Delft University of Technology

Computer Vision IN4393

REPORT

Final Project

Author:
Priadi Teguh Wibowo (4625862)
Marijn Otte (4222695)

Email:
PriadiTeguhWibowo@student.tudelft.nl
M.J.Otte@student.tudelft.nl

2 July 2017

Contents

1 Problem

The problem encountered in this project is emotion recognition based on human faces. From a video the (changing) emotion of the human visible in the video should be recognized. Emotion recognition can be used in a lot of applications. For example product testing can be made a lot easier by using emotion recognition. A new product can be given to a user, and by using emotion recognition it can be reduced whether the user likes the product. This method is very cheap, and also very reliable since the user cannot lie.

The approach to solve this problem will contain different steps:

1. Face Detection & Facial landmark detection
2. Feature extraction
3. Classifier training
4. Testing

2 Algorithms

In this section all algorithms used in the project will be explained, according to the steps explained in section 1.

2.1 Facial landmark detection

In order to detect the emotion of a human, first the face needs to be detected. It is also important to locate the coordinates from different parts of the head, like the mouth and the eyes. Facial landmark detection detects these different parts. The algorithm in this project detects in total 68 landmarks from mouth, brows, eyes, nose and jaw. The face detector is made using the Histogram of Oriented Gradients (HOG) feature, a linear classifier, an image pyramid and a sliding window detection scheme. An ensemble of regression trees is used to regress the location of the facial landmarks based on [1]. The coordinates of all landmarks of the testing set are saved in .txt files, so that facial landmark detection does not have to be executed every time the project is runned.

2.2 Feature extraction

Two different feature extraction methods have been tried: geometrical based feature extraction and texture based feature extraction. The implementation of geometrical based features is based on the eyes and the mouth, as proposed by Dileep et al. [4] The implementation of texture based feature will be based on Local Binary Pattern (LBP) histograms, as proposed by Ahonen et. al [5]. Meanwhile, the implementation of concatenated features (combination of geometric and texture based features) is based on Datta et. al [7].

2.2.1 Geometric Based Features

Using the coordinates of different parts of the face different geometrical features are calculated. They include the area of the mouth and the eyes as well as the perimeter of the mouth and the eyes.

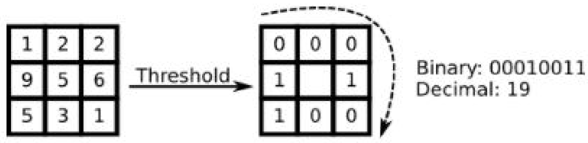

Figure 1: LBP Operator with 8 neighborhood

2.2.2 Texture Based Features

The LBP operator encodes for a pixel at position (x_c,y_c) with grayscale intensity i_c, i_p being the grayscale intensity of the the neighbor pixels and s(x) being 1 if $x \geq 0$ and 0 otherwise.

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(i_p - i_c)$$

The LBP operator that was used in paper [7] is a uniform pattern $LBP_{1,8}^{u2}$ which has radius = 1 and number of sampling points = 8. Uniform patterns are binary patterns that have at most 2 bit transitions when the 8-bit LBP code is interpreted as a circular string. For example, 11000010 is not a uniform pattern whereas 11110000 is (3 and 2 bit transitions respectively) [7]. With uniform patterns, the LBP histogram only has 59 histogram bins instead of 256. The 58 uniform binary patterns correspond to integer numbers, such as 0, 1, 2, 3, 4, 6, 7, 8, etc [8]. Meanwhile, all nonuniform patterns are assigned to the 59^{th} single bin.

To compute the texture based features, the face images is divided into number of cells or sub-images. Then $LBP_{1,8}^{u2}$ operator is implemented on each cells. Each LBP histogram from each cell is concatenated to form a feature vector.

2.2.3 Concatenated Features

Datta et al. [7] shows that using only geometric-based features is not good enough to distinguish all emotion classes. LBP histograms contain information about the distribution of micro-patterns such as edges, flat areas and wrinkles which represent some finer details of the face. By combining geometrical based and texture based features the expectation is that the performance of the classifier will be improved. The spatial uniform LBP histogram vector that contains the texture information and global description of the face is concatenated with geometric features as a single combined vector.This vector will contain all the features used by the classifier.

2.3 Classifier training

To classify the training data, a SVM classifier is used. 2-fold cross-validation is applied to test the classifier well. SVM is a often used linear classifier. There is no specific reason why this classifier is chosen. Future work can be done to analyze what is the best classifier for this project. The result of the classifier is saved into a datafile so classification of the trainingset only needs to be done once.

2.4 Distribution of algorithms

In this project some algorithms are taken over from existing libraries, while others are self written based on earlier research. For face detection and facial landmark detection dlib was used. [2]. Geometric feature extraction was done manually. To extracting the spatial LBP Histogram, the library from [6] is used. Meanwhile, the uniform pattern LBP is obtained manually. Concatenating the features is also done manually by combining both features. The SVM multiclass classifier from dlib [2] was used for classification.

3 Experiments & Results

The Cohn-Kanade dataset [3] was used in the experiments for training and testing the classifier. It contains 327 images of faces, with corresponding labels. The possible labels are: surprise, sadness, happiness, fear, disgust, neutral and anger. The goal of the experiments is to find the best combination of features. Experiments have been done on only using geometrical features, only texture based features and a combination of both. Columns 1 to 7 and rows 1 to 7 correspond to the emotions anger, neutral, disgust, fear, happiness, sadness and surprise respectively. Experiments based on 2-fold crossvalidation and based on live emotion recognition have been done.

3.1 Experiments based on cross-validation

To detect what combination of features performs best, 2-fold crossvalidation is applied. Different results of selecting different features can be found in tables 1 to 8.

3.1.1 Geometrical based features

In this section different combinations of geometrical based features are tested. The results can be found in tables 1 to 3.

26	0	6	0	2	0	10
4	0	1	0	2	0	11
5	0	20	0	27	0	6
0	0	2	0	5	0	17
0	0	2	0	52	0	14
15	0	3	0	1	0	9
1	0	1	0	16	0	64

Table 1: 2-fold Crossvalidation results based on mouth and eyes area

32	0	6	0	6	0	0
7	0	1	0	10	0	0
11	0	25	0	20	0	2
2	0	0	0	21	0	1
1	0	1	0	51	0	15
13	0	8	0	6	0	1
1	0	0	0	1	0	81

Table 2: 2-fold Crossvalidation results based on mouth area and perimeter

26	1	14	1	1	1	0
10	1	1	2	4	0	0
2	0	45	0	6	0	5
0	1	0	2	19	0	2
0	0	1	1	66	0	0
15	0	6	1	1	1	4
1	0	0	0	0	0	81

Table 3: 2-fold crossvalidation results based on mouth and eyes area and perimeter

As can be seen in tables 1 to 3, the classifier performs bad for classes 2, 4 and 6. This is because in the used dataset the amount of objects from classes 2, 4 and 6 is a lot smaller than the amount of objects from classes 1, 3, 5 and 7. It would be better to use a dataset in which each class has the same amount of objects.

It can be noticed that for the geometrical based features the combination of the following features performs best: moutharea, mouthperimeter, lefteyearea, lefteyeperimeter, righteyearea, righteyeperimeter. However, still some improvements can be made.

3.1.2 Texture based features

The results of experiment on texture based features are shown in tables 4 to 7. The experiment is conducted by changing the number of cells or sub-images of face images with constant radius 1 and sampling point 8.

10	1	12	2	5	8	6
1	12	1	2	0	1	1
3	0	47	1	3	2	2
3	2	2	4	8	2	3
0	0	1	0	67	0	0
10	1	5	0	0	9	3
2	1	2	4	2	0	71

Table 4: 2-fold crossvalidation results from 3x3 number of cells

7	3	12	5	2	11	4
1	11	1	2	1	1	1
1	0	42	0	7	2	6
5	2	4	3	5	2	3
0	0	0	0	67	0	1
10	0	1	1	0	13	3
0	1	2	3	1	1	74

Table 5: 2-fold crossvalidation results with 4x3 number of cells

21	0	8	1	2	5	7
1	11	0	3	1	1	0
3	0	48	1	7	1	2
4	0	2	3	5	1	5
0	0	0	0	67	0	1
8	1	3	1	0	10	5
0	1	0	2	1	0	78

Table 6: 2-fold crossvalidation results with 5x4 number of cells

23	1	7	2	2	5	4
1	12	0	3	1	1	0
6	0	46	0	3	0	3
1	2	2	4	9	2	4
0	0	0	1	66	0	1
10	1	1	1	0	10	5
0	1	1	2	0	0	78

Table 7: 2-fold crossvalidation results with 7x4 number of cells

As explained in the previous section that geometric-based features is poor in classifying classes 2, 4, and 6. By concatenating texture-based features with geometrical based features, the expectation is that the performance will increase. features. Hence, the LBP Histogram with 7x4 number of cells is chosen because it has the performance compared to other number of cells in classifying classes 2, 4 and 6, as shown by table 4, 5, 6 and 7.

3.1.3 Concatenated features

For concatenated features, six geometric features is combined with uniform pattern LBP histogram with 7x4 number of cells. The result of this combination is displayed by table 8.

33	0	7	2	2	2	0
4	12	0	1	1	1	0
6	0	48	0	3	1	2
0	0	1	9	14	0	0
1	1	0	0	66	0	0
12	1	2	1	0	10	2
0	1	0	0	0	0	81

Table 8: 2-fold crossvalidation results of concatenated features

As table 8 shows, classifying class 2, 4 and 6 is successfully increased compared to only using geometrically based features. Moreover, it also can improve geometric or texture based feature the classifier performance on classes 1 and 3.

3.2 Real-Life Testing

Real-Life testing was performed in order to test the classifier on other people using web camera. Real-life testing is performed on geometrical based features, texture based features and a combination of both. For each type of features the best combination was chosen according to sections 3.1.1, 3.1.2 and 3.1.3. Unfortunately it is not possible to express the results of real-life testing into numbers. It only possible to analyze different methods based on experience.

3.2.1 Geometrical based features

The classifier based on geometrical features performs well on the emotions anger, disgust and happy. Surprisingly the emotion ”surprise” was not detected since it performed really well when testing using cross-validation. No reason could be found why this emotion could not be detected.

3.2.2 Texture based features

The classifier based on texture features performs well on the emotions neutral, happy and surprise. These results underline the results from section 3.1.2.

3.2.3 Concatenated features

Since the geometrical and texture based features both perform well on different emotions, it is expected that concatenating the features of both approaches will increase the performance. Despite, the combination of features performs really bad. This is also surprisingly, since concatenating the features resulted in good results in section 3.1.3. No reason could be found why concatenating all features performs so bad.

3.3 Final system

Based on the results from sections 3.1 and 3.2 the decision is made to only use geometrical based features. Based on live tests it performed slightly better than texture based features. The final system can detect the emotions anger, disgust and happy well. Different improvements to the system that can be made will be discussed in section 4.

4 Conclusion & Discussion

In conclusion, several feature extraction algorithms are applied in our project, i.e. geometry-based features, texture-based features, and concatenated features in order to classify human emotion. Using 2-fold cross validation, it was found that geometry-based feature extraction is best in classifying anger, disgust, happy and surprise, texture-based feature extraction is better in classifying neutral, fear and sad emotion compared to geometry-based, and concatenated features can improve the classifier performance for all classes.

However, in live video using webcam, the SVM classifier with concatenated features performs quite bad. The classifier with texture-based features can distinguish neutral, happy and surprise and the classifier with geometric-based features perform slightly better and can distinguish anger, disgust, and happy emotion.

To improve the current system, several changes could be made. At first, a different dataset can be used. The currently used dataset has less objects in some classes (emotions) than in others. Because of this, classes containing less objects perform worse than classes containing more objects. A dataset in which each class has the same amount of objects would improve the performance.

Next, different feature based methods can be used. Currently texture- and geometrically based features are analyzed, but many different techniques exist to extract features from a face.

Also different classifiers can be analyzed. In this project svm is used. It could be that other (types) of classifiers would perform better.

Finally research can be done on concatenating the different features. The reason why the concatenation of geometrical- and texture based features performs so bad is unknown. By improving the technique of combining those features the system would probably perform a lot better.

References

[1] Kazemi, Vahid, and Josephine Sullivan. "One millisecond face alignment with an ensemble of regression trees." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.

[2] Dlib. http://www.dlib.net

[3] Cohn-Kanade database. http://www.consortium.ri.cmu.edu/ckagree/

[4] Dileep, M. R., and Ajit Danti. "Human Emotion classification based on Eyes and Mouth using Susan Edges."

[5] Ahonen, T., Hadid, A., and Pietikainen, M. Face Recognition with Local Binary Patterns. Computer Vision - ECCV 2004 (2004), 469–481.

[6] Philipp Wagner | November 08, 2011. (n.d.). Bytefish.de. Retrieved July 02, 2017, from http://www.bytefish.de/blog/local_binary_patterns/

[7] Datta, S., Sen, D., & Balasubramanian, R. (2016). Integrating Geometric and Textural Features for Facial Emotion Classification Using SVM Frameworks. Advances in Intelligent Systems and Computing Proceedings of International Conference on Computer Vision and Image Processing, 619-628. doi:10.1007/978-981-10-2104-6_55

[8] Local binary patterns. (2017, May 19). Retrieved July 02, 2017, from https://en.wikipedia.org/wiki/Local_binary_patterns

A Installation instructions

In the project OpenCV was used. Installation instructions can be found here:
`http://docs.opencv.org/2.4/doc/tutorials/introduction/windows_install/windows_install.html`.
Also dlib was used. Installation instructions can be found here:
`http://dlib.net/compile.html`
Finally Boost was used. Installation instructions can be found here:
`http://www.boost.org/doc/libs/1_55_0/doc/html/bbv2/installation.html`