

VNUHCM - University of Science  
Faculty of Information Technology  
Advanced Program in Computer Science

# CS412 - Report 1

## Face expression recognition (1)

December 25, 2016

### **Group members:**

1. 1351040 : Thai Thien
2. 1351059 : Ho Ngoc Huynh Mai

## **1 Introduction**

We choose to do Face expression recognition. Our implement will based on paper Robust Facial Expression Classification Using Shape and Appearance Features of SL Happy and Aurobinda Routray [1]. We adopt the pre-processing process, the feature extraction. But we do not intend to work on extract active patches. For the model, we plan to implement 2 model, a Support Vector Machine (as in [1]) and a simple Neural network.

## **2 Project Details**

The purpose of this project is implement the system to recognize facial expression, including, but not limited to anger, disgust, fear, happiness, sadness and surprise.

Input: A face of someone.

Output: The facial expression.

## 3 Methodology

### 3.1 Preprocessing

#### 3.1.1 Gaussian blur

By default, Opencv load image with BGR channel. We convert image to grayscale. Then apply Gaussian Blur with kernel size 5x5 and  $\sigma = 1$  to filter out noise.

#### 3.1.2 Face Detection

The purpose of this step is to detect the face in input image. We use image which is smoothed from the step above as input for face detection. We use Haar feature-based cascade classifiers [2] for face detection.

We use pre-trained data provided by OpenCV haarcascade\_frontalface\_default.xml.

#### 3.1.3 Extract and resize

We create the region of interest (ROI) from the face position detected by Haar classifiers. We create new image from region of interest and resize the image to 96x96.

#### 3.1.4 Result

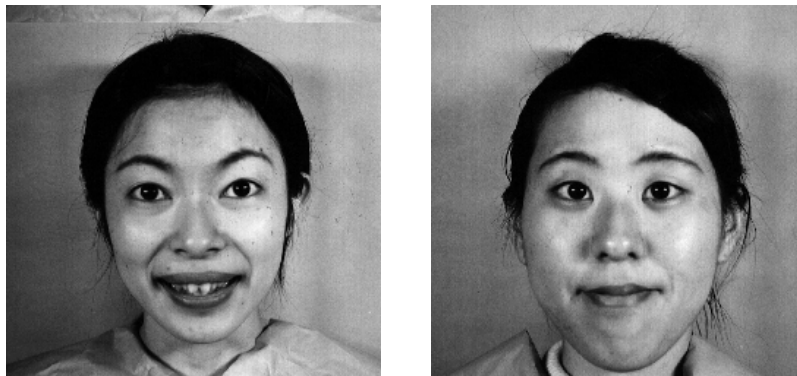


Figure 1: Before preprocessing

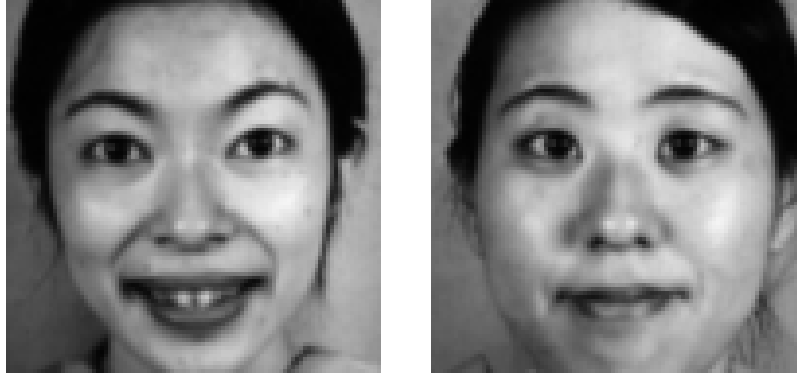


Figure 2: After preprocessing

### 3.2 Feature selection

This step we select feature to represent image.

Local Binary Pattern (lbp) is a type of texture descriptor[3].

Before construct lbp, the image must be convert into gray-scale. We already did that in 3.1

For each pixel in our image, we consider 8 neighbor pixels.

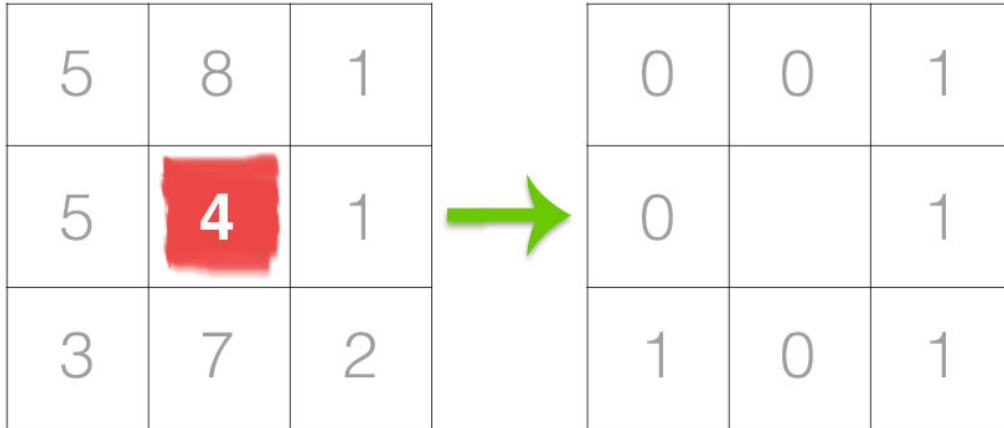


Figure 3: Local binary patterns (LBP)  
[4]

We compare the intensiy of the center pixel with its neighbor. Whenever the intensity of a neighbor pixel is greater or equal the middle pixel, we set to 1, else we set to 0. Then we take 8 bit represent of neighbor pixels and convert it into decimal representation. For example, in **figure 3**, we have  $0010111_2 = 23_{10}$ . Then we can store the values in another 2D array as in

**figure 4.** There are 8 neighbors pixels, therefore, there are  $2^8 = 256$  different pattern of lbp.

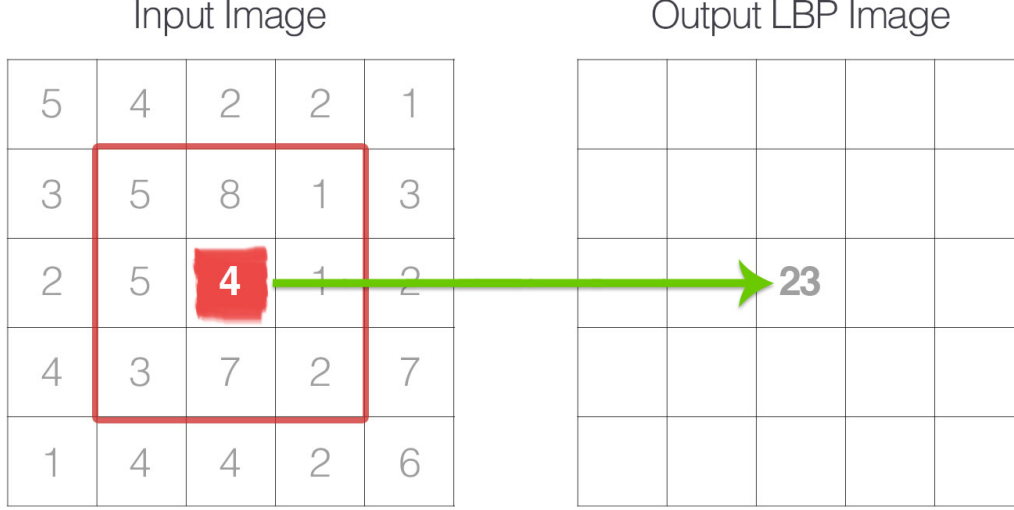


Figure 4: Local binary patterns (LBP)  
[4]

Finally, we calculate the histogram of each pattern occur. We treat this histogram as our feature vector.

For Local binary patterns implementation, we use scikit-image library [5]

### 3.3 The model

The model is a classifier model, which take feature vector as a input and output the class it belong to. There are 6 classes anger, disgust, fear, happiness, sadness and surprise.

For basic level, we implement a neural net, multi-layer perceptrons as the tutorial [6].

Then, we implement the One-Against-One Support Vector Machine. We need total 15 OAO SVM for 6 classes.

## 4 Data, Scenarios, and Models

Dataset: Cohn-Kanade (CK and CK+) [7], jaffe [8]

## References

- [1] S. L. Happy and A. Routray, “Robust facial expression classification using shape and appearance features,” in *Advances in Pattern Recognition (ICAPR), 2015 Eighth International Conference on*, pp. 1–5, Jan 2015.
- [2] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. I–511, IEEE, 2001.
- [3] M. Pietikäinen, “Local binary patterns,” *Scholarpedia*, vol. 5, no. 3, p. 9775, 2010.
- [4] “Local binary patterns with python & opencv.” <http://www.pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/>. Accessed: 2016-10-30.
- [5] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, “scikit-image: image processing in python,” *PeerJ*, vol. 2, p. e453, 2014.
- [6] “Neural networks.” [http://docs.opencv.org/2.4/modules/ml/doc/neural\\_networks.html](http://docs.opencv.org/2.4/modules/ml/doc/neural_networks.html). Accessed: 2016-10-30.
- [7] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, “The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pp. 94–101, June 2010.
- [8] M. J. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, and J. Budynek, “The japanese female facial expression (jaffe) database,” 1998.