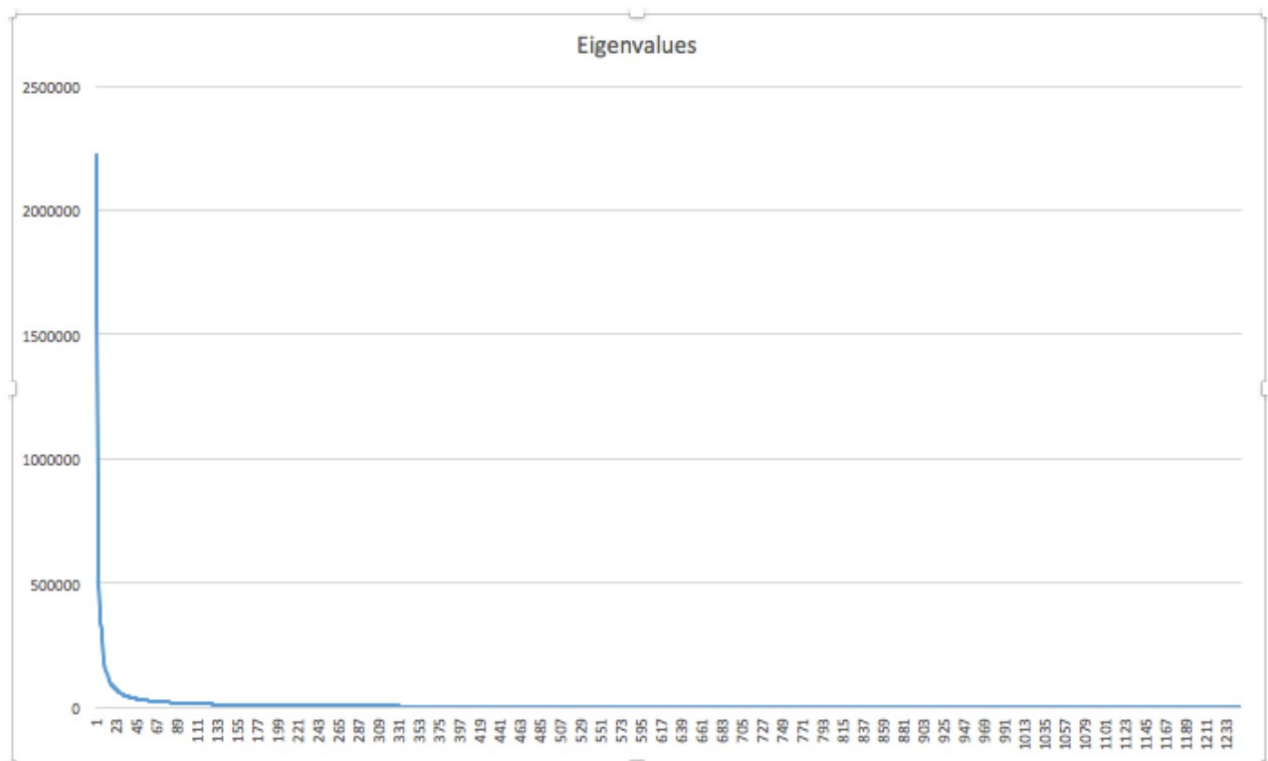


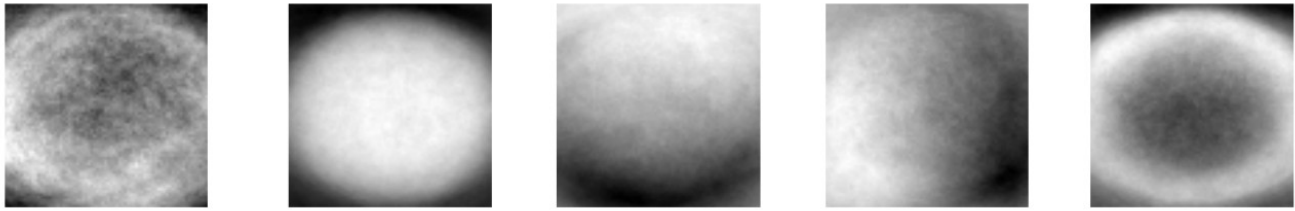
Assignment 3 - Part 2

Eigenfood, Haar-like Features, and Visual Bag-of-words

For each of these sections, the training and test images are cropped and scaled down to 73 x 73 which is the smallest dimension in one of the pictures. For Eigenfood, all the pixel values are placed in a matrix where column length is 73 x 73. That is each row of the picture has been shifted into one long vector. This matrix is then mean subtracted row wise, and a co-variance matrix calculated. A handy Eigen library is included in the code base, and is used to solve for Eigen vectors and values.



All of the eigenvalues from the training images are shown above. The highest is 2223452, and the lowest is 68. The values decrease sharply, and have a -107,001 slope over the first twenty. Over the last thousand, the slope levels off quite a bit around -5.5. This indicates fair results could be gathered with as few as 50 or so top vectors. A set of eigenvectors larger than 300-400 begins to seem redundant.



Above are the top five eigenvector represented as images. An algorithm scanned the vector for low and high values, and then scaled everything into a range of [0,255]. It seems round shapes dominated the training data in the form of bagels, hamburgers, pizza, bowls, plates, etc.

The final results scored a 23/250 on the test data using 250 Eigenvectors. Compared to the baseline score of 25/250, we see a near equal result. More importantly, the number of dimensions used to represent the images has been reduced over 2000%. And the processing time has also decreased. The complete training and testing of 'eigen' can be done in a sub interval of just training time of 'baseline'.

Haar filters are randomly generated by choosing a positive area $[5,25] \times [5,25]$, and then either one or two negative areas that fit inside some way. The maximum score of every filter is calculated using an integral map of each image. Then the best filters are selected by comparing each average score for all food categories to the average score of each category. The higher differences between a total average and a category average indicate that filter to be valuable. Once a particular category helps determine a good filter, that same category cannot be used in the following filter analyses to provide balance over all the categories. Generating more than 1000 filters did not seem to improve performance as redundancy begins to occur. The final results scored 25/250 using 150 of the best 1000 Haar filters. While this is actually a slight decrease from the baseline score, the number of reduced dimensions can be used to argue an improvement was made nevertheless.

For the bag-of-words, the SIFT algorithm generates feature points for every image. Our own implementation of k-means clustering than executes five times to extract 250 centroid points. The process is rather time intensive as it takes around 15 minutes to complete. For simplicity we calculated the k-mean centroid for the training set, and saved it as a constant file. Whenever you train and test the model with the given images it loads the centroid file and its values that were pre-computed. If you want to recalculate the centroid file the user could just change the function call to `runWordsFullTrain()`. Then each image is represented as a vector where each data is the number of times that centroid (or visual word) appears in the image. This approach scored a 65/250 on the test data, and showed the same improvement over the baseline as Eigenfood while using the same number of reduced dimensions. Overall, we conclude Eigenfood seems to be the best suited of these three approaches when factoring in both the quality of results, reduced dimensions, and computational time used.

The Eigen Confusion Matrix :

Confusion matrix:

	ha	ch	ch	sc	mu	pu	po	sp	fr	pa	br	sa	ku	wa	ba	sa	ta	ja	pi	ho	su	cr	br	la	ti
hamburger	5.	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	1
churro	1	0.	0	0	1	3	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1	0	1
chickennugget	0	0	0.	0	0	3	0	2	0	1	2	0	0	0	1	0	0	0	1	0	0	0	0	0	0
scone	0	0	1	1.	1	2	0	0	0	0	1	0	0	1	1	0	0	0	1	1	0	0	0	0	0
muffin	2	0	1	1	1.	0	1	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	1
pudding	1	0	0	0	0	1.	1	0	1	1	0	0	0	2	0	0	0	0	0	0	0	3	0	0	0
popcorn	0	0	0	0	1	3	1.	0	0	1	0	0	0	0	0	1	0	1	0	0	2	0	0	0	0
spaghetti	0	1	0	1	0	4	0	0.	2	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
frenchfries	0	0	1	1	0	2	0	0	1.	2	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0
paella	1	0	0	0	0	3	0	1	0	1.	0	0	0	0	1	0	0	0	0	1	0	0	1	1	0
brownie	1	0	1	1	0	0	0	1	0	0	2.	0	0	0	1	0	0	0	0	0	2	0	0	0	1
salad	1	1	1	0	0	1	0	0	2	0	1	0.	0	0	0	0	0	0	0	0	0	0	2	1	0
kungpaochicken	2	1	1	0	0	0	0	0	0	0	1	1	0.	2	0	1	0	0	1	0	0	0	0	0	0
waffle	1	0	0	0	0	0	0	0	0	1	1	0	0	1.	2	0	2	0	0	0	0	1	0	0	1
bagel	0	2	0	0	1	0	1	0	0	0	0	0	0	0	3.	0	0	0	0	1	1	0	0	0	1
salmon	1	0	0	0	0	2	0	0	3	0	0	0	0	1	0	1.	0	0	0	0	1	0	1	0	0
taco	0	0	0	1	0	0	0	1	0	0	2	2	0	1	0	0	1.	0	0	1	1	0	0	0	0
jambalaya	0	1	1	0	0	3	0	0	1	0	1	0	0	1	1	0	0	0.	0	1	0	0	0	0	0
pizza	0	0	1	0	0	2	0	0	0	1	0	0	0	1	0	1	0	1	2.	0	0	0	0	1	1
hotdog	0	2	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	1	0	1.	1	1	0	0	0
sushi	0	1	0	0	1	4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1.	0	2	0	0
croissant	1	2	1	1	0	0	0	0	0	0	2	0	0	0	1	0	0	1	0	0	0	0.	0	0	1
bread	0	1	0	1	1	1	0	0	0	1	0	0	0	1	0	0	0	0	1	1	1	0	0.	1	0
lasagna	0	1	0	1	0	4	0	0	1	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0.	0
tiramisu	1	0	0	0	0	1	0	0	0	0	1	1	0	1	2	0	0	0	0	0	1	0	0	0	2.

Classifier accuracy: 25 of 250 = 10% (versus random guessing accuracy of 4%)

The Bow Confusion Matrix :

Confusion matrix:

	ha	ch	ch	sc	mu	pu	po	sp	fr	pa	br	sa	ku	wa	ba	sa	ta	ja	pi	ho	su	cr	br	la	ti
hamburger	2.	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	1	0	0	3	0	0	0	0	0
churro	0	6.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	1
chickennugget	0	1	0.	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	1	0	2
scone	1	0	0	1.	0	0	2	2	1	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0
muffin	1	0	1	1	2.	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1
pudding	4	1	0	0	1	1.	0	0	0	0	0	0	0	0	2	0	0	0	0	1	0	0	0	0	0
popcorn	0	0	0	0	0	0	9.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
spaghetti	0	0	0	0	0	0	3	6.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
frenchfries	0	0	0	0	0	0	0	0	6.	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	1
paella	0	0	0	1	0	0	3	0	0	1.	0	1	0	0	0	0	0	0	0	0	0	0	0	2	2
brownie	1	0	0	0	0	0	0	0	0	0	0.	2	0	0	0	0	0	1	0	0	1	3	0	0	2
salad	0	0	0	0	1	0	0	1	0	1	0	2.	0	0	0	0	2	0	0	0	1	1	0	1	0
kungpaochicken	0	0	0	0	0	0	2	1	0	0	0	1	5.	0	0	0	0	0	0	0	0	0	0	0	1
waffle	0	1	0	1	0	0	0	1	0	1	0	1	0	1.	0	0	0	0	0	1	0	0	0	1	2
bagel	1	0	0	0	0	0	0	0	0	0	0	0	0	0	5.	0	1	0	0	0	0	0	0	3	0
salmon	1	1	0	0	1	0	3	1	0	0	0	0	0	0	0.	0	0	0	1	1	0	0	0	0	1
taco	0	1	0	2	1	0	1	0	0	1	0	2	0	0	0	0	1.	0	0	0	0	0	0	0	1
jambalaya	0	0	0	0	2	0	1	1	0	0	0	1	1	0	1	0	0	1.	0	0	0	0	0	1	1
pizza	0	0	0	0	0	0	1	0	0	1	0	1	2	0	0	1	0	0.	0	0	1	0	1	2	
hotdog	0	7	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1.	0	0	0	0	1	
sushi	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4.	0	1	0	1	
croissant	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0	1	0	2.	0	4	
bread	0	0	1	0	0	2	1	0	1	0	0	0	0	0	3	0	0	0	0	0	0	2.	0	0	
lasagna	0	0	0	0	0	1	1	0	0	1	3	0	0	0	0	0	0	0	0	1	1	0	2.	0	
tiramisu	0	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	5.	

Classifier accuracy: 65 of 250 = 26% (versus random guessing accuracy of 4%)

The Haar Confusion matrix:

Confusion matrix:

	ha	ch	ch	sc	mu	pu	po	sp	fr	pa	br	sa	ku	wa	ba	sa	ta	ja	pi	ho	su	cr	br	la	ti
hamburger	0.	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	3	1
churro	0	1.	0	0	0	1	0	0	0	0	0	0	0	1	0	0	2	0	0	0	1	1	1	1	1
chickennugget	0	0	2.	0	3	2	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0
scone	0	2	1	1.	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0	0	1
muffin	0	0	1	0	1.	1	0	0	0	0	1	0	0	0	1	0	1	1	0	0	0	1	0	1	1
pudding	0	0	0	0	0	4.	0	0	0	0	0	0	0	0	0	2	0	0	1	0	0	0	2	0	1
popcorn	0	0	0	0	0	0	3.	0	0	0	0	0	1	0	0	2	1	0	0	0	1	0	0	2	0
spaghetti	0	1	0	0	0	0	0	0.	0	0	0	0	0	0	0	1	2	1	0	0	1	0	0	4	0
frenchfries	0	2	0	0	0	0	0	0	0.	0	0	0	0	0	0	2	1	0	1	0	0	0	0	3	1
paella	0	2	0	0	0	0	0	0	0.	0	1	1	2	0	1	1	1	0	0	0	1	0	1	0	0
brownie	0	1	0	0	2	0	0	0	0	0	2.	0	2	1	0	0	0	0	0	0	0	0	0	2	0
salad	0	1	1	0	1	0	0	0	0	0	0	0.	1	0	0	1	1	0	1	0	0	2	0	1	0
kungpaochicken	0	0	1	0	0	0	0	0	0	0	2	1	1.	0	0	0	2	1	0	0	0	0	2	0	0
waffle	1	0	2	0	1	1	0	0	0	0	0	1	0	1.	0	1	2	0	0	0	0	0	0	0	0
bagel	1	1	1	0	0	1	0	0	0	0	0	0	0	0	2.	0	0	0	0	0	0	0	0	3	1
salmon	0	1	1	0	2	1	0	0	0	0	0	0	0	0	0	1.	0	1	0	0	0	0	2	1	0
taco	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0	1	1.	1	0	0	0	0	0	2	1
jambalaya	0	1	2	1	0	0	0	0	0	0	1	0	0	0	0	0	2	1.	0	0	1	0	0	1	0
pizza	0	2	0	0	0	0	0	0	0	1	0	0	0	0	0	3	0	0	0.	0	2	0	0	2	0
hotdog	0	1	1	0	0	1	0	0	0	0	0	0	1	0	0	0	3	1	0	0.	0	0	0	1	1
sushi	0	2	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0.	1	1	1	2
croissant	0	1	2	0	1	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0.	0	0	2
bread	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	1	3	1	2	0	0	0	0.	0	0
lasagna	0	4	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	1.	0
tiramisu	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0	0	1	1	0	1	0	0	0	0	3.

Classifier accuracy: 25 of 250 = 10% (versus random guessing accuracy of 4%)