Pattern Recognition practical 1

Maikel Withagen (s1867733) Steven Bosch (s1861948)

September 2015

1 Assignment 1

1.1

To compute the pair-wise correlation coefficients we used the following command:

Input

```
load('lab1_1.mat')
corrcoef(lab1_1)
```

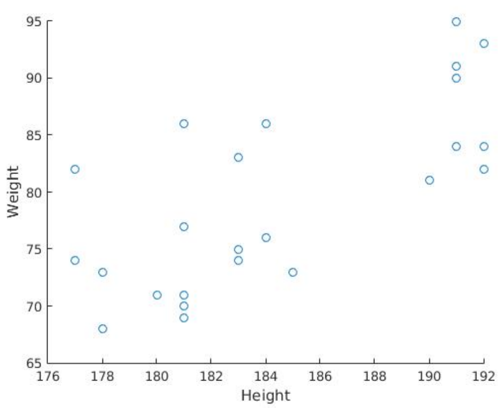
This yields us the following table of correlation coefficients:

Table 1: *Pair-wise correlation coefficients*

	Length	Age	Weight
Length	1	-0.0615	0.7156
Age	-0.615	1	0.5142
Weight	0.7156	0.5142	1

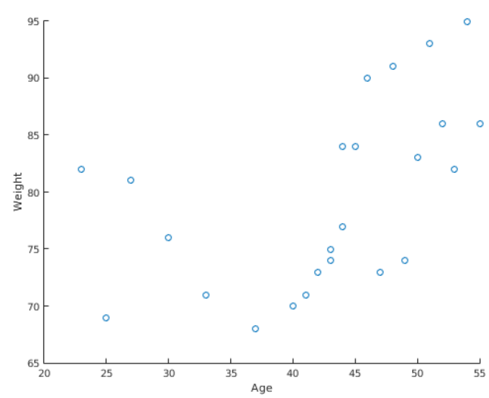
1.2

The two features for which the correlation is the largest are the first and third column, respectively the height and the weight.



(a) Scatterplot of weight to length

The two features for which the correlation is the second largest are the second and third column, respectively the height and the weight.



(b) Scatterplot of weight to age.

Figure 1

From a scatterplot alone it is hard to draw conclusions about any possible relationships between the different features. We do get indications though; figure 1a shows that there is likely to be a correlation between the weight and the height. An increase in weight seems to correspond to a (somewhat linear) increase in height. A similar kind of relationship can be seen in figure 1b, between the factors weight and age.

2 Assignment 2

2.1

The following subsections show the code we used to acquire the 1000 Hamming distances for set S and D.

a

To compute the set S we first create the set and fill it with zeros (line 1). Then for every element in the set we randomly pick a person and two random rows within that person and consequently load the actual content of these rows (lines 2-9). Then we compute the hamming distance between those two rows and store it within the appropriate place in the set (line 11). Finally the whole set is scaled so that every element in the set reflects the hamming distance between two elements instead of two rows.

Code for set S

```
1  hd_s = zeros(1,1000);
2  for i = 1:1000
3    person = randi([1,20]);
4    row1 = randi([1,20]);
5    row2 = row1;
6    while(row1 == row2)
7      row2 = randi([1,20]);
8    end
9    load(sprintf('person%02d.mat',person));
10
11   hd_s(i) = sum(abs(iriscode(row1,:)-iriscode(row2,:)));
12 end
13 hd_s = hd_s/30;
```

b

To compute the set D we do the same thing except for the fact that we compute the hamming distances between rows of two different persons:

Code for set D

```
hd_d = zeros(1,1000);
for i = 1:1000
  person1 = randi([1,20]);
  row1 = randi([1,20]);
  row2 = randi([1,20]);
  person2 = person1;
  while(person1 == person2)
    person2 = randi([1,20]);
  end
  load(sprintf('person%02d.mat',person1));
  x = iriscode(row1,:);
```

```
    load(sprintf('person%02d.mat',person2));
    y = iriscode(row2,:);
    hd_d(i) = sum(abs(x-y));
end
hd_d = hd_d/30;
```

2.2

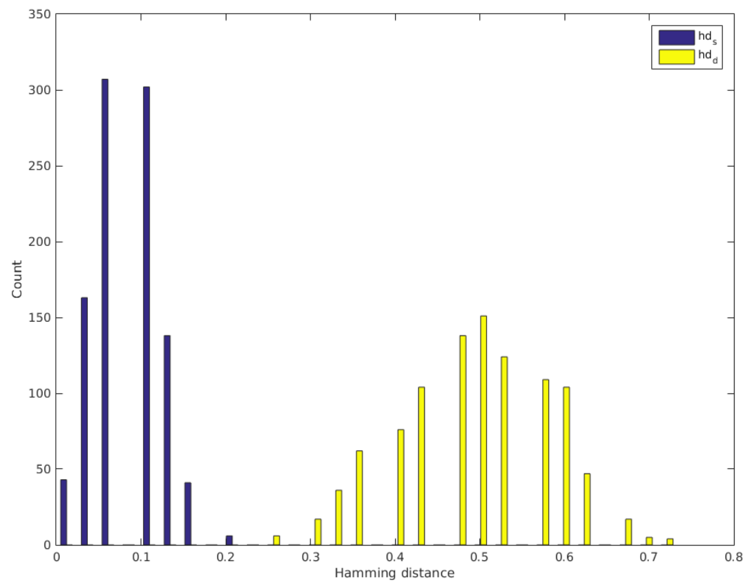


Figure 2: Histogram of sets S and D.

Figure 2 shows the histogram of sets S and D. The figure shows that the two distributions overlap very little, most of it around $hd = 5, 6$.

2.3

The means and variances of both of the sets are the following:

Set	Mean	Variance	Standard deviation
S	0.0825	0.0016	0.0398
D	0.4946	0.0079	0.0886

Table 2: Means and variances of both of the sets

The prior propability of two bits being the same between persons the following is: $1 - 0.4946 = 0.5054$. Using the formula $n = p(1-p)/\sigma^2$ we get $n = (0.4946 * (1 - 0.4946))/(0.0886^2) \approx 31.84$ statistically independent bits that are needed to encode an iris pattern. This means that the current number of bits in the iris vectors is insufficient.

2.4

We acquired the graph in figure 3 (see the appendix for the matlab code we used).

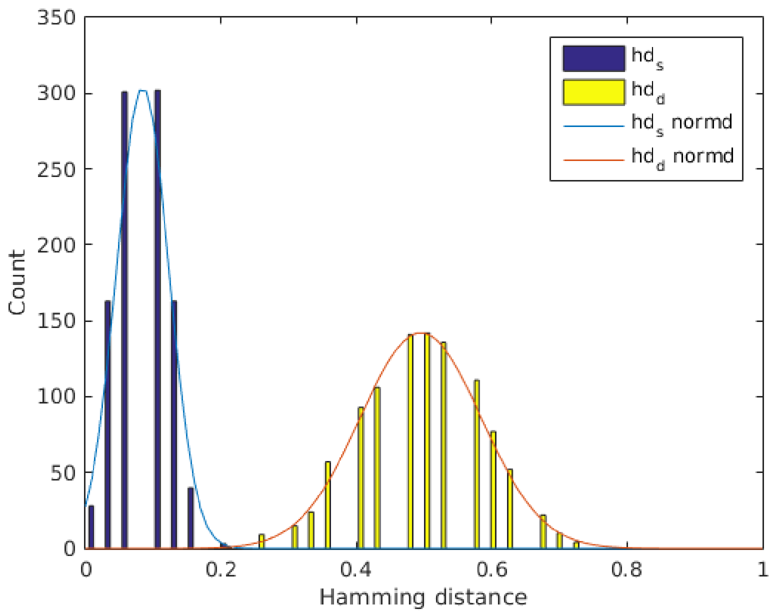


Figure 3: The histograms of sets S and D plotted with two Gaussian functions using the mean and variances.

Figure 3 shows the plot of the combined histograms and gaussian functions (note that the plot is slightly different from the previous subsections, because we reran the script in Matlab). The scaling is done by dividing the maximum value of the D and S sets by the maximum value of the initial Gaussian functions. This way the maximum value of the new Gaussian functions is the same as the maximum value of the D and S sets, so that the the Gaussians are nicely scaled to the histograms.

2.5

a

To estimate the value of the decision criterion we used the following code:

```
% Calculate the cumulative distribution function for the
% given x-value (range 0.000 - 1.000, with stepsize 0.001)
normc = normcdf([0:0.001:1],mean(hd_d),std(hd_d));
% Count the amount of x-values below the threshold value, minus 1 to correct
% from starting from 0.000, and divide by 1000 to get the decision criterion
d = (sum(normc(:) <= 0.0005)-1)/1000;
```

This yielded the value 0.185 as criterion. This means that below a hamming distance of 0.185, we will erroneously classify two irisses as being from the same person.

2.6

Using the criterion of 0.185, we get a false rejection rate of 0.00072, using the following code:

```
% Same as before, now for set S
normc = normcdf([0:0.001:1],mean(hd_s),std(hd_s));
% False rejection is integral of HD > d is 1 - normc(d)
1 - normc((d*1000)+1)
```

This means that up to 0.072% is falsely classified as being from two different persons.

2.7

Using the first part of the code in the appendix for assignment 2.6 we get a minimum hamming distance of 0, which we find in person 5, rows 1, 2, 3, 5, 6, 8, 10, 11, 12, 15, 17 and 18. Indeed, every known bit in testperson is the same as the bit in the same place in for example person 5, row 1:

It is clear that the the iris code of testperson most likely belongs to person 5. Now we use the second part of the code to calculate the average normalized hamming distance of the testperson to every row of person 5, which gives us a hamming distance of 0.0275.

Next we compute a set for person 5 just as we did for set S and set D. Finally we use this set and the HDmin value we got from the previous steps (0.0275) to calculate the integral below it, which gives us a significance value of $7.0872e - 07$. This means that we can most likely reject the null hypothesis that the persons are different.

3 Assignment 3

3.1 Background

The topic of biometric identification has been receiving a lot of attention lately. The use of fingerprints, facial features, and iris recognition are considered appropriate for identification. Of these three, fingerprints and iris recognition are considered to be more accurate than facial features, while facial featuresare more flexible, e.g. for the use in surveillance scenarios. A possible improvement for the performance of facial feature identification is the combination of information from multiple sources, for instance the ear. Ear images can be obtained in a similar manner to face images, and a recent study suggests that they are comparable in recognition power[3] and a combination of these sources gives a significant improvement over their individual performance.

The most common approach for using the ear as identification, is by using 2- or 3-dimensional images. These images can be compared by a number of methods e.g. PCA (Principal Component Analysis). A downside to these approaches is that it requires the images to be of a high quality, resulting in large amount of data (and thus needed processing power) as the need for high quality (expensive) cameras.

In this small report we will highlight two approaches that pose a different approach. The first study[2] poses a new technique that improves the robustness of ear recognition, while still using 2-dimensional images. The second study[1] proposes an entirely different approach where ear identification is based on the unique acoustic properties of ears.

3.2 Method

As said, identification by ear based on images usually happens by e.g. PCA, but unfortunately this method is very susceptible to misleading and occlusion (earrings and such). [2] make use of a large gallery of (identified) ear images to which they compare probe ear images. All the images of the gallery are analyzed for feature points with the SIFT[4] method. A to be identified image is analyzed in the same way and the gallery is searched for images with corresponding SIFT feature points. A temporary model is created from the matches in the database, and the transformation (rotation, angle, etc.) of the ear is calculated. By using this transformation data, more matches in the database are searched for, and the temporary model is updated until no more matches are found. The temporary model is then compared against the gallery images, and identification happens by selecting the identity of the most corresponding gallery image.

Akkermans and Schobben (2015) made use of the unique acoustic properties of the ear as a method of identification. Three different setup were tested; A headphone with microphones, Earphone pieces with microphones and a mobile telephone with an extra microphone. An explanation of the transformation of the received echo after sending in a tonal wave via a speaker, to to a feature vector goes beyond the scope of this report. An interesting point to mention is the fact that they only used low-cost equipment the experiment, thereby showing that their suggested method is able to perform without the need of high quality equipment with high costs.

3.3 Achieved Results

Achieved results are reported in percentages of correct identification, both false positives and false negatives fall are considered incorrect identification. The first box in Table 3 displays the results of manually selecting common ear images from the gallery and selecting the best fitting identity for the probe image, and the results of the suggested SIFT approach as described in Bustard (2008). The second box shows the results of the three setups as described in Akkermans and Schobben (2015).

3.4 Conclusion

It it shown that the proposed SIFT approach performs equally to the frequenctly used PCA approach combined with a manual selection of common images from the database. Therefore, the SIFT approach

Method	Recognition rate
Manual + PCA	96%
SIFT	96%
Headphone	98.6%
Earphone	98.1%
Mobile phone	92.8%

Table 3

seems to be an improvement over the status quo, due to the reduced needs for processing power and quality equipment. However, no mention was made about e.g. the time needed to perform the identification. To be able to fully compare the two methods, first all of their characteristics should be known. Making use of the acoustic properties of the ear radically differs from the other proposed techniques, but it shows promising results. However, the testing conditions were chosen to be as optimal as possible, and further research must show if this performance can also be achieved in less optimal circumstances.

In conclusion we can say that the attention in this field produces ever increasing results, and the use of the ear as identification can certainly be deemed possible in the future. However, more research is needed to further improve and test the currently known techniques.

References

[1] Anton HM Akkermans, Tom AM Kevenaar, and Daniel WE Schobben. Acoustic ear recognition for person identification. In *Automatic Identification Advanced Technologies, 2005. Fourth IEEE Workshop on*, pages 219–223. IEEE, 2005.

[2] John D Bustard and Mark S Nixon. Robust 2d ear registration and recognition based on sift point matching. In *Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on*, pages 1–6. IEEE, 2008.

[3] Kyong Chang, Kevin W Bowyer, Sudeep Sarkar, and Barnabas Victor. Comparison and combination of ear and face images in appearance-based biometrics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9):1160–1165, 2003.

[4] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. IEEE, 1999.

Appendix

Code for assignment 2.4:

```
hold off;

% Making the histogram
hd = [hd_s;hd_d].'
hist(hd,30); xlabel('Hamming distance'), ylabel('Count');
hold on;

% Create scaled Gaussian plots
x2 = sum(hd_s(:) == mode(hd_s));
normd = normpdf([0:0.01:1],mean(hd_s),std(hd_s));
plot([0:0.01:1],normd*(x2/max(normd)));

x1 = sum(hd_d(:) == mode(hd_d));
normd = normpdf([0:0.01:1],mean(hd_d),std(hd_d));
plot([0:0.01:1],normd*(x1/max(normd)));

legend('hd_s', 'hd_d', 'hd_s normd', 'hd_d normd');
```

Code for assignment 2.6:

```
% Load the testperson's iriscode
load(sprintf('testperson.mat'));
x = iriscode(1,:);

% Initialize variables
HDmin = 30; % HDmin starts at the maximum HD
person = 0; % The person with the minimum HD
row = 0; % The row within that person with HDmin

% Loop over all persons
for i = 1:20
    load(sprintf('person%02d.mat',i));
```

```
    % Loop over all rows within the persons
    for j = 1:20
        y = iriscode(j,:);
        HDtemp = 0; % HD for this specific row
        % Loop over every element, only add the HD to HDtemp if the element in
        % testperson is not 2 (since that bit is unknown)
        for k = 1:30
            if x(1,k) ~= 2
                HDtemp = HDtemp + abs(x(1,k)-y(1,k));
            end
        end
        % If the HD of the current row is the lowest, replace HDmin with
        % that and update the person and row
        if HDtemp <= HDmin
            HDmin = HDtemp;
            person = i;
            row = j;
            if HDmin == 0; % print the persons and rows with the lowest HDmin possible
                person
                row
            end
        end
    end
end
% Normalize HDmin to the amount of known bits in testperson (which is 20,
% because there are 10 unknown bits)
HDmin = HDmin/20;


% Now calculate the normalized HDmin for the comparison of testperson with
% every row in person 5.

HDtemp = 0;
load(sprintf('person05.mat'));

for j = 1:20
    y = iriscode(j,:);
    % Loop over every element, only add the HD to HDtemp if the element in
    % testperson is not 2 (since that bit is unknown)
    for k = 1:30
        if x(1,k) ~= 2
            HDtemp = HDtemp + abs(x(1,k)-y(1,k));
        end
    end
end
HDmin = HDtemp/400 % Normalize HDmin

% Calculate the set hd_5 just as with setS and setD

hd_5 = zeros(1,1000);
for i = 1:1000
```

```
    row1 = randi([1,20]);
    row2 = row1;
    while(row1 == row2)
        row2 = randi([1,20]);
    end
    load(sprintf('person05.mat',person));

    hd_5(i) = sum(abs(iriscode(row1,:)-iriscode(row2,:)));
end
hd_5 = hd_5/30;

% Calculate the integral of the tail of the distribution from the computed
% HDmin

normc = normcdf([0:0.001:1],mean(hd_5),std(hd_5));
1 - normc((HDmin*10000)+1)
```