**Name: Srikant Iyengar**

**Net ID: sxi140530**

**Topic:**

Emotion Recognition

**Description:**

Given a video clip of a person’s face, determine the emotion expressed in the face. These emotions will be “coarse” emotions such as happy, angry, sad, etc.

**Libraries Used:**

Stasm 4.1.0, dirent.h, opencv

**Dataset Used:**

cohn-kanade

**Training Process:**

1. Using a python script attached in the submission, we generate a text file.

2. This text file contains the path for the image to be trained followed by a semicolon (;) and the label for the current image. For example: train\Anger/S011\_004\_03142515.png;0

3. Place the training dataset and the text file above within the data folder of the project. Also place the video to be used for detection within the data folder in the data folder.

**Approach:**

1. Extract useful features from an image. I used stasm to extract essential feature points in the face. I computed the distance between several aspects of the image based on the table as shown below and based on the comparison image in Fig 1.

|  |  |
| --- | --- |
| Number | Features |
| 1 | Distance between 1 and 2 |
| 2 | Distance between 1 and 6 |
| 3 | Distance between 2 and 7 |
| 4 | Distance between 4 and 5 |
| 5 | Distance between 3 and 4 |
| 6 | Distance between 3 and 5 |
| 7 | Distance between 8 and 9 |
| 8 | Distance between 6 and 8 |
| 9 | Distance between 7 and 9 |
| 10 | Distance between 6 and 10 |
| 11 | Distance between 7 and 11 |
| 12 | Inter-angle 8, 10 and 12 |
| 13 | Inter-angle 8, 10 and 12 |

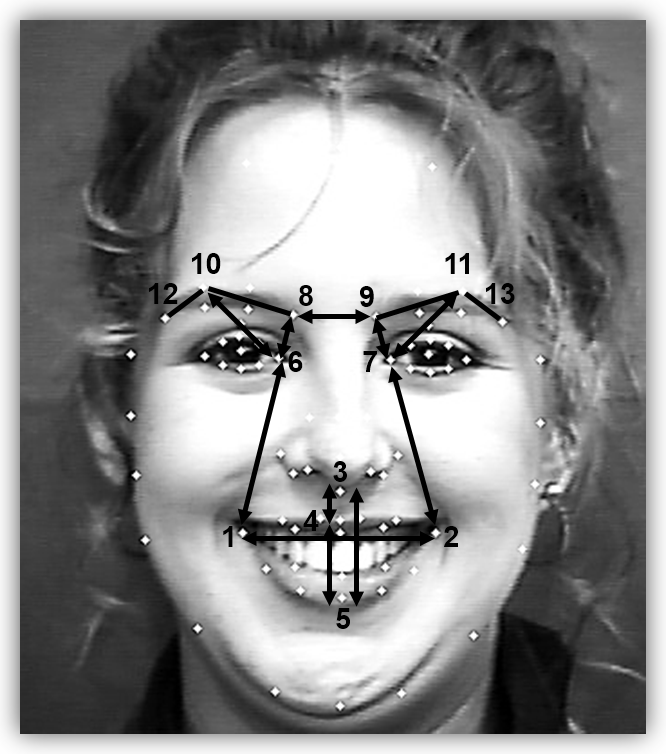
****2.

Fig 1: Points describing the features used for training the svm.

2. This creates a matrix with the number of rows equivalent to number of training examples used and the number of columns is equal to 13 where each values in the 13 columns are computed as per the table above.

3. Use the training matrix generated above to train the svm model. I assign a label for each emotion we represent. Thus we have label 0 for anger, 1 for Happy and 2 for sad respectively. The number of rows in the label matrix is same ass the number of training examples used.

4. Extract each frame from the video and compute the matrix as explained above. This frame will have a matrix of dimension 1\*13.

5. Using this matrix predict the label using the svm predict function. Set the prediction output in each frame as shown in the screenshots below.

6. Do this for each video frame.

7. For live video detection, simply replace the above process for live video frame instead of stored video frame.

**Screenshots:**







References:

<http://www.milbo.org/stasm-files/stasm4.pdf> - Stasm Manual

<http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html#introductiontosvms> – Documentation for svm tutorials

<http://docs.opencv.org/2.4/modules/contrib/doc/facerec/tutorial/facerec_video_recognition.html>

Emotion recognition tutorials provided by professor.