CV Assignment #3

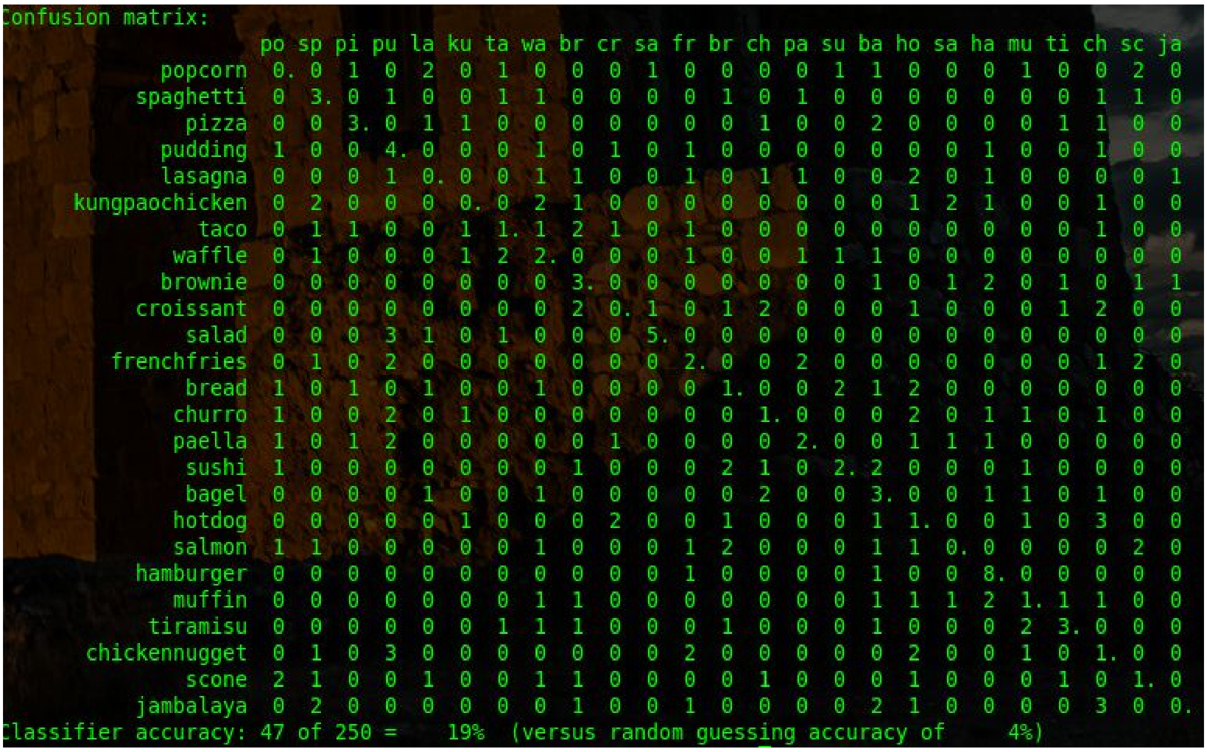
Project Members: Tejashree Khot, Srivatsan Iyer, Sanket Mhaiskar

Simple Baseline

Credits for SVM: https://www.cs.cornell.edu/people/tj/svm_light/svm_multiclass.html

1. The process involves resizing the image to 40x40 pixels. With 3 R, G, B channels, each image can be represented as a 4800 feature vector.
2. This is passed on to SVM for classification.

Parameters	Correct Recognition count	Accuracy
Full RGB channels (4800 vectors)	47 out of 250	19%
Greyscale (1600 vectors)	24 out of 250	9.6%



```
Confusion matrix:
               po sp pi pu la ku ta wa br cr sa fr br ch pa su ba ho sa ha mu ti ch sc ja
       popcorn 0. 0  1  0  2  0  1  0  0  0  1  0  0  0  0  1  1  0  0  0  1  0  0  2  0
     spaghetti 0  3. 0  1  0  0  1  1  0  0  0  0  1  0  1  0  0  0  0  0  0  0  1  1  0
         pizza 0  0  3. 0  1  1  0  0  0  0  0  0  0  1  0  0  2  0  0  0  0  1  1  0  0
       pudding 1  0  0  4. 0  0  0  1  0  1  0  1  0  0  0  0  0  0  0  1  0  0  1  0  0
       lasagna 0  0  0  1  0. 0  0  1  1  0  0  1  0  1  1  0  0  2  0  1  0  0  0  0  1
kungpaochicken 0  2  0  0  0  0. 0  2  1  0  0  0  0  0  0  0  0  1  2  1  0  0  1  0  0
          taco 0  1  1  0  0  1  1. 1  2  1  0  1  0  0  0  0  0  0  0  0  0  0  1  0  0
        waffle 0  1  0  0  0  1  2  2. 0  0  0  1  0  0  1  1  1  0  0  0  0  0  0  0  0
       brownie 0  0  0  0  0  0  0  0  3. 0  0  0  0  0  0  0  1  0  1  2  0  1  0  1  1
     croissant 0  0  0  0  0  0  0  0  2  0. 1  0  1  2  0  0  0  1  0  0  0  1  2  0  0
         salad 0  0  0  3  1  0  1  0  0  0  5. 0  0  0  0  0  0  0  0  0  0  0  0  0  0
   frenchfries 0  1  0  2  0  0  0  0  0  0  0  2. 0  0  2  0  0  0  0  0  0  0  1  2  0
         bread 1  0  1  0  1  0  0  1  0  0  0  0  1. 0  0  2  1  2  0  0  0  0  0  0  0
        churro 1  0  0  2  0  1  0  0  0  0  0  0  0  1. 0  0  0  2  0  1  1  0  1  0  0
        paella 1  0  1  2  0  0  0  0  0  1  0  0  0  0  2. 0  0  1  1  1  0  0  0  0  0
         sushi 1  0  0  0  0  0  0  0  1  0  0  0  2  1  0  2. 2  0  0  0  1  0  0  0  0
         bagel 0  0  0  0  1  0  0  1  0  0  0  0  0  2  0  0  3. 0  0  1  1  0  1  0  0
        hotdog 0  0  0  0  0  1  0  0  0  2  0  0  1  0  0  0  1  1. 0  0  1  0  3  0  0
        salmon 1  1  0  0  0  0  0  1  0  0  0  1  2  0  0  0  1  1  0. 0  0  0  0  2  0
     hamburger 0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1  0  0  8. 0  0  0  0  0
        muffin 0  0  0  0  0  0  0  1  1  0  0  0  0  0  0  0  1  1  1  2  1. 1  1  0  0
      tiramisu 0  0  0  0  0  0  1  1  1  0  0  0  1  0  0  0  1  0  0  0  2  3. 0  0  0
 chickennugget 0  1  0  3  0  0  0  0  0  0  0  2  0  0  0  0  0  2  0  0  1  0  1. 0  0
         scone 2  1  0  0  1  0  0  1  1  0  0  0  0  1  0  0  0  1  0  0  0  1  0  1. 0
     jambalaya 0  2  0  0  0  0  0  0  1  0  0  1  0  0  0  0  2  1  0  0  0  0  3  0  0.
Classifier accuracy: 47 of 250 =    19%  (versus random guessing accuracy of     4%)
```

Extra:

- There is a huge improvement in accuracy when all the 3 channels are included.
- On the down side, including the RGB channels takes a lot of time to train, (including the fact that SVM takes a lot of time to optimize.)

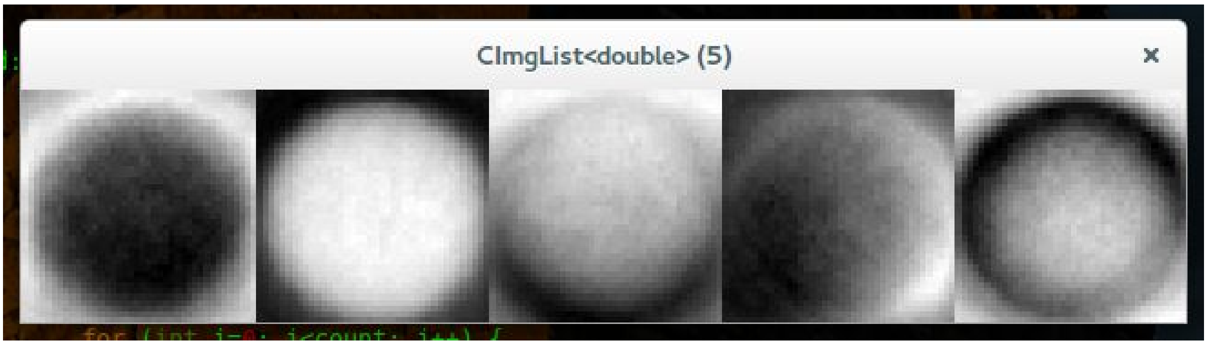
Model Folder: ./baseline-svm/

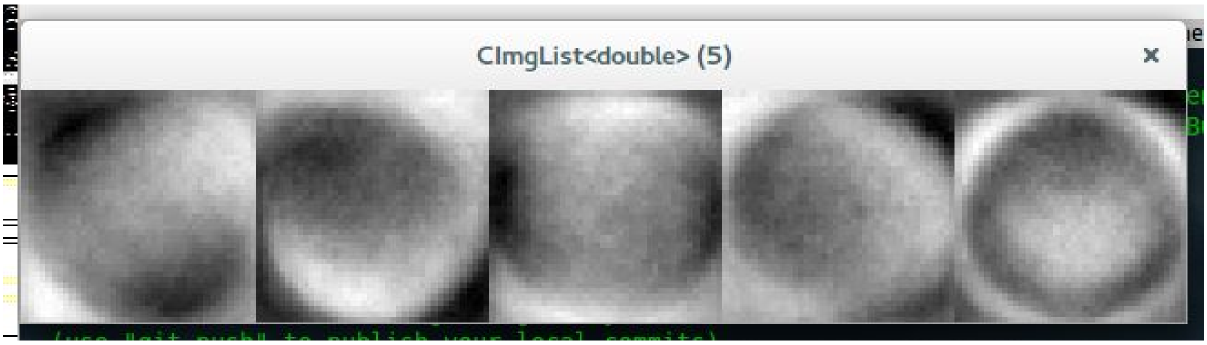
EigenFood SVM:

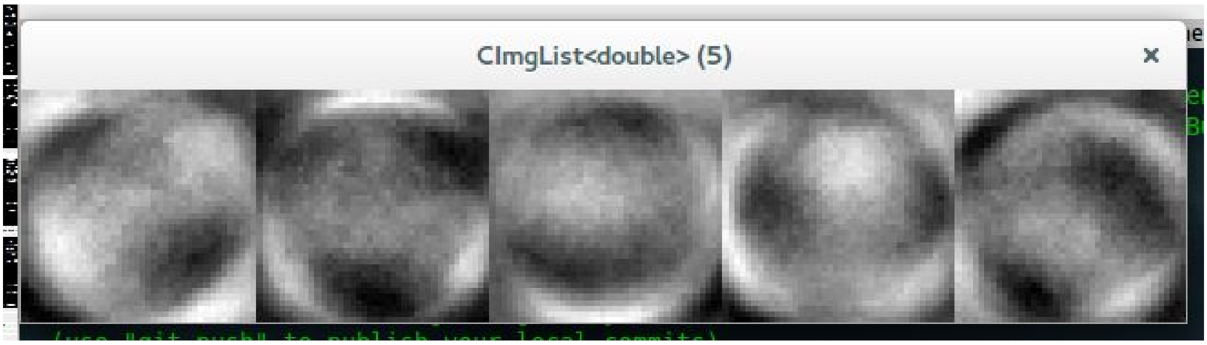
1. We resized image(40X40) and converted to gray scale.
2. We then calculated average image over the whole training data and subtracted this average image from each image to create difference image from each training image.
3. Unrolled each image and made a new matrix where each of these unrolled image is a row.
4. Calculated transpose of this matrix and on multiplication of matrix and its transpose we get covariance matrix.
5. Used eigen decomposition on this covariance matrix.
6. Selected top k eigenvectors whose eigenvalues were relatively larger than the rest.
7. Calculated coefficient for top k eigenvectors such that their summation can get us the original training image.
8. Used these coefficients to create training vectors for the SVM.
9. For testing, we map all images to eigenspace and calculate their coefficients. A vector of these coefficient is input to the SVM.

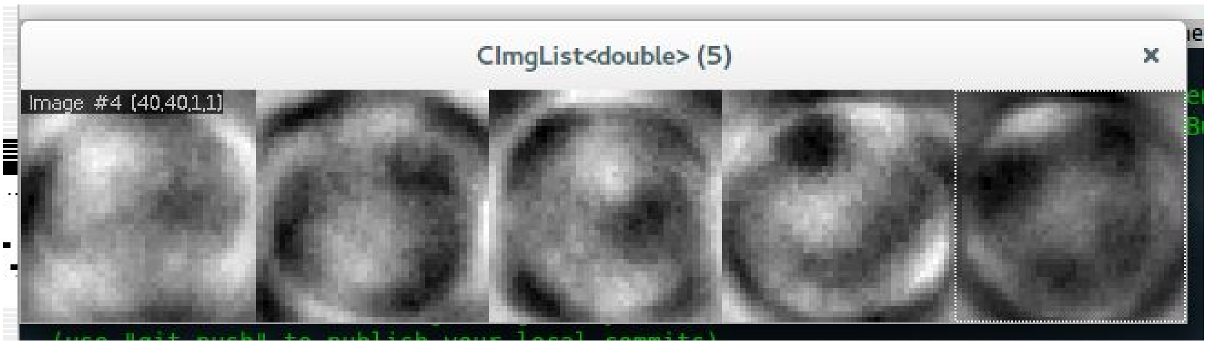
The eigen drops very rapidly at first and very slowly. We picked only the top 20 Eigen vectors.

The top 20 Eigen Food images look like below (in order):

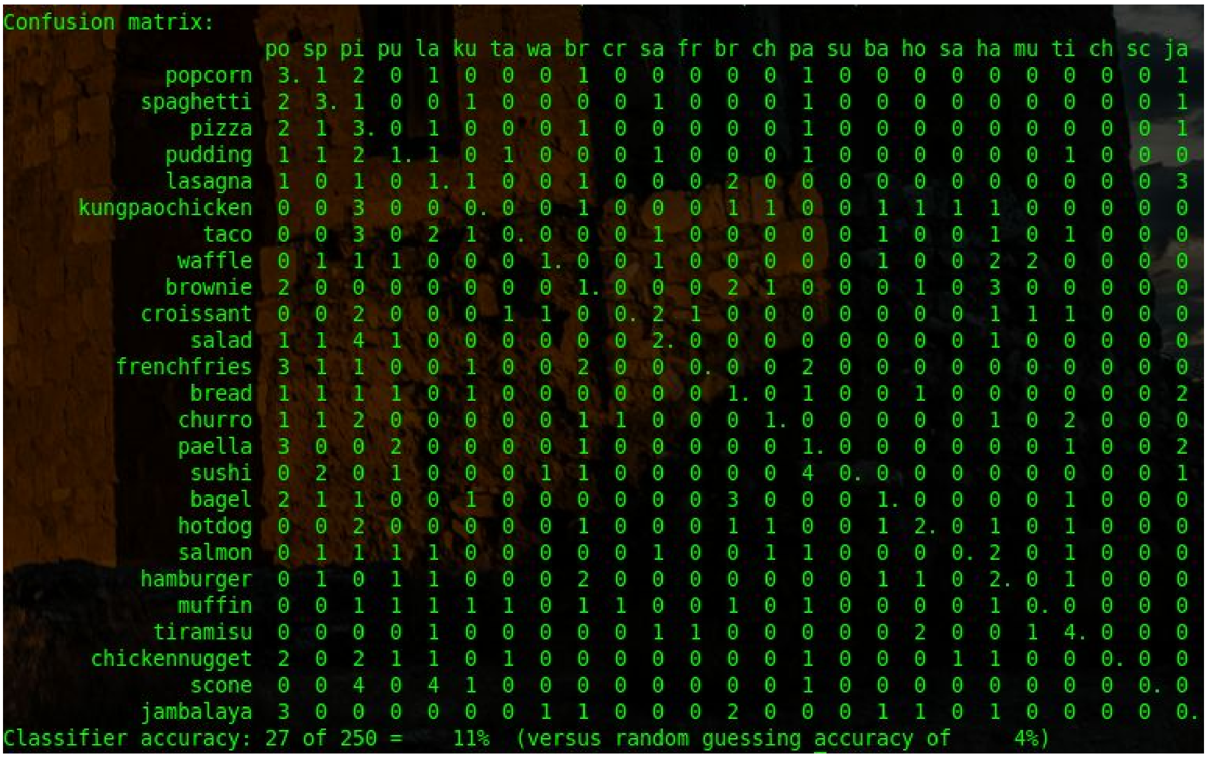








Resize 40x40, Top 40 eigen vectors	9.6% (24 out of 250)
Resize 40x40, Top 20 eigen vectors	11%
Resize 40x40, Top 16 eigen vectors	9.6% (24 out of 250)



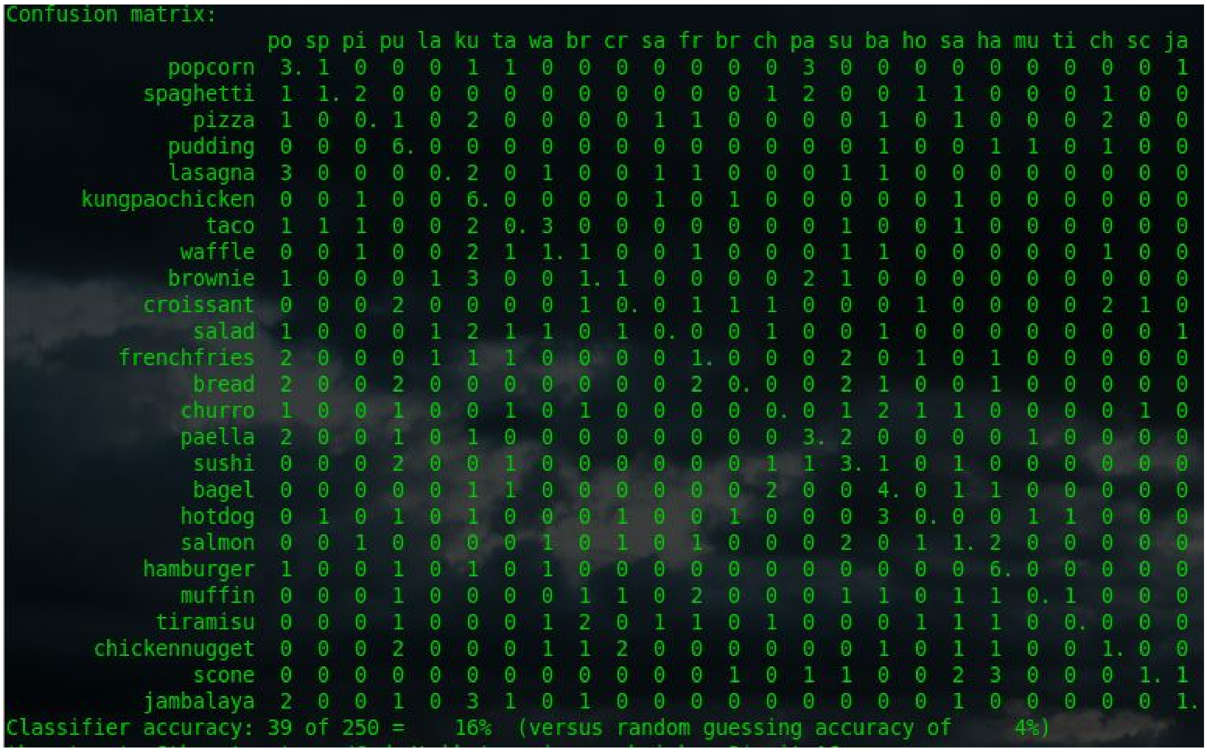
```
Confusion matrix:
               po sp pi pu la ku ta wa br cr sa fr br ch pa su ba ho sa ha mu ti ch sc ja
       popcorn 3. 1  2  0  1  0  0  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  1
     spaghetti 2  3. 1  0  0  1  0  0  0  0  1  0  0  0  1  0  0  0  0  0  0  0  0  0  1
         pizza 2  1  3. 0  1  0  0  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  1
       pudding 1  1  2  1. 1  0  1  0  0  0  1  0  0  0  1  0  0  0  0  0  0  1  0  0  0
       lasagna 1  0  1  0  1. 1  0  0  1  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  3
kungpaochicken 0  0  3  0  0  0. 0  0  1  0  0  0  1  1  0  0  1  1  1  1  0  0  0  0  0
          taco 0  0  3  0  2  1  0. 0  0  0  1  0  0  0  0  0  1  0  0  1  0  1  0  0  0
        waffle 0  1  1  1  0  0  0  1. 0  0  1  0  0  0  0  0  1  0  0  2  2  0  0  0  0
       brownie 2  0  0  0  0  0  0  0  1. 0  0  0  2  1  0  0  0  1  0  3  0  0  0  0  0
     croissant 0  0  2  0  0  0  1  1  0  0. 2  1  0  0  0  0  0  0  0  1  1  1  0  0  0
         salad 1  1  4  1  0  0  0  0  0  0  2. 0  0  0  0  0  0  0  0  1  0  0  0  0  0
   frenchfries 3  1  1  0  0  1  0  0  2  0  0  0. 0  0  2  0  0  0  0  0  0  0  0  0  0
         bread 1  1  1  1  0  1  0  0  0  0  0  0  1. 0  1  0  0  1  0  0  0  0  0  0  2
        churro 1  1  2  0  0  0  0  0  1  1  0  0  0  1. 0  0  0  0  0  1  0  2  0  0  0
        paella 3  0  0  2  0  0  0  0  1  0  0  0  0  0  1. 0  0  0  0  0  0  1  0  0  2
         sushi 0  2  0  1  0  0  0  1  1  0  0  0  0  0  4  0. 0  0  0  0  0  0  0  0  1
         bagel 2  1  1  0  0  1  0  0  0  0  0  0  3  0  0  0  1. 0  0  0  0  1  0  0  0
        hotdog 0  0  2  0  0  0  0  0  1  0  0  0  1  1  0  0  1  2. 0  1  0  1  0  0  0
        salmon 0  1  1  1  1  0  0  0  0  0  1  0  0  1  1  0  0  0  0. 2  0  1  0  0  0
     hamburger 0  1  0  1  1  0  0  0  2  0  0  0  0  0  0  0  1  1  0  2. 0  1  0  0  0
        muffin 0  0  1  1  1  1  1  0  1  1  0  0  1  0  1  0  0  0  0  1  0. 0  0  0  0
      tiramisu 0  0  0  0  1  0  0  0  0  0  1  1  0  0  0  0  0  2  0  0  1  4. 0  0  0
 chickennugget 2  0  2  1  1  0  1  0  0  0  0  0  0  0  1  0  0  0  1  1  0  0  0. 0  0
         scone 0  0  4  0  4  1  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0. 0
     jambalaya 3  0  0  0  0  0  0  1  1  0  0  0  2  0  0  0  1  1  0  1  0  0  0  0  0.
Classifier accuracy: 27 of 250 =     11%  (versus random guessing accuracy of     4%)
```

Model Folder: ./eigen-svm/

Haar Like Features

1. We generated random points and random window sizes to calculate the Haar Like Wavelets.
2. Calculated the Summed Area Tables for every image i.e Integral Images.
3. We then calculated Hx and Hy i.e two Haar projects with different window orientations.Every feature value was the magnitude of Hx and Hy values.
4. We then trained the SVM using the feature vector generated and tested it using the same random points and window size values generated for training. We have tried changing the number of random samples generated to different configurations and below were the results :

Parameters	Image Size	Accuracy
No of features: 6400	40x40	39 of 250 = 16%
No of features: 1600	100x100	35 of 250 = 14%



```
Confusion matrix:
               po sp pi pu la ku ta wa br cr sa fr br ch pa su ba ho sa ha mu ti ch sc ja
       popcorn 3. 1  0  0  0  1  1  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0  1
     spaghetti 1  1. 2  0  0  0  0  0  0  0  0  0  0  1  2  0  0  1  1  0  0  0  1  0  0
         pizza 1  0  0. 1  0  2  0  0  0  0  1  1  0  0  0  0  1  0  1  0  0  0  2  0  0
       pudding 0  0  0  6. 0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  1  1  0  1  0  0
       lasagna 3  0  0  0  0. 2  0  1  0  0  1  1  0  0  0  1  1  0  0  0  0  0  0  0  0
kungpaochicken 0  0  1  0  0  6. 0  0  0  0  1  0  1  0  0  0  0  0  1  0  0  0  0  0  0
          taco 1  1  1  0  0  2  0. 3  0  0  0  0  0  0  0  1  0  0  1  0  0  0  0  0  0
        waffle 0  0  1  0  0  2  1  1. 1  0  0  1  0  0  0  1  1  0  0  0  0  0  1  0  0
       brownie 1  0  0  0  1  3  0  0  1. 1  0  0  0  0  2  1  0  0  0  0  0  0  0  0  0
     croissant 0  0  0  2  0  0  0  0  1  0. 0  1  1  1  0  0  0  1  0  0  0  0  2  1  0
         salad 1  0  0  0  1  2  1  1  0  1  0. 0  0  1  0  0  1  0  0  0  0  0  0  0  1
   frenchfries 2  0  0  0  1  1  1  0  0  0  0  1. 0  0  0  2  0  1  0  1  0  0  0  0  0
         bread 2  0  0  2  0  0  0  0  0  0  0  2  0. 0  0  2  1  0  0  1  0  0  0  0  0
        churro 1  0  0  1  0  0  1  0  1  0  0  0  0  0. 0  1  2  1  1  0  0  0  0  1  0
        paella 2  0  0  1  0  1  0  0  0  0  0  0  0  0  3. 2  0  0  0  0  1  0  0  0  0
         sushi 0  0  0  2  0  0  1  0  0  0  0  0  0  1  1  3. 1  0  1  0  0  0  0  0  0
         bagel 0  0  0  0  0  1  1  0  0  0  0  0  0  2  0  0  4. 0  1  1  0  0  0  0  0
        hotdog 0  1  0  1  0  1  0  0  0  1  0  0  1  0  0  0  3  0. 0  0  1  1  0  0  0
        salmon 0  0  1  0  0  0  0  1  0  1  0  1  0  0  0  2  0  1  1. 2  0  0  0  0  0
     hamburger 1  0  0  1  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0  6. 0  0  0  0  0
        muffin 0  0  0  1  0  0  0  0  1  1  0  2  0  0  0  1  1  0  1  1  0. 1  0  0  0
      tiramisu 0  0  0  1  0  0  0  1  2  0  1  1  0  1  0  0  0  1  1  1  0  0. 0  0  0
 chickennugget 0  0  0  2  0  0  0  1  1  2  0  0  0  0  0  0  1  0  1  1  0  0  1. 0  0
         scone 0  0  0  0  0  0  0  0  0  0  0  0  1  0  1  1  0  0  2  3  0  0  0  1. 1
     jambalaya 2  0  0  1  0  3  1  0  1  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  1.
Classifier accuracy: 39 of 250 =    16%  (versus random guessing accuracy of     4%)
```

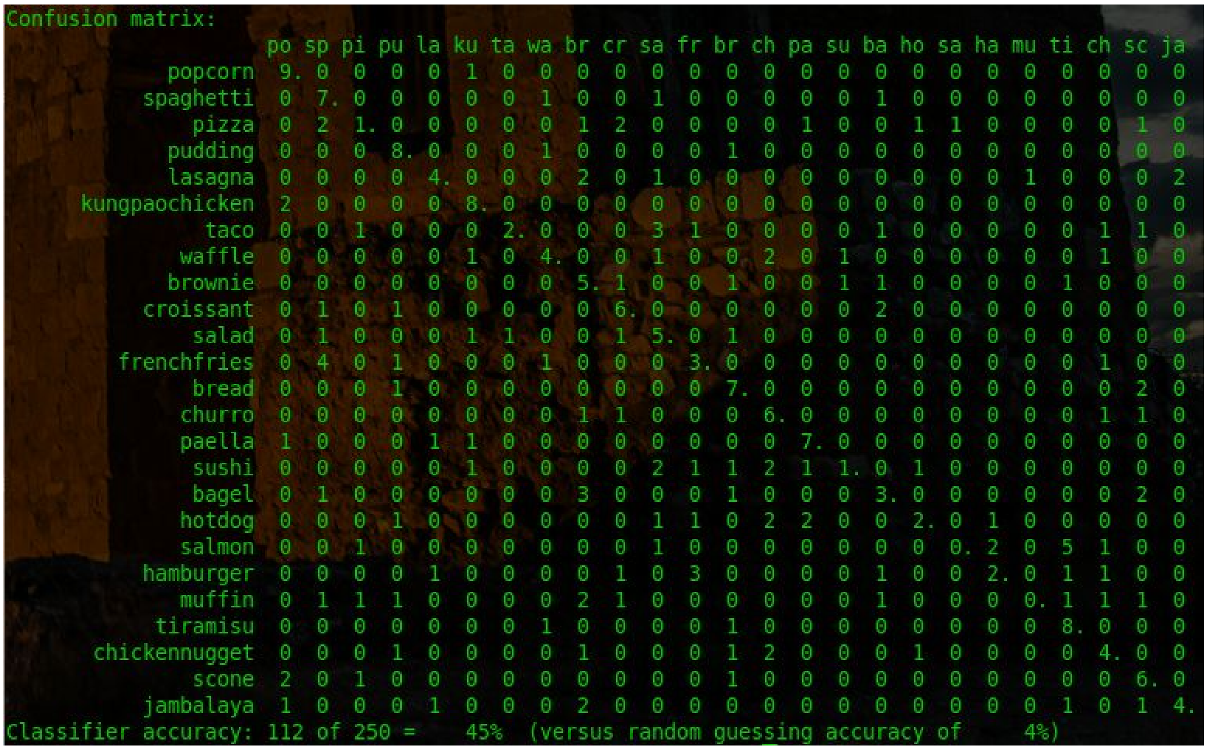
References :

1) http://www.iri.upc.edu/people/mvillami/icpr06.html
2) https://computersciencesource.wordpress.com/2010/09/03/computer-vision-the-integral-image/

Bag of Words

1. We iterate over every image in the dataset, and extract the SIFT features from it. Each SIFT feature is a 128 dimensional vector.
2. We take all the SIFT vectors of every image, put it together, and run K-means on it. Once we run clustering, we have **visual words**.
3. Next, for every image in the training dataset, we iterate over each the SIFT vector and try to find the nearest visual word using Euclidean distance.
4. Since each SIFT vector can be associated with a visual word, we can create a histogram among these visual words, and return the frequency (histogram) as the feature vector for SVM classification.
5. The visual words is written out to a file. This would be the core model for Bag of Words.
6. In the testing phase, we read the visual words from the file, and try to recreate the histogram for the given input image. This histogram (represented as a feature vector) is used as an input to SVM for classification.

Parameters	Correct count	Accuracy
Cluster count = 30	89 of 250	36%
Cluster count = 100	112 of 250	45%



```
Confusion matrix:
               po sp pi pu la ku ta wa br cr sa fr br ch pa su ba ho sa ha mu ti ch sc ja
       popcorn 9. 0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
     spaghetti 0  7. 0  0  0  0  0  1  0  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0  0
         pizza 0  2  1. 0  0  0  0  0  1  2  0  0  0  0  1  0  0  1  1  0  0  0  0  1  0
       pudding 0  0  0  8. 0  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0
       lasagna 0  0  0  0  4. 0  0  0  2  0  1  0  0  0  0  0  0  0  0  0  1  0  0  0  2
kungpaochicken 2  0  0  0  0  8. 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
          taco 0  0  1  0  0  0  2. 0  0  0  3  1  0  0  0  0  1  0  0  0  0  0  1  1  0
        waffle 0  0  0  0  0  1  0  4. 0  0  1  0  0  2  0  1  0  0  0  0  0  0  1  0  0
       brownie 0  0  0  0  0  0  0  0  5. 1  0  0  1  0  0  1  1  0  0  0  0  1  0  0  0
     croissant 0  1  0  1  0  0  0  0  0  6. 0  0  0  0  0  0  2  0  0  0  0  0  0  0  0
         salad 0  1  0  0  0  1  1  0  0  1  5. 0  1  0  0  0  0  0  0  0  0  0  0  0  0
   frenchfries 0  4  0  1  0  0  0  1  0  0  0  3. 0  0  0  0  0  0  0  0  0  0  1  0  0
         bread 0  0  0  1  0  0  0  0  0  0  0  0  7. 0  0  0  0  0  0  0  0  0  0  2  0
        churro 0  0  0  0  0  0  0  0  1  1  0  0  0  6. 0  0  0  0  0  0  0  0  1  1  0
        paella 1  0  0  0  1  1  0  0  0  0  0  0  0  0  7. 0  0  0  0  0  0  0  0  0  0
         sushi 0  0  0  0  0  1  0  0  0  0  2  1  1  2  1  1. 0  1  0  0  0  0  0  0  0
         bagel 0  1  0  0  0  0  0  0  3  0  0  0  1  0  0  0  3. 0  0  0  0  0  0  2  0
        hotdog 0  0  0  1  0  0  0  0  0  0  1  1  0  2  2  0  0  2. 0  1  0  0  0  0  0
        salmon 0  0  1  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0. 2  0  5  1  0  0
     hamburger 0  0  0  0  1  0  0  0  0  1  0  3  0  0  0  0  1  0  0  2. 0  1  1  0  0
        muffin 0  1  1  1  0  0  0  0  2  1  0  0  0  0  0  0  1  0  0  0  0. 1  1  1  0
      tiramisu 0  0  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0  0  8. 0  0  0
 chickennugget 0  0  0  1  0  0  0  0  1  0  0  0  1  2  0  0  0  1  0  0  0  0  4. 0  0
         scone 2  0  1  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  6. 0
     jambalaya 1  0  0  0  1  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  1  0  1  4.
Classifier accuracy: 112 of 250 =     45%  (versus random guessing accuracy of     4%)
```

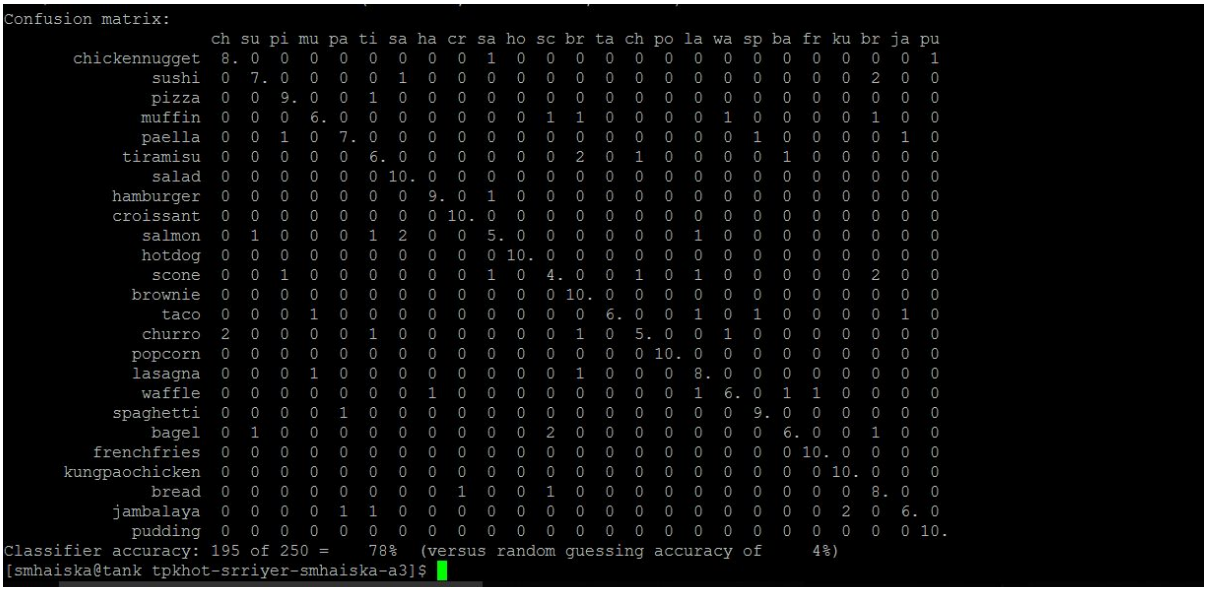
Model Folder: ./sift-svm/   and ./visual_words

Deep Neural Networks

Reference: "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks", Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, Yann LeCun http://arxiv.org/abs/1312.6229

NOTE: The model files for this algorithm was greater than 50MB, we had to zip it up to be able to push to origin. Please unzip deep-svm

1. In this method, we try to invoke an external program that will run a given image up to 12th layer. The program outputs the the feature vector at the layer into STDOUT.
2. These feature vectors are captured and used in the SVM classifier.



```
Confusion matrix:
                     ch su pi mu pa ti sa ha cr sa ho sc br ta ch po la wa sp ba fr ku br ja pu
       chickennugget  8. 0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
               sushi  0  7. 0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0
               pizza  0  0  9. 0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
              muffin  0  0  0  6. 0  0  0  0  0  0  0  1  1  0  0  0  0  1  0  0  0  0  1  0  0
              paella  0  0  1  0  7. 0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1  0
            tiramisu  0  0  0  0  0  6. 0  0  0  0  0  0  2  0  1  0  0  0  0  1  0  0  0  0  0
               salad  0  0  0  0  0  0 10. 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           hamburger  0  0  0  0  0  0  0  9. 0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
           croissant  0  0  0  0  0  0  0  0 10. 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
              salmon  0  1  0  0  0  1  2  0  0  5. 0  0  0  0  0  0  1  0  0  0  0  0  0  0  0
              hotdog  0  0  0  0  0  0  0  0  0  0 10. 0  0  0  0  0  0  0  0  0  0  0  0  0  0
               scone  0  0  1  0  0  0  0  0  0  1  0  4. 0  0  1  0  1  0  0  0  0  0  2  0  0
             brownie  0  0  0  0  0  0  0  0  0  0  0  0 10. 0  0  0  0  0  0  0  0  0  0  0  0
                taco  0  0  0  1  0  0  0  0  0  0  0  0  0  6. 0  0  1  0  1  0  0  0  0  1  0
              churro  2  0  0  0  0  1  0  0  0  0  0  0  1  0  5. 0  0  1  0  0  0  0  0  0  0
             popcorn  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10. 0  0  0  0  0  0  0  0  0
             lasagna  0  0  0  1  0  0  0  0  0  0  0  0  1  0  0  0  8. 0  0  0  0  0  0  0  0
              waffle  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  1  6. 0  1  1  0  0  0  0
           spaghetti  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  9. 0  0  0  0  0  0
               bagel  0  1  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  6. 0  0  1  0  0
         frenchfries  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10. 0  0  0  0
      kungpaochicken  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10. 0  0  0
               bread  0  0  0  0  0  0  0  0  1  0  0  1  0  0  0  0  0  0  0  0  0  0  8. 0  0
           jambalaya  0  0  0  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  6. 0
             pudding  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10.
Classifier accuracy: 195 of 250 =    78%  (versus random guessing accuracy of     4%)
[smhaiska@tank tpkhot-srriyer-smhaiska-a3]$
```

Image resized to 231 x 231 .
Accuracy : 195 0f 250 = 78%

Model Folder: ./deep-svm/

Comparison

Quantitative comparison:

Type	Parameters	Correct Recognition count	Accuracy
Baseline	Full RGB channels (4800 vectors)	47 out of 250	19%
Baseline	Greyscale (1600	24 out of 250	9.6%

	vectors)		
Eigen Food	Resize 40x40, Top 40 eigen vectors	24 out of 250	9.6%
Eigen Food	Resize 40x40, Top 20 eigen vectors	27 of 250	11%
Eigen Food	Resize 40x40, Top 16 eigen vectors	24 out of 250	9.6%
Haar-Like	No of features: 6400, 40x40 resize	39 of 250	16%
Haar-like	No of features: 1600, 100x100	35 of 250	14%
Bag of Words	Cluster count = 30	89 of 250	36%
Bag of Words	Cluster count = 100	112 of 250	45%
Deep Learning	Resize: 231x231.	195 0f 250	78%

- Apart from the accurary reported above, there are quite a few qualitative data points where the classifiers differ:
  - Time take to train: Baseline takes the least amount to time taken to train, followed by Haar-like. Bag of words and deep neural networks take the largest amount of time.
  - Parameters: Classifier such as baseline classifier are simple to tune. However classifiers such as EigenFood or Bag of words extract the feature vectors from non-image space (SIFT-space, or eigen-space). The parameters we set over here are not so intuitive.