

Real Time Facial Expression Recognition App Development on Mobile Phones

Humaid Alshamsi¹ and Veton Kepuska¹Hongying Meng^{1,2}

¹Dept. Electronic and Computer Engineering Florida Institute of Technology Melbourne FL, USA

² Dept. Electronic and Computer Engineering Brunel University London

ABSTRACT

Facial expression has made significant progress in recent years with many commercial systems are available for real-world applications. It gains strong interest to implement a facial expression system on a portable device such as tablet and smart phone device using the camera already integrated in the devices. It is very common to see face recognition phone unlocking app in new smart phones which are proven to be hassle free way to unlock a phone. Implementation a facial expression system in a smart phone would provide fun applications that can be used to measure the mood of the user in their daily life or as a tool for their daily monitoring of the motion in psychology studies. However, traditional facial expression algorithms are normally computing extensive and can only be implemented offline at a computer. In this paper, a novel automatic system has been proposed to recognize emotions from face images on a smart phone in real-time. In our system, the camera of the smart phone is used to capture the face image, BRIEF features are extracted and k-nearest neighbor algorithm is implemented for the classification. The experimental results demonstrate that the proposed facial expression recognition on mobile phone is successful and it gives up to 89.5% recognition accuracy.

Keywords: *Machine Learning; Facial Expression; Image Processing; Mobile Computing*

1. INTRODUCTION

Face recognition technology has a wide range of applicability from auto unlock in smart phones and psychological usage in the past few years [12]. It is a very good example on the success of computer vision on embedded devices. Facial expression recognition in such is a similar idea that will have numerous uses in the future. Facial expressions are one of those things that hold great importance to human in communication as they tend to convey emotions, energies and expressions without the use of words in communication. Darwin stated to the biological roots of facial expressions and the important role it play in the survival of the species including human beings. Ekman [8] defined 6 basic emotional facial expression, happy, sad, disgust, anger, surprise and fear and states how these expressions are universal. The deepest emotions are embedded in the facial expressions of human beings. Ekman states that all the expressions are a blend of all these basic expressions which sometimes makes it difficult to understand. Facial expressions basically function with the activation/dilatation of one or more of the total forty-three facial muscles.

2. RELATED WORKS

Facial analysis gained a lot of popularity in the efforts and research in the past years. The applications of facial analysis include recognition, expression analysis, simulation and in the animation industry [1, 2]. Humans naturally are equipped in recognizing emotions via facial expressions. Somewhat, universally six basic expressions are identifiable and research purpose. Paul Ekman and his colleagues announced the Facial Action Coding System (FACS) covering all possible ones in fixed images [3]. This system has been used vastly by many researchers, especially those with a psychological background

and focus of interest and many facial expressions are coded by the FACS system. The FACS system is based on human observations and the labelling process is manual. For many years, researchers designed systems to classify facial expressions in an automated way. A lot of definitions of emotion have been given in previous years.

Numerous approaches were invented using still images and videos for facial expression recognition systems. The first to introduce local parametric motion representations are Black and Yacoob then preceding the information to the required classifier [4]. The low dimensional manifold technique for representing, trailing, and identifying facial expressions was created by Chang [5]. They utilized the Lipschitz embed system alignment for facial features for the manifold construction. The Gabor wavelets were subjugated to track the changes in facial expressions that were further examined temporal area for the expression recognition by Valstar and colleagues initially [6]. Some rare expressions of the face were studied using the templates of variable intensity by Kumano and his associates stating the differences between several classification templates [7]. The usage of Adaboost classifier and Support Vector machines rarely gained benefit after this [8]. Later on Torre's methods were known for graphical spectral techniques in this field [9]. Zeng's understanding was prominent where the separation of non-emotional and emotional facial expression was validated [10]. Sometimes non emotional are just commonly used habits of people. Recent research is focus on emotion recognition in the wild that was evidenced by the series of international challenges of EmotiW (2013[13], 2014[14] and 2015[15]) In these challenges, various automatic emotion recognition systems have been built to recognizing emotions from still images and short videos in a certain degree that might be used in real world applications.

However, none of all these methods can be used in portable devices such as mobile phones in real-time.

3. FACIAL EXPRESSION RECOGNITION

The first step towards the design is the pre-processing which includes detection and tracking the face movements by using HAAR cascades that is supported by open CV [20]. Then the face image goes on the second step of feature extraction. The core element of the design is the classification. This is where the K-NN algorithm is utilized to compare the histogram values with the stored values in the training set. The database used is the Cohn-Kanade (CK) database [19], a very big database. Cohn Kanade's database is one of the well-known databases of video recordings of facial expressions. The top of the facial expressions in those recordings is fully described by the activation of the Action Units (AUs). The database currently has 780 images for six emotional face images for eight people for the emotions such as happy, sad, angry, fear, disgust and surprise. After classification, it will show up the emotion bar of the facial expression image. The results of the application are very accurate and very few errors till date. The database is used for the testing of the proposed system. Then the application is built on an Android platform (eclipse).

A. Overview of the System

Figure 1 shows a block diagram of emotional face recognition on mobile phone. The three main stages in the system are: (i) feature extraction and (ii) feature classification. The feature extraction stage involves pre-processing stages such as acquiring a sequence of images (15 frames/second) using a video camera and detecting the facial region of the image and standardizing the properties for lighting the image, this application BRIEF feature extraction is being used. The feature classification process involves the K-nearest neighbor (KNN) algorithm; it is widely accepted as a powerful nonparametric pattern classification method in image processing.

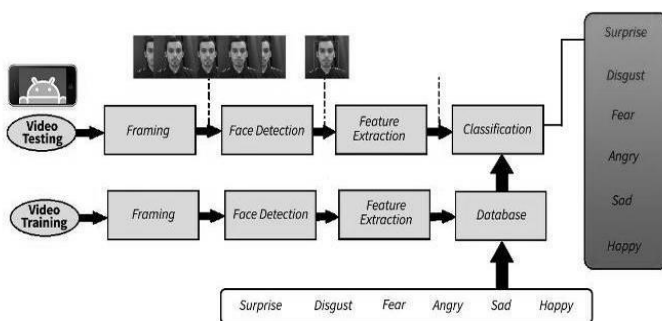


Figure 1: Overview structure of the facial expression recognition.

B. Feature Extraction

Binary Robust Independent Elementary Features (BRIEF) [16, 18] is explained in this paper. It is one of the simplest binary descriptor that functions based on the comparison between two smoothed pixel intensities and assigns binary number 0 or 1 based on their relative intensities. For example,

if the first smoothed pixel has more intensity than that of the second, a binary 1 is assigned in the descriptor string for the pair of smoothed pixel. On the other hand, if the intensity is less, than a binary 0 is assigned in the descriptor string. The selection of smoothed pixels for comparison is random.

Firstly, an image is divided into patches; each patches are selected around some interesting key points of the image. Basically, the key points are those that exhibit variation in intensities around them, not those with monotonous intensities all around. For example, it is unlikely to consider the middle of a sky in an image as an interesting point because of little or no variation in the intensities around it; the edge of a cloud in the sky should be preferred as key points because of obvious variation in intensities between white and blue color of the cloud and sky. Each patch contains, say $(60 \times 60$ or $50 \times 50)$ pixels.

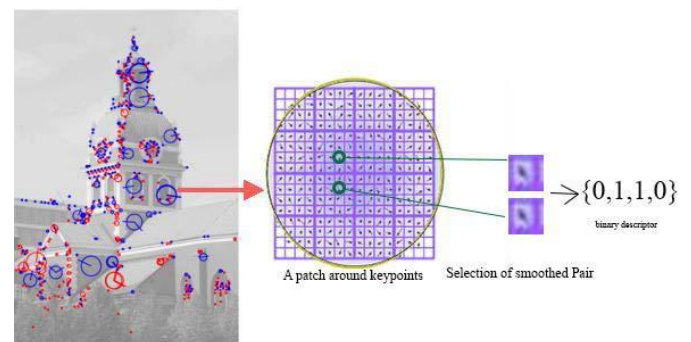


Figure2: Key point selection and pair comparison

Once the patches are selected, smoothed pixels are selected randomly as the second step of BRIEF. For example, if first pixel has a coordinate of $(-1, 10)$, the second might be chosen to be $(-2, 20)$ or $(-3, 30)$. Thirdly, they are compared and their binary descriptor value is registered in a Boolean string that contains all the binary descriptors of all the pairs from the same patch [figure 2]. Therefore, for each patch there is an individual Boolean string. The distance between two descriptors is computed as the number of different bits between them, which can be calculated as sum (XOR (descriptor1, descriptor2)).

BRIEF targets real-time applications leaving a large part of the available CPU power for subsequent tasks, but also permits running feature point matching algorithms on computationally weak devices such as mobile phones.

Many different types of distributions of tests were considered in [11]; here we use one of the best performers, a Gaussian distribution around the center of the patch. More specifically, we define test τ on patch p of size $S \times S$ as

$$\tau(p; x, y) := \begin{cases} 1, & \text{if } P1 < P2 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

The image is divided into patches of size $S1 \times S2$. They have been selected by choosing rows and columns of partition to be 7 and 6 respectively. For each patch p , two points $P1 = (x1, y1)$ and $P2 = (x2, y2)$ are generated from i.i.d. uniform distribution and the following test is performed in figure 3, we take our BRIEF descriptor to be the n_d –dimensional bit

string. This test is repeated $n_d = 256$ times to get 256-bit vector.

$$f_{n_d}(p) := \sum_{1 \leq i \leq n_d} 2^{i-1} \tau(p; x_i, y_i) \quad (2)$$

The design for BRIEF is not logically invariant. However, this is depicted in our results on basis of 5 samples of test data to get better result; that it does tolerate minute rotations. On a binary string if an image patch is to be represented, BRIEF is the descriptor because it relies on a comparatively lower number of intensity difference tests. The benefits of this descriptor are as follows: i) Quick assembly of way more efficient relative to its competitors; ii). Relative to its competitors it does provide higher recognition rates; iii). It is not mandatory to provide invariance in relation to large in-plane rotations.

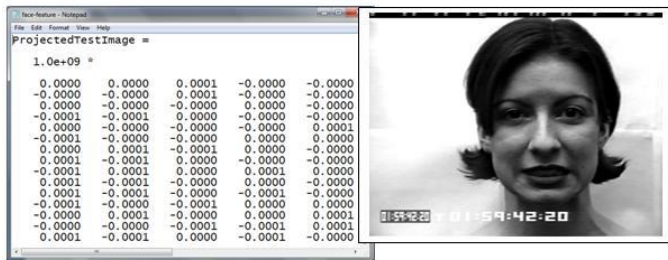


Figure 3: BRIEF feature example

C. K-Nearest Neighbor Classification

The K-Nearest Neighbors (KNN) algorithm was developed by Cover and Hart [3]. All these features make KNN ideal for classification problems. Classification can be achieved with minimal data distribution information. KNN has completely overcome the issues of probability densities, contributing to its wide use as a classification algorithm. It is a supervised learning algorithm [1]. This algorithm defines the nearest in the light of Euclidean distance. The Euclidean distance among records are given by

If $X_1 = (X_{11}, X_{12}, \dots, X_{1n})$ and $X_2 = (X_{21}, X_{22}, \dots, X_{2n})$ then

$$\text{Distance}(X_1, X_2) = \sqrt{\sum_{i=1}^n (X_{1i} - X_{2i})^2} \quad (3)$$

In the above equation, X_1 and X_2 are two records with n attributes; the distance formula is used to determine the distance between patterns X_1 and X_2 . In the K-nearest neighbor classification algorithm, the classification for the results of recent instance enquiry is dependent upon the category of the K-closest neighbor. Training examples have the form of n -dimensional numeric characteristics and every example correspond to a direction towards n -dimensional pattern place. The general concept of this algorithm can be explained as follows. KNN is a simple classification algorithm estimating the nearest neighbor dependent upon the space between the enquiry examples and the training test samples [11].

4. SYSTEM EVALUATION

The proposed system was evaluated in a public dataset firstly. It will demonstrate its efficiency on facial expression and validate the method.

A. Dataset

In the recent past there was a tendency towards resolutely including these kinds of problem pictures; the rationale is towards building a truly effectual and globally representative database and the screening process has to cope with these kinds of problems. Cohn and Kanade recommend that the subject filtration should not take place until after extraction of features, but be addressed by the data collection and extraction process [4].

In the recent past there was a tendency towards resolutely including these kinds of problem pictures; the rationale is towards building a truly effectual and globally representative database and the screening process has to cope with these kinds of problems. Cohn and Kanade recommend that the subject filtration should not take place until after extraction of features, but be addressed by the data collection and extraction process [4].

The data used in this application is stored in the Cohn-Kanade database [10]. In 2000, the Cohn-Kanade (CK) database was published with the intention to promote investigation into automatically recognizing person facial expressions. I have taken about 780 images of different people with their different emotions and identify six types of emotions (Happy, Sad, Fear, Surprise, anger, Disgust) which will be compared with the actual images.

The selection of images from the database is as follows:

The numbers of images of each emotion that have been assembled to be used in this project are (Happy 130, Sad 130, Fear 130, Surprise 130, Anger 130 and Disgust 130). These images were taken with high frame rate cameras.

B. Performance

The performance of BRIEF feature extracted and KNN algorithm has been examined.

The result of this testing will be stored in the external three files. The following tables display the results that are collected from the three external files, the first table demonstrates the system results of the first group testing which contain 390 images and these images was chosen randomly from the main file that has 780 images, the second table demonstrates the system results of the second group testing which contain 390 images which is the second half of the main file database, and the third table which is the last one demonstrates the system results of the third group testing which contain 780 images that add the first group and the second group test images. The format that been used to calculate the accuracy for these three tables is:

$$\text{Accuracy} = \frac{\text{correct Predicted}}{\text{Total number of images}} \times 100\% \quad (4)$$

Note that the numbers of the test images for each emotion are equal because the images for testing have been selected

randomly from the test file. Moreover, the system may test an image more than once.

TABLE I. CONFUSION MATRIX ON THE CLASSIFICATION

| Predicted \ True | Fear | Surprise | Happy | Sad | Angry | Disgust |
|------------------|------|----------|-------|-----|-------|---------|
| Fear | 116 | 2 | 1 | 3 | 4 | 4 |
| Surprise | 0 | 118 | 4 | 3 | 2 | 3 |
| Happy | 3 | 4 | 115 | 2 | 2 | 4 |
| Sad | 5 | 2 | 3 | 115 | 2 | 3 |
| Angry | 3 | 2 | 2 | 1 | 119 | 3 |
| Disgust | 2 | 4 | 2 | 3 | 4 | 115 |

5. IMPLEMENTATION ON THE PHONE

A. Feature Extraction

This section of the report pertains to the part of the code related to feature extraction and method which has been utilized to extract the features of the images. After extracting the features of the images, these features are stored to make a database so that it can further be utilized for testing purposes.

When the main body of the code is run, it asks for the training images and then asks for the test images. When the images are loaded in the system, these images presented to feature extraction module.

B. Testing of the System

The developed training database was tested on the test images to see whether the database is formed correctly and the system is working well. In order to test the system, there must be some classifier to compare the test images with that of training images database. Testing has been done by developing code in MATLAB and was tested on three databases of 390, 390 and 780 images each. Following is the code which is used to implement the testing using K-NN classifier and the results will be depicted.

C. Testing on the Phone

BRIEF has emerged as an important feature descriptor. Its value lies in its computational complexity as well as its recognition rate in most cases. A variant of the BRIEF descriptor named BRIEF-64 (i.e. 64 bytes descriptor) has been implemented. It has been shown in [5, 7] that a 64 bytes descriptor is sufficient to obtain very good matching results.

The input for feature extraction is the region of interest i.e. the face. This region is converted to grayscale. Gaussian smoothing [5, 7] is applied on the resulting image i.e. it is filtered using a Gaussian low-pass filter with variance of Gaussian kernel taken as 2 [5, 7] and window size of 9x9 [5]. The testing of BRIEF descriptors works on pixel level and is thus very sensitive to noise. By pre-smoothing the patch, this sensitivity can be reduced, increasing the stability and repeatability of the descriptors. Matlab simulation result of filtering is shown in Figure 5.



Figure 4: Testing result from the mobile phone

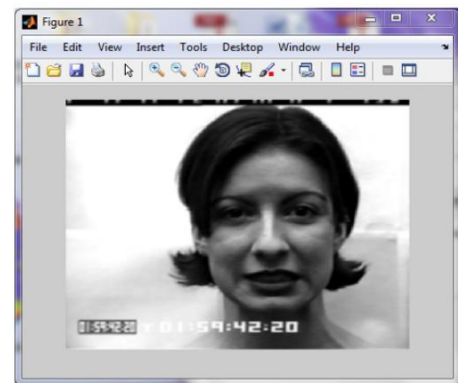


Figure 5: Image after Gaussian smoothing

The image is divided into patches of size $S_1 \times S_2$. They have been selected by choosing rows and columns of partition to be 7 and 6 respectively [17]. For each patch p , two points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ are generated from i.e. uniform distribution [16, 18].

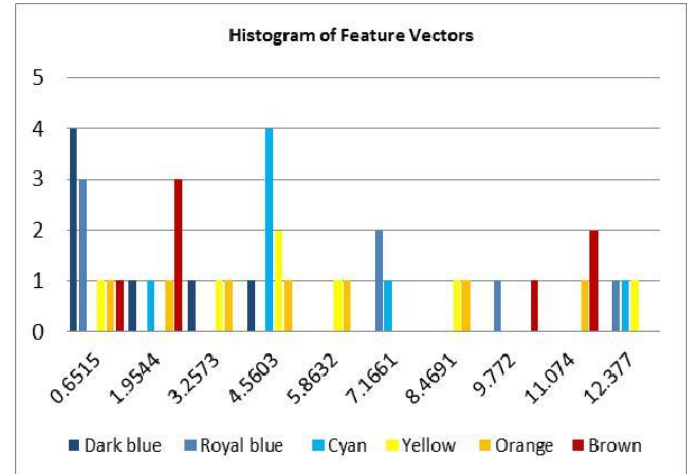


Figure 6: Histogram of BRIEF descriptors

Histogram of the values of BRIEF descriptors for all the patches is then plotted to visualize the values of the features in a graphical manner. The number of bins can be varied as per requirement. The color code is as follows:

- Dark blue represents column number 1.
- Royal blue represents column number 2.

- Cyan represents column number 3.
- Yellow represents column number 4.
- Orange represents column number 5.
- Brown represents column number 6.

The absence of a color in any bin out of 10 means there is no element in that bin. A bar of height 1 means that one element of the specified column exists in that particular bin. Similarly, a bar of height 2 means that two elements of the specified column exist in that particular bin and so on. A sample histogram is shown in figure 6. Due to the randomness in selection of two points for binary tests, the resulting histogram is bound to show different values on different runs of the software but statistically they represent the same image.

D. Testing summary

The final step in this project is to assess and evaluate the project performance; to measure how many of the requirements for emotional face recognition on mobile phones can be achieved. Actually, testing has been continuously addressed from the early implementation stage until the final stage.

Firstly, the testing of each function is carried out individually. It is tested to ensure that the algorithm and each line code works correctly. Sometimes, a small sample of the image data is used to test the code and sometimes a different kind of data set is built. For example, a database of numeric matrix is built to calculate easily the results of the tested function manually and to compare the target results with the code results.

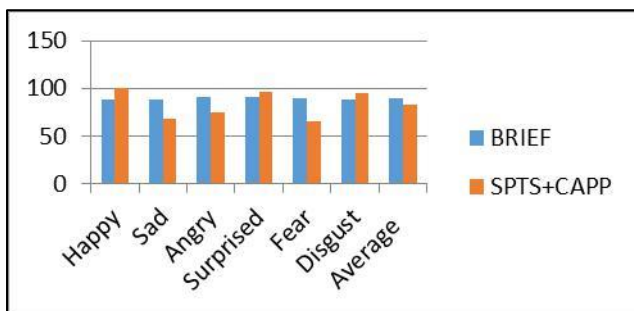


Figure 7: Performance comparison on BRIEF and SPTS+CAPP features

Secondly, after completing a certain stage, the performance of that stage is tested. Furthermore, after integrating the system stages, the overall system performance was tested. In these phases, sometimes an implemented algorithm that is not useful for this project is discovered; then one is forced to return to the starting point. For example, the K-nearest neighbor algorithm is described three times in three different papers. All these attempts failed in our hands. The problem was with the huge number of multi-class features that need to be trained. To solve this problem, attention was turned to the Android platform tools and a Matlab tool that can be used with the project data. The Android platform was used to program the application and

the Matlab tool used to test the application. Eventually, after many attempts, the optimal solution was found.

TABLE II. CLASSIFICATION ACCURACY

| | BRIEF | SPTS+CAPP |
|------------------|-------|-----------|
| Happy | 88.5 | 100 |
| Sad | 88.5 | 68 |
| Angry | 91.5 | 75 |
| Surprised | 90.7 | 96 |
| Fear | 89.2 | 65.2 |
| Disgust | 88.5 | 94.7 |
| Average | 89.5 | 83.2 |

Finally, the entire project was tested to guarantee that the project met the requirements and achieved its aim. This is the most significant step, which ensures that the study algorithm is efficient and powerful; that it could recognize emotional face expressions from a mobile phone.

6. CONCLUSION AND DISCUSSION

By comparing the results of both methods, it is evident that my system developed using BRIEF feature outperforms the other as its results were more than 85% in the detection range while in the other paper the average detection rate is about 66.7% for CAPP features and around 53.7% for SPTS features (results are for the first six emotions). When they combined the features (SPTS + CAPP) the detection rate of the system was increased to around 83.2% but it was still lower than our system [9]. By using hybrid features detection rates can be increased. While comparing each emotion detection results of system (using BRIEF features with 780 images) with one given in the reference (using SPTS + CAPP features) following results are obtained: Angry - 83% (75.00%), Disgust - 86.1% (94.7%), Fear - 83% (65.2%), Happy - 87.6% (100%), Sadness - 84.6% (68.00%), Surprised - 81.5% (96%). As we see from individual emotion results that in most of the emotions the referenced system outperforms ours, but overall their average is a little lower than our result. As you can see my result after add 390 images the first database and 390 images the second database (using BRIEF feature) following result are obtained: Angry - 119 (91.5%), Disgust - 115 (88.5%), Fear - 116 (89.2%), Happy - 115 (88.5%), Sad - 115 (88.5%), Surprised - 118 (90.7%) and the average for these six emotions is 89.5%. Table 4; shows the comparison between the two work that been explained above.

Although the proposed method has been successfully implemented on smart phone for real-time facial expression recognition and a satisfied performance has been achieved, the work can be improved further by adding new features such as selected pre-trained facial expression deep learning features. It can also be very interesting to try other machine learning methods for the classification. The classifier should be run real-time on the smart phone, but the training can be down offline.

ACKNOWLEDGMENT

The work of Hongying Meng was supported by Brunel Research Initiative & Enterprise Fund on the project entitled "Automatic Emotional State Detection and Analysis on Embedded Devices".

REFERENCES

- [1] Pantic, M., Pentland, A., Nijholt, A., Huang, T.S.: Human Computing and Machine Understanding of Human Behavior: A Survey, in Artificial Intelligence for Human Computing, 2007.
- [2] Li, S.Z., Jain, A.K.: Handbook of Face Recognition, ISBN: 0-387-40595-X, 2005.
- [3] Ekman, P.: Strong evidence for universals in facial expressions: a reply to Russell's mistaken critique. Psychol. Bull. 115(2), 268–287, 1994.
- [4] Black, M. J., Yacoob, Y.: Recognizing facial expressions in image sequences using local parameterized models of image motion. Inter. Journal of CV, 25(1), pp. 23–48, 1997.
- [5] Hu, C., Chang, Y., Feris, R., Turk. M.: Manifold based analysis of facial expression, Image and Vision Computing 24, pp. 60–614, 2006.
- [6] Valstar, M.F., Pantic, M.: Fully automatic facial action unit detection and temporal analysis. Proceedings of IEEE Int'l Conf. Computer Vision and Pattern Recognition, 2006.
- [7] Kumano, S., Otsuka, K., Yamato, J., Eisaku, S., Sata, Y.: Pose-Invariant facial expression recognition using variable intensity templates. Asian Conf. on Computer Vision, 2007.
- [8] Bartlett, S., Littlewort, G., Frank, G., Lainscsek, C., Fasel, I., Movellan, J.: Fully automatic facial action recognition in spontaneous behavior. In Proc. Conf. Automatic Face & Gesture Recognition, pp. 223--230, 2006.
- [9] Torre, F., Campoy, J., Ambadar, Z., Cohn, J.F.: Temporal Segmentation of Facial Behavior, International Conference on Computer Vision, October, 2007.
- [10] Zeng, Z., Fu, Y., Roisman, G.I., Zhen, W.: Spontaneous emotional facial expression detection. Journal of Multimedia, 2006.
- [11] Sakarkaya, M.; Yanbol, F.; Kurt, Z., "Comparison of several classification algorithms for gender recognition from face images," Intelligent Engineering Systems (INES), 2012 IEEE 16th International Conference on , vol., no., pp.97,101, 13-15 June 2012.
- [12] Dieter Findling, Rainhard, and Rene Mayrhofer. "Towards Pan Shot Face Unlock". Int J of Pervasive Comp & Comm 9.3 (2013): 190-208. Web.
- [13] Dhall, R. Goecke, J. Joshi, M. Wagner and T. Gedeon, Emotion Recognition In The Wild Challenge 2013, ACM ICMI 2013
- [14] A. Dhall, R. Goecke, J. Joshi, K. Sikka and T. Gedeon, Emotion Recognition In The Wild Challenge 2014: Baseline, Data and Protocol, ACM ICMI 2014.
- [15] Dhall, Abhinav et al. "Video And Image Based Emotion Recognition Challenges In The Wild: EmotiW 2015". ResearchGate. N.p., 2015. Web.
- [16] Calonder, Michael, et al. "Brief: Binary robust independent elementary features." Computer Vision–ECCV 2010. Springer Berlin Heidelberg, 2010. 778-792.
- [17] Abbo, Anteneh A., et al. "Mapping facial expression recognition algorithms on a low-power smart camera." Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on. IEEE, 2008.
- [18] Calonder, Michael, et al. "BRIEF: Computing a local binary descriptor very fast." Pattern Analysis and Machine Intelligence, IEEE Transactions on 34.7 (2012): 1281-1298.
- [19] "The Extended Cohn-Kanade Dataset (CK+): A Complete Dataset For Action Unit And Emotion-Specified Expression". N.p., 2016.
- [20] Opencv.willowgarage.com. 2013. FaceDetection - OpenCV Wiki. [online] Available at: <http://opencv.willowgarage.com/wiki/-FaceDetection>.