Facial Emotion Recognition Project Report

Feiyi Ouyang 5253190

1 Introduction

If you open up the photo library of one's smartphone, you won't be surprised that faces comprise the majority of individual image databases. Indeed, faces are probably the most frequent object on social medias like Facebook. Facing with such a big database, how to recognize faces automatically has always been a hot topic in computer science.

Face recognition, the strategy to classify and identify faces, has been widely used in various applications, especially for safety and security concerns. Police, for example, are using face recognition systems to identify criminals. Using faces to log in may become the mainstream of authentification in the future, concerning its accuracy, convenience, and safety.

Facial expression recognition, a subfield of face recognition, is gaining more and more attention. With a growing number of robots living together with humans, it is important to teach machines to recognize human emotions so as to facilitate human-computer interaction. In addition, recognizing facial expressions has been widely used in diagnosis and study of psychological diseases, like autism whose facial expressions are rigid and simple compared to normal people.With these backgrounds, our project aims to develop a program to recognize human facial expression.

2 Background and Related Work

Like recognition of other objects, the key challenge in face recognition is the invariance issue. Images including faces vary in terms of external factors like illuminance and pose [5]. In some cases, the external variances are even larger than internal variances, that is, variances of faces features [3]. Thus. research in the area typically restrains the input images. One way is to put constraints on input images regarding their poses, illuminance, etc; the other to preprocess the image, like equalizing histograms, to reduce variances caused by environments.

When it comes to classifying facial expressions, there are two basic ideas to extract facial features, locally and globally. One is to identify local features such as distances between ears, eyes, and mouths. To obtain local features, one way is to detect and compare the contours of facial areas like eyes, mouths, noses [7]. The idea is consistent with the Action Unit (AU) idea of Facial Actions Coding System (FACS), which has been widely used in psychological studies to recognize human emotions. However, extracting features from facial parts proves to be difficult from our experiment as the edges of separate facial parts are vague, therefore hard to extract the exact shape, which is exactly what we plan to rely on to classify facial expressions. The other way to represent local features is using SIFT features. However, the problem is that good SIFT features of different persons can hardly be matched. To solve the problem, the dense SIFT (DSIFT) algorithm was proposed, which fixs the feature locations over the whole image at which SIFT features are described.

Another direction of research analyzes faces as a whole, detecting the *patterns* of different facial expressions. The most straightforward idea is to calculate pixel correlations in the image space. However, this approach uses raw data with high dimensions, therefore is low in efficiency and speed. Consequently, dimension reduction methods, like principle component analysis (PCA), are widely used in this area. Slightly different in the constraints of projection from high dimensional image space to low dimensional subspace, Fisherface and Eigenface algorithms are good representatives of this idea [2,6].

To sum up, out of our literature research and initial trials, we focused on three potential approaches: DSIFT, Eigenface, and Fisherface.

3 Relavant Approaches

3.1 DSIFT: paper [1]

- Problem Specification: When basic SIFT feature detectors are applied to the entire face, the number and locations of detected keypoints change with illuminance, occlusion, and poses, making it hard to detect facial emotions. The paper aims to overcome the drawback by using Deformable Parts Model (DPM) algorithm to detect ROIs and Dense Scale Invariant Feature Transform (D-SIFT) algorithm to extract features.
- Preliminaries:
  - Viola-Jones framework
    To detect faces out of images, the framework applies five Haar-like patterns including two edge features, tow line features and four rectangle features to images and adopts Adaptive Boost Learning algorithm to select features and train the framework.
  - DPM
    DPM assumes an object is constructed by its parts. Thus, the detector will first found a match of its whole, and then use its part models to fine-tune the result. For each location, a score is assigned, so that a varied set of objects or multiple instances of the same object can be identified. The ultimate score for each root location is calculated as:

    $$\sum_{i=0}^{n} F_i \phi(H, p_i) + \sum_{i=1}^{n} a_i(x_i, y_i) + b_i(x_i^2, y_i^2)$$

    where F_i represents the filter for the i'th part of the image, $\phi(H, p_i)$ represents HOG features in sub-window specified by the location of i'th part p_i, a_i and b_i are two-dimensional deformation parameters specifying deformation costs.
  - D-SIFT features
    Obtained by extracting SIFT features at regular points and calculating orientation descriptors, so that key points do not depend on matching.
  - SVM classifier
    SVM can either be used in identification, to return the identity based on the best estimate, or verification, to give positive or negative responses to the input identity associated with the input image. In this paper, the former application of SVM was used, and different SVMs were used to identify different facial parts. The resulting values were conditionally interpreted to find the identity.
- Approach: The authors first used the Viola-Jones framework to detect faces. After that, ROIs were identified by DPM. Then D-SIFT features were extracted. Lastly, SVM classifier was used with inputs of D-SIFT features. The algorithm was tested on Caltech, MIT India, and LFW for face recognition purposes,

and the accuracies were 88.5%, 89.3%, 82.6%, respectively. The authors also found that the proposed model outperformed Dense SIFT along, and SIFT algorithm in face recognition.

- Critiques: The paper provides a novel idea to extract and classify facial features from a "local feature" perspective, overcoming the limitations of traditional local feature methods which focused on detecting exact shapes of facial parts. It can be compared with "global pattern" methods to test which one achieves better performance.

3.2 Eigenface and Fisherface 1: paper [2]

- Problem Specification: To develop a face recognition algorithm to identify faces insensitive to large variations in lighting and facial expression
- Preliminaries
  - Projection of Lambertian surface: One important observation is that all of the images of a Lambertian surface, faken from a fixed viewpoint, but under varying illumintion, lie in a 3D linear subspace of the high-dimensional image space. Thus, projecting faces from the high-dimensional image space to a significantly lower dimensional feature space can reduce variation in lighting direction and facial expression while maintain discriminability.
  - FisherFace VS EigenFace The projection directions of FisherFace method are nearly orthogonal to the within-class scatter. Using Fisher's Linear Discriminant (FLD), the method maximizes the ratio of between-class scatter to that of within-class scatter.
    The EigenFace method uses principle components analysis for dimensionaility reductionm yielding projection directions that maximize the total scatter across all classes, thus retaining unwanted variations due to lighting and facial expressions.

- Approach: The authors compared several methods for face recognition under variation using both a subset of the Harward Database and a database created at Yale. The details of methods are listed as follows:

  1. Correlation
     Normalize images to have zero mean and unit variance, then assign the image to the label of the closest point in the learning set where distances are measured by the correlation of images. The normalization process makes the method independent of light source intensity, but calculating correlation is computatively expensive and requires large amounts of data to train.
  2. EigenFaces
     Suppose the k^{th} image is a vector represented by $\mathbf{x}_k$, the feature vector $\mathbf{y}_k$ is defined by the following linear transformation:
     $$\mathbf{y}_k = \mathbf{W}^T \mathbf{x}_k$$
     The total scatter matrix S_T is defined as
     $$S_T = \sum_{k=1}^{N} (\mathbf{x}_k - \mu)(\mathbf{x}_k - \mu)^T$$
     The total scatter matrix of projected samples is $W^T S_T W$, and W_{opt} is chosen to maximize the determinant of the matrix:
     $$W_{opt} = arg \max_W |W^T S_T W| = [\mathbf{w}_1 \, \mathbf{w}_2 ... \mathbf{w}_m]$$
     where $\mathbf{w}_i$ is n-dimensional eigen vectors of S_T corresponding to the m largest eigen values.
     The drawback of the method, as said before, is that it not only maximizes the between-class scatter but also within-class scatter.

3. FisherFaces
   The idea is to use class specific linear methods like FLD for dimensionality reduction. It selects W in such a way that the ratio of between-class scatter $S_B = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T$ and within-class scatter $S_W = \sum_{i=1}^{c} \sum_{x_k \in X_i} N_i(\mathbf{x}_k - \mu_i)(\mathbf{x}_k - \mu_i)^T$, where μ_i is the mean image of class X_i and N_i is the sample number of class X_i. Simillarly, the W_{opt} is chosen to maximize the ratio of the determinant of the between-class matrix and within-class matrix:

$$W_{opt} = arg \max_{W} \frac{|W^T S_B W|}{|W^T S_W W|} = [\mathbf{w}_1 \, \mathbf{w}_2 ... \mathbf{w}_m]$$

   where $\mathbf{w}_i$ is generalized eigen vectors of S_B and S_W corresponding to the m largest generalized eigen valies λ_i:

$$S_B \mathbf{w}_i = \lambda_i S_W \mathbf{w}_i$$

   corresponding to the m largest eigen values.

- Critiques: The paper found that the error rate of identifying faces in various illuminance and facial expressions was smallest for the FisherFace method. Note that in this paper, the images were classified according to identity, and deviation in facial expressions is "unwanted variance". In our task, however, faces will be labeled according to facial expressions with other sources of varinces as "unwanted". Thus, the methods introduced in this paper should be suitable for emotion recognition tasks.

3.3 Eigenface and Fisherface 2: paper [5]

- Problem Specification: To develop robust facial emotion recognition algorithm, further, to develop an emotion recognition system.
- Preliminaries: Similar to those in paper [2].
- Approach: In the paper, the authors first saved training images along with their labels in MATLAB matrix formats. The images were frontal face images with the same backgrounds as well as constraints like without glasses, without facial hair. In the pre-processing stage, the images were converted to grayscale to get rid of hue and diffusion information and were resized to the same length and width in pixels. Then the eigenfaces were generated following steps:
  - Find the mean of training matrixes
  - Subtract the mean from training matrixes
  - Obtain the covariance matrix
  - Calculate eigenvectors and eigenvalues
  - Choose the eigenvectors that have the highest eigenvalues as the eigenfaces

  The recognization is implemented in the following steps:
  - Find feature vector of the test image
  - Calculate Euclidean Distances between weight vectors with training image's feature vectors
  - Classify the test image as the same category with the closest training image
  - Choose a threshold. If the smallest distance is less than the threshold, the test image represents the same face as the training image; if the smallest distance is larger than the threshold, the test image represents an unknown face.

- Critiques: Testing on JAFFE (Japanese Female Facial Expressions) dataset, the prototype achieves an accurate rate over 90%. Looking closely, we found that happy, anger, and sad faces were best recognized with accuracy of 100%, and surprise faces were worst recognized with a mere accuracy of 75%
  One thing to mention is that the training images and test images are from the same species, in this case, Asian. However, people from different species differ in facial expressions even for the same emotion. Thus, it is worthwhile to train and test facial expression of people from different species.

4 Our Approaches

We found the Karolinska Directed Emotional Faces (KDEF) database online. The set contains 70 individuals wearing T-shirts with uniform colors, each displaying 7 different emotional expressions, each expression being photographed from 5 different angles. Each image was taken under a soft, even light, using of a grid to center participants' faces to position eyes and mouths in fixed image coordinates [4].

We tried 3 approaches to achieve the goal (refer to the supporting material for the specific task assignment):

1. Extracting contour properties

   Inspired by the FACS idea, we planned to extract features of contours of eyes, mouths, noses and classify emotions according to the AU table [7] at the initial stage. We used cascades and then canny edge detectors in OpenCV to extract contours, but soon found it impractical. Different from objects like buildings with sharp, straight, and clear contours, contours of eyes, lips, and noses are smooth, continuous, and multi-layered, thus, difficult to be extracted. For example, in Figure 1, even though with the best threshold parameter, the extracted contour of the lip was incomplete. Looking close, it is clear that the edge of the lower lip is very vague, which causes much trouble for the gradient-based algorithm to detect. We also tried the snake algorithm, but little consistency was achieved.

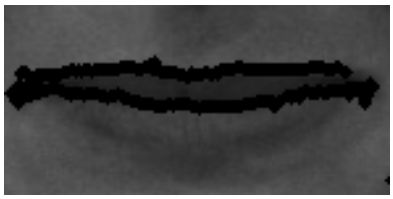

Figure 1: The best contour extracted for the mouth area

2. Extracting DSIFT features of facial parts

   We first tried a simple DSIFT idea, applying the DSIFT features over the whole image and passing these DSIFT descriptors as inputs to KNN. The method, however, didn't work well. Obviously, although it reduced the dimension of the original image space, 128 (the dimension of one SIFT descriptor) X 100 (number of SIFT features) = 12800 features were still needed to represent an image, which was much larger than the number of samples that we could provide.

   To further reduce the dimension, we first segmented facial parts from face images [2]. To do this, we applied trained cascade classifiers in OpenCV. Although we found specific cascades for different facial parts like mouth cascade, eyes cascade, nose cascade, etc, our experiment showed that only mouth cascade worked well, which detected eyes and mouths together. This is not surprising as mouths and eyes share the same oval shape, so we took the 3 detected mouth areas as right eyes, left eyes, and mouths, respectively (Figure 2). To improve the detector performance, we added size constraints as well as some location constraints

(like to restrain the upper and lower bounds and relative locations) to the cascade detector based on our observation. Additionally, after writing the detected facial parts to image files, we manually deleted the incorrect results detected, like removing mouths from eyes.

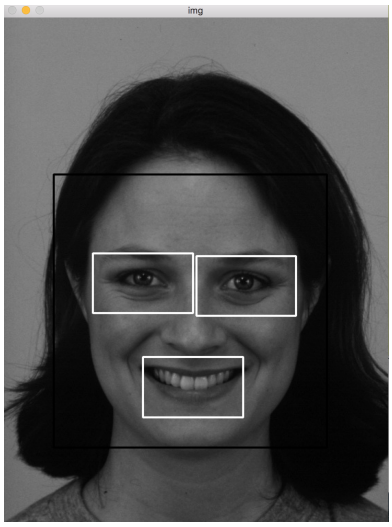

Figure 2: Facial parts detected by the mouth cascade

Next, we applied DSIFT algorithms to the facial parts. Before that, we preprocessed images to improve the contrast by histogram equalization and to reduce the effects of illuminance by normalization. We then use the DSIFT features to represent each facial parts and trained three SVM classifier for the three parts, respectively. We used 5 emotions (Happy, Surprise, Angry, Disgusting, Sad) from 119 people as the training set (5*119=595 samples), and used the 5 emotions from 16 people as a testing dataset. We achieved correct rates of 23.8%, 23.3%, 23.7% for the right eye, left eye, and mouth classifiers.

The results are not satisfying and we thought it could be caused by the following reasons:

(a) The extracted facial parts are not enough to represent the information of facial expression. Although eyes and mouths are probably the most important source of expression judgment, information from these individual parts alone may not be enough. Other feature like the nose part can also be important in analyzing facial expressions.

(b) We didn't take the conditional probabilities into consideration. In the original paper, the authors mentioned that they judged the emotions from the results of three SVM predictions conditionally. However, as the correct rates of SVMs for individual facial parts are low, we don't think the conditional judgment will help a lot.

3. FishFace and EigenFace
We tested the FisherFace and EigenFace algorithms on combinations of emotion sets which included happy and neutral along with a combination of other emotions. We selected the set of happy, surprise, disgust, and neutral faces which achieved the highest accuracy. Specifically, we detected face areas on images using the face cascade detector and preprocessed face images as in the previous approach (resize, equalize, and normalize). Then we passed the face images as inputs to the fisher detector and the Eigen detector provided by OpenCV (we planned to rewrite the fisher detector and Eigen detector codes but we run out of time as we tried many implementation approaches). To test the accuracy systematically, we used 5-fold cross-validation, selecting one-fifth of the samples as test data and the other four-fifths as training data for 5 times. The results are listed as below:

Detector	1st fold	2nd fold	3rd fold	4th fold	5th fold	Average
FisherFace_OpenCV	87.5%	84.96%	92.04%	84.96%	77.48%	85.388%
EigenFace_OpenCV	83.93%	84.07%	88.5%	84.07%	73.87%	82.888%

To test the algorithm on more natural images, we trained the model with a data set comprised of preprocessed emotion images searched from Google. The accuracy was lower but still two times of the average level (25%):

Detector	1st fold	2nd fold	3rd fold	4th fold	5th fold	Average
FisherFace_OpenCV	52.0%	52.17%	69.57%	62.5%	62.5%	59.748%
EigenFace_OpenCV	44.0%	47.83%	56.52%	62.5%	58.33%	53.826%

Performance of our eigenface model:

Detector	Dataset	1st fold	2nd fold	3rd fold	4th fold	5th fold	Average
FisherFace_OpenCV	KDEF	87.5%	84.96%	92.04%	84.96%	77.48%	85.388%
EigenFace_OpenCV	KDEF	83.93%	84.07%	88.5%	84.07%	73.87%	82.888%
EigenFace_Rewrite	KDEF	77.0%	83.1%	81.4%	78.8%	79.5%	80.0%
FisherFace_OpenCV	GoogleImage	52.0%	52.17%	69.57%	62.5%	62.5%	59.748%
EigenFace_OpenCV	GoogleImage	44.0%	47.83%	56.52%	62.5%	58.33%	53.826%

Achieving good classification results, we took it a step further by building a video-based face recognition program. We extracted frames from cameras, preprocessed the image, and use the trained model to classify the face image shown to the camera. We found that the dynamic detector could well distinguish positive emotions (like happy) from negative emotions (sad, disgust, surprise), but it could not fine-tune to recognize the exact negative emotion. We also tried testing the robustness of the detecter. We found that the detector worked equally well for faces of different species (Caucasian faces an4d Asian faces). The detector was also found to be robust in environments with different illuminance. What's more, although not trained on images with different poses, the detecter succeeded in detecting faces with poses slightly obviating from the frontal pose. As all the training images are without glasses, the detector could not detect emotions of faces with glasses, either.

In a word, our algorithm achieves a high accuracy to classify emotions on the KDEF dataset. Although the accuracy is not as high when we apply the model to detect facial expressions in the video, it is functional to recognize human moods and robust across different conditions. A possible application of the program can be used for machines like chatting robots to push contents music, posts, news to users' lists in response to human moods. Future investigations could focus on developing algorithms to detect facial expressions in faces with occlusions and to improve performances on classifying facial expressions on the fly.

5 References

[1] Arun BP, Roy K, Ahamed T, Mohanraj V, Vaidehi V, Kumar R. Facial feature extraction and recognition using Deformable Parts Model and DSIFT. InSignal Processing, Communication and Networking (ICSCN), 2015 3rd International Conference on 2015 Mar 26 (pp. 1-6). IEEE.

[2] Belhumeur PN, Hespanha JP, Kriegman DJ. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Transactions on pattern analysis and machine intelligence. 1997 Jul;19(7):711-20.

[3] Huang GB, Mattar M, Berg T, Learned-Miller E. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. InWorkshop on faces in'Real-Life'Images: detection, alignment, and recognition 2008 Oct.

[4]Lundqvist D, Flykt A, hman A. The Karolinska Directed Emotional Faces (KDEF). CD ROM from Department of Clinical Neuroscience. Psychology section, Karolinska Institutet; 1998. ISBN 91-630-7164-9;.

[5] Tan X, Triggs B. Fusing Gabor and LBP feature sets for kernel-based face recognition. InInternational Workshop on Analysis and Modeling of Faces and Gestures 2007 Oct 20 (pp. 235-249). Springer Berlin Heidelberg.

[6] Thuseethan S, Kuhanesan S. Eigenface based recognition of emotion variant faces. Browser Download This Paper. 2016 Mar 22.

[7] Tian Y, Kanade T, Cohn JF. Facial expression recognition. InHandbook of face recognition 2011 (pp. 487-519). Springer London.

6 Supporting Material

The flowchart recording our exploration path: