Proceedings of the Second International Conference on Inventive Systems and Control (ICISC 2018)
IEEE Xplore Compliant - Part Number:CFP18J06-ART, ISBN:978-1-5386-0807-4; DVD Part Number:CFP18J06DVD, ISBN:978-1-5386-0806-7

Facial emotion recognition in real-time and static images

Shivam Gupta
Electronics Engineering Department,
Harcourt Butler Technical University.
Kanpur, India
Email: sadam9099@gmail.com

ABSTRACT

Facial expressions are a form of nonverbal communication. Various studies have been done for the classification of these facial expressions. There is strong evidence for the universal facial expressions of eight emotions which include: neutral happy, sadness, anger, contempt, disgust, fear, and surprise. So it is very important to detect these emotions on the face as it has wide applications in the field of Computer Vision and Artificial Intelligence. These fields are researching on the facial emotions to get the sentiments of the humans automatically. In Robotics, emotions classification can be used to enhance human-robot interactions since the robot is capable of interpreting a human reaction [13]. In this paper, the emotion detection has been done in both real-time and static images. In this project, we have used the Cohn-Kanade Database (CK) and the Extended Cohn-Kanade (CK+) database, which comprises many static images 640 x 400 pixels and for the real-time using the webcam. The target expression for each sequence in the datasets are fully FACS (Facial action coding system) coded and emotion labels have been revised and validated [3]. So for emotion recognition initially we need to detect the faces by using HAAR filter from OpenCV in the static images or in the real-time videos. Once the face is detected it can be cropped and processed for further detection of facial landmarks. Then using facial landmarks the datasets are trained using the machine learning algorithm (Support Vector Machine) and then classified according to the eight emotions [1]. Using SVM we were getting the accuracy of around 93.7%. These facial landmarks can be modified for getting better accuracy.

Keywords: FACS, HAAR, Support vector machine, OpenCV, k-means clustering.

1. INTRODUCTION

An emotion can be defined as the physiological and mental state which is subjective and private. It involves a lot of behaviors, actions, thoughts, and feelings. In the field of computer vision these emotions, play an important role for different research purposes. In this project, the images datasets were organized properly and then sorted accordingly. Then the face was detected in all the images using the Haar filters [20] in OpenCV as it has few face recognized classes and then detect, crop and save faces. The classification is done using the supervised learning (SVM) [15] in this project as it gave better accuracy. The training and classification sets were created and we randomly sample and train on 80% of the data and classify the remaining 20%. This whole work was done on the Cohn-Kanade datasets of static images.

This study has also been extended in the real time as well in which like we detect the emotions in the static images. Now the webcam will be running a video and the faces are going to be detected in the frames according to the facial landmarks which will contain the eyes, eyebrows, nose, mouth, corners of the face. Then the features were extracted from these facial landmarks (dots) faces which will be utilized for the detection of the facial emotions [4]. This was optimized by calculating the centre of gravity of all these dots and calculating the distance between the centre of gravity and the corresponding dots. After the extraction of these features, the machine learning algorithms were applied for training and classifying the different emotions [5]. Then the performance was evaluated in the real-time and the static images.

2. MATERIALS AND DATASETS

We have used the Cohn-Kanade Database (CK) and the Extended Cohn-Kanade Database (CK+) of the Carnegie Mellon University (CMU) which contains the sequence of static images from neutral to a particular emotion. In this project, we have implemented it in Python. OpenCV (open source computer vision) was required as this project is based on the computer vision. For the Real-time webcam is required which is going to record the video and the facial emotions will be detected. We have used python 2.7 (Anaconda + Jupyter notebook is a nice combo-package). Dlib library was installed (a C++ library for extracting the facial landmarks). CMake and Boost-Python were also used in this research-based project.

3. METHODOLOGY

3.1 In Static Images

3.1.1. Data organization

A face consists of some features on it which play an important role in the detection of the emotions on it. The emotion recognition system is divided into 3 stages: face detection, feature extraction, and emotion classification. We have encoded our eight emotions in the datasets as {0= happy, 1=sadness, 2=fear, 3=anger, 4=surprise, 5=disgust, 6=contempt, 7=neutral}. Initially, the faces are detected in all the sequences of the Cohn-Kanade Database. First, we have organized the dataset by preparing two folders called "emotions source" and "images source" in the directory we are working and put all folders containing the text files with FACS in a folder called "emotions source" and put the folders containing the images in a folder called "images source". We have also created a folder named "sorted images set", to collect our sorted emotion images. Within this folder, we have folders for the emotion labels ("happy", "disgust", etc.). Each image sequence consists of the development of an emotional expression, starting from a neutral face and ending with some particular emotion. So, from each image sequence, our focus is to extract two images that are one neutral (the first image) and one with an emotional expression (the last image) in the sequence.

3.1.2. Extracting Faces

The classifier will work properly if the images contain only the faces so the images were processed accordingly for the detection of the faces and then were converted to grayscale and were cropped and were stored in some specific folder [7]. We have used a HAAR filter from OpenCV for automatic detection of faces. As OpenCV contains 4 pre-trained classifiers, so it is better we detect as many faces as possible. We have Created another folder called "Extracted faces datasets", and subfolders within this folder for each emotion ("sadness", "happy", etc.).

3.1.3. Training and classification

The dataset has been organized and is ready to be recognized, but initially, we need to actually train the classifier what particular emotions look like. The approach we have used is to split the complete dataset into a training set and a classification set. We use the training set to teach the classifier to recognize the labels to be predicted, and used the classification set to estimate the performance of the classifier. After creating the training and classification set, we randomly sample and train on 80% of the data and classify the remaining 20%, and repeat the process 20 times [15]. After training the fisher face classifier the emotions were predicted.

3.2. In Real-Time

In real time the emotions can also be detected but it becomes quite complex when compared to the static images as in the real-time the webcam is recording a video which is a collection of many frames, not just a single frame. In this case, we have used the Facial Landmarks [11] approach to detect emotions, continuously which is more robust and powerful than the fisherface classifier [12] which was used in the sequences of static images but it also required some more features and modules. After installing the Dlib libraries, CMake and boost python build the libraries.

3.2.1. Testing the landmark detector

Initially, we need to operate the webcam on the computer which is going to record the real-time video. The faces will be detected in each frame of the webcam video and then the further processing will be done on those detected faces. In real-time also we have used OpenCV to operate webcam. Then after that, the image is processed by converting to grayscale, optimising the contrast with an adaptive histogram equalization.

3.2.2. Extracting features from the faces

In the feature extraction stage, the faces detected in the previous stage are further processed for identification of eye, eyebrows, nose, corner of face and mouth regions. Initially, the likely Y coordinates of the eyes were identified with the use of the horizontal projection. Then the areas around the y coordinates were processed to identify the exact regions of the features. Finally, a corner point detection algorithm was used to obtain the required corner points from the feature regions.

This will result in a lot of dots on the faces in the webcam video outlining the shape and all the "moveable parts" [19]. The latter is, of course, important because it is what makes emotional expressions possible. These dots are very important for the extraction of the features for the training and classification using the Machine learning algorithms. Then we implemented the ways to transform these nice dots overlaid on faces in the webcam video into features to feed the classifier. Features are small information that is used to describe the object or object state that we are trying to classify into different categories. The facial landmarks from the image material tell about the position of all the "moving parts" of the depicted face, the things we need to express an emotion.

We calculated the coordinates of all facial landmarks [16]. These coordinates are the first collection of features, and this might be the end of the road. We derived other measures from this that will inform the classifier more about what is being calculated on the faces detected. We tried to extract more information from what we have. Feature generation is always a better way for classification because it brings you closer to the actual data [16]. The coordinates may change as my face moves to different parts of the frame. I could be expressing the same emotion in the top left of an image as in the bottom right of another image, but the resulting coordinate matrix would express different numerical ranges. To get rid of these numerical differences originating from faces in different places of the image we normalized the coordinates between 0 and 1. This was done by the following equation (1) :

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

The more precise way which we have implemented to calculate the position of all points relative to each other. To do this we calculated the mean of both axes, which results in the point coordinates of the sort-of "centre of gravity" of all face landmarks. We can then get the position of all points relative to this central point. Note that each line has both a magnitude (distance between both points) and a direction (angle relative to image where horizontal=0°), in other words, a vector. The problem we need to fix is that Faces may be tilted in the webcam which might confuse the classifier. We can correct for this rotation by assuming that the bridge of the nose for most people is more or less straight, and offset all calculated angles by the angle of the nose bridge. This rotates the entire vector array so that tilted faces become similar to non-tilted faces with the same expression.

Finally, comes to the training and classification The main focus was to read the existing dataset into a training and prediction set with corresponding labels, trained the classifier (we used Support Vector Machines with the linear kernel from SKLearn, polynomial or RBF(Radial basis function kernel)) [18] , and evaluated the result. This evaluation was done in two steps; initially, we get an overall accuracy after ten different data segmentation, training and prediction run, second, we will evaluate the predictive probabilities.

3.2.3. Support Vector Machines

We used Support Vector Machines (SVMs) as a classification (also known as supervised learning) method in order to classify these eight facial emotions. SVMs are learning methods, which aim to find the optimal separating plane that analyzes data and recognize pattern used for regression analysis. In SVM, P data is classified to which class it belongs, by points with a (P − 1) dimensional hyperplane, which is called a linear classifier. The optimal hyperplane that separates the clusters of vectors is found by SVM modeling. The cases with one category of the target variable are on one side of the plane and cases with the other category are on the other side of the plane. Figure 1 illustrates the working principle of SVM.

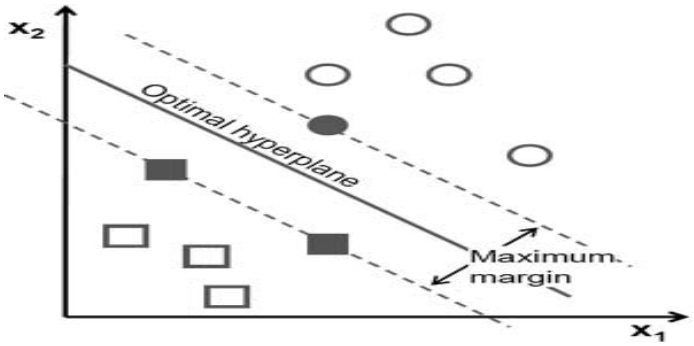


Figure 1. The working principle of SVM.

A good separation between the two possible classes is achieved by building a maximal margin hyperplane. The margin maximizes the distance between the classes and the nearest data point of each class. In general, the larger the margin is, the lower the generalization error of the classifier. Figure 2 shows the trade-off margin choice. In addition, SVMs handles the separation by a kernel function [18] to map the data into a different space with a hyperplane. SVM gives the flexibility for the choice of the kernel, as shown in Figure 3. Linear, polynomial and radial can be taken as an example for a kernel function. The choice of a kernel depends on the problem we are trying to model.

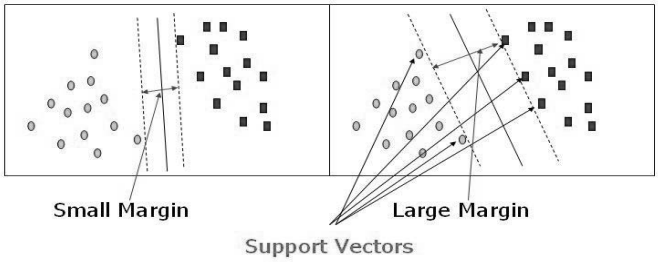


Figure 2. The trade-off margin choice

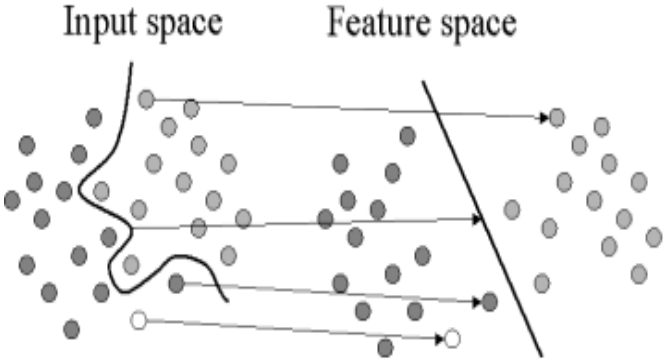


Figure 3. Choice of Kernel function

4. RESULTS

4.1. In static Images

In the sequence of static images in the Cohn-Kanade datasets, the emotions were detected with a great accuracy. In this sequence, all images were sorted and the faces were detected from the haar filters in OpenCV and were converted to grayscale and were cropped and were stored. So figure 4 shows some of the images from the datasets which contained cropped images of faces only which predicted the emotions perfectly with an accuracy of 93% by SVM classification.

(a) Happy (FACS-0)

(b) Sadness (FACS-1)

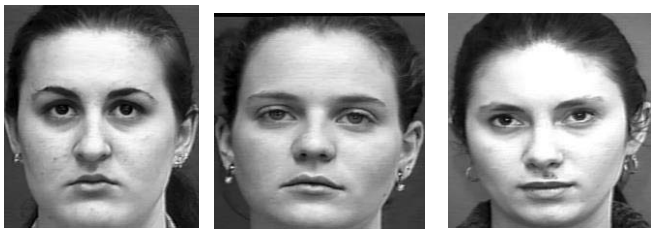
(c) Fear (FACS-2)

(d) Angry (FACS-3)

(e) Surprise (FACS-4)

(f) Disgust (FACS-5)

(g) Contempt (FACS-6)



(h) Neutral (FACS-7)

Fig. 4 The cropped images of faces predicting the emotions respectively mentioned in (a), (b), (c), (d) (e), (f), (g), (h), (i).

These results show the emotion with their FACS values. The data was sampled as per 80% training and 20% testing for high accuracy. In some cases like in the happy images it was detecting angry or neutral and in surprise images happy and fear. These were the little errors in the prediction which reduced the accuracy to 92.1%. But these small errors are possible when we are dealing with the large data. The results were stored in different folders with their emotion names after prediction with their FACS value within that folder. So this was the automatic detection of the emotion in static images. We can try with different datasets and evaluate the accuracy accordingly.

4.2. REAL-TIME

In Real-time the webcam recorded the video and detected the faces in the yellow box and extracted the facial landmarks with red dots and then calculated the centre of gravity with the blue dot [6]. Then the vectors were calculated with red lines from that centre of gravity to each and every dot as shown in the figure. The figure 5 shows the screenshot of the results with these facial landmarks and then predicting the emotion by the SVM classification [17].

(a) (b)

(c)

Figure 5 (a) The face detection with the facial landmarks detected with red dots

(b)The facial landmarks with the centre of gravity (COG)

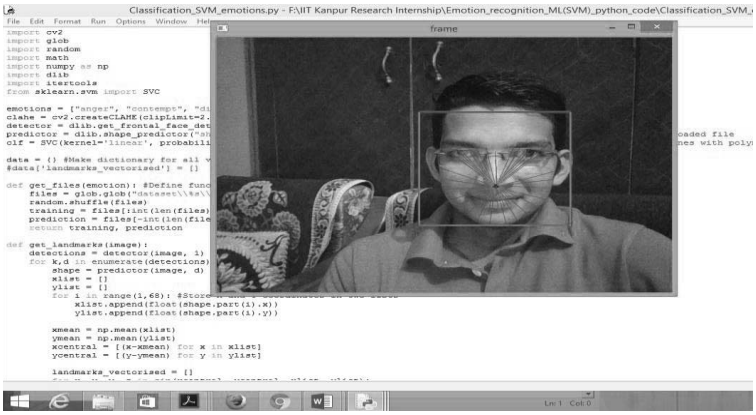
(c)The vectors from COG to every dot

5. DISCUSSION

The linear SVM classification predicted the emotion with great accuracy. But we also implemented with polynomial SVM [18] but the accuracy was less than the linear one. We have also tried with k-means clustering (unsupervised learning) [14] and Random Forest Classifier. But in our case SVM were predicting the emotions with better accuracy as compared to the other classification algorithms. The tip of the nose can be chosen as the central point in real time but would also throw extra variance in the mix due to short, long, high- or low-tipped noses [9]. Extra variance is introduced in the "centre point method"; so when the head moves away from the camera, the centre of gravity shifts accordingly, but we analyzed this is less than when we are using the nose-tip method because most faces more or less face the camera in our sets. The figure 6 shows the screenshot of the output in which the face is detected in red square and the facial landmarks are calculated with the red dots with the centre of gravity on the nose tip with the blue dot. Then the vector lengths and the angles are calculated as the features extraction. Then it was trained with SVM. It showed the FACS value as 4 which means happy. It was tested with different faces in the real-time with different emotions on the face it was detecting correctly with an accuracy of approximately 93.6%.

(a)

(b)



(c)

Figure 6 (a) The face detection with the facial landmarks detected with red dots
(b)The facial landmarks with the centre of gravity (COG on Nose Tip)
(c)The vectors from COG to every dot

6. PERFORMANCE EVALUATION

After the detection of the emotions, it is important that how much accurately our classifier is predicting. So table 1 shows the accuracy in which the column shows the actual emotion and the row show the emotions predicted then the accuracy is calculated for every emotion individually by dividing the number of emotions correctly detected to the total no of images. So the overall accuracy was coming around 92.1%. As the classification is done with different types of classification algorithms [10]. So table 2 shows the comparative analysis of accuracy in each case. So we found that linear SVM was giving the maximum accuracy of 94.1% when all the features were considered.

In static images

Emotion	Happy	Sad	Fear	Angry	Contempt	Disgust	Surprise	Neutral	Accuracy
Happy	**95**	**0**	**0**	**3**	**1**	**0**	**0**	**1**	**95%**
Sad	**1**	**85**	**2**	**0**	**3**	**0**	**1**	**0**	**92.3%**
Fear	**2**	**2**	**82**	**0**	**0**	**4**	**0**	**0**	**91.1%**
Angry	**2**	**1**	**3**	**78**	**0**	**3**	**0**	**0**	**89.6%**
Contempt	**0**	**2**	**1**	**3**	**64**	**0**	**3**	**0**	**87.7%**
Disgust	**1**	**3**	**2**	**0**	**0**	**84**	**0**	**1**	**92.3**
Surprise	**2**	**0**	**2**	**0**	**2**	**1**	**91**	**0**	**92.9**
Neutral	**0**	**2**	**1**	**1**	**0**	**0**	**1**	**81**	**91.2%**
Overall Accuracy = (660/717*100%) = 92.1%									

Table 1 Accuracy for different emotions

In Real Time

	ACCURACY		
Classification algorithm	With all features	With only vector length and angles	With just raw coordinates
Linear SVM	**94.1%**	**92.3%**	**91.5%**
Polynomial SVM	**91.2%**	**89.8%**	**89.9%**
K-Means Clustering	**88.7%**	**87.4%**	**86.6%**
Random forest classifier	**88.1%**	**81.5%**	**78.9%**

Table 2 Accuracy for different classifiers and features.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we presented the fully automatic recognition of facial emotions using the computer vision and machine learning algorithms which classify these eight different emotions. We tried many algorithms for the classification but the best which came out of the results was the support vectors machines with the accuracy of around 94.1%. Our results imply that user independent, fully automatic real-time coding of facial expressions in the continuous video stream is an achievable goal with present power of the computer, at least for applications in which frontal views can be assumed using the webcam. This machine learning based system for the emotion recognition can be extended to the deep learning system using the Convolutional Neural networks which will have many layers and the chances of getting much higher accuracy is there around 99.5% [2]. This project can be extended in which it will detect as many emotions of different people in one frame in the real-time videos [8]. Emotion recognition is going to be very useful in the near future in the research field of robotics and artificial Intelligence for example if a robot can sense the sentiment of any human and that robot can act accordingly without any intervention of any other humans. This automatic machine learning system for emotion recognition can also be extended with the detection of mixed emotions other than these eight universal emotions.

REFERENCES

[1] Patrick Lucey, Jeffrey F. Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar. "Recognizing Facial Expression: Machine Learning and Application to Spontaneous Behavior". Computer Vision and Pattern Recognition 2005.

[2] I. Cohen, N. Sebe, F. Cozman, M. Cirelo, and T. Huang. "Learning Bayesian network classifiers for facial expression recognition using both labeled and unlabeled data". Computer Vision and Pattern Recognition., 2003.

[3] P. Ekman and W. Friesen. Facial Action Coding System: A Technique for the Measurement of Facial Movement. Consulting Psychologists Press, Palo Alto, CA, 1978.

[4] A. Kapoor, Y. Qi, and R.W.Picard. Fully automatic upper facial action recognition. IEEE International Workshop on Analysis and Modeling of Faces and Gestures., 2003.

[5] M. Pantic and J.M. Rothkrantz. Automatic analysis of facial expressions: State of the art. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(12):1424–1445,2000.

[6] I. Cohen, N. Sebe, A. Garg, L. Chen, and T.S. Huang. Facial expression recognition from video sequences: Temporal and static modeling. Computer Vision and Image Understanding, 91(1-2):160–187, 2003.

[7] A.J. Colmenarez and T.S. Huang. Face detection with information based maximum discrimination. In IEEE Conference on Computer Vision and Pattern Recognition, pages 782–787, 1997.

[8] I.A. Essa and A.P. Pentland. Coding, analysis, interpretation, and recognition of facial expressions. IEEE Trans. on Pattern Analysis and Machine Intelligence, 19(7):757–763,1997.

[9] M.-H. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images: A survey. IEEE Trans. on Pattern Analysis and Machine Intelligence, 24(1):34–58, 2002.

[10] Nazia Perveen, Nazir Ahmad, M. Abdul Qadoos Bilal Khan, Rizwan Khalid, Salman Qadri." Facial Expression Recognition through Machine Learning" International Journal of Scientific & Technology Research Volume 5, ISSUE 03, MARCH 2016 ISSN 2277-8616.

[11] Ghimire, D., and Lee, J., 2013, Geometric feature-based facial expression recognition in image sequences using multi-class AdaBoost and support vector machines Sensors,13(6), 7714-7734.

[12] Abidin, Z., and Harjoko, A., 2012, A neural network based facial expression recognition using fisherface, International Journal of Computer Applications, 59(3), 30-34.

[13] Giorgana, G., and Ploeger, P. G., 2012, Facial expression recognition for domestic service robots, InRobo Cup 2011: Robot Soccer World Cup XV, pp. 353-364.

[14] Rituparna Halder, Sushmita Sengupta, Arnab Pal, Sudipta Ghosh, Debashish Kundu." Real-Time Facial Emotion Recognition based on Image Processing and Machine Learning". International Journal of Computer Applications (0975 – 8887) Volume 139 – No.11, April 2016.

[15] Claudia Lainscsek, Mark Frank, Ian Fasel, Marian Stewart Bartlett, Gwen Littlewort, Javier Movellan, "Recognizing Facial Expression: Machine Learning and Application to Spontaneous Behaviour", vol. 02, no. , pp. 568-573, 2005, doi:10.1109/CVPR.2005.297

[16] Bhuiyan, M. A.-A., Ampornaramveth, V., Muto, S., and Ueno, H., Face detection and facial feature localization for human-machine interface. NII Journal, (5):25–39, 2003.

[17] Philipp Michel, Rana El Kaliouby "Real-Time Facial Expression Recognition in Video using Support Vector Machines" ICMI '03 Proceedings of the 5th international conference on Multimodal interfaces; ISBN:1-58113-621-8.

[18] Amari, S. and Wu, S. (1999). Improving support vector machine classifiers by modifying kernel functions. Neural Networks, 12(6):783–789.

[19] Bartlett, M., Littlewort, G., Fasel, I., and Movellan, J.(2003). Real-time face detection and facial expression recognition: Development and applications to human-computer interaction. In Computer Vision and Pattern Recognition Workshop, 2003. CVPRW

[20] Kotsia, I., Buciu, I., and Pitas, I. (2008). An analysis of facial expression recognition under partial facial image occlusion. Image and Vision Computing, 26(7):1052– 106.