Assignment 4: CS 215

Devansh Jain	Harshit Varma
190100044	190100055

November 13, 2020

Question 5

Instructions for running the code:

1. Unzip and `cd` to `q5/code`, under this find the file named `q5.m`
2. Run the file, required plots for each digit will be generated and saved to the `q5/results` folder
   `comp_x.jpg` - Comparing two images side-by-side.

We extract the images of handwritten digits from dataset `mnist.mat` and recast the training data into double and normalize it.
Normalization helps to reduce errors which otherwise occur due to high range of values in covariance matrix.
We also reshape the images into a vector for ease of computation.

```
DATA_PATH = "../data/mnist.mat";
load(DATA_PATH, "-mat"); % Load data
N = length(digits_train);
WIDTH = size(digits_train, 1);
SIZE = WIDTH^2;
% Reshape, Recast, Normalize image intensity
train_data = cast(reshape(digits_train, [SIZE N]), 'double')/255;
```

From PCA, we know that along the Principal modes of variation, variance is maximum. We are asked to form a 84-dimensional hyperplane within a 784-dimensional Euclidean space.
Therefore, we need to choose 84 orthonormal vectors to define such a hyperplane along with a shifted origin (which would be the sample mean).
From our knowledge of PCA, we can see that these 84 vectors would be none other than eigenvectors corresponding to the highest 84 eigenvalues obtained from Eigen decomposition of sample covariance.
The required function for computing the hyperplane is `reduction_basis` which returns the 84-dimensional basis and its origin for each digit.
Usage : `[Q_all, mean_all] = reduction_basis(train_data, labels_train, 84);`

```
function [Q_all, mean_all] = reduction_basis(data, labels, no_pmv)
    % no_pmv is the number of principal modes of variation
    Q_all = zeros(10, 784, no_pmv);
```

```
    mean_all = zeros(10, 784);
    for digit=0:9
        count = sum(labels==digit);
        digit_data = data(:, labels == digit);
        mean = sum(digit_data, 2)/count; % sample mean
        cov = (digit_data-mean)*(digit_data'-mean')/(count-1); % sample cov

        % Eigendecomposition of cov: eigs() is much much faster than eig()
        [Q, L] = eigs(cov, no_pmv);
        % Get "no_pmv" (84) largest eigenvalues and corresponding eigenvectors
        % Q represents "no_pmv" (84) columns representing a 84-dimensional basis

        Q_all(digit+1, :, :) = Q;
        mean_all(digit+1, :) = mean;
    end
end
```

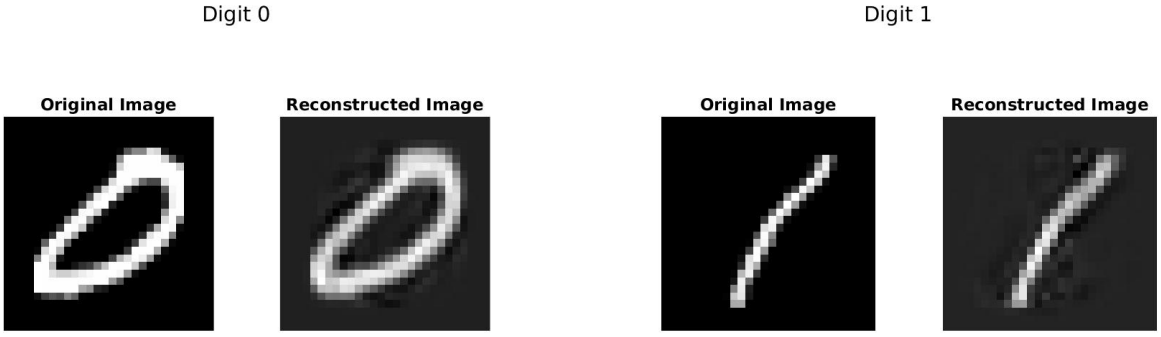
We could have used SVD algorithm instead of Eigen Decomposition to obtain the principal modes of variations and generate the hyperplane.

Once, we have obtained the 84-dimensional basis, we center the image and then compress it.
While regenerating, we uncompress using the 84-dimensional basis and then add the mean to it.
Remark: We need to first center the image about the sample mean and then reduced its dimensionality so as to variation of the data is about the mean.
The required function for compressing the image is `reduce_dim` which returns a vector containing the 84 coordinates to represent the image. Usage : `reduced_data = reduce_dim(test_data, Q, mean);`

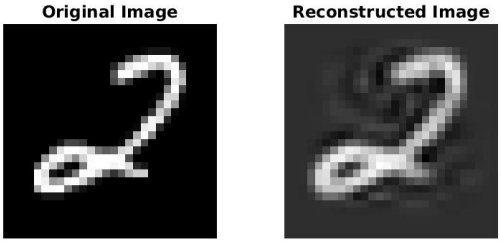
```
function [reduced_data] = reduce_dim(digit_img, Q, mean)
    reduced_data = Q'*(digit_img-mean);
end
```

The required function for regenerating the image is `reconstruct_img` which returns the reconstructed image as a vector. Usage : `reconstruct_img(reduced_data, Q, mean);`

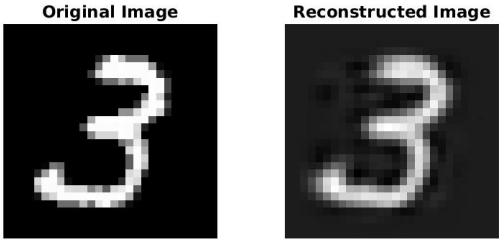
```
function [reduced_img] = reconstruct_img(reduced_data, Q, mean)
    reduced_img = mean + Q*reduced_data;
end
```



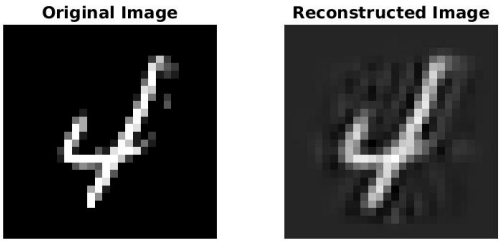
Digit 2



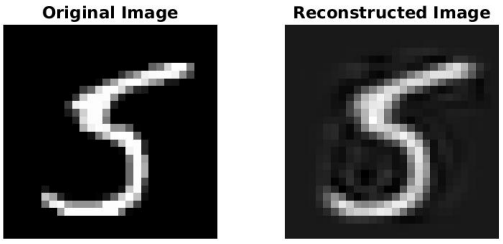
Digit 3



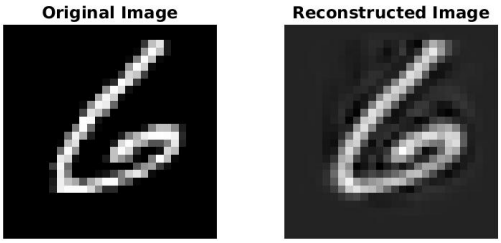
Digit 4



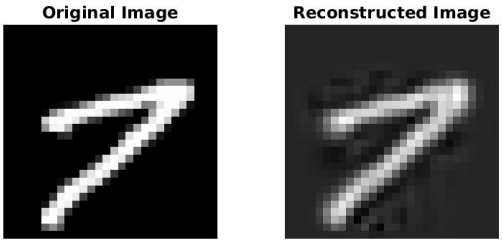
Digit 5



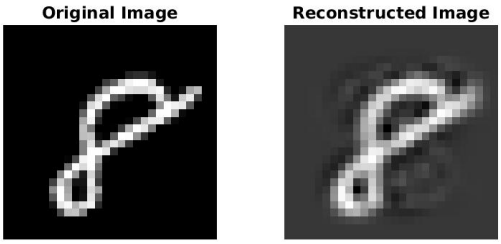
Digit 6



Digit 7



Digit 8



Digit 9