Syddansk Universitet

Data Preprocessing

Bjarki Sigurdsson
Abdulrahman Abdulrahim
Miguel de la Colina

March 21, 2017

Abstract

The purpose of the exercise was to examine the effects of preprocessing on kNN classification. We specifically tested Principal Component Analysis (PCA) and Min-Max Normalization. Other preprocessing methods used were those provided in the previous exercise by the `loadImage.R` script, namely image smoothing and DPI downsampling.

1 PCA

For this part of the exercise we used PCA to reduce the dimensionality of the data and observed the effect on kNN classification of using the first principal components resembling 80, 90, 95 and 99% of the total variance. We tested for classification accuracy and computation times for varying k.

1.1 Single-person data

Information on the first 10 principal components, in order, is shown in table 1. These correspond to the data from member 1 in group 5. This is how we determine the breakpoints for 80, 90, 95 and 99% of the total variance. They are 13, 23, 36 and 77, respectively. This is also illustrated in figure 1.

Table 1: The first 10 principal components.

PC#	Standard Deviation	Proportion of Variance	Cumulative PV
1	.7646	.1876	.1876
2	.6878	.1518	.3994
3	.6365	.1300	.4695
4	.4804	.0741	.5435
5	.4290	.0591	.6026
6	.3723	.0445	.6471
7	.3162	.0321	.6791
8	.2864	.0263	.7055
9	.2689	.0232	.7287
10	.2662	.0228	.7514

Figure 2 shows the results from setting k to 50, varying the number of principal components according to the breakpoints mentioned and measuring the average execution time of the kNN classification step. We see that there is a linear relationship between the two.

Figure 3 shows the classification accuracy for varying numbers of principal components and varying k. We see that the behaviour is similar for varying k and that the effect of using more principal components is negligible above 90% of total variance.

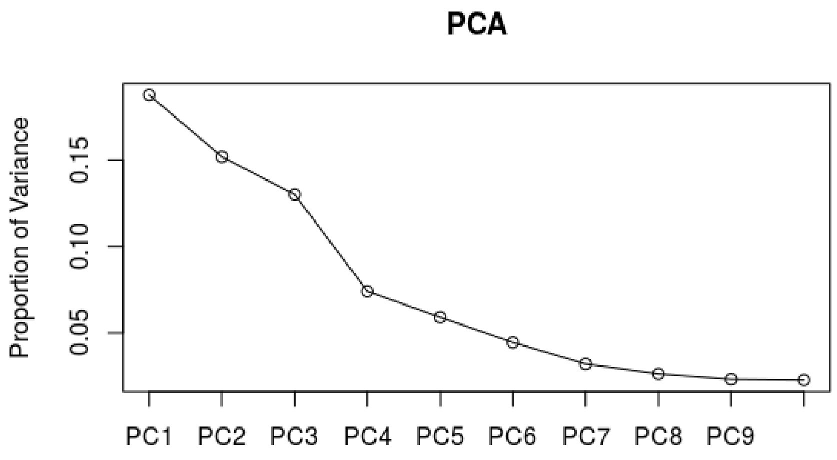


Figure 1: Scree-plot of the first 10 principal components.

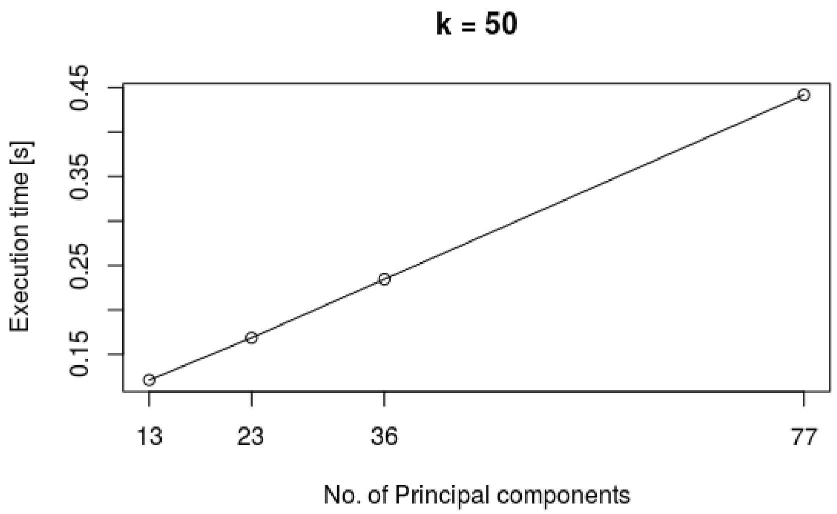


Figure 2: Timing results with a varying number of principal components.

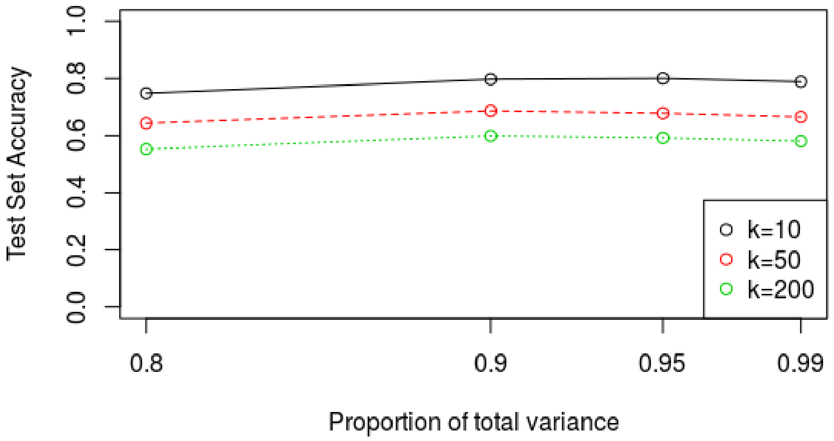


Figure 3: Classification accuracy results for varying k and Proportion of Variance.

1.2 MULTI-PERSON DATA

Information on the first 10 principal components, in order, is shown in table 2. In this case the breakpoints are 20, 31, 45 and 89 for 80, 90, 95 and 99%, respectively. This is illustrated in figure 4.

Table 2: The first 10 principal components.

PC#	Standard Deviation	Proportion of Variance	Cumulative PV
1	.8013	.1336	.1336
2	.7457	.1157	.2493
3	.5915	.0728	.3220
4	.5542	.0639	.3859
5	.5369	.0600	.4459
6	.4652	.0450	.4909
7	.4434	.0409	.5318
8	.4335	.0391	.5709
9	.3806	.0301	.6010
10	.3757	.0294	.6304

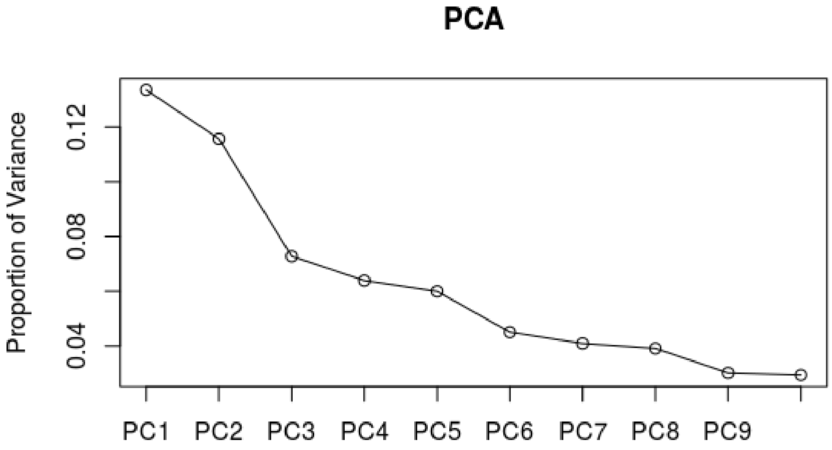


Figure 4: Scree-plot of the first 10 principal components.

Figure 5 shows the results from setting k to 50, varying the number of principal components according to the breakpoints mentioned and measuring the average execution time of the kNN classification step. Again we see that there is a linear relationship between the two. This time around the timings were less accurate due to time constraints and the size of the dataset.

Figure 6 shows the classification accuracy for varying numbers of principal components

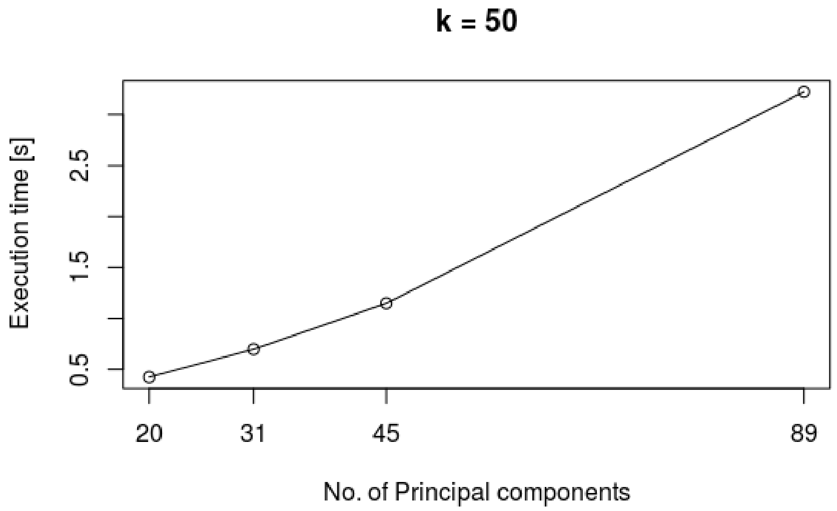


Figure 5: Timing results with a varying number of principal components.

and varying k. This is similar to what we saw for the single-person dataset.

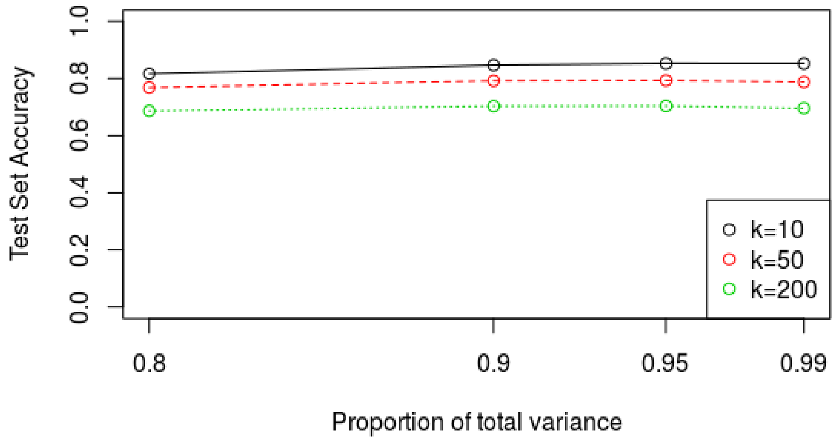


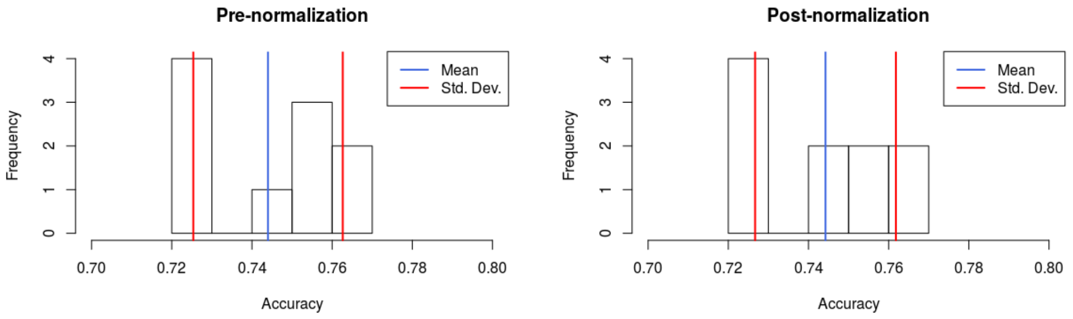
Figure 6: Classification accuracy results for varying k and Proportion of Variance.

2 Normalization

In this section we analyze the effects of applying Min-Max Normalization before the PCA step (pre-normalization) or after (post-normalization). To this end, we perform a cross-validation of the kNN classifier and compare the results.

2.1 Single-person data

Figure 7 shows the results of cross-validation on pre-normalized and post-normalized data. We see that the results are near-identical. This is logical as PCA depends on the eigenvalue decomposition of the covariance matrix and is therefore not significantly altered by normalization via robust methods.



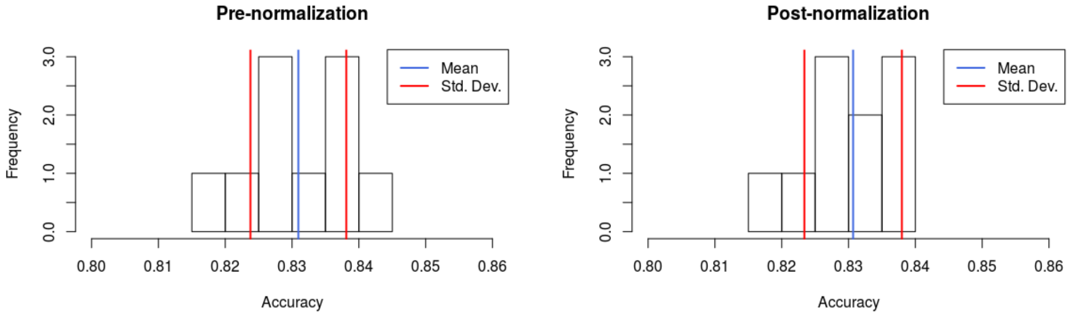
(a) Pre-normalization

(b) Post-normalization

Figure 7: Histogram of cross-validation results with k=50 using 36 principal components (95% of total variance).

2.2 Multi-person data

Figure 8 shows the results of cross-validation on pre-normalized and post-normalized data. Again the results are near-identical.



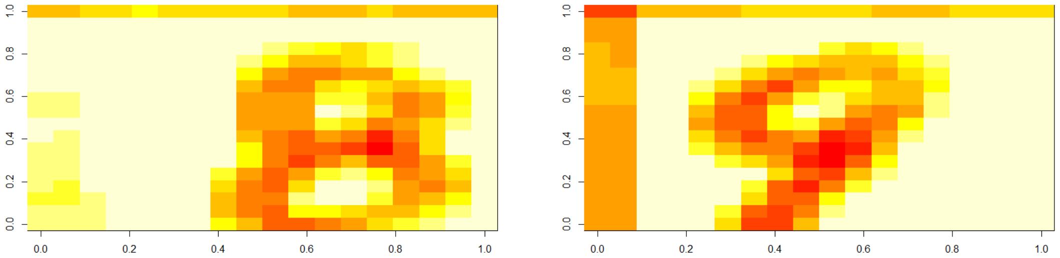
(a) Pre-normalization

(b) Post-normalization

Figure 8: Histogram of cross-validation results with k=50 using 45 principal components (95% of total variance).

3 Reconstruction

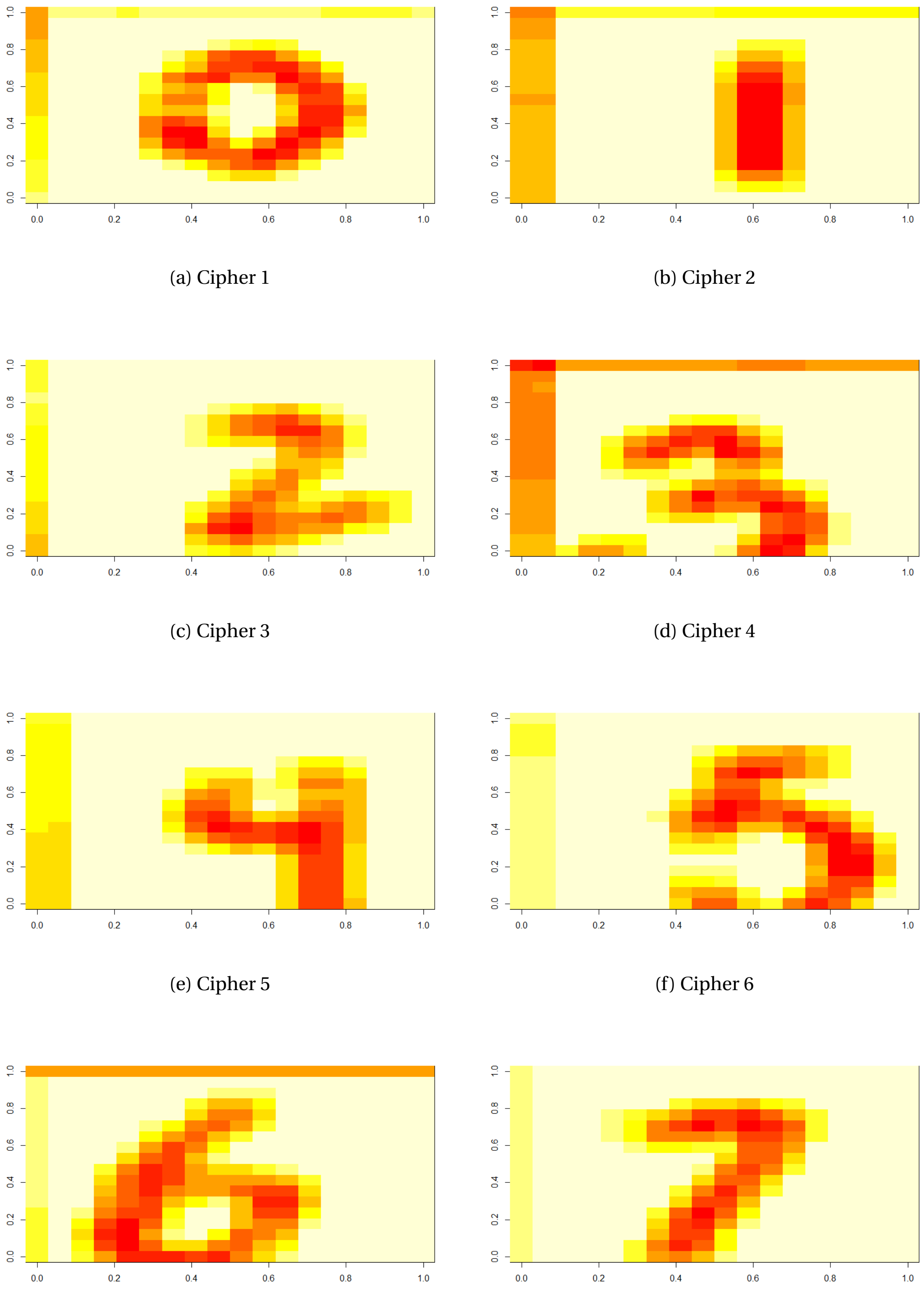
In this section we will be using the eigenvectors that we got from through PCA in order to reconstruct the images and compare them to the original ciphers. So first we plotted an image of each one of the original ciphers as we can see in figures 9 and 10.



(a) Cipher 9

(b) Cipher 10

Figure 9: Original ciphers.



(a) Cipher 1

(b) Cipher 2

(c) Cipher 3

(d) Cipher 4

(e) Cipher 5

(f) Cipher 6

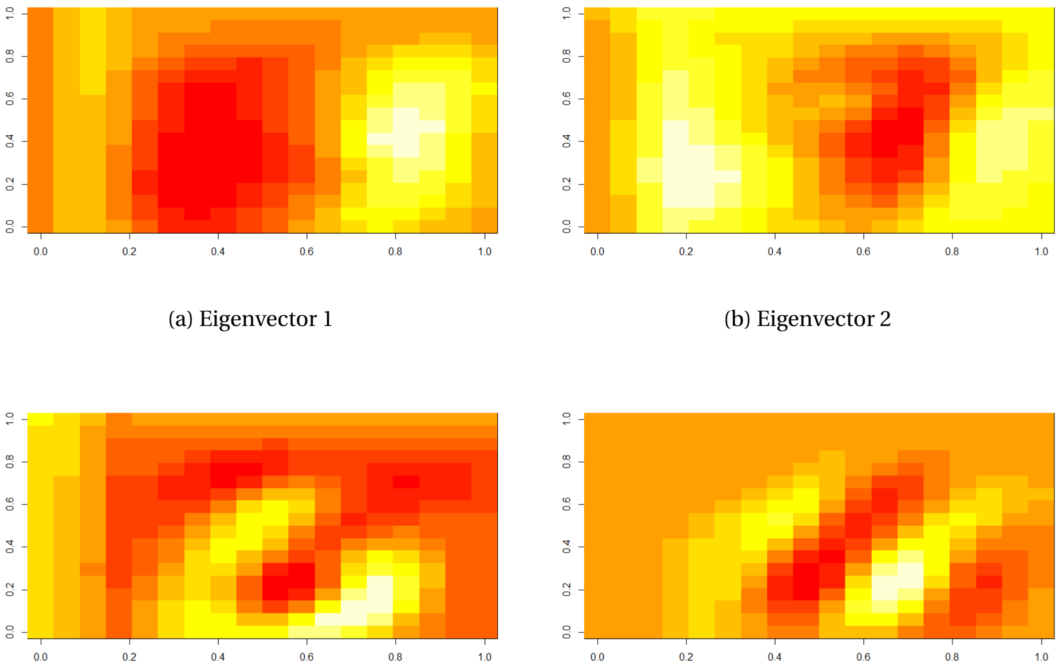
(g) Cipher 7

(h) Cipher 8

Figure 10: Original ciphers.

3.1 EIGENVECTORS

Then we plotted the first 10 eigenvectors that we got from running the pca algorithm through our original data set as we can see in figures 11 and 12. What we can see when we look at the eigenvectors is high pixel values in areas that have high variances



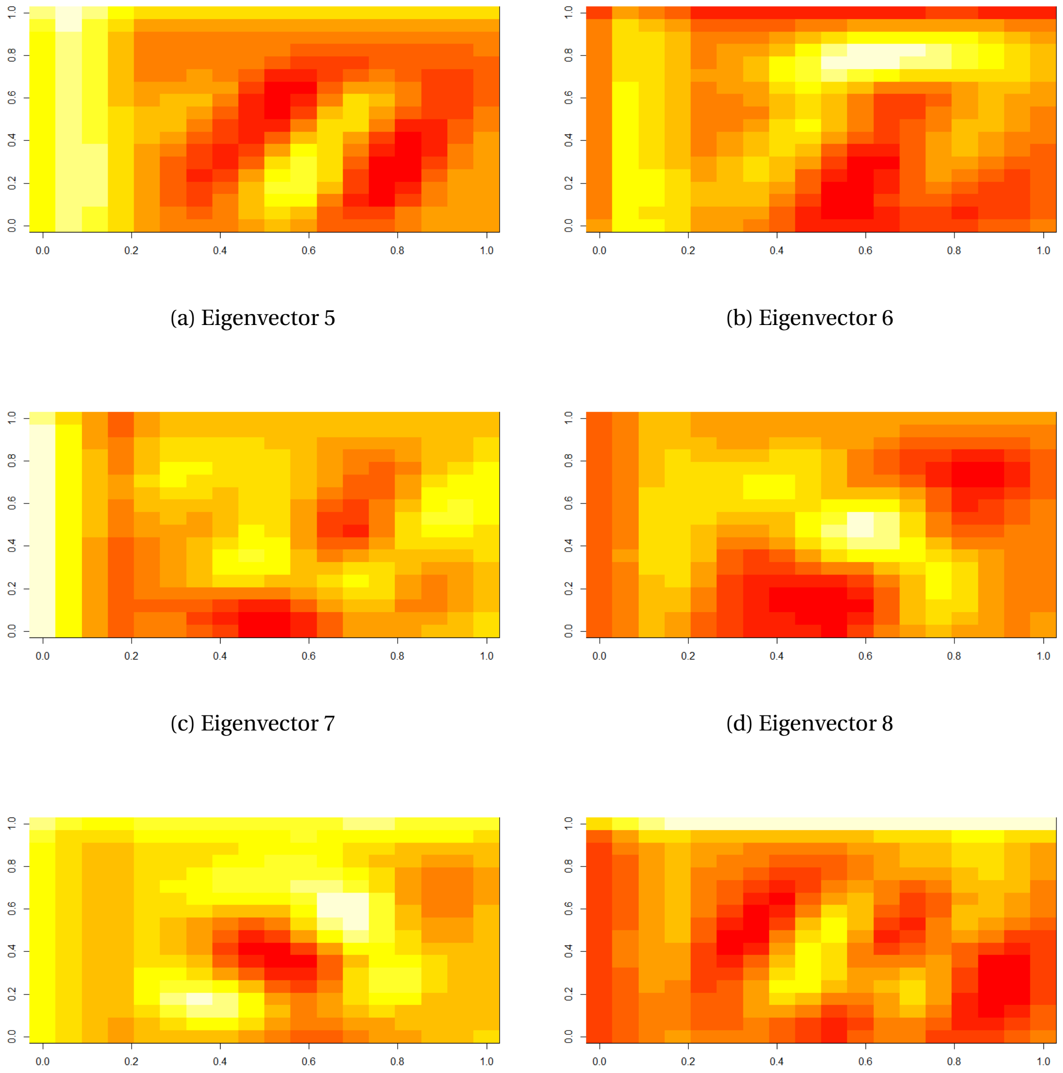
(a) Eigenvector 1

(b) Eigenvector 2

(c) Eigenvector 3

(d) Eigenvector 4

Figure 11: Eigenvectors



(a) Eigenvector 5

(b) Eigenvector 6

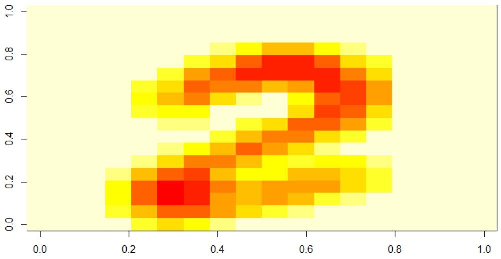
(c) Eigenvector 7

(d) Eigenvector 8

(e) Eigenvector 9

(f) Eigenvector 10

Figure 12: Eigenvectors.



(a) Reconstruction

Figure 13: Reconstruction with all PC's

3.2 Reconstruction with all PC's

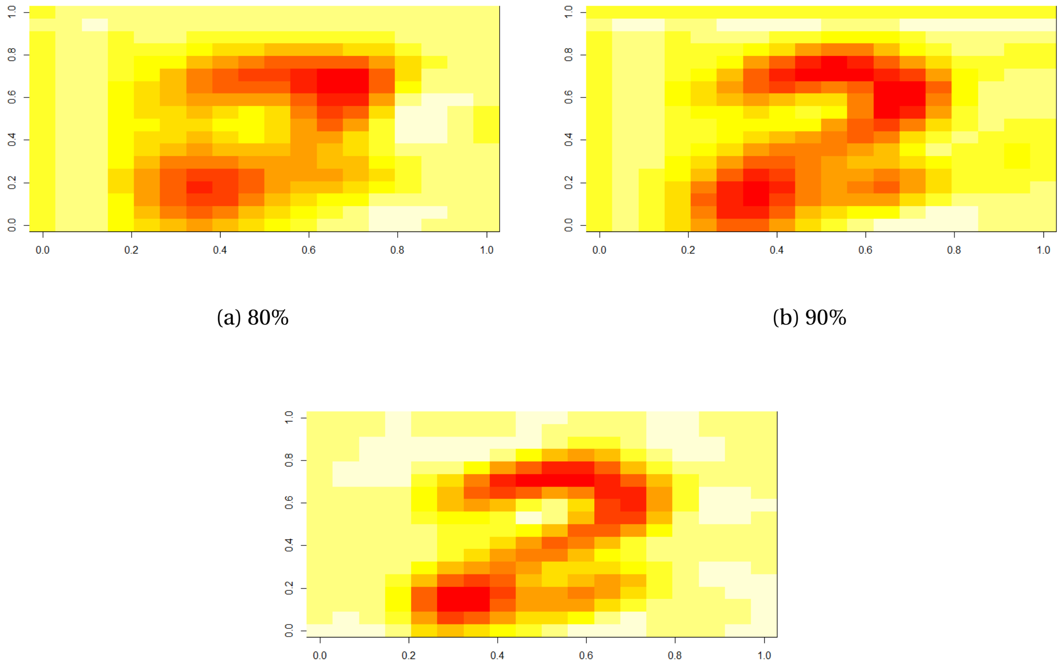
What we can see in figure 13 is a reconstruction of one Cipher based on the eigenvectors that were created when we used the PCA algorithm, this one is made using all the Principal Components which is why it really resembles the original ones.

3.3 Reconstruction with 80%, 90% and 95% of variance

As we know the first Principal components are the ones that have the highest variance which means that the first Principal component has the most variance and the second one has the second highest variance and so on so to achieve 80%, 90% and 95% we will only need to use a fraction of the percentages in this case 19, 30 and 43 respectively.

As we can see we still get something similar as the original cipher in all of the cases but we can also see that there is a huge improvement with higher variance and although 90% looks almost the same as the one that is using all the Principal Components it's still not as good and we get a better result when we use all of them.

PC	1	2	3	4	5	6	7	8	9	10
1	-0.0062	-0.0252	0.08513	-0.01152	0.1068	-0.03209	0.07201	-0.0396	0.0802	0.06937
2	-0.0069	-0.03585	0.0799	-0.01319	0.08456	0.00585	0.0911	-0.0448	0.0517	-0.02189



(a) 80%

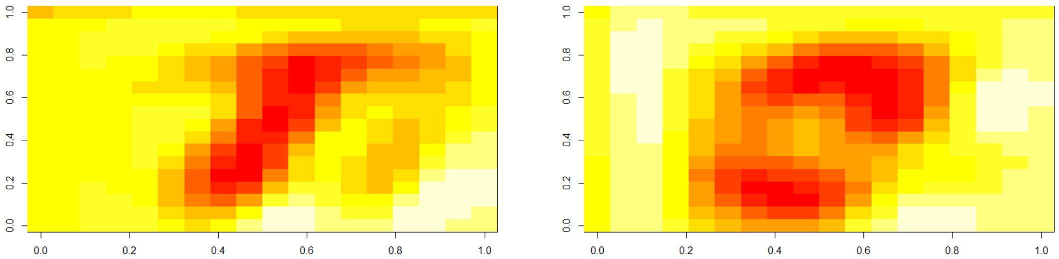
(b) 90%

(c) 95%

Figure 14: Reconstruction with different variances

3.4 Compare 10 first scores

In figure 15 we can see two different reconstructions of ciphers but only using it's first 10 scores and we can see that although it's hard to tell what original cipher they're supposed to be they still manage to look differently between them so you can see there is actually quite a difference. And as we can see in the table were we have the first 10 scores we see that although some of them are similar there are also some that have a huge difference.



(a) First Cipher

(b) Second Cipher

Figure 15: Comparison of first scores in two ciphers