Problem 1. Apply (1) decision trees, (2) bagged and (3) boosted decision trees to the digit datasets from Homework 2. (You may use the standard libraries of Matlab or download Matlab code from the Web.) Use appropriate cross validation on the training set. Compare performance.

***Code:***

train = load('train79.mat');

train=train.d79;

test = load('test79.mat');

test=test.d79;

label = vertcat(ones(1000,1)\*1, ones(1000,1)\*-1);

%% Decision trees

DT = fitctree(train,label, 'CrossVal','on');

crossValFun = @(x)sum(x.IsBranch);

crossValResult = cellfun(crossValFun, DT.Trained);

figure;

histogram(crossValResult)

%% Bagged trees

BAT = fitcensemble(train,label,'Method','Bag','NumLearningCycles',500,'Kfold',8);

kFoldLossFunBaT=kfoldLoss(BAT,'mode','cumulative');

kFoldLossBaT = kFoldLossFunBaT(end)

figure(1)

plot(kFoldLossFunBaT,'r.');

xlabel('Learning Cycles');

ylabel('Loss Rate');

%% Boosted trees

BOT = fitcensemble(train,label,'Method','AdaBoostM1','NumLearningCycles',500,'Kfold',8);

kFoldLossFunBoT=kfoldLoss(BOT,'mode','cumulative');

kFoldLossBoT = kFoldLossFunBoT(end)

figure(2)

plot(kFoldLossFunBoT,'r.');

xlabel('Learning Cycles');

ylabel('Loss Rate');

***Analysis:***

For the cross validation for bagged tree and boosted tree,

The final error rate obtained by cross validation:

|  |  |
| --- | --- |
| Classifier | Cross validation error |
| Decision trees | 0.0460 |
| Bagged trees | 0.0245 |
| Boosted trees | 0.0160 |

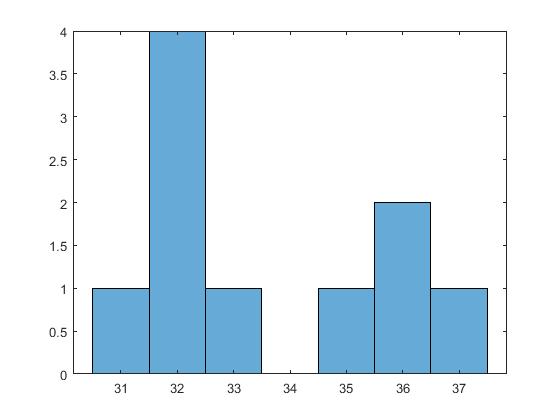


Figure 1 The histogram of isBranch

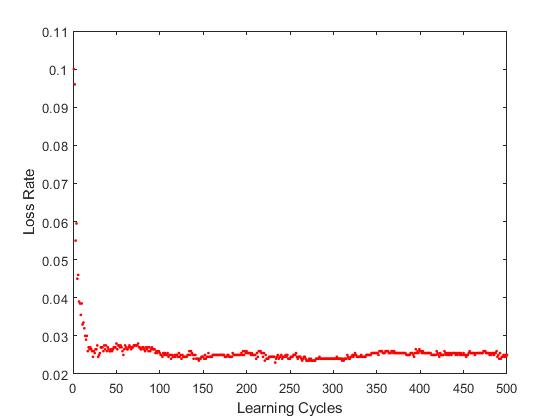


Figure 2 Bagged tree's loss rate with learning cycles

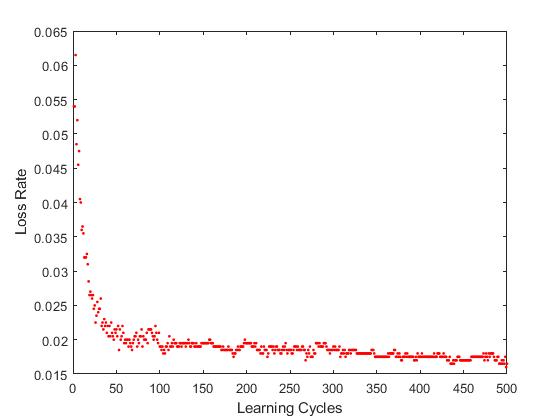


Figure 3 Boosted tree's loss rate with learning cycles

Problem 2.

1. Implement PCA and apply it to the digit data, reducing the dimension to two. Visualize the data after dimensionality reduction using colors for different classes.

***Code:***

test=load('test79.mat');

test=test.d79;

train=load('train79.mat');

train=train.d79;

label = vertcat(ones(1000,1)\*1, ones(1000,1)\*-1);

N=2000;

d=784;

k=2;

train\_7 = train (1:1000,:);

train\_9 = train (1001:2000, :);

[PCA, newTrain] = PCA\_eig(train, k)

%% Visualization

figure(1)

scatter(newTrain(1:1000,1),newTrain(1:1000,2),'.b');

hold on

scatter(newTrain(1001:2000,1),newTrain(1001:2000,2),'.r');

***Analysis:***

Figure 4 shows the distribution of data after PCA in a 2-D space. We can see two classes are somewhat separable: sevens are red dots which resides in the top, while nines are blue which resides in the bottom. However, we can also observe there is a huge mixed cluster in the middle. This is due to the massive information loss during the process of PCA.

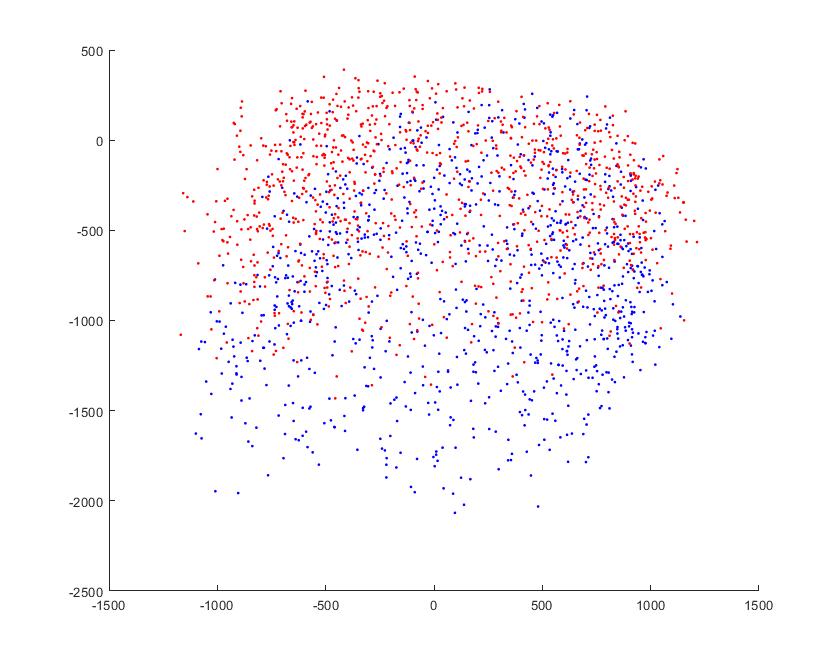


Figure 4 The visualization of reduced data after PCA in 2D space. (Red: seven; Blue: nine)

1. Produce pictures of “eigendigits” for the dataset combining both classes and for each class separately. Observations?

***Code:***

figure(2)

subplot(1,2,1)

x = reshape (PCA(:,1),28,28);

pcolor(x);

title('Eigendigit 1');

daspect([1 1 1])

subplot(1,2,2)

x = reshape (PCA(:,2),28,28);

pcolor(x);

title('Eigendigit 2');

daspect([1 1 1])

***Analysis:***

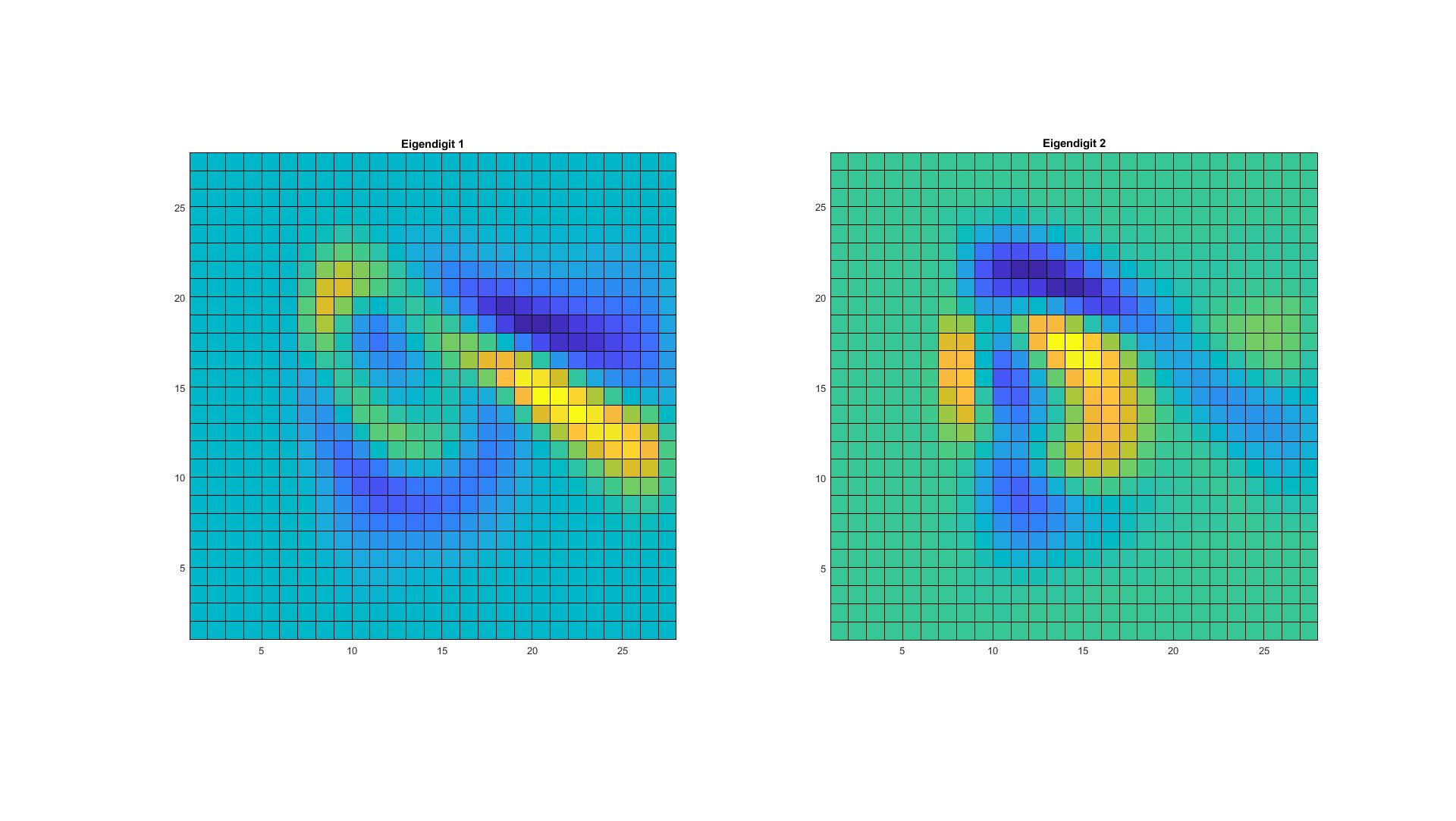


Figure 5 Eigendigits for dataset combining both classes.

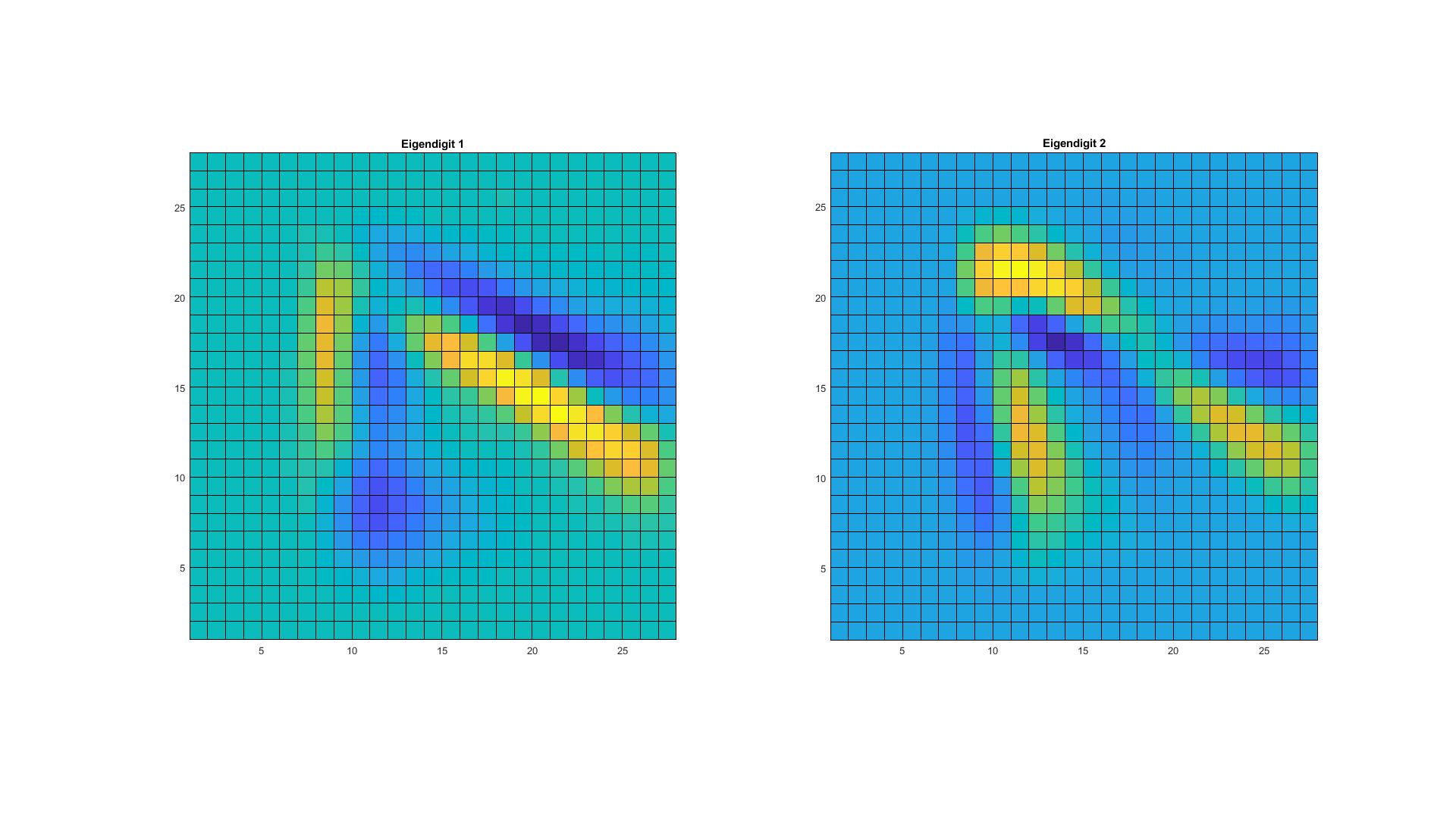


Figure 6 Eigendigits for dataset of sevens

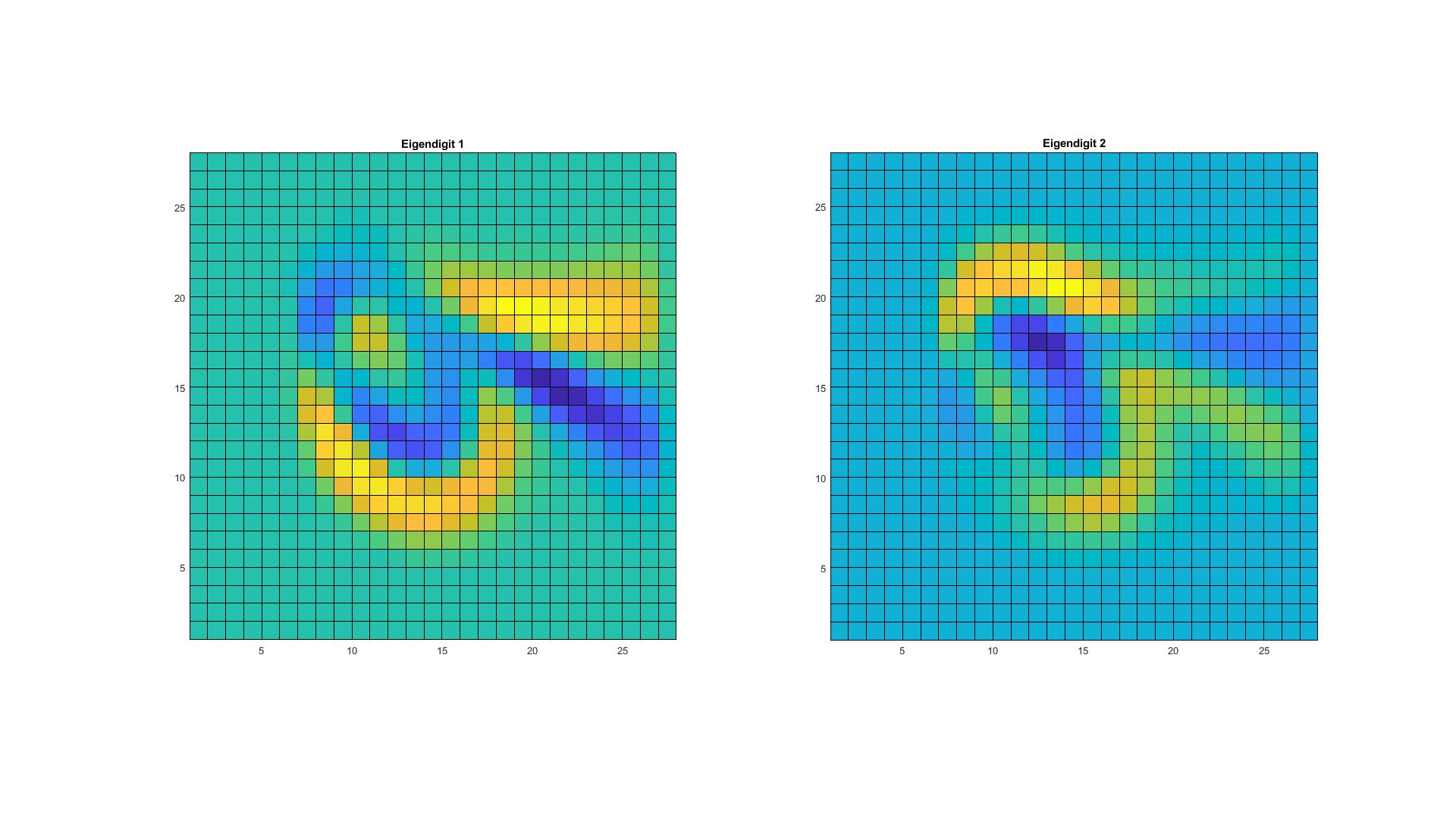


Figure 7 Eigendigits for dataset of nines

Problem 3. Apply k-means clustering to the digits data set for k = 2, 5, 10, 50. How well does it identify the different digits? (Note that clustering is unsupervised – how do you compare classification and clustering results?)