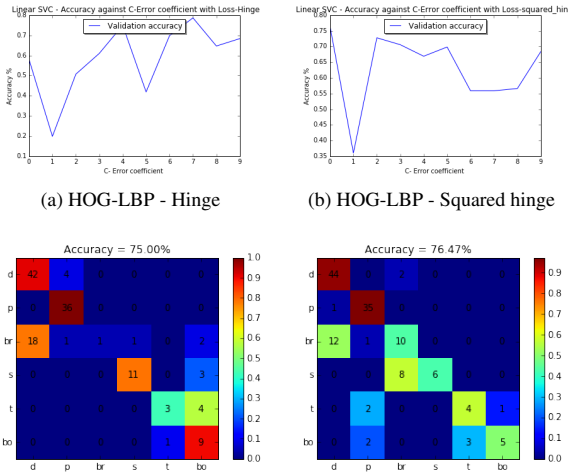
We make the following combination of features:

(1) Linear SVC - HOG and LBP - For the combination of HOG+LBP, The optimal set of parameters were observed were C=3,loss=squared hinge.See [Figure 9]
(2) Linear SVC - SIFT and Color Histogram - The optimal set of parameters were observed were C=10,loss=squared hinge.See [Figure 10].
(3) Linear SVC - HOG and Color Histogram - The optimal set of parameters were observed were C=100,loss=squared hinge.
(4) Decision trees - HOG and LBP - The optimal set of parameters were observed were max depth=11, max leaf nodes=30, criterion , max-features=7.See [Figure 11]

5.4 Ensemble methods

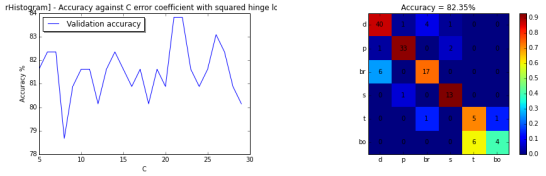
5.4.1 Adaboost. We achieved highest accuracies when we used the combined features with ensemble methods.

(1) Adaboost-Decision Trees - HOG and LBP - We used the optimal set of parameters that were observed with running decision trees for HOG+LBP previous and then we fine tuned parameters for Adaboost. Optimal value for n estimators=400. We achieved accuracy of about 97.06%See [Figure 12]
(2) Adaboost- Decision Trees - HIST and HOG - We used the optimal set of parameters that were observed with running decision trees for HOG+HIST previous and then we fine tuned parameters for Adaboost. Optimal value for n estimators=350. We achieved accuracy of about 94.85 %. HIST and HOG did not give high accuracy independently. This shows the power of ensemble methods. See [Figure 13].
(3) Adaboost-Decision Trees - SIFT and HIST - We used the optimal set of parameters that were observed with running decision trees for SIFT and Color histogram previous and then we fine tuned parameters for Adaboost. Optimal value for n estimators=375. We achieved accuracy of about 93.38%. See [Figure 14].



(a) HOG-LBP - Hinge
(b) HOG-LBP - Squared hinge
(c) Confusion Matrix - Hinge
(d) Confusion Matrix -Squared Hinge

Fig. 9: Linear SVC with HOG-LBP



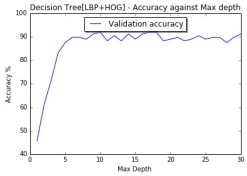
(a) Linear SIFT-HIST
(b) Linear SVC+ SIFT-HIST Confusion Matrix

Fig. 10: Linear SVC with SIFT- Color Histogram

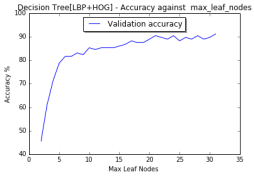
We can see that confusion among donuts, pizzas and burgers decreases drastically by using ensemble methods. Also burritos in the training and test have weird shapes and hence it can give a lot of confusion with some of the other items like tacos.

5.4.2 Random Forests. The RandomForest Classifier was tuned for different number of estimators and for both criteria.

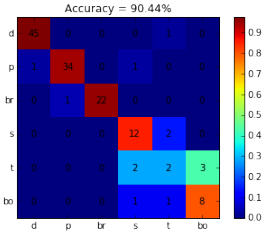
(1) Random forests - HOG and LBP - We fine tuned for parameters num-estimators. We chose a num-estimators=14. The final accuracy obtained was 96.32%. See [Figure 15].
(2) Random forests - HOG and Color Histogram - We fine tuned for parameters num-estimators. We chose a num-estimators=16. The final accuracy obtained was 91.18% See [Figure 16].



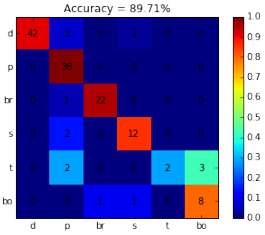
(a) Changing max depth



(b) Changing max leafnodes

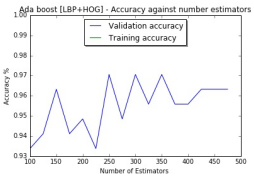


(c) Confusion Matrix - Gini

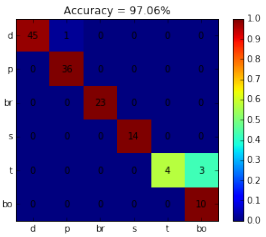


(d) Confusion Matrix - Entropy

Fig. 11: Decision Trees with LBP +HOG

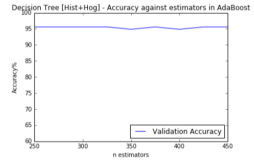


(a) Changing num estimators

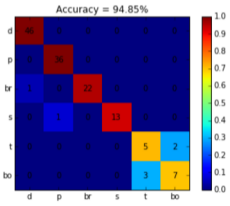


(b) Confusion Matrix

Fig. 12: Adaboost with LBP +HOG



(a) Changing num estimators



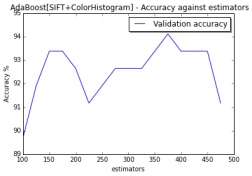
(b) Confusion Matrix

Fig. 13: Adaboost with HIST +HOG

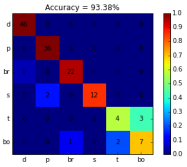
(3) Random forests - SIFT and HIST - We fine tuned for parameters num-estimators. We chose a num-estimators=17. The final accuracy obtained was 90.44% See [Figure 17].

5.5 Deep Convolutional Neural Net

The VGGNet with softmax regression gives 94.4% accuracy on validation dataset. Figure 19 shows the validation error with increase in the number of epochs.

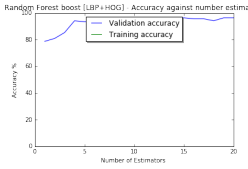


(a) Changing num estimators

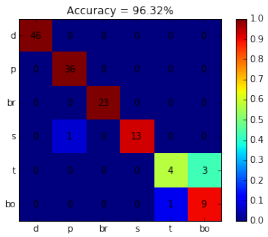


(b) Confusion Matrix

Fig. 14: Adaboost with SIFT +Color Histogram

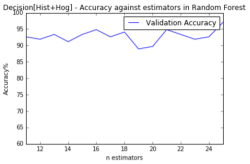


(a) Changing num estimators

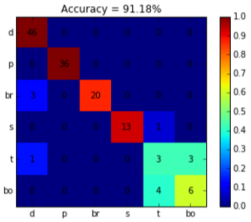


(b) Confusion Matrix

Fig. 15: Random Forests with LBP +HOG



(a) Changing num estimators



(b) Confusion Matrix

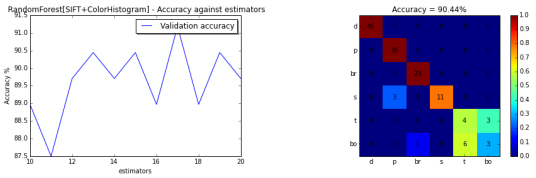
Fig. 16: Random Forests with HIST +HOG

6. RESULTS

In the below section, we tabulate the results [See Table 1 on page 6] obtained with different features. Color-histogram performed well with Linear SVC giving an accuracy of 79%. While LBP taken individually did not perform as good as HOG, taken together they were the highest performing. Both the ensemble methods performed similarly. For Adaboost it so happens that after increasing the depth, over-fitting starts and after more iterations it almost starts predicting all categories as 0.

6.1 Observations from Test

We tried running some of the models against actual test data obtained from restaurants. Some of the images have multiple food items or other miscellaneous objects in them. see Figure 19 on page 6.
We observed that restaurant images performed pretty decently with some of the top-performing models on validations[see Table 2 on page 6 for test results obtained] We noticed that fea-



(a) Changing num estimators (b) Confusion Matrix

Fig. 17: Random Forests with HIST +SIFT

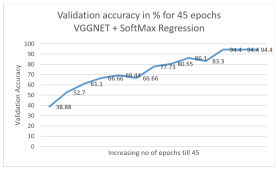
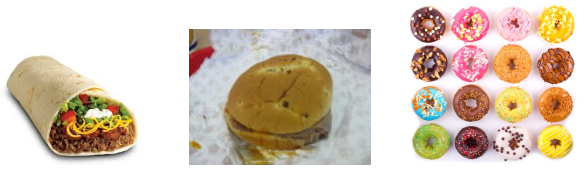


Fig. 18: Validation accuracy of VGGNet + SoftMax Regression



(a) Burrito (b) Burger (c) Donut

(d) Pizza (e) Sandwich (f) Tacos

Fig. 19: Test images from restaurants

Validation Accuracy% with different models	
SVC with SIFT	64.7%
SVC with HOG	63.9%
SVC with LBP	48.5%
SVC with HIST	79.4%
Decision Trees with Color Histograms	76.3%
Decision Trees with SIFT	56.6%
Decision Trees with HOG	83.82%
Decision Trees with LBP	74.2%
Linear SVC with Color Histograms + SIFT	82.35%
Linear SVC with HOG + LBP	50.7%
Decision Trees with HOG + LBP	91.91%
Ada boost with Color Histograms + SIFT	93.38%
Ada boost with HOG + LBP	96.32%
Ada boost with HOG + HIST	94.85%
Random forest with Color Histograms + SIFT	90.44%
Random forest with HOG + LBP	97.05%
Random forest with HOG + HIST	91.85%

Table 1. : Prediction accuracy for Validation data

ture combination of LBP and HOG worked well and gave an accuracy of around 55%. While adding another feature Color histogram brought down the accuracy significantly(around 25%) Interestingly, ensemble methods did not do as well as expected. Adaboost with [LBP+HOG] gave a mere 39.04% compared to just Linear SVC of the same features. Although color histograms did not prove to be a useful feature, Adaboost with [LBP+HOG+Color HIST] gave an accuracy of 56%.
Looking at the confusion matrices[see Table 20 on page 7], we see that food item with max number of images contributed a lot to accuracy. In this case, we had almost 80 sandwich images while the total test data was just 146. We believe that If we increase our dataset by a few hundred more, our system would perform closer to validation accuracies.

Test Accuracy% with different models	
SVC with LBP+HOG	55.47%
SVC with HIST+LBP	24.66%
Decision with LBP+HOG	52.05%
Ada boost with LBP+HOG	39.04%
SVC with LBP+HOG+Color histogram	35.61%
Decision tree with LBP+HOG+Color histogram	23.29%
Adaboost with LBP+HOG+Color histogram	56.61%
Random Forest with LBP+HOG+Color histogram	29.45%

Table 2. : Prediction accuracies for Test data (Restaurant images)

7. FUTURE WORK

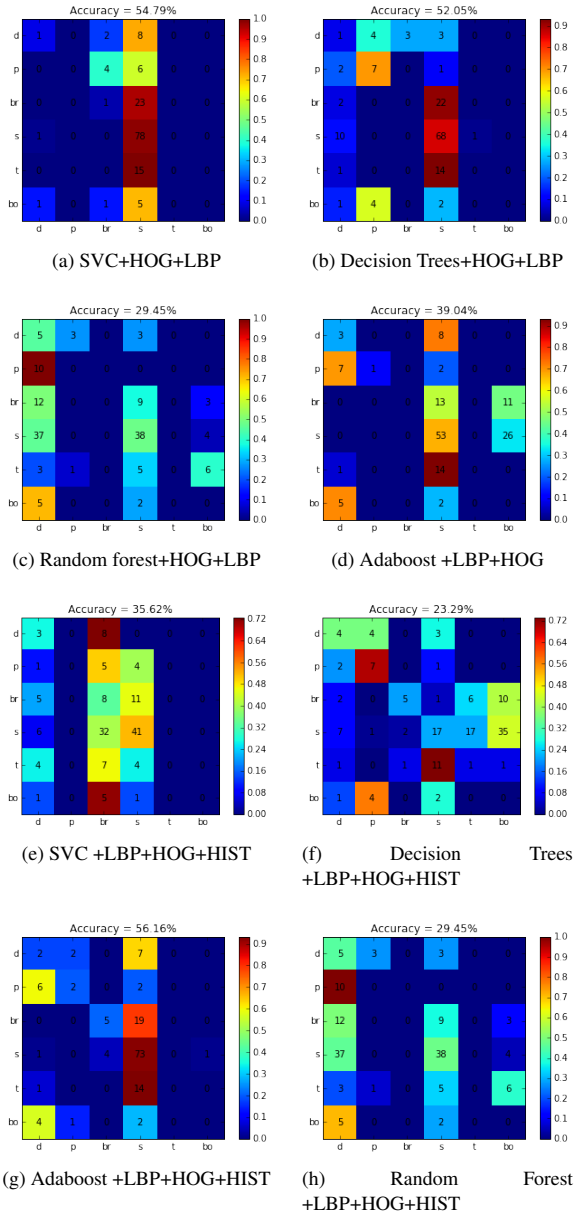
We observed that with a very good data set, features like LBP and HOG performed very well with boosting techniques. It would be great to have restaurant quality images tagged with calorie content. In that case, the second step would be a regression problem, trying to calculate calories for an image with SIFT/HOG features.
Also Semantic Texton Forests, technique can be used mentioned in [10] to train on identifying different ingredients in the fast food images. Further the problem can be extended to multi label classification where multiple food items in a single image can be identified.
We would like to thank our project advisor Brian D'Alessandro for guiding us in the right direction in our project.

8. REFERENCES

[1] Chen, M., Dhingra, K., Wu, W., Yang, L., Sukthankar, R. and Yang, J. *PFID: Pittsburgh fast-food image dataset.* In Image Processing (ICIP), 2009 16th IEEE International Conference on (pp. 289-292). IEEE.

[2] Kingma, Diederik, and Jimmy Ba. *"Adam: A method for stochastic optimization."* . arXiv preprint arXiv:1412 A

[3] Collobert, Ronan, Koray Kavukcuoglu, and Clment Farabet. *"Torch7: A matlab-like environment for machine learning"* . BigLearn, NIPS Workshop. No. EPFL-CONF-192376. 2011.



(a) SVC+HOG+LBP (b) Decision Trees+HOG+LBP

(c) Random forest+HOG+LBP (d) Adaboost +LBP+HOG

(e) SVC +LBP+HOG+HIST (f) Decision Trees +LBP+HOG+HIST

(g) Adaboost +LBP+HOG+HIST (h) Random Forest +LBP+HOG+HIST

Fig. 20: Test images from restaurants

[4] Simonyan, Karen, and Andrew Zisserman. *"Very deep convolutional networks for large-scale image recognition."* . arXiv preprint arXiv:1409.1556(2014).

[5] Dalal, Navneet, and Bill Triggs. *"Histograms of oriented gradients for human detection"* . Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.

[6] Lowe, David G. *"Object recognition from local scale-invariant features."* . Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Computer vision, 1999. The proceedings of the seventh IEEE international conference on. Vol. 2. Ieee, 1999.

[7] Ojala, Timo, Matti Pietikinen, and Topi Menp. *"Multiresolution gray-scale and rotation invariant textureclassification with local binary patterns."* . IEEE Transactions on24.7 (2002): 971-987.

[8] Bradski, Gary, and Adrian Kaehler *"Learning OpenCV: Computer vision with the OpenCV library."* . Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society O'Reilly

[9] Pedregosa, Fabian, et al. *Scikit-learn: Machine learning in Python.* The Journal of Machine Learning Research 12 (2011): 2825-2830.

[10] Yang, Shulin, et al. *Food recognition using statistics of pairwise local features.*Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010.