Classroom Data Collection and Analysis using Computer Vision

Jiang Han
Department of Electrical Engineering
Stanford University

Abstract

This project aims to extract different information like faces, gender and emotion distribution from human beings in images or video stream. Based on those collected data we may be able to obtain some useful feedback or information, which can be valuable guidance on how we can improve the class education quality. In this project, a few computer vision topics were touched like face detection, gender classification and emotion analysis. Technical details tested include feature extraction strategies like Bag of Words, HOG, LBP, key-point detection, feature reduction etc. Machine learning classifier tested including Nave Bayes, KNN, Random Forest and SVM. Classifier parameters are tuned to achieve the best accuracy. The system is able to achieve accuracy of 0.8673 for gender classification and 0.5089 for emotion analysis (0.6073 when we prune particular class). Real life image and video stream tests also verified the validity and robustness of the system.

1. Introduction

Motivation of this project came from my personal experience. One time when I was taking a class at Stanford CEMEX Auditorium, I often saw a TA came to the second floor and count how many students presented at the class. Attendance was not strictly required for this class, so this data is only used by the instructor to have a better understanding of current instruction status. At that time, I thought it will be great if we can do this by simply taking a photo.

When I take CS231A this year, this old idea came to me so I immediately decided to do some related work. Instead of only counting number of students, I decided to get more interesting information like gender and emotion. By detecting faces and doing gender classification, we can have a rough number of students attendance and their gender distribution. Emotion analysis maybe more useful in analyzing the class quality. Later on from Section 4, it is shown that we are able to get the emotion distribution across time with a video stream. If some emotion like "Surprise" appears a lot in the distribution, the class instructions may not be as clear as it should be.

So far most camera system on mobile phone has embedded face detection algorithm, but they have not put all three topics including face detection, gender analysis and emotion analysis all together. Thus implementing such a demo system will be very exciting to me.

Good thing is there has already been a lot researches on the above three areas. As a classical topic in computer vision, a lot of approaches have been proposed for face detection. Authors from [1] [2] concluded there are four types of approaches for face detection: knowledge based method, feature based method, template matching approach and appearance based method. In particular, an outstanding face detection algorithm was proposed by Viola and Jones [3]. Viola-Jones algorithm applys adaboosting and cascading classifiers and has advantages like fast speed, suitable for scaling, which makes it as the embedded face detection algorithm by toolbox from OpenCV and Matlab. For gender classification, authors from [4] also roughly divide it as feature-based and appearance-based approaches. Later on, Lian etc. [5] used LBP feature with SVM and were able to achieve very high accuracy. Also, Baluja etc. [6] applied Adaboost classifier and able to achieve more than 93% accuracy. Similar to gender classification, emotion analysis is also a classifying problem, but with multiple classes instead of two, which increases difficulty in accuracy. Authors from [7] presents an approach using means of active appearance model to do both gender and emotion classification. SVM was used to label four emotion types (happy, angry, sad and neutral). Y Kim etc. [8] applied deep learning techniques to overcome the linear feature extraction limitation in emotion detection, which is able to boost performance. In all, gender classification and emotion analysis have been very hot topics in researches from both feature extraction and learning model optimization aspects [9].

In this report, Section 2 shows problem statement, which briefly describes the system framework. Section 3 is technical content, which introduces data set used, evaluation metric, and shows different approaches tried related feature and classifier selection. Multiple numerical simulation results are also shown in Section 3. Section 4 is experimen-

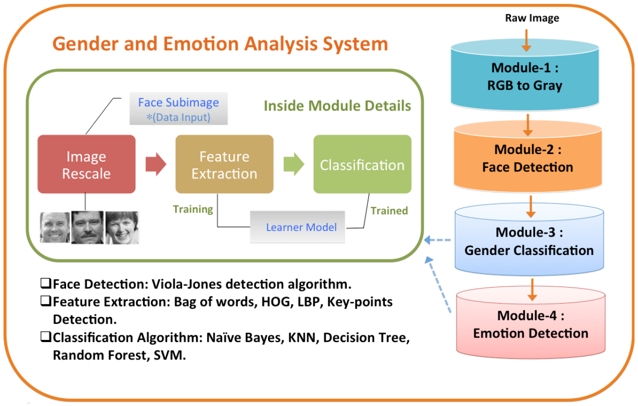
Raw Image
Gender and Emotion Analysis System
Module-1 : RGB to Gray
Face Subimage
*(Data Input)
Inside Module Details
Module-2 : Face Detection
Image Rescale
Feature Extraction
Classification
Training
Trained
Learner Model
Module-3 : Gender Classification
Module-4 : Emotion Detection
❑Face Detection: Viola-Jones detection algorithm.
❑Feature Extraction: Bag of words, HOG, LBP, Key-points Detection.
❑Classification Algorithm: Naïve Bayes, KNN, Decision Tree, Random Forest, SVM.

Figure 1. Framework of system.

tal setup and results, which analyzes on class F1 score, also shows the system performance on real life images and video stream. Finally, conclusions and future work are shown in Section 5.

2. Problem Statement

Main framework of system design is shown in Fig.1, which includes four modules to process the input image. Image is transformed to gray scale image at the first beginning since color information is not that important in this classification problem. Then face detection is applied for the image to locate all the human faces positions. Inside each face box, gender classification and emotion analysis engine works to generate corresponding labels. As Fig.1 shows, Module-3 and module-4 shares very similar inside core, which includes:

- Image rescale: The subimage inside face box needs to be rescaled for three reasons: (1) This is necessary and will make things much easier to generate consistent feature dimension later on. (2) The source image data set for training and testing of gender analysis was different in scale size. (3) Face box derived from Module-1 may be different in sizes due to different face size.

- Feature extraction: After rescaling, this step generates features with consistent dimension. Feature extraction can use algorithms like Histogram of Gradient (HOG), Local Binary Pattern (LBP), Bag of Words (BoW), etc.

- Model training/classification: Based on the feature vectors output from feature extraction, we are able to train the classifier with input image training data. Training step may take long time due to data size. But once the model is trained, we are able to use it to do classification on the testing image directly.

From Fig.1 we can see that each module may have multiple algorithm candidates to implement, while being able to design a system with good trade-off between performance and complexity is one of the project target.

Gender	Male	Female
Training	9,993	10,992
Testing	3,040	2,967

Table 1. Data set for gender classification

Emotion	Angry	Disgust	Fear	Happy
Training	3,995	436	4,097	7,215
Testing	958	111	1,024	1,774
Emotion	Sad	Surprise	Neutral	
Training	4,830	3,171	4,965	
Testing	1,247	831	1,233	

Table 2. Data set for emotion analysis

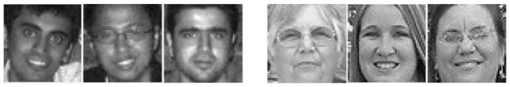

Figure 2. Samples from gender data set (Male: left three images. Female: right three images).

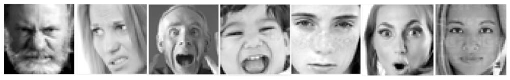

Figure 3. Samples from emotion data set (left to right: Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral).

3. Technical Content

3.1. Data set and initial analysis

The used data set for gender classification and emotion analysis is shown in Table 1 and 2.

The data set for gender classification is extracted partly from Image of Group (IoG) data set [10], the original data set includes more properties on each person like: face position, eye position, age, gender, pose. In this project, we only care about the age property. Thus I split the data into four folders: “Male_Training”, “Female_Training”, “Male_Testing” and “Female_Testing”. From Table I we can see that both male and female image number was roughly balanced to guarantee the best model training result. With this split, we can use Matlab command *imageSet* easily load corresponding images. And the total number of images used for both training and testing is 26,992. One thing I notice is that source image for genders is not scaled to the same size. This is one of the reasons why we add “Image Rescale” before the feature extraction step to make the input image 48 by 48 gray image. Fig.2 shows six gender sample images, which includes three male and

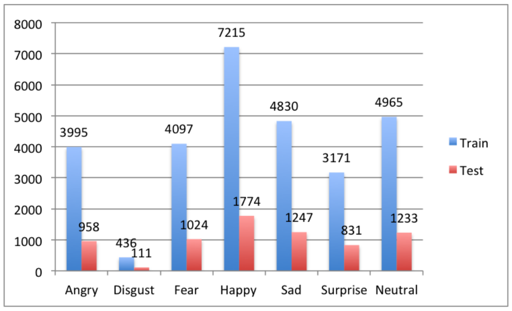

Figure 4. Bar chart of emotion data set.

three female. Also we note that this data set includes people from different races and different age.

Furthermore, Table 2 shows the data set for emotion analysis. The data set is from ICML [11], which has 7 categories of emotions including: Angry, Disgust, Fear, Happy, Sad, Surprise and Neutral (both for training and testing). Fig.3 shows samples of the seven types of emotions. We notice that this data set includes emotions from different gender and ages, which is good to train a robust model. However, emotion of human beings is very complicated and vague. For example the “Surprise” image of Fig.3 may also be treated as “Angry” in reality. Furthermore, Fig.4 shows the bar chart of different emotion types image number (including both training and testing). Notice that this data set is mostly balanced, except that “Disgust” type has significantly lower number than other categories, which explains why “Disgust” class has the lowest F1 score in later testing.

3.2. Evaluation metric

To evaluate the performance of classification, accuracy (ACC) [12] is sued as the metric, which defines as:

$$ACC = (TP + TN)/(P + N) \tag{1}$$

Here, P and N are the total number of positive and negative samples. TP and TN are the true positive and true negative samples number. Therefore, ACC value of 100% indicates that the system is able to predict labels exactly same with the ground truth values. In this project, ACC is used to evaluate different feature extraction and machine learning algorithms.

In addition, I also used precision, recall and F1 [13] to do the analysis on each type of the prediction classes, which is shown in Section 4.

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{2}$$

Here, $precision$ is defined as true positive (tp) over tp plus false positive (fp). $recall$ is defined as tp over tp plus false negative (fn).

3.3. Face Detection

For face detection, initially I was using the similar method from our problem set 3 with HOG + SVM + sliding window. I also tried the following dynamic boxing method to adjust window size to fit the face scaling.

$$size = minSize + 2^N \times step \tag{3}$$

Here, size value is with constraint of $size \leq maxSize$. N is the N-th time of window expanding, $step$ is a value controls the window expanding speed. The advantage of this strategy is that from $minSize$ to $maxSize$, we at most need $N = \lceil log_2 \frac{maxSize - minSize}{step} \rceil$ times of expanding. And we are giving smaller expanding speed when the window size is still small. Hence, instead of doing linear time of expanding and apply SVM at each position, here we have cost at $O(log)$ level, eventually we choose the face window size with the biggest prediction score (only if this score is bigger than SVM threshold).

However the test shows that sliding window scheme is quite slow because we need to run SVM multiple times. Considering face detection is not the main part of this project, I turned to use the MATLAB embedded $vision.CascadeObjectDetector()$ to do the face detection, which is using Viola Jones object detection framework. Viola Jones algorithm is much faster and good at detecting scaled faces [3].

3.4. Gender Classification and Emotion Analysis

I put gender classification and emotion analysis in the same subsection since from Fig.1, we see that the two modules share very similar inside blocks. Therefore in this subsection, I’m going to introduce the following three blocks: image rescale, feature extraction and training/classification.

3.4.1 Image rescale

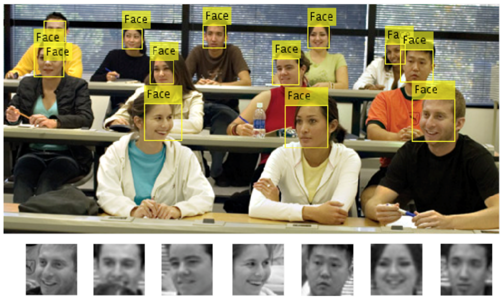

Figure 5. Face rescale example on testing image (not all faces are shown here).

Image scale is necessary for both training and classification. For the training set, each image of emotion analysis was originally given with 48 by 48 gray scale, which is ok to use directly. But for gender classification data, it was come with RGB images with different scale size. Thus rescale is necessary to make the training images into fixed size and gray scale. This will make much easier in the feature extraction step to obtain feature vector with consistent dimension.

For the classification step, there may be multiple faces marked from the original image, and each comes with different box size. Thus we do RGB to gray scale and rescale to 48 by 48 before we apply classifier. Fig.5 shows an example of classification rescale inside box.

3.4.2 Feature extraction (BoW)

Note in order to save space, for feature extraction part from 3.4.2 to 3.4.5, the table ACC values are based on testing of gender classification using SVM. However, emotion analysis data also gives similar conclusion.

For the feature extraction, I started with bag of words. Matlab provides some embedded functions like "bagOfFeatures", "trainImageCategoryClassifier", "imageCategoryClassifier", etc. to be used. The default feature vector is based on SURF, I also tried dense SIFT features. Based on the extracted feature vector from each images, K-means is applied to the feature space with entire training images. Here, K defines the vocabulary size for the histogram. Eventually the image feature is defined as a histogram which defines the nearest cluster center distribution for every image.

Table 3 shows the ACC of gender classification with both SURF and dense SIFT features. From later on ACC of SVM we can see that this performance is even slightly worse than Naive Bayes. It seems reasonable to me since the testing objects are all faces with same structures (eyes, nose, mouth etc.). Clustering those features may lose some details. The scenario of gender classification and emotion analysis is different from scenarios where bag of words are most used (for example object classification like cups, ships, etc.). In addition, BoW is giving me very slow training speed with around 3 million feature vectors to e clustered. Thus BoW was not selected after testing.

Feature	SURF	Dense SIFT
BoW Test ACC	0.7137	0.7326

Table 3. BoW ACC performance of gender classification on SURF and dense SIFT features (vocabulary size is 300).

3.4.3 Feature extraction (HOG and LBP)

HOG and LBP were tested after BoW. HOG is a very well-know feature descriptor in computer vision [14], which accumulates local gradient information . There are several parameters we can tune for HOG, like "Cell Size", "Block Size", "Block Overlap", "Number of Bins". In my test, I kept default value of "Block Overlap" and "Number of Bins" since that is the typical settings. "Cell Size" and "Block Size" will be more important parameters which can control feature vector size and testing performance. Here, "Cell Size" defines the box to calculate histogram of gradients. Smaller cell size will give us better chance to catch small-scale details. On the other hand, increasing the cell size will be able to capture large-scale spatial information. "Block Size" defines number of cells inside the block, smaller block size may reduce the influence due to illumination changes of HOG features [15].

Paras	Cell:8,8. Block:2,2	Cell:4,4. Block:2,2
Feature Size	900	4356
HOG ACC	0.8369	0.8500
Paras	Cell:8,8. Block:3,3	Cell:16,16. Block:2,2
Feature Size	1296	144
HOG ACC	0.8337	0.7718

Table 4. Gender classification HOG ACC performance with various cell/block settings.

Paras	Cell:8,8.	Cell:12,12.	Cell:16,16.	Cell:24,24
Size	2124	944	531	236
ACC	0.8638	0.8390	0.8274	0.7864

Table 5. Gender classification LBP ACC performance with various cell settings.

In addition to HOG, LBP is another feature I found out to be very useful, the performance is no worse than HOG. The principle of LBP is different from HOG, instead of using gradient information. LBP compares the pixels value with its neighbors, and based on the binary comparison result to construct the histogram. LBP can be easily extended to rotation invariant version [16].

Table 4 and Table 5 shows the HOG and LBP testing ACC with different feature dimension of gender classification (emotion analysis data testing result is having similar trend, thus not listed here). By setting cell/block size we are able to obtain different feature dimension. Unsurprisingly higher feature dimension is able to give better ACC but may also slow down the system significantly due to increased complexity for machine learning models.

3.4.4 HOG and LBP feature combination

To get the best trade-off between performance and speed, and based on the research fact that combination of HOG and LBP features is able to improve the detection performance [17]. I joined HOG feature (Cell:8,8, Block: 2,2, dimension of 900) together with LBP feature (Cell: 12,12, dimension of 944) to boost ACC. Table 6 shows the combination feature result.

Paras	HOG	LBP	Joined
Feature size	900	944	1844
Test ACC	0.8369	0.8390	0.8673

Table 6. LBP and HOG feature combination result.

Table 6 shows the ACC performance of combination between HOG and LBP. From which we can see that the original feature dimension for HOG and LBP were both around 900, ACC performance were between 0.83 - 0.84. By joining HOG and LBP together, we are able to get significant 0.03 ACC boosting. Even though we have doubled feature dimension after combination, but this performance is still higher than HOG or LBP alone with similar size. Because LBP and HOG is using different principles to construct features, this kind of combination is able to get diversity gain.

3.4.5 Feature dimension reduction

Consider feature size is important to system speed, a reasonable prune or feature dimension reduction will be very helpful. Especially when we use real-time system, we would rather lose small performance to have more smooth experience for users.

The way I did to reduce feature dimension was to only extract HOG/LBP features from areas around key-points. Initially I tried several famous key-point detection methods as following:

Harris: detects corners using HarrisStephens algorithm.

SURF: detect blobs using Speeded-Up Robust Features.

MSER: detect regions using Maximally Stable Extremal Regions.

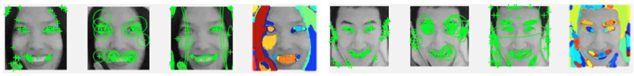

Figure 6. Key-point detection result (each person left to right: Harris, SURF, MSER, MSER Region)

Fig.6 shows key-point detection result using Harris, SURF, MSER. Those detection method will return different number of points for different images. Since we are not using BoW, we need to construct a consistent feature dimension. I tried different ways to do this like doing K-means, or select strongest K points out of N key-points. However, the testing result gives significant ACC loss comparing to sliding window scheme.

Therefore, instead of using corner or blob key-point detections, which returns mostly different physical positions in the image. I used a fixed key-point feature extraction, which extracts fixed number of key-points on face. It turns out we are able to use this method reduces feature dimension with small performance loss.

Specifically, we use Matlab CascadeObjectDetector system object to detect nose on the face. If object returns the nose position successfully, we select K points evenly round the circle (with predefined radius) with nose as the center. If the nose is not detected (CascadeObjectDetector may fail with the 48 by 48 low resolution image), we simply use the center of image as the circle center. Here, K can be selected with different values to get the best trade-off between ACC performance and complexity.

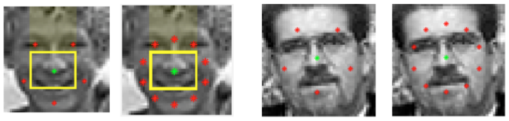

Figure 7. Circle key-point detection (circle center as nose or image center, K = 5 and K = 10).

In Fig.7, we show the results of circle based key-point detection. Matlab CascadeObjectDetector is able to return nose position on left two images, but failed on two images on the right side (under which situation we use image center directly). Also, $K = 5$ and $K = 10$ detection are shown here. After this, we can only extract HOG/LBP features around those key-points.

K	5	10	15	20
Feature size	20%	40%	60%	80%
ACC Loss	5.6%	2.68%	1.41%	1.16%

Table 7. Key-point based feature reduction performance.

Table 7 shows the testing result with different K values. We can see that with only 40% of feature dimension, we are only losing 2.68% of the ACC performance. For real time systems, people may would rather to satisfy this 2.68% ACC to get more smooth using experience.

3.4.6 Model Training and Classification

After feature extraction method is selected, we can now test on different machine learning classifiers. In this project, I tried different models including: Naive Bayes (NB), K Nearest Neighbors (KNN), Random Forest and Support Vector Machine (SVM). Also note that to get better system speed, I used same features for gender classification and emotion analysis. However, we need to choose most suitable learner for gender and emotion classification.

Naive Bayes: NB method is based on Bayes rule to calculate the probability of each classes. NB also naively assumes the independence of each features.

K Nearest Neighbors: KNN is taking the majority of labels from the K number of nearest neighbors. KNN distance

	Gender Classification	Emotion Analysis
Test ACC	0.7495	0.3589

Table 8. Naive Bayes ACC performance.

Neighbor Number	1	5	10	20
Gender-ACC	0.7446	0.8017	0.8062	0.8190
Emotion-ACC	0.5306	0.4820	0.4727	0.4583

Table 9. KNN ACC performance.

Tree Number	20	60	100	300
Gender-ACC	0.7696	0.8022	0.8102	0.8235
Emotion-ACC	0.4388	0.4859	0.4965	0.5074

Table 10. Random Forest ACC performance.

	C=0.0008	C=0.01	RBF	Gaussian
Gender-ACC	0.8673	0.8608	0.4939	0.4950
Emotion-ACC	0.5089	0.5022	0.2064	0.2017

Table 11. SVM ACC performance.

Pruned	0	1	2	3
SVM-ACC	0.5672	0.5205	0.5911	0.4752
Pruned	4	5	6	
SVM-ACC (s)	0.6073	0.5207	0.5558	

Table 12. SVM ACC with class selection.

Classifier	NB	KNN	Random Forest	SVM
Gender Time (s)	1.71	63.3	57.1	170.8
Emotion Time (s)	4.12	103.2	231.2	457.2

Table 13. Time cost for data set training.

can be calculated with different metric like Euclidean distance, Hamming distance, etc. (K value can be tuned for KNN).

Random Forest: random forest is an ensemble model based on decision tree, where decision tree trains and tests based on attribute split, and label with leaf node. (Tree number can be tuned.)

Support Vector Machine: well-know method to split samples with minimum distance maximized. Matlab also provides ClassificationECOC classifier to support multi-class classification with SVM. (C value can be tuned, which controls overfitting, different kernels can be tried.)

To select the best model, we need to run and tune each of the classifiers. Note that for random guess, gender classification will have 50% ACC, and emotion analysis will have 14.28% ACC (7 classes in total).

Table 8 shows NB ACC testing results of both gender classification and emotion analysis. The advantage of NB is that it's running super fast, which is the fastest model among all models, but the ACC is poor with only 0.7495 for gender and 0.3589 for emotion. Note that gender has higher ACC since it only has two labels while emotion has 7 labels.

Table 9 shows the testing result of KNN with different K values. For gender classification, we can see that when $K = 20$, we get gender ACC of 0.8190. But the boost from $K = 10$ to $K = 20$ is relatively small, meaning that the neighbors ranked 10 to 20 are contributing limited influence. However for emotion analysis, ACC is the best when $K = 1$, and performance is reducing significantly when we increase K value. This means that for emotion analysis data, neighbors outside the first 1 are introducing more noise than positive contributions. Note that comparing with NB method, by using KNN, we are able to boost gender ACC from 0.7495 to 0.8190, and emotion ACC from 0.3589 to 0.5312.

Table 10 shows the testing result of random forest, I tested on different tree size. Here, we can see that gender ACC is increasing from 0.7696 all the way to 0.8235 when tree number is 300. Emotion ACC increases from 0.4388 to 0.5074. Also we notice that from tree size of 100 to 300 the improvement is relatively small, which means the model has almost converged. It has been proved that random forest is able to prevent over-fitting, thus bigger number of tree size should converge to some value. A reasonable tree size should be chosen to get the best trade-off between performance and complexity.

Table 11 shows ACC performance of SVM. It turns out that SVM is having less improvement in emotion analysis than gender classification, this mainly because emotion analysis itself is not a binary classification problem as gender. Also, parameter C value seems not influencing ACC that much. I also tried different kernels like RBF and Gaussian. In both gender and emotion problem, RBF and Gaussian kernel are performing very badly, which definitely not a good kernel choice. Considering there are some overlaps/similarity between different emotions, and some emotion type may have negative influence, i.e., cause some false positive to other emotions. I also tested on pruning class labels. Table 12 shows the SVM ACC result when pruning different emotions. We can see that with specific prune, we are able to boost ACC to more than 0.60.

Table 13 shows the training time cost for each models, which indicates the following training time relation: NB < KNN < Random Forest < SVM. However, longer train time does not necessarily indicates longer test time.

4. Experimental Setup and Results

Note in section 3, I already showed the majority of numerical testing results (like different feature performance, different classifier performance) in multiple tables. In this section, we will shown some experiment results in addition to that.

The simulation tool used for this project was mainly Matlab. The total .m files number is around 20. I also used JAVA and Python for some data/image parsing. I have several main functions to test on BoW features, data processing, gender detection, emotion analysis, etc. Also other helper functions to do feature extraction, classification, etc. Vlfeat tool was also used in order to test on dense SIFT feature.

For convenience, I parsed and split images into the following format:

```
emotion-train/test
 |__0-Angry
 |__1-Disgust
 |__2-Fear
 |__3-Happy
 |__4-Sad
 |__5-Surprise
 |__6-Neutral
gender-train/test
 |__Male
 |__Female
```

Here, each of the classes of emotion analysis and gender classification has its own corresponding folder, which makes very easy to load images using Matlab command imageSet.

4.1. Class F1 score analysis

In Section 3, we already showed majority of numerical results including test on feature extraction, feature reduction, different classifier models etc. In this part, I'm going to show how robust the system is on each type of classes. Instead of using accuracy, precision, recall and F1 score are used (definition can be found in Section 3.2).

Table 14 shows the precision, recall and F1 of gender classification. We can see that both of male and female type have very close performance on the three metrics. Thus we can conclude the system has no bias, and will have very similar good performance on male and female.

Table 15 shows the precision, recall and F1 of emotion analysis. Different from table 14, here we notice each of the class is highly biased. Among them, "Happy" and "Surprise" have the highest F1 score, indicating those two types of emotions will have the best performance in real life testing. Some emotions have relatively low F1, like "Angry", "Fear" and "Sad", but this is understandable since those emotions were essentially kind of vague. As shown later on in section 4.2 and 4.3, we can tolerate some overlapping among them. In addition, "Disgust" has the lowest F1 score, this is also reasonable since we have much less image training data for "Disgust" emotion (as shown in Fig.4 and Table 2). Note it also has very high precision of 0.9091 and very low recall of 0.1802, which meas it may be difficult for the system to retrieve "Disgust" from testing image, but once it is marked, with 90.91% probability that will be correct.

Gender	Male	Female
Precision	0.8745	0.8602
Recall	0.8615	0.8733
F1	0.8679	0.8667

Table 14. Precision, recall and F1 of gender classification.

Emotion	Angry	Disgust	Fear	Happy
Precision	0.4004	0.9091	0.3730	0.6789
Recall	0.3779	0.1802	0.2754	0.7627
F1	0.3888	0.3008	0.3169	0.7183
Emotion	Sad	Surprise	Neutral	
Precision	0.3719	0.6982	0.4405	
Recall	0.3841	0.6041	0.5345	
F1	0.3779	0.6477	0.4830	

Table 15. Precision, recall and F1 of emotion analysis.

However, those numerical results are tested on the testing data set, whose resolution was intended to be low and face expression was very complex. My feeling when testing on real life image or video stream is that, the system is working far better than the numerical performance on testing data set (shown in section 4.2 and 4.3).

4.2. Real Life Image Test

The numerical ACC results from section 3 should already be enough to verify the correctness of the system. But to have a more straightforward view of the performance. This subsection shows some test result on images and videos.

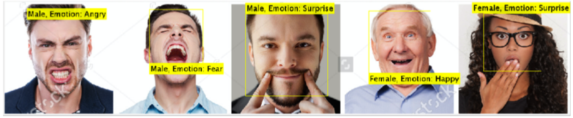

Figure 8. Emotion image-1 testing result.

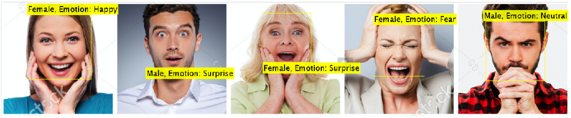

Figure 9. Emotion image-2 testing result.

In order to test the emotion classification result, I found several images (Fig.8 and Fig.9) [18] with human faces of various expressions. Also, the image includes people with different races and ages. Note the five faces of Fig.8 and Fig.9 were passed to the system inside one image. Hence, corresponding back to Fig.1, the process will be:

- Face detection module circles out five face box areas.
- Rescale image inside face box. Then feature extraction generates feature vectors for each face with consistent dimension.
- Run gender classifier with the provided feature.
- Run emotion classifier with the provided feature.
- Add face box, gender and emotion label to image.

From Fig.8 and Fig. 9 we can see that we actually have a very good test result on both gender and emotion classification. For gender classification, only one of the 10 faces had error. For emotion analysis, we marked five different emotions labels, which are: Angry, Fear, Surprise, Happy and Neutral. I would say almost all the emotion classification results look reasonable to me. However, human being's face expression is very complicated. It's even vague for us sometimes to judge other's emotion through face expression. For example, the third face on Fig.9 could be explained as either “Surprise” or “Happy” by different people.

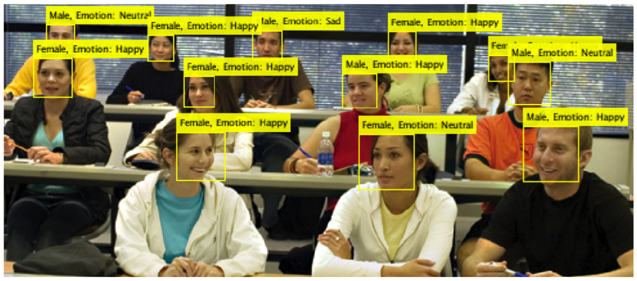

Figure 10. Classroom image testing result.

In addition, Fig.10 shows an example image result of classroom students (original image from [19]). Fig.10 includes scaling and rotation. Different students may have different face size in the image, depending on their distance to the camera in 3D coordinates. This issue can be handled by Viola-Jones detection algorithm. But we do lose some details of the face with smaller face scale. Another issue is rotation, we notice that different from Fig.8/9, where everyone was looking at the camera center directly. None of the people in Fig.10 is looking at the camera, we also have several rotated faces. However, there were also a few training data comes with rotation, thus We are able to do the correct prediction in Fig.10. Almost all the emotion prediction in Fig.10 seems reasonable to me, except one person marked with “Sad”, this may due to the scaling of face. And the gender classification only has one error in the image, since the gender features for the person seems kind of vague.

I also tested the system on a lot of my personal images, my feeling is that the system is giving much better performance than the ACC value showed in table of Section 3. For gender classification, we have quite high ACC here because people in real lift images mostly have more clear features than training data. While for emotion analysis, as shown in Section 4.1. The system is giving much higher F1 score on emotions like “Happy”, “Surprise”, which are more common emotions in real life.

4.3. Video Stream Test

In addition to image, I also tested the system on video stream to see how well it can handle continuous expression change. Different from image, which is static, video is a more flexible method to do the gender and emotion testing, since we can show different expressions and observe how the system handles those changes.

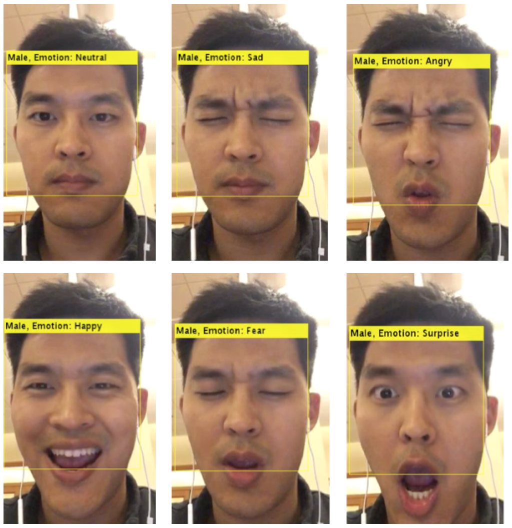

Figure 11. Emotion samples from video.

We can use Matlab vision.VideoFileReader to get each of the video frame. We can also use sparse sampling rate, instead of doing classification on each frame, do it for each of the N frames. This may lose some resolution of classification but definitely be able to speed up the system.

After training from section 3.4.6, classifiers are ready to use (which can be saved as .mat format). Thus in real system testing, we can load classifier into memory directly, no training needed.

I recorded a video with length of 94 frames in total, which lasts for 15 seconds. Fig.11 shows 6 sample frames from the video stream. Each of the sample has different emotion type. The result looks quite reasonable to me. One interesting thing we notice is face box excludes mouth under some situations (like “Fear” and “Surprise”), but we are still able to get the correct prediction, indicating the system is robust. Again, emotion prediction itself is kind of subjective or vague in real life. It seems the major difference between “Angry” image and “Sad” is on the mouth feature, but both predictions are reasonable to me.

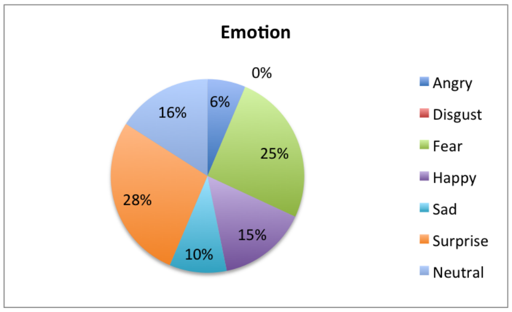

Figure 12. Emotion distribution during video sampling time.

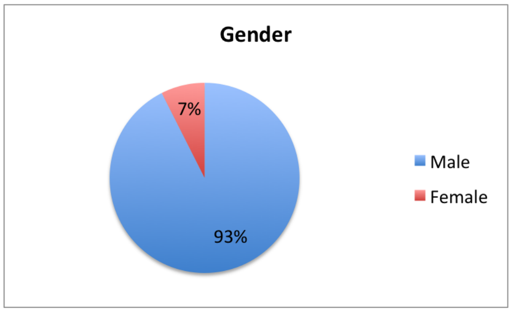

Figure 13. Gender distribution during video sampling time.

Instead just focusing on one single image, a more interesting or useful analysis would be doing this on entire video stream time. Fig.12 shows the emotion distribution on the 94 frames of this 15 seconds video. We can clearly see the proportion of each type of emotions. In this demo video, I was “Angry” for 6% of the time, never been “Disgust”, “Fear” for 25%, “Happy” for 15%, “Sad” for 10%, “Surprise” for 28% and “Neutral” for 16%. We notice that emotion “Disgust” never appears in this video stream, this is because training data for “Disgust” was significantly smaller than others (based on Table 2 and Figure 4). Also from 4.1, we can see “Disgust” has a very low recall, which means it’s relatively difficult recognize this emotion from image, but once it’s recognized, it will mostly be correct (based on the high precision from 4.1). System detects 6 types of emotions in this short video because I was changing my expression frequently on purpose. In reality, this kind of emotion distribution maybe useful to evaluate the quality of a class.

Fig.13 shows the gender prediction distribution during the video time. From which we can see that most of the time (93%) the system is able to make prediction with correct gender. The gender prediction engine seems quite accuracy. It should also be very robust since I did a lot of exaggerated face expressions during this video. Otherwise we should look forward to even higher accuracy.

5. Conclusion and Future Work

In this project, we touched topics like face detection, gender classification and emotion analysis. Different feature extraction method like BoW, LBP and HOG were tested. Key-point detection based feature dimension reduction was considered to reduce complexity. Multiple classifiers like NB, KNN, Random Forest and SVM were tested. Parameters for each model were tuned to get the best performance. Numerical results indicate that we are able to get 0.8673 accuracy on gender classification and 0.5089 on emotion analysis (0.6073 when we prune particular class). Further analysis was done on precision, recall and F1 for each classes. Testing on real life images and video stream also demonstrates the validity of the system.

Personally speaking, this is a very exciting project, which lets me familiar with different vision algorithms and how to connect them with machine learning tools. Being able to develop a demo system that can be used immediately is very interesting. I had a lot of fun to test on my different personal pictures and photos.

ACC on emotion analysis is a part that can be improved in future work. Also current emotion analysis has biased performance on different type of emotions. Introducing deep learning concept should be very help to improve this multiclass problem.

Also, to be better used in real life scenarios like class quality analysis. More information can be collected, like human poses, age information, human recognition etc. A good model to use the collected information generate an overall summary score (like group analysis) will also be very interesting.

Code link: Follow this link.

Code link also submitted through Google Form.

References

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

[2] Zhang C, Zhang Z. A survey of recent advances in face detection[J]. 2010.

[3] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features[C]//Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. IEEE, 2001, 1: I-511-I-518 vol. 1.

[4] Mkinen E, Raisamo R. An experimental comparison of gender classification methods[J]. Pattern Recognition Letters, 2008, 29(10): 1544-1556.

[5] Lian H C, Lu B L. Multi-view gender classification using local binary patterns and support vector machines[M]//Advances in Neural Networks-ISNN 2006. Springer Berlin Heidelberg, 2006: 202-209.

[6] Baluja S, Rowley H A. Boosting sex identification performance[J]. International Journal of computer vision, 2007, 71(1): 111-119.

[7] Saatci Y, Town C. Cascaded classification of gender and facial expression using active appearance models[C]//Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on. IEEE, 2006: 393-398.

[8] Kim Y, Lee H, Provost E M. Deep learning for robust feature generation in audiovisual emotion recognition[C]//Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013: 3687-3691.

[9] Fasel B, Luettin J. Automatic facial expression analysis: a survey[J]. Pattern recognition, 2003, 36(1): 259-275.

[10] Gallagher A, Chen T. Understanding images of groups of people[C]//Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009: 256-263.

[11] https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data.

[12] https://en.wikipedia.org/wiki/Accuracy_and_precision.

[13] https://en.wikipedia.org/wiki/F1_score.

[14] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]//Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. IEEE, 2005, 1: 886-893.

[15] http://www.mathworks.com/help/vision/ref/extracthogfeatures.html

[16] Ahonen T, Hadid A, Pietikainen M. Face description with local binary patterns: Application to face recognition[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2006, 28(12): 2037-2041.

[17] Wang X, Han T X, Yan S. An HOG-LBP human detector with partial occlusion handling[C]//Computer Vision, 2009 IEEE 12th International Conference on. IEEE, 2009: 32-39.

[18] http://www.shutterstock.com/s/emotions/search.html

[19] https://jonmatthewlewis.wordpress.com/2012/02/24/posture-and-gestures-in-the-classroom-and-on-the-date/