Assignment 3

CSCI B657 – Computer Vision

Jsureshk-bde-sriksrin

Bipra De
Jayalakshmi Suresh
Srikanth Srinivas Holavanahalli

1. Files Submitted

Sr. No.	File Name	Included/Modified
1	a3.cpp	Modified
2	Makefile	Modified
3	Baseline.h	Included
3a	model_file_baseline	Included
4	Eigen.h	Included
4a	Model_file_eigen	Included
5	haar.h	Included
5a	Model_file_haar	Included
6	Bow.h	Included
6a	model_file_bow	Included
6b	centroids_bow.dat	Included
7	CNN.h	Included
7a	Model_file_deep	Included

- Baseline.h - contains code for core logic for baseline code of part1
- Eigen.h – contains the code for core logic for eigen implementation of part 2
- haar.h – contains the code for core logic for viola jones implementation of part 2
- bow.h – contains the code for core logic of bag of words implementation of part 2
- CNN.h – contains the code for core logic of deep learning implementation of part 3

2. How to run the code?

- Part 1
    1) ./a3 train baseline
    2) ./a3 test baseline
- Part 2 – Eigen
    1) ./a3 train eigen
    2) ./a3 test eigen
- Part 2 – Haar
    1) ./a3 train haar
    2) ./a3 test haar
- Part 2 – Bag of words
    1) ./a3 train bow
    2) ./a3 test bow

- Part 3 – Deep learning
    1) ./a3 train deep
    2) ./a3 test deep

3. Results, Analysis and Implementation summary

SVM

For using SVM Multiclass, we have prepared the training and test file in the below format:

<Class label> <Feature1>:<Feature1_value>
<Feature2>:<Feature2_value>.....<FeatureN>:<FeatureN_value>

Note:

1. Class label must be numeric
2. Feature number must be greater than 0 (in our experiment it is the pixel index of the image flattened to a single row with index starting from 1)
3. Feature values are the pixel values of the image at the corresponding index position

NOTE: The matrix dimensions below are specified in CImg format of (width,height)

Part 1

Train phase

- We first convert the CImg matrix of an image into grayscale or leave it as colored depending upon our experimentation
- We then resize the Cimg matrix to dimension (size*size)
- Then we unroll the CImg matrix to a row matrix of dimension
    - (size*size) x 1 for gray scale
    - (size*size*3) x 1 for colored
- We then prepare a training file for SVM in the format mentioned earlier
- Next, we make a system call for executing “svm_multiclass_learn” on the training file and generate a learned model

Test phase

- We repeat the steps 1 - 3 for a test image
- Then we write the test image to a file in the format required by SVM mentioned earlier
    - We have kept the class label fixed for all the test images to some arbitrary number (2 in our case) as we are not using SVM to compute accuracy
- Next, we make a system call for executing “svm_multiclass_classify” on the test file and generate the output file.

- We then parse the output file to get the predicted class numerical id and do a look up on the map “classValue_className_map” to get the class name (i.e. folder name to which the image belongs) corresponding to the numerical class id predicted and then return it

Experimentation

- We have run the experiment with colored as well as grey scaled image
- We also varied the resolution of the resized image

How well does your program work, both quantitatively and qualitatively? Does it make a difference if you use color or not?

	Colored	**Gray Scale**
Resized to 40 x 40	Classifier Accuracy : 19%	Classifier Accuracy : 7.6%
Resized to 80 x 80	Classifier Accuracy : 18%	Classifier Accuracy : 10%

- Using Colored images clearly gave a better accuracy in our experimentation.
- It takes approx 6 minutes to train and test SVM .

Result Screenshots:

Colored Image resized to 80 x 80

```
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  0. 0  0  1  0  0  0  0  0  1  2  0  0  0  0  0  0  0  1  0  2  0  0  0  3
              sushi  0  3. 0  1  0  0  0  0  0  0  0  0  1  0  1  1  0  0  0  2  0  0  1  0  0
              pizza  1  0  1. 0  1  1  0  0  0  0  0  0  0  1  1  1  1  0  0  2  0  0  0  0  0
             muffin  1  0  0  1. 0  1  0  1  0  0  1  1  1  0  0  0  0  1  1  1  0  0  0  0  0
             paella  0  0  0  0  1. 0  0  2  0  1  1  0  0  0  0  0  1  0  0  0  0  0  1  0  3
           tiramisu  0  0  1  2  0  2. 0  0  0  0  0  0  1  1  0  0  0  1  0  1  0  0  1  0  0
              salad  0  0  0  0  0  0  5. 0  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  3
          hamburger  0  0  0  0  0  0  0  8. 0  0  0  0  0  0  0  0  0  0  0  1  1  0  0  0  0
          croissant  2  0  0  0  0  1  1  0  0. 0  1  0  1  0  2  0  0  0  0  0  0  1  1  0  0
             salmon  0  0  0  0  0  0  2  0  0  0. 1  1  0  1  0  1  0  1  1  1  0  0  1  0  0
             hotdog  2  0  0  0  0  0  0  0  0  1  0  3. 0  1  0  0  0  0  0  0  1  0  1  0  1  0
              scone  1  0  0  1  0  1  0  0  0  0  0  1. 1  0  2  0  1  1  1  0  0  0  0  0  0
            brownie  0  0  0  1  0  0  0  1  0  2  0  1  2. 0  0  0  0  1  0  1  0  0  0  1  0
               taco  0  0  1  0  0  0  1  1  0  0  0  0  1  2. 0  0  1  1  0  0  1  0  0  1  0
             churro  1  0  0  0  0  0  0  1  0  0  2  1  0  0  1. 0  0  0  0  0  0  1  0  0  3
            popcorn  0  0  1  0  0  0  1  0  0  0  0  3  0  0  0  0. 1  0  0  0  0  0  2  1  1
            lasagna  0  0  0  0  1  0  0  1  0  1  1  0  1  0  0  0  0. 0  0  2  1  0  0  1  1
             waffle  1  0  0  0  1  1  0  1  0  0  0  0  0  0  0  0  0  3. 0  1  1  0  1  0  0
          spaghetti  0  0  0  0  1  0  0  0  0  0  0  1  0  1  0  0  0  1  3. 1  1  0  0  0  1
              bagel  1  0  1  1  0  1  0  1  0  0  0  0  0  0  0  2  0  0  0  0  3. 0  0  0  0  0
        frenchfries  2  0  1  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0  0  1  0  1. 0  0  0  3
      kungpaochicken  2  0  0  0  0  0  0  1  0  0  0  0  2  0  0  0  0  1  2  0  0  0. 0  2  0
              bread  1  3  0  0  0  0  0  0  0  0  0  1  0  0  0  1  1  1  1  0  0  0  0  1. 0  0
          jambalaya  2  0  0  0  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0  1  2  1  0  0  2. 0
            pudding  1  0  0  0  0  0  0  1  1  0  0  0  0  0  1  1  0  1  0  0  1  0  0  0  3.
Classifier accuracy: 46 of 250 =    18%  (versus random guessing accuracy of     4%)
```

Grey Scale image with resized to 80 x 80

```
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  0. 0  1  0  0  0  0  0  0  0  0  0  2  0  0  0  1  0  2  1  0  0  0  0  3
             sushi   0  1. 0  1  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  2  0  5
              pizza  1  0  1. 0  1  1  0  0  0  0  0  0  0  1  0  0  1  0  0  1  0  0  0  0  3
            muffin   1  0  0  1. 0  1  2  2  0  1  0  0  1  0  0  1  0  0  0  0  0  0  0  0  0
             paella  0  1  0  0  2. 0  0  2  0  0  1  0  0  0  0  0  0  0  0  1  0  0  0  0  3
           tiramisu  0  1  0  0  0  2. 1  1  0  0  0  0  1  0  0  0  0  1  0  2  0  0  0  0  1
              salad  2  0  0  0  0  1  0. 1  0  0  0  0  1  0  0  0  1  0  0  0  2  0  1  0  1
          hamburger  0  1  1  0  0  0  0  6. 0  0  0  0  0  1  0  0  0  0  0  1  0  0  0  0  0
          croissant  1  0  0  0  0  1  0  1  0. 0  0  1  2  1  0  0  0  0  0  1  0  0  1  1  0
             salmon  1  0  0  0  0  0  1  1  0  0. 1  0  0  1  0  0  0  1  0  1  2  0  0  0  1
            hotdog   1  1  0  0  0  0  0  0  0  1  0  1. 1  0  0  2  0  0  0  0  0  0  1  1  0  1
             scone   1  0  1  1  0  0  0  0  0  0  0  1  1. 1  0  0  0  0  1  0  1  0  0  0  0  2
           brownie   0  0  0  0  0  0  0  1  1  0  0  1  1  1. 0  0  0  0  0  2  2  0  0  0  1  0
              taco   0  0  0  0  0  0  0  0  1  1  0  0  1  1  1. 1  0  0  1  0  0  2  0  0  0  1
            churro   0  0  0  1  0  1  0  1  0  0  2  0  0  1  0. 0  0  0  0  0  0  0  1  0  3
           popcorn   1  2  0  0  1  0  0  0  0  0  0  1  0  0  0  1. 0  0  0  0  1  1  0  0  2
           lasagna   2  0  0  0  0  0  1  0  0  0  1  1  0  0  0  0  0. 0  0  1  1  0  0  1  2
            waffle   0  0  0  0  1  1  0  2  0  0  0  0  1  2  0  0  0  1. 1  1  0  0  0  0  0
         spaghetti   1  1  0  1  0  0  0  0  0  0  0  1  0  0  1  0  0  1  0. 0  2  0  0  0  2
             bagel   0  0  0  1  0  0  0  2  0  0  1  0  0  0  2  1  0  0  0  3. 0  0  0  0  0
        frenchfries  0  0  1  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0  0  1  3. 0  1  0  2
     kungpaochicken  1  0  1  0  0  0  1  2  0  0  0  0  1  1  1  0  0  1  0  1  0  0. 0  0  0
              bread  0  1  1  1  1  0  0  0  0  0  1  0  0  0  1  0  1  1  0  0  0  0  0. 0  2
          jambalaya  2  1  0  0  0  0  0  0  0  0  1  0  0  0  0  0  1  1  1  2  0  0  0  0. 1
            pudding  2  0  0  0  1  0  0  0  3  0  0  0  0  0  0  1  0  2  0  0  0  0  0  0  1.
Classifier accuracy: 26 of 250 =    10%  (versus random guessing accuracy of     4%)
```

Part 2

Eigenfood

Train phase

- We first convert the CImg matrix of an image into grayscale
- We then resize the Cimg matrix to dimension (size*size)
- Then we unroll the CImg matrix to a row matrix of dimension (size*size) x 1 for gray scale

PCA

- Then we prepare a CImg matrix of dimension (number of train images) * (size*size) where size is the width and height of a resized image. [**imageMatrix**]
- We then find the mean of each row of the imageMatrix and prepare a mean matrix of dimension (size * size) x 1 [**meanMatrix**]
- Now, do matrix subtraction imageMatrix - meanMatrix [meanSubtractedImageMatrix]
- Next, we compute the covariance matrix of dimension (size*size) x (size*size) [**covarianceMatrix**]
- meanSubtractedImageMatrix * meanSubtractedImageMatrix.get_Transpose()
- Then, we compute the eigen vector and eigen value from the covariance matrix
- covarianceMatrix.symmetric_eigen(eigenVal,eigenVec);
- We then call the **generateEigenFace**() function to generate eignen faces from the eigen vector

- Then, we extract a matrix from the eigen vector of the dimension k * (size * size) where k is the magic parameter for selecting top k eigen values. We then transpose this matrix. [**dimensionAfterPCA**]
- The dimensionAfterPCA is saved to a file "PCA.txt" to be used later in classify method.
- We then multiply imageMatrix with dimensionAfterPCA and take a transpose of the result .This gives us all train images with top k features. [**imageMatrixAfterPCA**]
- We then write this matrix to a train file in the format required by SVM specified above.
- To assign the class label to each image in the train file, we used the vector classNumber that stores the numeric id of each class
- We then perform a system call to run "svm_multiclass_learn" on the training file and generate a model file "model_file_eigen"

Test phase

- We repeat the steps 1 - 3 for a test image and transpose it. So the dimension of the resultant matrix is (size*size) x 1 [**test_image**]
- We then read the dimensionAfterPCA matrix from the file "PCA.txt"
- We multiply the dimensionAfterPCA matrix with test_image matrix and take a transpose of the result [**imageMatrixAfterPCA**]
- Next, we prepare a train file in SVM format from the imageMatrixAfterPCA
- We have kept the class label fixed for all the test images to some arbitrary number (2 in our case) as we are not using SVM to compute accuracy
- Next, we make a system call for executing "svm_multiclass_classify" on the test file and generate the output file.
- We then parse the output file to get the predicted class numerical id and do a lookup on the map "classValue_className_map" to get the class name (i.e. folder name to which the image belongs) corresponding to the numerical class id predicted and then return it

Experimentation

- We have run the experiment with different values for K that specified the number of top eigne values to select
- We also varied the resolution of the resized image

Results

	Resized to 40 x 40	**Resized to 60 x 60**
k=20	8.8%	6.8%
k=80	13%	8.4%

Screenshot

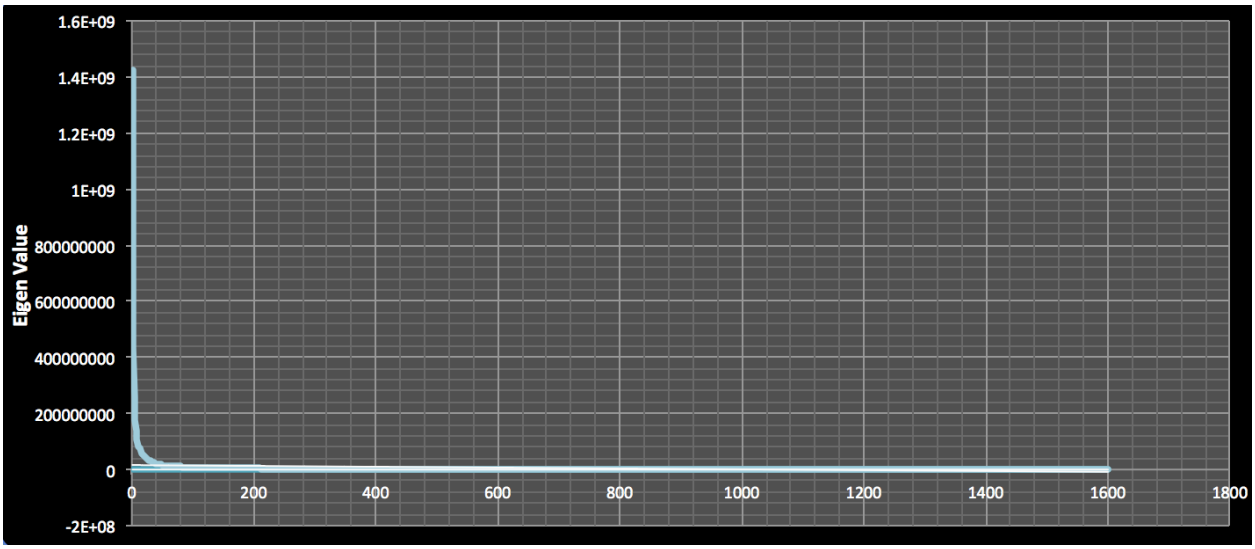
Selecting top 80 eigen values and image resized to 40 x 40

```
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  0. 0  0  0  1  1  2  0  0  0  1  0  1  1  0  0  0  0  1  1  0  0  0  0  1
             sushi   0  0. 0  0  2  0  0  0  0  0  0  1  1  0  0  0  1  0  1  1  1  1  1  0  0
              pizza  0  0  3. 0  1  1  0  0  0  1  0  0  0  0  0  0  1  0  0  0  1  0  1  0  1
            muffin   0  1  0  0. 1  1  0  3  0  1  0  0  0  0  0  0  0  0  0  0  0  1  1  0  1
             paella  1  0  1  1  2. 0  0  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  1  1  0
           tiramisu  1  0  0  2  0  2. 1  1  0  0  0  0  1  0  0  1  0  0  0  0  0  1  0  0  0
              salad  2  0  0  0  0  2  1. 2  0  0  0  0  1  0  0  0  0  0  0  0  1  1  0  0  0
          hamburger  0  0  0  2  0  0  1  2. 1  0  0  1  1  0  1  1  0  0  0  0  0  0  0  0  0
          croissant  0  1  0  0  0  1  1  1  0. 0  0  0  1  1  0  0  0  1  3  0  0  0  0  0  0
             salmon  1  1  0  1  0  0  2  1  0  1. 0  0  0  0  0  1  1  0  0  1  0  0  0  0  0
            hotdog   0  0  0  1  1  0  1  3  1  0  2. 0  0  0  0  0  0  0  0  1  0  0  0  0  0
              scone  0  0  1  0  0  0  1  0  0  0  1  2. 0  0  0  0  2  0  1  1  1  0  0  0  0
            brownie  0  0  0  1  0  0  2  2  0  0  0  0  2. 0  0  1  0  0  0  0  0  2  0  0  0
               taco  1  1  0  0  0  0  0  2  0  0  0  0  1  1. 0  1  0  0  1  0  0  0  0  2  0
             churro  0  1  0  1  0  1  0  1  1  0  1  0  1  0  0. 0  0  0  1  0  0  0  0  0  2
            popcorn  0  0  0  0  1  0  0  0  0  0  0  2  0  1  0  5. 0  0  0  1  0  0  0  0  0
            lasagna  0  1  0  0  1  0  0  1  1  0  0  0  1  0  1  1  0. 1  0  1  0  0  0  1  0
             waffle  0  0  0  1  1  1  1  0  0  0  0  0  0  1  0  0  0  2. 0  0  0  2  1  0  0
           spaghetti  0  1  0  0  0  0  0  0  0  0  0  0  1  1  1  0  1  0  1  3. 0  1  0  0  0
              bagel   0  1  0  2  0  1  0  1  0  0  0  0  1  0  0  1  0  1  1  1. 0  0  0  0  0
        frenchfries   0  0  0  0  1  0  0  0  0  0  0  2  0  1  2  0  2  1  0  0  0. 0  1  0  0
      kungpaochicken  0  0  0  1  1  1  1  0  0  0  0  0  2  0  1  0  0  0  0  0  0  3. 0  0  0
              bread   0  2  2  0  0  1  0  0  1  0  1  0  0  1  0  2  0  0  0  0  0  0  0. 0  0
           jambalaya  0  0  0  0  0  0  1  0  0  0  1  0  2  0  1  2  0  2  0  0  0  1  0  0. 0
             pudding  0  0  0  0  1  0  1  0  0  0  1  0  0  1  0  1  0  0  3  0  1  0  0  0  1.
Classifier accuracy: 33 of 250 =    13%  (versus random guessing accuracy of     4%)
[bde@silo jsureshk-bde-sriksrin-a3]$ clear
```

What do the top few Eigenvectors look like, when plotted as images?

Kindly see the images generated in the project directory with filename as EigenFace_<image number>.png

How quickly do the Eigenvalues decrease?



Eigenvalues have a sharp decrease after K=135, but the behavior is erratic. It rises and falls again as we keep increasing the k.

Viola Jones

Brief outline of the algorithm

Training phase

- Step 1: Resize the image to 40*40 size using resize() function
- Step 2: Generate 4500 2-rectangles. The rectangles are generated randomly across 40*40 region by randomly selecting x, y values and also randomly selecting height and width.
- Step 3: The above generated rectangles are same across all images
- Step 4: For each rectangle, the sum of all pixels under that rectangle is calculated [white rectangle]. The grey rectangle is generated by adding the height to the white rectangle so that the grey rectangle is just below white rectangle. Sum of all pixels under grey rectangle is calculated.
- Step 5: Difference of the sums of the 2 rectangles are calculated and appended as a feature value in the train_haar.dat
- Step 6: The train_haar.dat is appended values for all images across all folders.

Testing Phase

- Step 1: The test image is resized to 40*40
- Step 2: Using the previously generated rectangles generate the 4500 features and appended to test_haar.dat
- Step 3: Use system call to give test.dat file as an input to svm_classify along with previously created model_file
- Step 4: Repeat the steps 1-3 for all the test images

Results:

- Following the screenshot of the result obtained for images,
    - Greyscale images of size 40*40
    - 4500 2-rectangles

```
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  0. 1  2  0  0  0  0  2  0  2  0  0  0  0  0  1  0  0  1  0  0  0  0  1
             sushi  0  2. 0  2  0  0  0  0  0  0  1  0  1  0  1  1  0  0  0  0  0  0  1  0  1
             pizza  1  1  1. 0  1  1  0  0  0  0  0  0  0  0  0  0  1  0  0  3  0  1  0  0  0
            muffin  0  1  0  1. 0  1  0  2  0  0  0  1  1  0  0  0  0  0  0  0  0  0  1  1  1
            paella  1  0  0  0  0. 0  0  3  0  0  1  0  0  0  0  0  2  0  0  1  1  0  0  0  1
          tiramisu  1  1  0  0  0  3. 0  0  0  2  0  0  0  0  0  0  0  0  0  1  0  0  2  0  0
            salad  1  0  1  0  0  0  0. 2  0  0  0  0  0  0  0  1  1  0  0  0  0  2  0  1  0  1
         hamburger  0  2  0  0  0  1  0  4. 0  0  0  0  0  0  0  0  0  0  0  0  0  1  2  0  0
         croissant  0  2  0  0  0  0  2  1  0. 0  0  0  0  1  0  0  0  0  0  1  0  0  3  0  0
            salmon  1  0  0  0  1  1  0  0  0  1. 1  0  0  0  0  1  0  1  0  0  1  0  2  0  0
            hotdog  0  1  1  0  0  0  0  1  0  2  0. 0  1  0  0  1  0  1  0  0  0  1  1  0  0
             scone  0  1  2  0  0  0  0  1  1  1  1  0. 0  0  0  0  0  2  0  1  0  0  0  0  0
           brownie  1  0  0  1  0  0  1  1  0  0  0  1  2. 0  0  2  0  0  0  1  0  0  0  0  0
              taco  0  0  0  1  0  1  1  0  0  0  1  0  0  1. 0  0  0  1  1  0  1  1  0  1  0
            churro  0  1  0  0  0  1  0  1  1  0  0  0  1  1  0. 1  0  0  0  0  0  0  1  0  2
           popcorn  0  0  1  0  1  0  0  1  0  2  0  1  0  0  1  0. 1  0  0  0  1  1  0  0  0
           lasagna  0  0  3  0  0  0  2  0  0  0  0  0  0  0  0  0  1. 0  0  0  0  1  0  1  2
            waffle  0  0  0  0  2  1  1  1  0  0  0  1  0  1  0  0  0  0. 0  0  0  0  1  1  1
         spaghetti  0  1  0  0  1  1  1  0  0  0  0  0  0  0  1  0  1  0  0  0. 0  0  0  1  2  1
             bagel  0  0  0  0  0  0  0  3  0  0  0  0  0  1  0  1  0  0  1  1  1. 1  0  1  0  0
       frenchfries  0  0  0  0  2  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  1  3. 1  0  1  1
     kungpaochicken  1  0  1  1  0  0  1  1  0  0  0  0  1  0  0  1  0  0  0  1  1  1. 0  0  0
             bread  0  1  1  1  0  0  0  0  1  0  1  0  0  1  0  0  1  2  0  0  1  0  0. 0  0
         jambalaya  0  0  0  0  0  2  1  0  0  0  0  0  0  0  0  1  0  1  0  0  3  0  0  1  0. 1
           pudding  0  1  3  0  1  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  1  0  3.
Classifier accuracy: 24 of 250 =   9.6%  (versus random guessing accuracy of     4%)
```

- Following the screenshot of the result obtained for images,
    - Greyscale images of size 40*40
    - 4500 2-rectangles

```
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  0. 1  1  0  0  0  1  1  1  1  0  0  0  0  0  1  1  0  1  0  0  1  0  0  0
             sushi  0  3. 0  1  0  1  1  0  0  0  1  0  1  0  0  0  1  1  0  0  0  0  0  0  0
             pizza  1  2  2. 0  1  0  0  1  0  1  0  0  0  0  0  0  0  0  0  1  0  1  0  0  0
            muffin  0  1  1  0. 0  1  1  1  0  1  0  1  1  1  0  0  0  0  0  0  0  0  1  0
            paella  2  1  0  0  0. 0  0  1  0  1  1  0  0  0  0  2  0  0  0  0  1  0  0  0  1
          tiramisu  0  2  0  1  0  1. 0  0  1  0  1  0  0  1  0  0  0  0  0  0  0  0  2  1  0
            salad  0  0  0  0  0  1  1. 0  1  0  0  0  0  3  1  0  0  0  1  0  0  1  1  0  0
         hamburger  0  3  0  0  0  1  0  2. 1  0  0  0  1  0  0  0  0  0  0  1  1  0  0  0  0
         croissant  0  1  0  0  0  1  1  1  0. 0  0  0  0  1  1  0  0  0  0  1  0  0  3  0  0
            salmon  0  0  1  0  2  0  0  0  0  1. 1  0  1  0  0  0  1  0  0  0  1  0  1  0  1
            hotdog  0  2  0  0  0  0  0  0  0  3  0. 0  1  0  0  1  1  0  0  0  0  0  1  0  1
             scone  0  0  2  0  0  0  1  0  1  0  2  0  0. 0  0  0  0  1  1  0  0  1  0  0  0  1
           brownie  0  1  0  1  0  0  1  1  0  0  0  1  3. 0  0  1  0  0  0  1  0  0  0  0  0
              taco  0  0  0  0  0  1  1  0  0  1  1  0  1  1. 0  0  0  1  1  0  1  0  0  1  0
            churro  0  0  0  0  0  3  0  1  0  0  0  1  1  0  1. 0  0  0  0  0  0  0  1  0  0  2
           popcorn  0  0  0  0  0  0  0  1  0  3  1  1  0  0  0  0. 1  0  1  0  1  1  0  0  0
           lasagna  0  0  2  0  0  0  1  1  0  0  0  0  1  0  0  0  1. 0  0  0  0  0  0  2  2
            waffle  0  1  2  0  0  0  2  1  0  0  0  0  2  0  0  0  0. 0  0  0  0  0  1  1  0
         spaghetti  0  1  0  0  1  0  1  0  0  0  1  0  0  0  0  0  0  2  0. 0  0  0  0  2  2
             bagel  0  1  0  0  0  0  0  2  0  0  0  0  1  0  0  1  0  2  0  2. 0  0  0  1  0
       frenchfries  0  0  1  1  1  0  0  0  0  2  0  0  0  1  0  0  0  0  0  0  2. 1  0  0  1
     kungpaochicken  0  1  1  0  0  0  0  1  0  0  0  0  1  0  0  0  0  2  0  1  1  0. 1  1  0
             bread  0  2  1  1  1  0  0  0  0  0  1  0  0  0  1  0  1  1  0  0  1  0  0. 0  0
         jambalaya  1  0  0  0  0  0  0  0  0  0  0  0  1  0  1  1  1  1  0  1  2  0  1  0. 0
           pudding  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  1  1  0  0  0  2  0  0  0  4.
Classifier accuracy: 24 of 250 =   9.6%  (versus random guessing accuracy of     4%)
```

Bag of words

Brief outline of the algorithm

Training phase

- For each image in the training image, sift descriptors are extracted.
- A bank of sift vectors from all the training images is formed and is clustered into k clusters using k means.
- Then, the k dimensional feature vector for each training image is formed by putting the number of sift descriptor from the training image in a cluster as a feature. That is, if 20 of the sift vectors of training image 1 is in cluster 0, then the 0th feature value of training image is 20.
- Then, the training data and the k centroids are written into two different files.

Testing Phase

- In the test mode, it loads the k centroids.
- It obtains all the sift descriptors of the test image and assigns each to the cluster it belongs to.
- It then computes the count of sift descriptors of the test image in each cluster and forms a feature vector for the test image. That is, if 20 sift descriptors of test image 1 belongs to cluster 0, then the 0th feature is 20.
- It runs the SVM model built during train on the test image and predicts the label.
- It takes about 15 mins for classifying all the images.

Assumptions and obstacles

- It takes from 25 to 35 minutes to train depending on the k value. Major portion of the time is spent on getting the sift descriptors of the images, which can be reduced by sub-sampling the image, but to have an enriched data, we used all the sift descriptors of all the images.

- A major obstacle was to figure out when to stop k means. To give a little room for convergence and to save from running a really long time, convergence of kmeans was as follows. If 90% of the centroid values in the next iteration is within a Euclidean distance of 15, it converges.

- Another difficulty was the data structure, because when we have the bank of sift descriptors, we couldn't just discard information about the file it belongs to. We need the information about the origin of sift descriptor to be able form the training features.

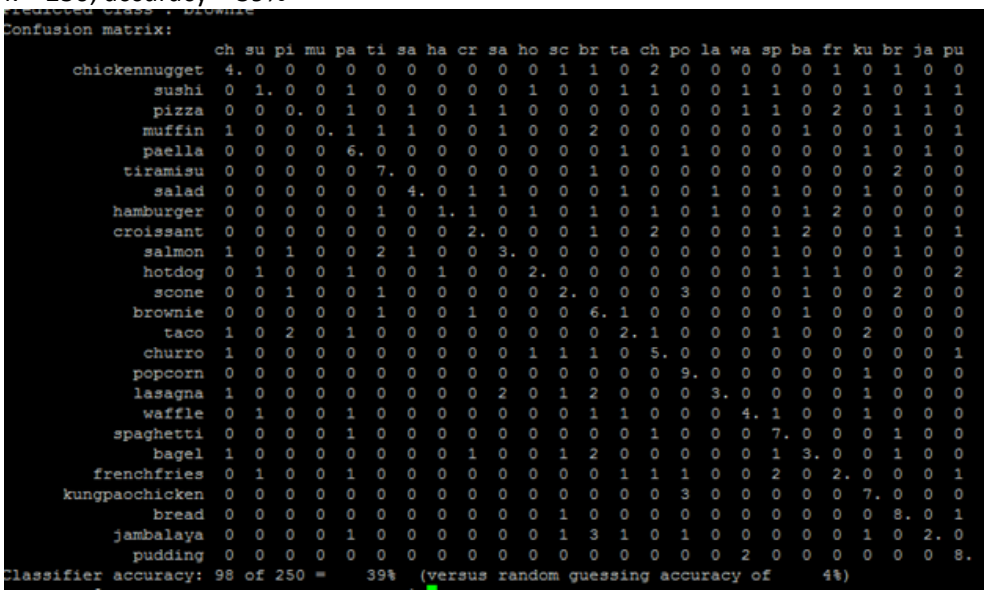
- Thus, the algorithm performs k means with clusters represented in a data structure that stores the coordinates of the sift descriptors in the original data bank.

- The data bank is a map of vectors of vectors of sift descriptors: map<string, vector<vector<SiftDescriptor> > >

- The label or class maps to a vector of files. Each entry in the vector of files is a vector of sift descriptors which are that image's sift descriptor.

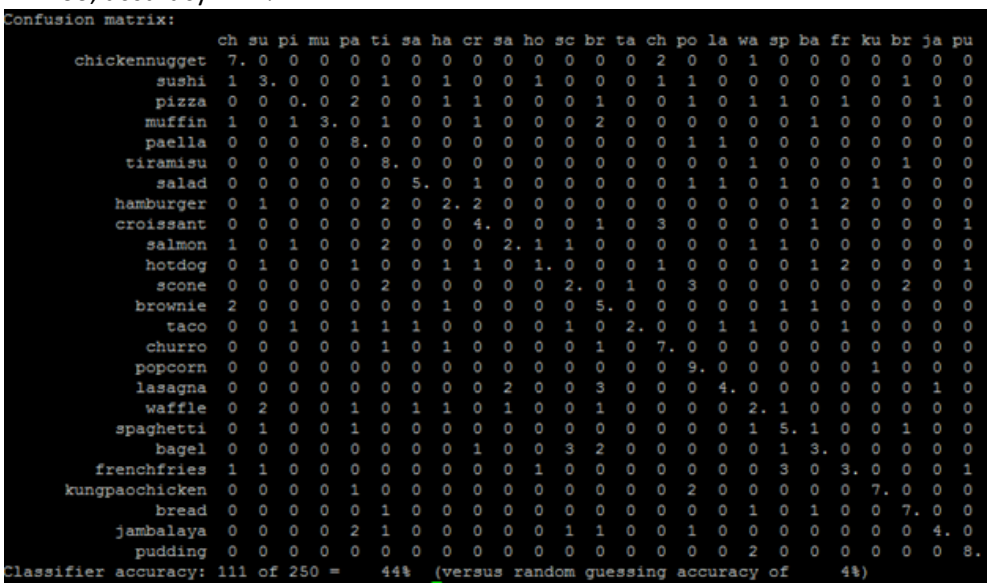
Results:

k = 250, accuracy = 39%



```
Predicted class : brownie
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  4. 0  0  0  0  0  0  0  0  0  0  1  1  0  2  0  0  0  0  0  1  0  1  0  0
             sushi   0  1. 0  0  1  0  0  0  0  0  1  0  0  1  1  0  0  1  1  0  0  1  0  1  1
              pizza  0  0  0. 0  1  0  1  0  1  1  0  0  0  0  0  0  0  1  1  0  2  0  1  1  0
            muffin   1  0  0  0. 1  1  1  0  0  1  0  0  2  0  0  0  0  0  0  1  0  0  1  0  1
            paella   0  0  0  0  6. 0  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0  1  0  1  0
          tiramisu   0  0  0  0  0  7. 0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  2  0  0
             salad   0  0  0  0  0  0  4. 0  1  1  0  0  0  1  0  0  1  0  1  0  0  1  0  0  0
         hamburger   0  0  0  0  0  1  0  1. 1  0  1  0  1  0  1  0  1  0  0  1  2  0  0  0  0
         croissant   0  0  0  0  0  0  0  0  2. 0  0  0  1  0  2  0  0  0  1  2  0  0  1  0  1
            salmon   1  0  1  0  0  2  1  0  0  3. 0  0  0  0  0  0  0  0  1  0  0  0  1  0  0
            hotdog   0  1  0  0  1  0  0  1  0  0  2. 0  0  0  0  0  0  0  1  1  1  0  0  0  2
             scone   0  0  1  0  0  1  0  0  0  0  0  2. 0  0  0  3  0  0  0  1  0  0  2  0  0
           brownie   0  0  0  0  0  1  0  0  1  0  0  0  6. 1  0  0  0  0  0  1  0  0  0  0  0
              taco   1  0  2  0  1  0  0  0  0  0  0  0  0  2. 1  0  0  0  1  0  0  2  0  0  0
            churro   1  0  0  0  0  0  0  0  0  0  0  1  1  1  0  5. 0  0  0  0  0  0  0  0  1
           popcorn   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  9. 0  0  0  0  0  1  0  0  0
           lasagna   1  0  0  0  0  0  0  0  0  0  2  0  1  2  0  0  0  3. 0  0  0  0  1  0  0  0
            waffle   0  1  0  0  1  0  0  0  0  0  0  0  1  1  0  0  0  4. 1  0  0  1  0  0  0
         spaghetti   0  0  0  0  1  0  0  0  0  0  0  0  0  0  1  0  0  0  7. 0  0  0  1  0  0
             bagel   1  0  0  0  0  0  0  0  1  0  0  1  2  0  0  0  0  0  1  3. 0  0  1  0  0
        frenchfries  0  1  0  0  1  0  0  0  0  0  0  0  0  1  1  1  0  0  2  0  2. 0  0  0  1
      kungpaochicken 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0  0  0  7. 0  0  0
              bread  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  8. 0  1
           jambalaya 0  0  0  0  1  0  0  0  0  0  0  1  3  1  0  1  0  0  0  0  0  0  1  0  2. 0
            pudding  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  8.
Classifier accuracy: 98 of 250 =    39%  (versus random guessing accuracy of     4%)
```

k = 100, accuracy = 44%



```
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  7. 0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  1  0  0  0  0  0  0  0
             sushi   1  3. 0  0  0  1  0  1  0  0  1  0  0  0  1  1  0  0  0  0  0  0  1  0  0
              pizza  0  0  0. 0  2  0  0  1  1  0  0  0  1  0  0  1  0  1  1  0  1  0  0  1  0
            muffin   1  0  1  3. 0  1  0  0  1  0  0  0  2  0  0  0  0  0  0  1  0  0  0  0  0
            paella   0  0  0  0  8. 0  0  0  0  0  0  0  0  0  0  0  1  1  0  0  0  0  0  0  0
          tiramisu   0  0  0  0  0  8. 0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1  0  0
             salad   0  0  0  0  0  0  5. 0  1  0  0  0  0  0  0  0  1  1  0  1  0  0  1  0  0  0
         hamburger   0  1  0  0  0  2  0  2. 2  0  0  0  0  0  0  0  0  0  0  0  1  2  0  0  0  0
         croissant   0  0  0  0  0  0  0  0  4. 0  0  0  1  0  3  0  0  0  0  1  0  0  0  0  1
            salmon   1  0  1  0  0  2  0  0  0  2. 1  1  0  0  0  0  0  0  1  1  0  0  0  0  0
            hotdog   0  1  0  0  1  0  0  1  1  0  1. 0  0  0  1  0  0  0  0  1  2  0  0  0  1
             scone   0  0  0  0  0  2  0  0  0  0  0  2. 0  1  0  3  0  0  0  0  0  0  2  0  0
           brownie   2  0  0  0  0  0  0  1  0  0  0  0  5. 0  0  0  0  0  0  1  1  0  0  0  0  0
              taco   0  0  1  0  1  1  1  0  0  0  0  1  0  2. 0  0  1  1  0  0  1  0  0  0  0
            churro   0  0  0  0  0  1  0  1  0  0  0  0  1  0  7. 0  0  0  0  0  0  0  0  0  0
           popcorn   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  9. 0  0  0  0  0  1  0  0  0
           lasagna   0  0  0  0  0  0  0  0  0  0  2  0  0  3  0  0  0  4. 0  0  0  0  0  0  1  0
            waffle   0  2  0  0  1  0  1  1  0  1  0  0  1  0  0  0  0  2. 1  0  0  0  0  0  0  0
         spaghetti   0  1  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  1  5. 1  0  0  1  0  0
             bagel   0  0  0  0  0  0  0  0  1  0  0  3  2  0  0  0  0  0  1  3. 0  0  0  0  0
        frenchfries  1  1  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  3  0  3. 0  0  0  1
      kungpaochicken 0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  7. 0  0  0
              bread  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1  0  1  0  0  7. 0  0
           jambalaya 0  0  0  0  2  1  0  0  0  0  0  1  1  0  0  1  0  0  0  0  0  0  0  4. 0
            pudding  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  8.
Classifier accuracy: 111 of 250 =    44%  (versus random guessing accuracy of     4%)
```

k = 50, accuracy = 38%

```
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  3. 0  0  1  0  0  1  0  0  0  0  0  1  0  1  0  0  0  0  0  1  0  0  1  1
             sushi  0  3. 0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  1  0  0  3  0  0  1  0
             pizza  0  1  0. 0  1  0  0  1  1  0  0  1  0  0  0  0  0  1  2  0  0  0  1  1  0
            muffin  1  0  1  0. 1  0  1  0  0  0  0  1  2  1  0  0  0  0  0  1  0  0  0  0  1
            paella  0  0  1  0  6. 0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1  0  1  0
          tiramisu  1  0  0  0  0  6. 0  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0  1  0  0
             salad  0  0  0  0  0  0  4. 0  1  0  0  0  0  0  0  2  0  0  2  0  0  1  0  0  0
         hamburger  0  1  0  0  0  2  0  3. 1  0  0  0  0  1  0  0  0  0  0  1  1  0  0  0  0
         croissant  0  0  0  0  0  1  0  0  3. 0  0  0  1  0  2  0  0  0  1  1  0  0  0  0  1
            salmon  0  0  1  0  0  1  0  0  0  2. 1  0  0  0  1  0  0  1  0  0  0  0  2  1  0
            hotdog  0  1  0  0  0  0  1  1  1  0  0. 0  0  0  1  0  0  0  1  0  2  0  1  0  1
             scone  0  0  0  0  0  1  0  0  0  0  0  5. 0  1  0  2  0  0  0  0  0  0  1  0  0
           brownie  0  0  0  2  0  0  0  0  1  0  0  0  4. 0  0  0  0  0  1  1  0  0  1  0  0
              taco  1  0  1  0  0  1  1  0  0  1  0  0  0  1. 0  0  1  0  2  0  0  0  0  1  0
            churro  0  0  0  0  1  1  1  0  0  0  0  1  1  0  4. 0  0  0  0  0  0  0  0  0  1
           popcorn  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  9. 0  0  0  0  0  1  0  0  0
           lasagna  0  0  0  0  0  2  0  0  0  0  0  0  3  0  0  1  2. 0  0  0  0  1  0  1  0
            waffle  0  2  0  0  1  0  0  1  0  0  0  0  0  0  1  0  0  2. 0  0  0  1  0  1  1
         spaghetti  0  0  0  0  2  0  0  0  0  0  1  0  0  0  0  0  0  0  5. 0  0  0  2  0  0
             bagel  0  0  0  0  0  0  0  0  0  1  1  0  1  2  0  0  0  1  0  1  3. 0  0  0  0
       frenchfries  0  0  0  0  0  0  0  0  0  0  1  0  0  0  1  0  0  0  4  0  4. 0  0  0  0
     kungpaochicken  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  8. 0  0  0
             bread  0  0  0  0  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0  1  0  0  7. 0  0
         jambalaya  0  0  0  0  2  1  0  0  0  0  0  0  1  1  0  0  1  0  0  0  0  0  1  0  3. 0
           pudding  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  8.
Classifier accuracy: 95 of 250 =    38%  (versus random guessing accuracy of     4%)
```

k = 25, accuracy = 33%

```
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  4. 0  0  0  1  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  1  1  1  0  0
             sushi  0  1. 0  0  1  1  0  0  1  0  0  0  0  2  0  0  1  0  0  0  2  0  0  0  1
             pizza  0  0  0. 0  1  2  0  0  1  1  0  0  1  0  0  1  0  0  1  1  1  0  0  0  0
            muffin  0  0  0  0. 1  1  0  0  2  0  0  1  2  0  1  0  0  0  0  0  0  0  1  0  1
            paella  0  0  0  0  5. 0  0  0  0  0  0  0  0  1  0  1  0  1  0  0  0  0  1  0  1  0
          tiramisu  1  0  0  0  0  4. 0  0  1  1  0  0  2  0  0  0  0  0  0  0  0  0  1  0  0
             salad  0  0  0  0  0  0  2. 0  0  1  0  0  0  1  0  2  1  0  2  0  0  1  0  0  0
         hamburger  0  1  0  0  0  2  0  0. 1  0  0  0  1  0  0  0  0  0  0  0  1  2  0  0  2
         croissant  0  0  1  1  0  0  0  0  3. 0  0  0  0  0  1  0  0  1  0  0  0  0  2  0  1
            salmon  0  0  0  0  0  2  1  0  0  1. 2  1  0  1  0  0  0  1  0  0  0  0  1  0  0
            hotdog  0  0  0  0  1  0  1  0  1  0  1. 0  0  0  2  0  0  0  0  1  2  0  0  0  1
             scone  1  0  1  0  0  1  0  0  0  0  0  2. 0  1  0  2  0  0  0  0  0  0  2  0  0
           brownie  0  1  0  0  0  0  0  0  0  1  1  0  0  5. 0  1  0  0  0  0  1  0  0  0  0  0
              taco  0  0  0  0  1  1  0  0  0  0  0  0  0  0  2. 0  0  2  1  2  0  1  0  0  0  0
            churro  0  0  0  0  0  1  1  0  1  1  0  1  1  0  2. 0  0  0  0  0  0  1  0  0  1
           popcorn  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  9. 0  0  0  0  0  0  1  0  0  0
           lasagna  1  0  0  0  0  0  0  0  0  0  0  0  2  1  0  0  4. 0  0  0  0  0  0  2  0
            waffle  0  1  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  4. 0  0  0  3  0  0  1
         spaghetti  0  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  7. 0  0  0  0  0  1
             bagel  0  0  0  0  0  0  0  0  0  2  0  0  3  3  0  0  0  1  0  1  0. 0  0  0  0  0
       frenchfries  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  1  5  0  3. 0  0  0  0
     kungpaochicken  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  1  0  0  0  0  7. 0  0  0
             bread  0  0  0  0  0  1  0  0  0  0  0  0  1  0  0  0  1  0  0  0  0  0  0  7. 0  0
         jambalaya  0  0  0  0  1  0  0  0  0  0  0  0  1  2  1  0  2  0  0  0  0  0  1  0  2. 0
           pudding  0  0  0  0  1  0  0  0  0  0  0  0  1  0  0  0  0  0  1  0  0  0  0  0  0  7.
Classifier accuracy: 82 of 250 =    33%  (versus random guessing accuracy of     4%)
```

We seem to get the best results at a k = 100. The lower value gives us neither a wide range nor diverse features. The larger k values leads to a distribution of sift features in the clusters that doesn't suffice for a good model.

Part 3

CNN

Brief outline of the algorithm

Training phase

- Step 1: Resize the input image to 231*231 size using resize() function and save as file.bmp(bmp format is used since the image pixels are not lost) size 231 is chosen as per the weblink provided
- Step 2: Use system call to provide the file.bmp to overfeat package and save the output in an overfeat_results.txt file along with class value.
- Step 3: Extract the features from overfeat_results.txt file by skipping w*h dimensions
- Step 4: Append the output from step 3 to train.dat file
- Step 5: Repeat steps 2-4 for all the input images across all folders.
- Step 6: Use system call to give train.dat file as an input to svm_learn

Testing phase

- Step 1: Resize the input image to 231*231 size using resize() function and save as file
- Step 2: Use system call to provide the file.bmp to overfeat package and save the output in an overfeat_results_test.txt file
- Step 3: Extract the features from overfeat_results.txt file by skipping w*h dimensions
- Step 4: Append the output from step 3 to test.dat file
- Step 5: Use system call to give test.dat file as an input to svm_classify along with previously created model_file
- Step 6: Repeat the steps 1-5 for all the test images

Results

Following the screenshot of the result obtained for images,

- Greyscale images of size 231*231

- CNN layer 18

```
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  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
             sushi  0  7. 0  1  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  1  0  0
             pizza  1  0  5. 0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  1  0  0  1
            muffin  1  1  0  4. 0  0  0  0  1  0  0  1  0  0  0  0  0  1  0  1  0  0  0  0  0
             paella  0  0  0  0  7. 0  0  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0  1  0
          tiramisu  0  1  0  1  0  8. 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
             salad  0  0  1  0  0  0  4. 0  0  0  3  0  0  1  0  0  0  0  0  0  0  1  0  0  0
         hamburger  0  0  0  0  0  0  0  8. 1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
         croissant  0  0  0  0  0  0  0  0  9. 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
            salmon  0  0  0  0  0  1  1  0  1  3. 1  0  0  0  0  1  2  0  0  0  0  0  0  0  0
            hotdog  0  0  0  0  0  0  0  1  0  0  6. 0  0  0  0  0  0  0  0  1  2  0  0  0  0
             scone  0  0  1  0  0  1  0  0  0  0  0  1. 0  0  0  2  1  0  0  0  0  0  3  1  0
           brownie  0  1  0  1  0  0  0  0  1  0  0  0  6. 0  0  0  0  0  1  0  0  0  0  0  0
              taco  0  0  1  0  0  0  2  0  1  0  0  0  0  4. 0  0  0  0  0  0  0  1  1  0  0
            churro  1  0  0  0  0  0  0  0  0  0  2  0  1  0  6. 0  0  0  0  0  0  0  0  0  0
           popcorn  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  9. 0  0  0  0  0  1  0  0  0
           lasagna  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  1  6. 0  0  0  0  0  1  0  0
            waffle  1  0  1  0  0  0  1  1  0  0  0  1  0  1  0  0  0  3. 0  1  0  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  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  7. 0  0  1  0  0
       frenchfries  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  3  0  6. 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  1  0  0  0  1  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  7. 0  0
         jambalaya  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  1  0  1  0  0  0  1  0  5. 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: 158 of 250 =    63%  (versus random guessing accuracy of     4%)
[sriksrin@silo jsureshk-bde-sriksrin-a3]$
```

Following the screenshot of the result obtained for images,

- Color images of size 231*231
- CNN layer 18

```
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. 1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0
             sushi  0  8. 0  0  0  0  0  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
             pizza  0  0  4. 0  1  0  0  0  0  1  0  0  0  0  0  0  1  0  0  0  1  0  0  1  0
            muffin  0  0  0  5. 0  0  0  0  0  0  0  1  0  0  0  0  1  1  0  0  0  0  2  0  0
             paella  0  0  0  0  6. 0  1  0  0  0  0  0  0  0  0  2  0  1  0  0  0  0  0  0  0
          tiramisu  0  1  0  1  0  7. 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
             salad  0  1  1  0  1  0  6. 0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0
         hamburger  0  0  0  0  0  0  0  8. 1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
         croissant  0  0  0  0  0  0  0  0  9. 0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0
            salmon  0  0  0  0  0  2  0  0  1  3. 0  0  0  0  0  1  2  1  0  0  0  0  0  0  0
            hotdog  0  0  0  0  0  0  0  0  0  0  5. 0  0  0  0  0  0  1  0  3  1  0  0  0  0
             scone  1  0  1  0  0  0  0  1  0  0  0  2. 0  0  0  1  0  0  0  2  0  0  2  0  0
           brownie  0  1  0  0  0  0  0  0  0  0  0  0  5. 0  0  0  0  0  1  0  2  0  0  1  0
              taco  0  0  0  0  0  0  0  0  0  0  0  0  0  5. 0  0  0  3  0  0  0  1  0  0  1
            churro  1  0  0  0  0  0  0  0  0  0  2  0  0  0  7. 0  0  0  0  0  0  0  0  0  0
           popcorn  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  9. 0  0  0  0  0  1  0  0  0
           lasagna  0  0  0  2  0  2  0  2  0  0  0  0  0  0  0  2  2. 0  0  0  0  0  0  0  0
            waffle  0  0  0  0  0  0  1  1  0  0  0  0  0  0  1  1  0  0  3. 0  1  2  0  0  0  0
         spaghetti  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  1  0  0  8. 0  0  0  0  0  0  0
            bagel  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  7. 0  0  2  0  0
       frenchfries  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  2  0  6. 0  0  0  0
     kungpaochicken  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  9. 0  0  0
             bread  0  1  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  1  0  0  1  1  0  5. 0  0
         jambalaya  0  0  0  0  1  1  0  0  0  0  0  0  0  0  1  0  4  0  0  0  0  0  2  0  1. 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  0 10.
Classifier accuracy: 148 of 250 =    59%  (versus random guessing accuracy of     4%)
[sriksrin@silo jsureshk-bde-sriksrin-a3]$
```

Following the screenshot of the result obtained for images,

- greyscale images of size 231*231
- CNN layer 16

```
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 sm
      chickennugget  1. 1  0  4  0  0  0  0  2  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0
             sushi  1  4. 0  1  1  1  0  0  0  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0
             pizza  0  0  0. 3  0  0  1  0  0  0  0  0  0  0  0  2  1  0  1  0  1  0  0  0  1  0
            muffin  1  0  0  8. 0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0
            paella  0  0  0  3  5. 0  0  0  0  0  0  0  0  0  0  0  1  1  0  0  0  0  0  0  0  0
          tiramisu  1  0  0  4  0  0. 0  0  2  0  0  0  1  1  0  0  0  0  1  0  0  0  0  0  0  0
             salad  0  0  0  0  0  0  1. 0  0  0  3  0  0  0  0  1  0  0  3  0  0  2  0  0  0  0
         hamburger  1  0  0  2  0  0  0  1. 0  1  0  0  0  1  1  0  0  0  0  1  2  0  0  0  0  0
         croissant  1  0  0  3  1  0  0  0  0. 0  1  1  0  0  0  1  0  0  0  2  0  0  0  0  0  0
            salmon  0  0  0  5  0  0  0  0  0  0. 2  0  1  0  0  1  1  0  0  0  0  0  0  0  0  0
            hotdog  0  0  0  2  0  0  0  1  1  0  1. 0  1  0  0  1  0  2  1  0  0  0  0  0  0  0
             scone  1  0  0  3  0  0  0  0  0  0  0  0. 0  0  0  0  0  0  2  0  1  0  2  0  1  0
           brownie  1  0  0  3  0  0  0  0  0  0  1  0  2. 0  0  0  1  0  0  2  0  0  0  0  0  0
              taco  0  0  0  1  0  0  0  0  0  0  0  0  1  3. 0  1  1  1  0  0  1  1  0  0  0  0
            churro  2  0  0  1  0  0  0  0  0  0  2  0  1  0  0. 0  1  0  1  2  0  0  0  0  0  0
           popcorn  0  0  0  3  0  0  0  0  0  0  0  0  0  0  5. 0  2  0  0  0  0  0  0  0  0  0
           lasagna  1  1  0  3  0  0  0  0  1  0  1  0  0  1  0  1  0. 0  0  0  0  0  0  1  0  0
            waffle  2  0  1  4  0  0  0  0  0  0  0  0  0  0  0  2  0  0. 1  0  0  0  0  0  0  0
          spaghetti  1  1  0  1  1  0  0  0  0  0  1  0  0  0  0  1  0  0  3. 1  0  0  0  0  0  0
             bagel  1  0  0  4  0  0  0  0  0  0  0  0  0  0  0  1  0  0  2  0. 1  0  1  0  0  0
        frenchfries  0  0  0  1  1  0  0  0  0  0  0  0  0  0  2  0  1  1  0  1  0  2. 1  0  0  0
      kungpaochicken  1  0  0  3  1  0  0  0  0  0  0  0  0  1  0  2  0  1  0  0  0  1. 0  0  0  0
             bread  0  1  0  1  0  0  0  0  0  0  1  0  0  0  0  2  2  1  0  0  1  0  1. 0  0  0
          jambalaya  1  2  0  5  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  1. 0  0
            pudding  1  1  0  0  0  0  1  0  0  1  0  0  0  0  0  0  0  0  3  2  0  0  0  0  1. 0
         smallTrain  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0.
Classifier accuracy: 40 of 250 =    16%  (versus random guessing accuracy of   3.8%)
```

Following the screenshot of the result obtained for images,

- Color images of size 231*231
- CNN layer 16

```
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  0. 0  0  0  3  0  0  1  0  1  0  0  0  0  1  0  0  0  1  1  0  0  0  0  2
             sushi  0  2. 0  1  2  0  0  1  0  1  0  0  0  0  0  0  1  0  0  0  0  1  1  0  0
             pizza  0  0  2. 0  0  0  0  2  0  0  2  0  0  0  0  0  1  0  1  0  0  0  1  0  1
            muffin  0  0  0  1. 2  0  2  2  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  2  0
            paella  0  0  0  1  5. 0  0  0  0  1  0  0  0  0  0  0  2  0  0  0  0  0  0  1  0
          tiramisu  0  1  1  0  0  0. 0  3  0  0  0  0  1  0  0  0  0  0  0  1  1  1  0  1  0
             salad  0  0  1  0  1  0  1. 1  0  0  1  0  2  0  0  1  0  0  2  0  0  0  0  0  0
         hamburger  0  0  1  0  1  0  0  4. 0  0  1  0  1  0  0  0  0  0  0  1  0  1  0  0  0
         croissant  0  0  0  0  0  0  0  1  0. 0  0  0  0  0  0  0  1  1  5  1  1  0  0  0  0
            salmon  0  0  1  0  0  0  0  0  0  1. 0  0  0  0  2  0  2  0  0  1  2  0  0  1  0
            hotdog  0  0  0  0  1  0  0  0  0  1  3. 0  0  0  0  0  0  1  1  1  0  2  0  0  0
             scone  0  2  2  0  0  0  0  0  0  0  0  0. 0  0  0  0  1  0  1  2  0  1  1  0  0
           brownie  0  2  0  2  0  1  0  1  0  0  1  0  0. 0  0  1  0  0  0  1  1  0  0  0  0
              taco  0  1  0  0  0  0  0  0  0  0  0  0  0  0. 2  1  0  0  0  2  1  2  0  1  0
            churro  0  0  0  0  0  0  1  1  0  1  1  0  0  0  0. 0  2  1  0  3  0  0  0  0  0
           popcorn  0  0  1  1  1  0  0  0  0  0  0  0  0  0  0  3. 0  2  1  0  0  0  0  1  0
           lasagna  0  0  0  2  1  1  0  0  0  0  0  0  0  1  0  0  1  1. 0  0  0  2  0  0  1  0
            waffle  0  1  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  1  1. 1  0  1  2  0  1  1
          spaghetti  0  0  1  1  0  0  0  1  0  2  0  0  1  0  0  0  0  0  2. 0  0  1  0  1  0
             bagel  0  0  0  3  0  0  1  0  0  0  0  0  0  0  0  1  0  0  1  0. 1  0  1  1  1
        frenchfries  0  1  1  0  0  0  0  1  0  0  0  0  0  0  0  1  0  0  0  0  2. 2  1  1  0
      kungpaochicken  0  1  0  0  1  0  0  1  0  0  0  0  0  0  0  1  0  1  1  1  0  2. 0  1  0
             bread  0  1  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0  0  1  2  0  1  2. 1  0
          jambalaya  0  0  0  0  3  1  0  2  0  0  0  0  0  0  1  1  1  0  0  0  1  0  0  0. 0
            pudding  0  0  0  0  0  0  0  4  0  1  0  0  0  0  0  0  0  0  1  0  1  2  0  0  1  0.
Classifier accuracy: 32 of 250 =    13%  (versus random guessing accuracy of     4%)
```

Assumption to run part 3:

Overfeat package must be present in**. /overfeat/bin/linux_64/overfeat**

Comparison:

Method Name	**Greyscale(Accuracy, Time-training + testing)**	**Color(Accuracy, Time - training + testing)**
CNN Deep Learning	**63% ,** 28-30 min	**59%,** 34-35 min
Viola Jones	9.6% 22-15 min	9.6% 22-25 min
Eigen	13% 10-15 min	
Bag of words	44%, k = 100, 25-30 min	
Baseline	10% 8-10 min	19% 10-12 min

From the above quantitative results, we can infer the below qualitative results as:

- CNN deep learning method beats other methods by a large extent
- CNN provides feature values very well distinguished beyond the support vectors that SVM can learn an predict with high accuracy
- In case of bag of words, the performance is much better than that of viola jones, eigen or the base line algorithms as the features in bag of words take the sift descriptors
- Sift descriptors are features based on edge detection and thus, form a better and more pronounced feature for a classification process
- The process of k mean to the sift descriptors to form bag of parts gives us a feature set that is based on edges in the training images and relationships among them
- In case of the baseline and eigen, the features used are the direct pixel values, which are not as informative as the features we get from these other algorithms