Emotion Detection Using OpenCV and Geometric Facial Features for Automatic Facial Recognition

Jonathan Hicks Earlham College
801 National Road West
Richmond, Indiana
jghicks13@earlha.edu

1. INTRODUCTION

Currently there is a vast amount of research focusing on facial recognition and its capabilities. However, there is not as much focus on using facial recognition for emotion detection. Using source code from OpenCV and by creating a database of images portraying different emotions, I will develop an application that determines human emotion in an image. An application such as this could have many benefits, including interrogations of criminals and witnesses, or an interactive software that helps children with autism detect emotion.

2. BACKGROUND

For this application, my work will focus entirely on emotion detection and will be an extension of current research. The first step in the process is to create a database. This database will contain images of people accurately portraying different human emotion and people pretending to portray the emotions. I am choosing to include both accurate representations of human emotions and feigned emotions because this will allow the application to clearly determine whether or not the emotion of the scanned face is genuine or false[1]. Once the database has been created, I can then start to apply my research. I have to make sure though that I am looking out for potential problems like an obscured face, bad lighting, or non-frontal face[2]. As people, we are able to overlook these problems but it is significantly more difficult for a machine.

In starting my research I will need to first implement the source code from OpenCV. OpenCV is a computer vision library that was created in 1999. It has since become very popular for facial recognition. As mentioned earlier, the plan is to directly take the open source code for facial recognition that is on their site. OpenCV also has a few databases that I could also directly use or at least use them as a reference for my own. The next step is to find the facial points by breaking down each facial feature into separate tasks and using geometric features. Using geometric features will allow me to easily identify the shape and location of the current feature I am focusing on[3].

SAC'15 April 13-17, 2015, Salamanca, Spain.

3. PROPOSAL

In order to reach the end goal for my research, I am proposing an emotion detection software that contains the following components. The first component is the database that was discussed previously in the background section. This database will contain images of people portraying both real emotion and feigned emotion. Although this is not technically part of the software, it is still of vital importance. The next component is the source code from OpenCV. This source code most important part of the software as it should be able to scan an image and determine whether or not a face is present. Because the time frame of this research is only a matter of a few months, I plan to use this source code directly instead of trying to develop my own facial recognition code. Doing so allows me to focus more on emotion detection itself and makes the end result more achievable.

For emotion detection I plan to use a similar approach that Majumder et al. used by breaking down the face into different sections and using geometric facial features to find the shape and location of each facial point.

4. DESIGN

4.1 Eyes

The eyes are an extremely important facial feature as they help align the face and allow for a good reference point to other facial features. To detect the eyes I plan to use a thresholding algorithm that takes in input from every pixel from the eye region[3]. This algorithm is represented as
$T_{local}(x,y) = \mu_{global}(x,y) + k * \sigma_{local}(x,y)$
$\mu_{global}(x,y) = (1/M*N)[\sum_{j=0}^{N}\sum_{i=0}^{M} f(i,j)]$
where $T_{local}(x,y)$ is the threshold value of the pixel at (x,y) and $\sigma_{local}(x,y)$ is the standard deviation. μ_{local} is the local mean and μ_{global} is the global mean and $M*N$ is the size of the eye.

4.2 Eyebrows

The location of the eyebrow is first found using facial geometry. Once located, the facial points are stored in a vector. The positioning of these facial points will aid in determining the emotion.

4.3 Nose

As the nose lies between the eyes and represents the approximate center of the face, the rough estimation of the nose should be simple. The thresholding algorithm from earlier can now be applied to the nose as well as a contouring method to find the two nostrils.

4.4 Lips

Majumder et al.[3] provide a step-by-step algorithm for estimating the location of the lips. The first step is to find the center of the eyes and the height of the nose. From there they form a rectangle around the projected region of the mouth by using the function $rect(x_I, y_I, h_I, w_I)$ where x_I and y_I are the coordinates of the top left corner, h_I is the height and w_I is the width.

5. TIMELINE

Starting this week I plan to begin working with the source code from OpenCV. The goal by the end of the next couple of weeks is to get that source code working properly. I anticipate this will take at least a few days of focus and debugging.

Once the source code is implemented and working then I have to start working on the database. I will start off by using the databases provided on the OpenCV website to make sure that everything is working as it should. From there I can start working on the emotion detection portion of the project and the rough draft of the paper as well.

By November 16th I hope to have a good portion of the emotion detection software completed and the rough draft finished. By a good portion I mean I hope to have the code implemented at the very least (working or not working). Once I have all of the code implemented and ready to go I have to then learn how to take that code and put it into an Android application. I am most concerned with this step as I have never done any mobile application software before. Then, by December 4th the goal is to have the full application finished and ready to present.

Ideally, the application will be nearly ready to go if not working as expected. However realistically, I want at least the code itself to be working apart from the application portion. If I can achieve that then I believe I can produce a quality paper by December 12th.

6. REFERENCES

[1] C. Padgett and G. Cottrell, "Representing face images for emotion classification," p. 895âĂŞ900.

[2] A. Samal and P. A. Iyengar, "Automatic recognition and analysis of human faces and facial expressions: a survey," *Pattern Recognition*, vol. 25, no. 1, p. 65âĂŞ77, 1992.

[3] A. Majumder, L. Behera, and V. K. Subramanian, "Emotion recognition from geometric facial features using self-organizing map," *Pattern Recognition*, vol. 47, no. 3, p. 1282âĂŞ1293, 2014.