Facial Expression Recognition using Supervised Learning for Gestures

Tejas Padliya
tejas.padliya@gmail.com

ABSTRACT

"Facial Expression Detection" is one of the most challenging tasks in the field of object detection. Object detection is a field of computer vision that deals with detection and recognition of semantic objects (such as cars, human face, buildings and etc) in images and videos. Nowadays, with advancements in hardware technologies and increase in image acquisition. Facial expression in human images can convey important information. Facial expression can be used for deciding marketing and teaching strategies. Facial expressions along with hand gestures can be used in VR applications. It can useful in providing inclusive healthcare services.

In this paper, we propose a novel approach to extract facial features and recognize facial expression from an image. We display our hybrid approach to detect and measure facial features such as eyes, nose, mouth and eyebrows. We also describe various feature points and their associated distance which can be used for expression recognition by matching techniques using Supervised Machine Learning. We have used the JAFFE (Japanese Female Facial Expression) database for this project. We classify each image into seven categories of facial expression viz. 1) Angry, 2) Disgust, 3) Fear, 4) Happy, 5) Neutral, 6) Sad and 7) Surprise.

General Terms:

Algorithms, Experimentation, Measurement, Artificial Intelligence, Theory

Keywords

Facial Expressions, Gestures, Supervised Learning, Edge Detection, Noise Removal, Vision

INTRODUCTION

Facial expression recognition is important in many fields. It can be used to study behavior and emotions of a person. Thus, it finds application in teaching field. Software can interact more efficiently by understanding facial expression of a student. Facial expression recognition also finds its application in sign language recognition. It can be used to classify songs into appropriate folders based on facial expression. Hand gesture recognition [11] can be applied in healthcare for remote patient monitoring, rehabilitation, and physical therapy exercises, enabling healthcare professionals to track patients' movements and provide feedback in real-time.

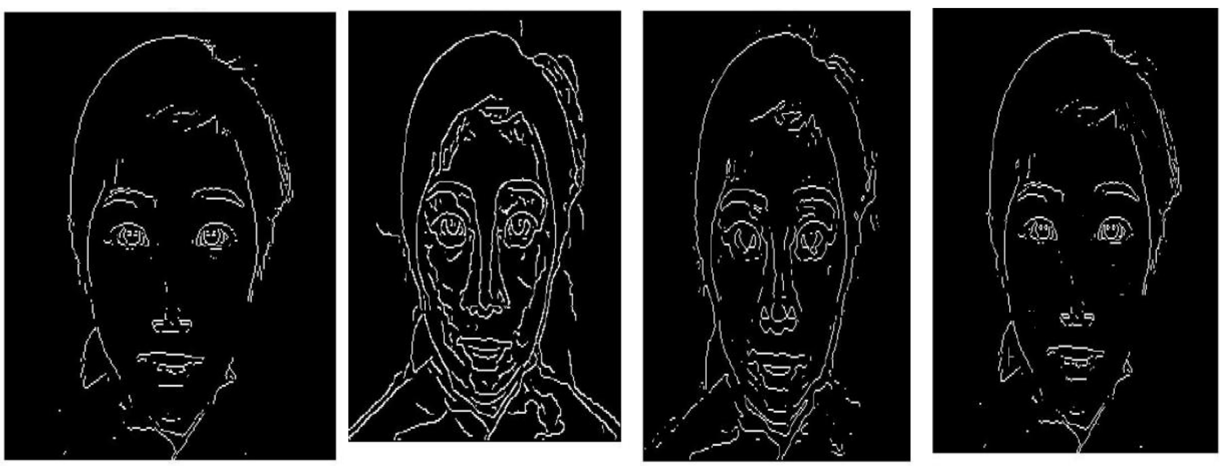
Our project can be classified into three components viz., 1) Detection of facial features, 2) Facial expression recognition and 3) hand gesture recognition. In our project, the images consist of neck and face region. To detect facial feature points, many solutions had been proposed. The paper Wilson et al [1] uses Haar classifier to detect facial features. The main component of the Haar classifier is the Haar-like features. Haar like features uses the difference in contrast values between adjacent rectangular region of pixels of the original images. Haar-like features are scale invariant. For RGB images, The authors of [4] and [5] propose to use color based skin segmentation to extract facial features and remove background from an image. Here hair region is also considered as background. Different regions can be removed by using connected component labeling. The author Yi et al [7] proposes to use skin color information to locate the facial features and then used adaptive canny edge detection for extracting the feature points to recognize the facial expression. For facial expression recognition training of images using feature points is required. The paper Turk et al [10] describes the use of eigenfaces for the face detection technique. Eigenface [12] is a conceptual term given to the faces detected using the eigenvectors. The basic approach followed to calculate the eigenvectors is that a probability distribution of all the vectors over the face images is calculated and then using that probability distribution, a covariance matrix calculated. These covariance matrices are then in turn used to calculate the eigenvectors which in turn is useful for the detection of faces. The paper Turk et al [10] describes in detail the steps followed by the authors for the detection of the faces. Initially, the training set of images are collected and the eigenfaces for each of these faces is calculated. The calculation of eigen faces involves the mathematical operations of multiplication and addition. Whenever a new face image comes up, its weights are calculated by comparing the new face with the already calculated eigenfaces. This step helps in detecting that the image is a face image or some other image. If the image is identified as a facial image, then it is further studied upon using the weights obtained to check and see if the face image is an image of a known person or not. Now, the paper [10] goes on to describe the identification step. This step makes use of pattern recognition algorithm. The pattern recognition approach (feature based template matching) checks for known patterns from the images and if there is a match in the pattern, it recognizes facial expression. The paper [14] describes the use of gabor wavelets for the face detection technique. Gabor wavelets [15] are basically calculated using fourier tranformations of the various facial features. Wavelets are nothing but mathematical functions which divides the data into various frequency range and then based on the frequency ranges a pattern matching algorithm is used to match the frequency distribution of the existing images and the new images. The main advatanges of using gabor wavelets are that it is robust against noise, it works even in the cases where the faces are illuminated, its is a fast recognition process, and so on. The important steps explained in the paper include: Image Acquiring, Feature Extraction and Neural Network Classification. The image acquisition phase captures the images of a person and various pre-processing techniques are used for noise removal. The gabor wavelets are then calculated for each of the important features points in the image. These gabor wavelets are used to determine the frequency distribution of those sections in the image. These features are then extracted as vector values for further processing. These feature points are then fed to a neural network in order to obtain a trained model of the feature points. The trained model is then used for testing the new unknown images to recognize the various feature points and thus based on that the corresponding face. Various mathematical calculation related to matrices and vectors are used in this paper. The last step in the paper [10] is the application of the pattern matching algorithm in order to find the common patterns between the unknown image and the trained image. The best way to train images invariant to scaling and rotation is using SURF (Speed Up Robust Features) or SIFT (Scale-invariant Feature Transform). The paper Raheja et al [3] makes use of the back propagation neural network algorithm to train the images. First, it applies all the preprocessing techniques to the images like image thresholding, histogram equalization and edge detection techniques. After that it trains the pre-processed images using neural networks. Now the new images are compared to the results of these trained images and the corresponding expression is detected. The paper Michel et al [5] uses SVM (Support Vector Machines) to train images. SVM is supervised binary linear classification algorithm. It constructs a hyperplane in multi-dimension space.

DATASET

The database of images we selected for our project is "The Japanese Female Facial Expression (JAFFE)". The database consist of 213 images posed by 10 Japanese female models. The 213 images consist of seven expressions viz. 1) Angry, 2) Disgust, 3) Fear, 4) Happy, 5) Neutral, 6) Sad and 7) Surprise. All images have pre-determined expression associated with it. All the images are grayscale images. They are of same size (256 * 256).

APPROACH

Initially we decided to use edge detection algorithms to detect edges of facial features as we are interested in their corner points. We thought of detecting edges and then segment the image using connected component labeling. We first smoothed the image by using median filter to remove noise. On filtered image we applied various edge detector operators to detect edges. Out of which 'Prewitt' gave us the best result as shown in the figure below.



Prewitt Edge Image Canny Edge Image **log Edge Image** Sobel Edge Image

The problem we faced here was that the edge discontinuity, to solve this problem we decided to apply edge linking algorithm or use morphological operations like dilations and erosions to fill the incomplete edges. When we tested our result on some randomly selected 5 images, The edge linking approach failed as the gap at the edges was very huge. So we were not able to proceed further to perform connected component labeling algorithm to separate (remove) the connected component with highest area and also to isolate the facial features.

We then developed a new hybrid approach to avoid edge discontinuity problems. Our new approach would work on different illuminated images and is independent of the small variation in the facial feature size. The proposed solution uses Matlab "Computer Vision System Toolbox". The new approach is shown in the figure 1 below.

The solution uses 'vision.CascadeObjectDetector' from "Computer Vision System" toolbox. The cascade object detector uses the Viola-Jones algorithm [8]. Viola-Jones algorithm was proposed in 2001 by Paul Viola and Michael Jones. It detects objects of various types in real time. It is a supervised learning algorithm for face detection. The features value calculated by it is similar to Haar basis function. It uses rectangular region to compute sum of pixels in a region. The concept of Haar-like features comes from Haar-like wavelets. The cascade object detector can be trained to a custom classifier to increase the detection accuracy or to recognize a particular object. The Matlab provides trained cascade models which can be used with the cascade detector to detect human faces, nose, eyes, eye pair, mouth and upper body.

The new approach

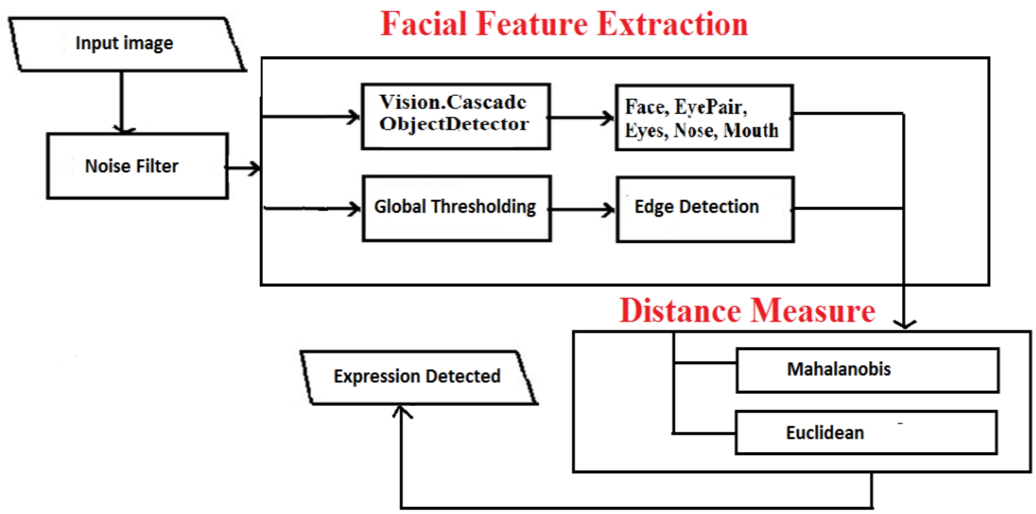


Figure 1. Final Algorithm

The smoothed image obtained by applying median filtering to the original image is given to vision.cascade object detector to detect facial features. We use face, eyes, nose, mouth as facial features which are detected using this cascade detector as shown in the figure 2. To detect face, we used 'FrontalFaceCart' model of vision.cascade object detector. This model detects faces that are forward facing and are in the upright position. This model is based on the classification and regression tree analysis (CART). We get the square bounded on the face part of an image. We then crop this face from the previous image. We get the height of the face (fHt). We then detect eye-pair using 'EyePairBig' model. We calculate the height (epH) and width (epW) from this. We then detect individual eyes using the single eye CART model. These models are based on CART trees. For the detection of left eye, we use 'LeftEyeCart' model. We calculate the distance as tempEyeLHeight and tempEyeLWidth. For the detection of right eye, we use 'RightEyeCart' model. We calculate the distance as tempEyeRHeight and tempEyeRWidth. At this stage, if we are not able to detect face or eye-pair, we don't detect single eyes, nose and mouth. After computing all eyes related parameters, We then crop the image such that now it only has region from below the eye to the lower end of the mouth. We then use 'Nose' model to detect nose. If the nose is detected then we remove it from the image and then pass the image to detect mouth. We then detect mouth using 'Mouth' model with 'MergeThreshold' parameter as 32'. In certain images like "KM.SU1.14.tiff " mouth is not detected as it is almost round and wide open. To detect mouth in such cases, we do global thresholding on the previous image obtained and then apply 'Prewit' edge detector to detect edges. Edges will be mostly of mouth and nose. We then apply circular Hough transform to detect mouth of such images. We then compute mouth width (mWt) and (mHt) using radius and center.

Based on these feature point values, we then calculated five feature point distance values which are used to determine the actual (averaged) computational vector for each of the expressions. These five special values are mouth width (mWt), mouth height (mHt), left eye height gap (eHtL), right eye height gap (eHtR) and eye width gap (eGap). The formulas for calculating left eye height gap (eHtL), right eye height gap (eHtR) and eye width gap (eGap) are mentioned below.

eHtL = tempEyeLHeight – epH;
eHtR = tempEyeRHeight – epH;
eGap = epW – (tempEyeLWidth + tempEyeRWidth)

The figure 2 shows how face, eyepair, eyes are detected using vision.cascade object detector. The figure 3 shows how all the feature distances are measured. The figure 4 shows the algorithm flow for mouth detection for the "NA.HA3.204.tiff" image. The figure 5 shows the algorithm flow for mouth detection for the "KM.SU2.15.tiff" image.

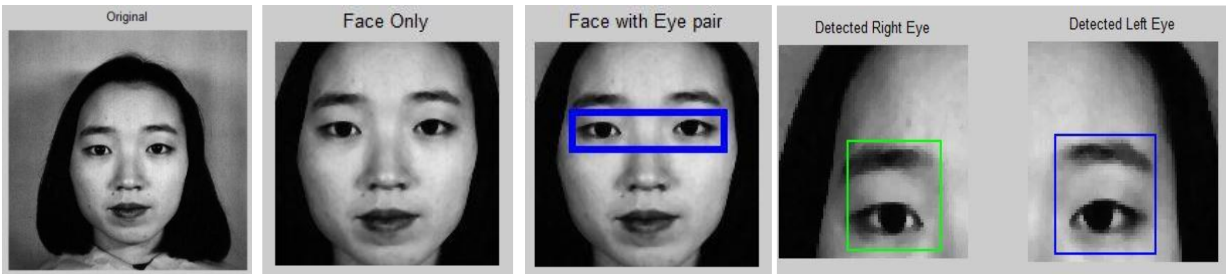


Figure 2. Face, EyePair and Eye Detection

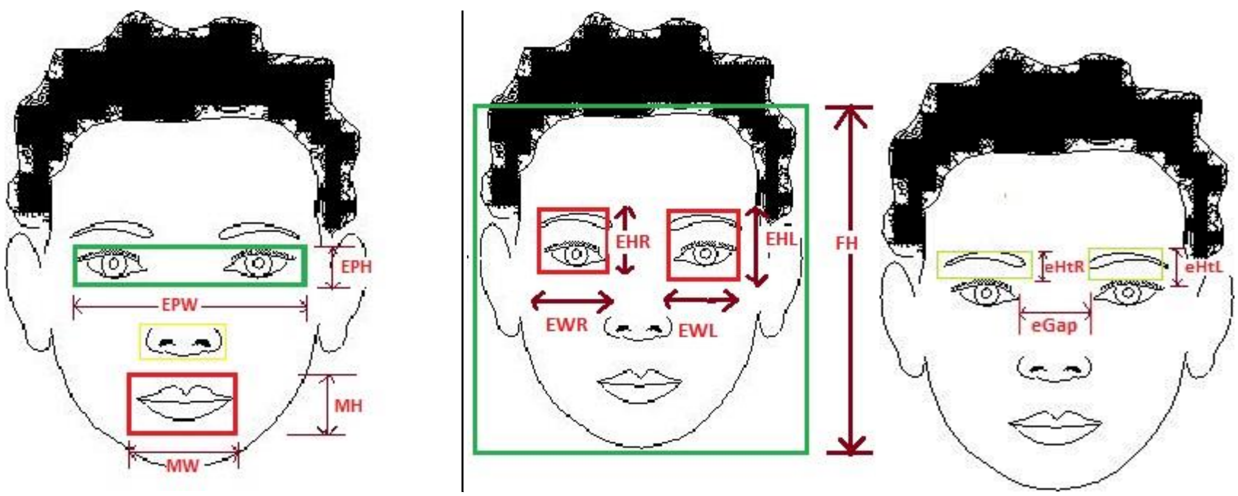


Figure 3. Feature Distance Measurement

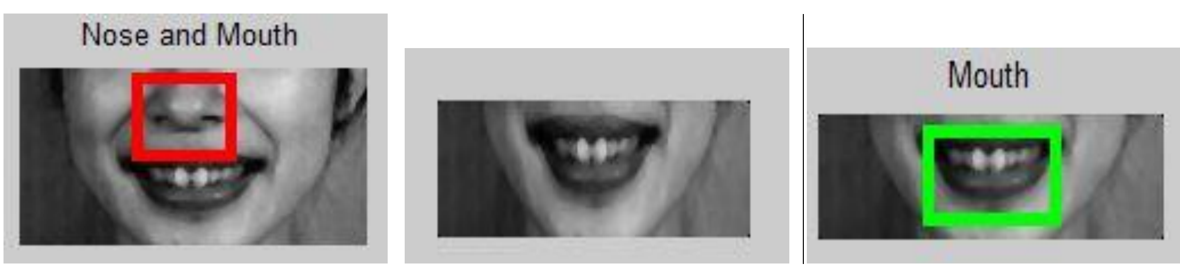


Figure 4. Nose and Mouth detection for "NA.HA3.204.tiff"

Figure 5. Nose and Mouth detection for "KM.SU2.15.tiff"

After finding these five features [mWt, mHt, eHtL, eHtR, eGap], We normalize this each feature values by their face height (fHt) for each image to increase accuracy and to let these feature vectors be independent of face size. As a result all width and height dimensions are normalized before passing to calculate a distance measure. For training, while doing it for the first time we create "trained.mat file which stores this five feature vectors for all the expression. When the test image comes next time for testing, there is no need to train data again and these feature vectors can be loaded from "trained.mat" file. This saves the computational cost. This also reduces the running time of the program. To recognize, i.e. to determine the facial expression of the test image, The feature vectors (having five values) for test image are calculated. Now these values are compared with the vectors of all the expression by using a distance measure to find similarities. The expression is identified if the distance between a particular expression feature vector and feature vector of test is minimum. This is similar to the concept of template matching in digital image processing. In template matching, the local weak features of test data are matched with the template data. Template matching requires large training data and normalization of data for better accuracy. Template matching. In our project, we are using a template based template matching as we are using the entire feature vector for finding expression. We are calculating the final feature vector for each expression by calculating the average of each feature vector value for all the images of that expression. For distance measure we are using two types of distance calculations viz. 1) Mahalanobis distance and 2) Direct distance (Euclidean distance). We are calculating the Euclidean distance in the Matlab by using 'norm' function. We have used our own implementation of Mahalanobis distance and are not using Matlab inbuilt function mahal(). Following are the final computed feature vector for each of the expressions. All the values are multiplied by 100 for the better representation purpose;

angryFeatureF = [32.51, 19.41, 7.02, 6.22, 22.63]   disgustFeatureF = [30.09, 19.05, 6.87, 6.51, 21.42]

fearFeatureF = [32.79, 19.98, 7.46, 7.88 , 19.64]   happyFeatureF = [34.25, 20.41, 8.43, 8.26, 18.43]

neutralFeatureF = [33.13, 19.84, 8.84, 8.66, 18]   sadFeatureF = [32.75, 19.57, 7.77, 7.56, 19.48]

surpriseFeatureF = [29.00, 20.03, 9.21, 8.83, 17.30]

RESULTS

We performed a series of evaluation on our facial feature detection and facial expression recognition result. We were able to recognize 206 faces and eye pairs from the 213 images in our data set. Out of these 206 images we were able to recognize nose and / or mouth of 201 images. So overall we were able to recognize all facial feature points for 201 images. Overall our facial feature detection accuracy is 97.57 %. There was no false mouth recognition in our approach as we removed some false positive nose and mouth detection using the principle of face symmetry and some careful use of if else statement in the code. We were not able to detect 5 faces and 7 eye pair as in those images the face is in tilted position. We were not able to detect certain eyes as they were very small for the (EyePairBig) classifier to detect it. However the EyePairSmall model for EyePair was able to detect it. But it was not able to detect some other eyes. Hence we decided to use the EyePairBig model for detecting the eye pairs. We were able to detect all individual eyes from 206 images by adjusting the parameter. Thus we are using 206 images to find facial expression. We tested our implementation for around 10 random images of each expression. The accuracy for each of the expression was not affected whether we choose the distance to be Euclidean distance or Mahalanobis distance. The results were same by using the both distance measures. Best expression to recognize was an angry as it had 100 % accuracy. For disgust we had 84 % accuracy. For happy and surprised we had 86 % and 80 % accuracy respectively. For fear and sad we had 75 % and 86 % accuracy. For neutral images we had 73 % accuracy. It is difficult to recognize fear and neutral images.

CONCLUSION

We performed the task of facial expression recognition by extracting and detecting facial features using supervised learning. We were fully successful in extracting facial features and finding the distance measure. We were successful in terms of accuracy for finding facial expression. We found that this approach is ideal for low computational resources availability versus using convolutional neural networks. It is ideal for recognition using mobile devices and secured as the facial expression is identified in the real time. We were able to normalize our five feature points parameter. We understood the need to normalize the feature vector for each expression by applying some weight to individual feature points in those expressions individually. Overall, facial expression and hand gesture recognition technology has the potential to revolutionize human-computer interaction and enable a wide range of innovative applications across different industries, offering more intuitive, natural, and accessible ways to interact with technology.

REFERENCES

[1] P. Wilson, T. Miyaki and J. Rekimoto.
Facial Feature detection using Haar classifiers
DOI=http://dl.acm.org/citation.cfm?doid=36652.112716

[2] Images Source
Japanese Female Facial Expression Images Database
DOI=Source Image Database

[3] J. Raheja and U. Kumar.
Human Facial Expression Detection From Detected In Captured Image Using Back Propagation Neural Network
DOI=http://airccse.org/journal/jcsit/0210ijcsit10.pdf

[4] J. Wang and E. Sung.
Frontal-view Face Detection and Facial Feature Extraction using Color and Morphological Operation
DOI=http://dx.doi.org/10.1016/S0167-86(9072-0

[5] P. Michel and R. Kaliouby
Real Time Facial Expression Recognition in Video using Support Vector Machines
DOI=http://cmu/pub/Michel-RTFaceSVM.pdf

[6] H. Sako and A. Smith
Real-Time Facial Expression Recognition based on Features' Positions and Dimensions
DOI=http://dl.acm.org/doi=839288.842203

[7] P. Yi, Z. Hui and Z. Yu
Real-time Facial Expression Recognition based on Adaptive Canny Operator Edge Detection
DOI=http://ieeexplore.ieee.jsp?arno=5474348

[8] Matlab Computer Vision System Toolbox
http://www.mathworks.com/detector-class.html[1]

[9] Face Recognition using Eigen Faces
M. Turk and A. Pentland
http://www.cs.ucsb.edu/CVPR91.pdf

[10] Face Recognition using Eigen Faces
M. Turk and A. Pentland
http://www.cs.ucsb.edu/CVPR91.pdf

[11] Gesture Recognition and Recommendations
T. Padliya
https://doi.org/10.13140/RG.2.2.22522.30406

[12] Eigen Face
http://en.wikipedia.org/wiki/Eigenface

[13] Haar like Features
http://en.wikipedia.org/wiki/Haar-like_features

[14] Face Recognition using Gabor Wavelets
V. Kumar and G. Murthy
http://vbettadapura.com/files/vtReport.pdf

[15] Gabor Wavelet
http://en.wikipedia.org/wiki/Gaborvelet

[16] Gabor WaveletHand Gesture Recognition for Patients Monitoring
K. Muthumeena, S. Santhiya S
https://jespublication.com/upload/2020-110748.pdf