# TNM098 - Lab 3

## Part 1: Image comparison

Twelve different images were given. The task was to compare the images based on different characteristics to find out what images were most similar.

### Method

The problem was solved using MATLAB. A feature vector was created with one characteristic in each dimension. The dimensions were:

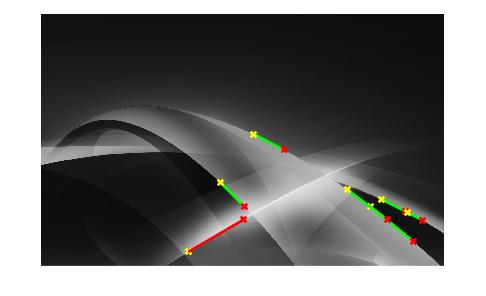
**1, 2, 3:** Mean color for each channel (R, G, B) in the whole image.

**4,5, 6:** Mean color for each channel (R, G, B) in a square of 200\*200 pixels in the center of the image.

**7:** Luminance distribution in a square of 200\*200 pixels in the center of the image.

**8, 9:** Angle and length for longest line.

**10, 11, 12, 13, 14, 15:** Angle and length for the second, third and fourth longest line.

The lines in the image was found using edge detection and Hough transform. First, edge detection method *Canny* was used with lower and upper thresholds as , which resulted in a binary file with the edges represented as ones and the rest as zeros. The edges were used in calculating the Hough transform, and from that finding the peaks and the lines in the images, see figure 1. The four longest lines in each image were saved, and the length and angle of the line compared to the x-axis were used as features in the feature vector. They were all weighted with a factor 3 for making the comparison less focused on color and luminance.

### Result

Two out of the three images with faces (1, 6, and 12) were calculated to be the most similar to another face. The last face was most like the earth (9), probably because luminance and color as well as line length and angles were very much alike. One of the two food images was matched to the other food. Two pairs of images were matched with each other, 7 and 10, and 9 and 12. The images that contained blueish landscapes (2, 7, 10) were match relatively good with each other. All matches to the images can be seen in table 1.

Figure 1 - Line detection using Hough transform.

Table 1 - All images and their calculated image to be most alike.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 6 | 10 | 5 | 8 | 11 | 12 | 10 | 11 | 12 | 7 | 4 | 9 |

## Part 2: Text comparison

Ten different text files were given. Two of the ten files contained content from some of the other files. For one text, one sentence was copied, and for another a whole paragraph was copied. The task was to find out which texts contained the plagiarized content.

### Method

The problem was solved using Python. First, the text in the files had to be cleaned by removing punctuation and making all letters to lowercase. Each sentence was put on a single line. From the cleaned texts tables were made by letting each sentence be the key and setting the value to 1 in the Python data type dictionary. Now the dictionaries could be used for checking plagiarized content.

For all dictionaries, every entry in them was checked against all entries in the other dictionaries. If a match was found and it was more than 20 characters long, it was registered as plagiarism.

### Result

The plagiarized material was taken from file 2 and could be found in file 6 and 8. The computational time on Intel Core i5, Windows 8.1 was 1.62682950241881 seconds.

#### **In both text nr 6 and nr 2:**

i tried out the scales and found that my involuntary host weighed over 195 pounds a good deal of it around the middle

#### **In both text nr 8 and nr 2:**

rutherford was pacing with surgical precision up and down my den

#### **In both text nr 8 and nr 2:**

he looked slightly more self possessed than the day before and seemed to be in excellent physical condition

#### **In both text nr 8 and nr 2:**

i guessed at the contour beneath my wadded black silk dressing gown and re considered my original plan to throw him bodily out of the house for having come without my invitation