CS364/AM792: Homework #5

Due on 20 May, 2020 at 17:00pm

Unathi K. Skosana

Problem 1

(a)

Following the prescribed procedure to construct the matrix X and its corresponding SVD, the singular values of X are plotted below.

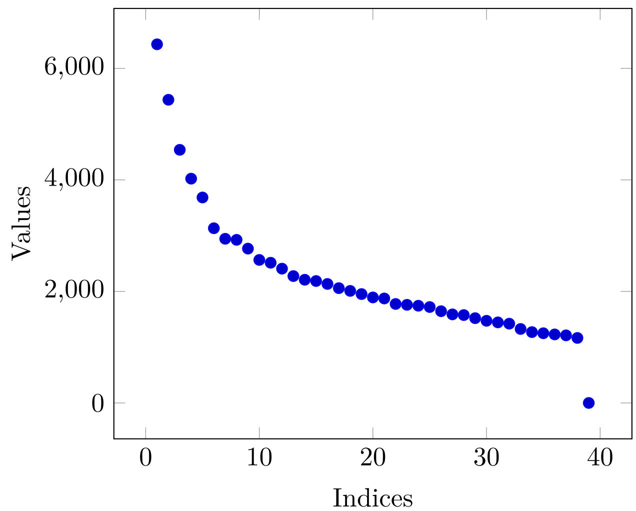
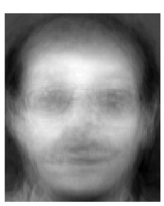
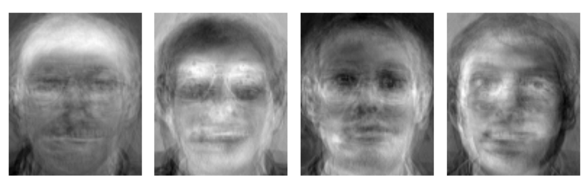


Figure 1: Singular values of X

From the plot above, it seems like > 2000 is a sensible threshold. This particular threshold gives an alpha value of $\alpha = 19$. Figure 2a and 2b show the average face $\mathbf{a}$ and the first five eigenfaces, rescaled and shifted back to grayscale.



(a) Average Face



(b) First Five Eigenfaces

(b)

The acquisition of U_α and $\mathbf{a}$ allows one to encode all the test images to their corresponding eigenface representations, $\mathbf{y}_i$. From this representation we can reconstruct a low dimension representation of a test image via

$$\widehat{\mathbf{f}}_i = \mathbf{a} + U_\alpha \mathbf{y}_i$$

Problem 1 continued on next page...

Figure 3 shows a sample of a few such low dimension representations and their originals, the image information seems to be captured well enough as some of the features of the original are recognizable.

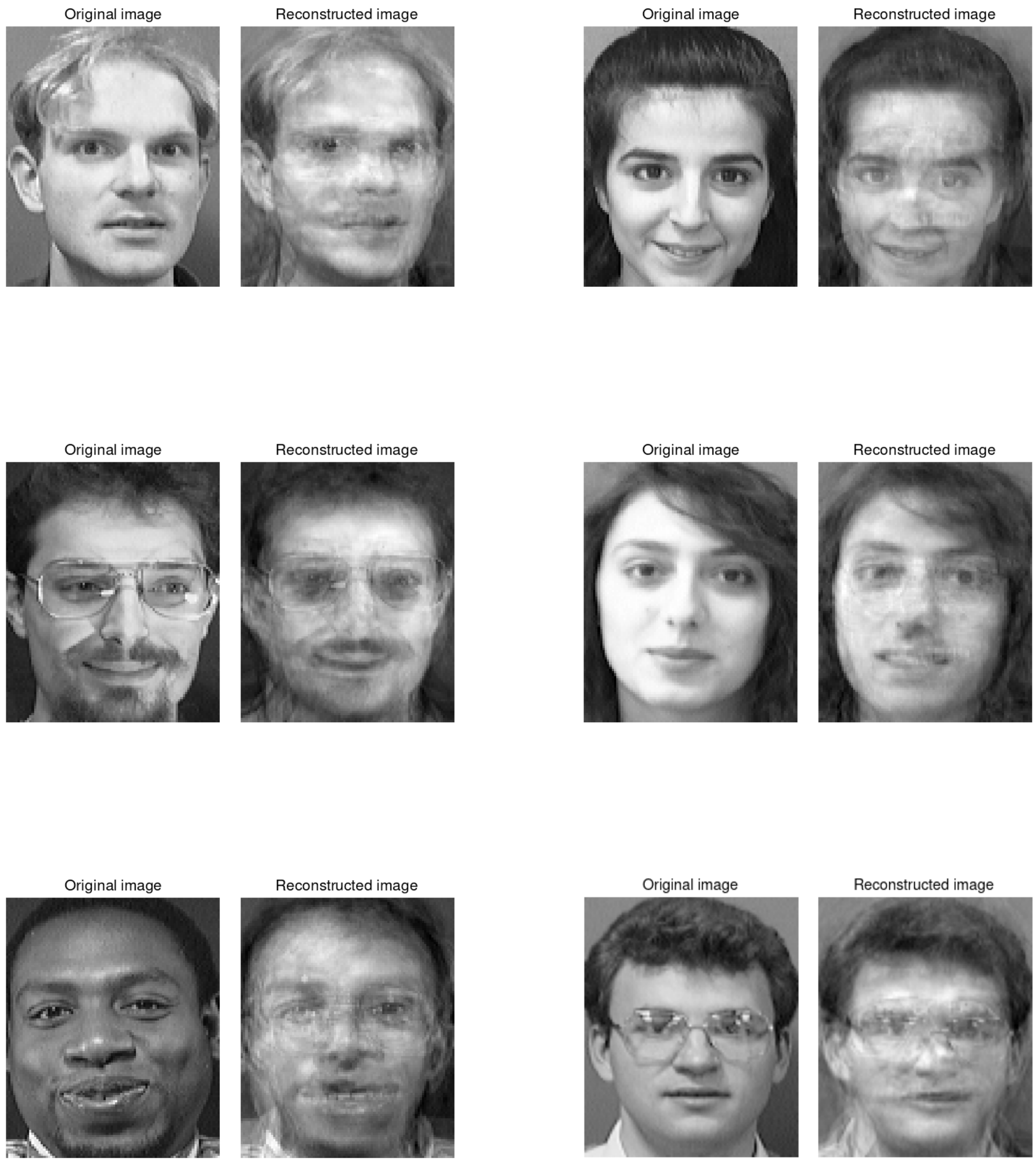


Figure 3: Originals and Low Dimensions Representations Of Test Images

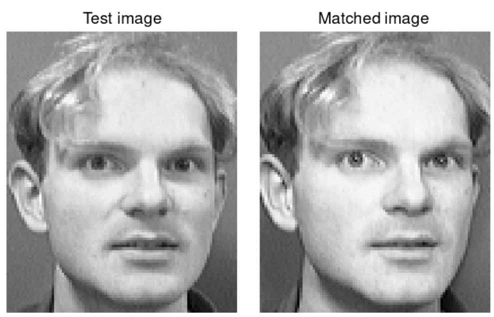
(c)

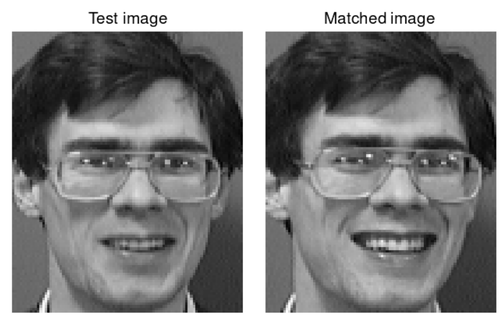
To accomplish classification, we first have to get the eigenface representations of the training and test set images, then for every image i in each test set, calculate

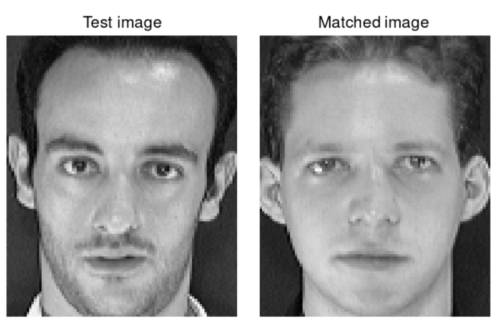
$$d_j = \|\mathbf{y}_i - \mathbf{y}'_j\|$$

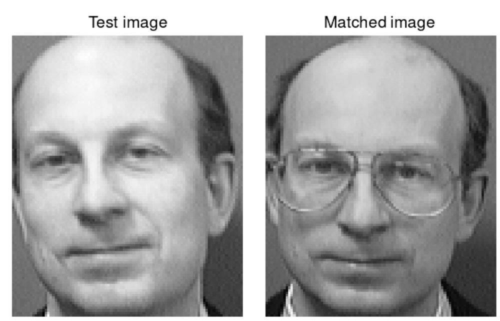
where $\mathbf{y}'_j$ is the eigenface representation of the j^{th} image in the training set.

The index i^* of the minimum value of $\mathbf{d}$ is the closest neighbor of the i^{th} test image. If $i = i^*$, then the image was correctly matched and incorrectly matched otherwise. Below are the test images with their corresponding matches as described.









Problem 1 continued on next page...

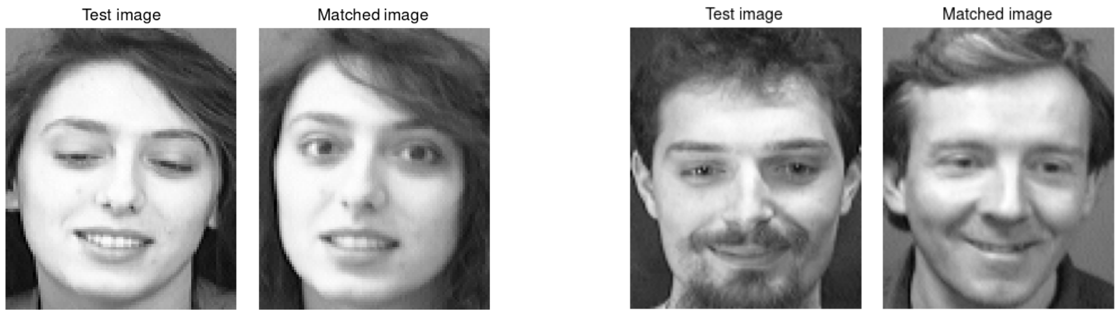


Figure 5: Test images And Their Closest Neighbors From The Training Set

The accuracy of the matching was roughly 78.75% and we can see some of the inaccuracies in the above sample.

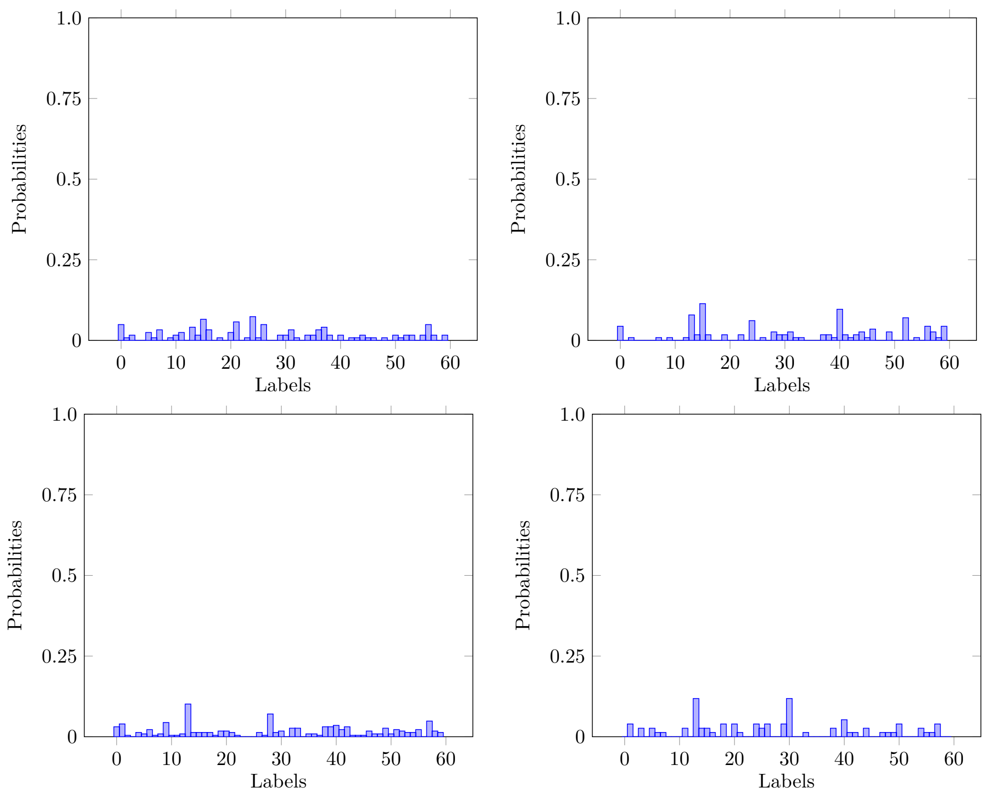
Problem 2

(a)

For this question, I used the SIFT features already provided in the accompanying material. Reading the dataset into python, I was able to arbitrarily assign labels to images according to what class they belonged to, i.e, Coast $\rightarrow 0$, Forest $\rightarrow 1$, Highway $\rightarrow 2$ etc. The class labels were assigned this sequence of numbers in ascending alphabetic order.

Collecting all descriptors of the training set into one giant array and feeding them to the `sklearn`'s `Kmeans` module for a fit with $k = 60$, giving a vocabulary of 60 words. From this vocabulary, we can run through image descriptors and use `kmeans.fit(descr)` to assign them to cluster center labels, thus the Bag Of Words representation of an image becomes a histogram of the cluster center labels the image's descriptors belong to.

Here are a few representative samples of histograms from the first four classes in the training set.



Problem 2 continued on next page...

(b)

Now from the bag of words representation of the training set, we assign to it labels according to what class each image belongs to, as described earlier. We can fit the training set histograms along side with their correct labels with `sklearn`'s `svm` module, initialized with

```
clf = svm.SVC(C=50.0, decision_function_shape='ovo', probability=False,
                    kernel='linear', cache_size=1000)
```

where C is the regularization parameter, here set to 50 (will elaborate later on the choice) and decision_function_shape specifies the classification approach for multi-class classification, in this case it is set to one versus one, and the kernel is specified as linear.

We can fit the training data with

```
clf.fit(flatten(train_bow_hist), flatten(train_labels))
```

where `train_bow_hist` is an array with shape

```
(num_classes=9, samples_in_each_class=100, number_of_vwords=60)
```

with `train_labels` having a similar shape, containing the labels for entries in `train_bow_hist`. The classifier can infer the number of classes by counting only unique elements in `flatten(train_labels))`.

Finally we can use the fitted classifier to classify the test data. The below table shows the classification report

class	precision	recall	f1-score	support
0	0.68	0.50	0.57	260
1	0.78	0.93	0.85	228
2	0.51	0.47	0.49	160
3	0.50	0.51	0.50	110
4	0.71	0.63	0.67	274
5	0.64	0.66	0.65	115
6	0.70	0.70	0.70	215
7	0.56	0.61	0.59	192
8	0.64	0.85	0.73	141
accuracy			0.65	1695
macro avg	0.64	0.65	0.64	1695
weighted avg	0.65	0.65	0.65	1695

Reporting an accuracy of $\sim 65\%$. The confusion matrix for this particular instance of the classifier is given by

$$M_{\mathcal{C}} = \begin{bmatrix} 129 & 15 & 44 & 2 & 31 & 2 & 3 & 26 & 8 \\ 0 & 213 & 0 & 0 & 7 & 0 & 2 & 2 & 4 \\ 33 & 0 & 75 & 0 & 9 & 1 & 7 & 22 & 13 \\ 1 & 0 & 1 & 56 & 1 & 24 & 7 & 12 & 8 \\ 26 & 30 & 12 & 0 & 172 & 0 & 6 & 14 & 14 \\ 0 & 0 & 2 & 27 & 2 & 76 & 4 & 1 & 3 \\ 0 & 9 & 4 & 12 & 8 & 10 & 150 & 15 & 7 \\ 2 & 4 & 7 & 7 & 11 & 3 & 30 & 118 & 10 \\ 0 & 1 & 3 & 8 & 2 & 2 & 4 & 1 & 120 \end{bmatrix}$$

Here are few samples of incorrectly classified test images

(a) Coast scene incorrectly classified as a highway

(b) Coast scene incorrectly classified as a suburb scene

(c) Forest scene incorrectly classified as a mountain scene

(d) Forest scene incorrectly classified as a street scene

(e) Highway scene incorrectly classified as a coast scene

(f) Highway scene incorrectly classified as a office scene

(g) Kitchen scene incorrectly classified as a office scene

(h) Kitchen scene incorrectly classified as a suburb scene

(i) Mountain scene incorrectly classified as a coast scene

(j) Mountain scene incorrectly classified as a suburb scene

Problem 2 continued on next page...

Here are few samples of correctly classified test images

(a) Office scene correctly identified

(b) Office scene office identified

(c) Store scene office identified

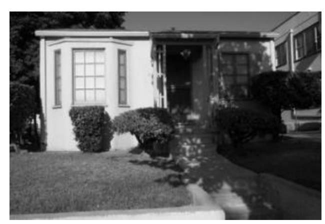
(d) Store scene office identified

(e) Street scene office identified

(f) Street scene office identified

(g) Suburb scene office identified

(h) Suburb scene office identified



Discusion:
On k-means clustering, I've noted that increasing k beyond $50-60$ doesn't marginally improve the end result (test accuracy), in fact there seems to be some point slightly above 60 where increasing k any further makes the result less accurate, thus it seems that that a locally optimal value of k lies just above 60. Training the 36 SVMs for classification while having a small $1.0 < C \leq 5.0$ (loose constraints or large margin) produced inaccurate results (36% – 40%). These results are improved when $5.0 \leq C \leq 50.00$ with a high of 65% test accuracy, similar to the k-means clustering scenario, increasing C any further becomes detrimental to the test accuracy. This is expected as this is a drawback of narrow margins fit too closely to the training set ; the classifier does not fare well when classifying new data.