Facial Features Analysis Algorithms Comparison

Yanling He
Department of Computer Science & Engineering
University of Washington
Seattle, WA 98105
heyl@cs.washington.edu

Xin Yang
Department of Computer Science & Engineering
University of Washington
Seattle, WA 98105
yx1992@cs.washington.edu

Abstract

Facial expression is a powerful communication tool in our daily life. As the computer and Internet becomes more popular today, more communications happen between the human and the machine. To understand and analyze the facial expression becomes a key point in human-computer interaction. In this project we are going to implement couple machine learning methods to recognize and classify the face image features, including facing position, facial expression and some wearing accessories and comparing the advantages and drawbacks of the methods for different data sets.

1 Introduction

Recently, because of the broadly use of computers and technology equipments there is an increasing research interest in facial recognition and facial features and expressions analysis. There are many algorithms has been developed in machine learning and computer vision areas, like Principal Component Analysis (PCA), kernel method, support vector machine (SVM), boosting solution and so on. There are also many facial feature analysis applications in different areas, including social communication, human-computer interaction, animation, gaming, celebrity detection and face searching. For different purpose, different algorithms may have different advantages.

Based on those backgrounds, this project focuses on implementing different face recognition and feature detection algorithms to compare the performances between them. The project uses two major different pipeline, first is using PCA to simplify the features of image data, then apply different training methods (logistic regression with 5-fold cross validation and SVM with different kernels). Second one is using K-NN to learn the features directly from the image data.

The project also applies algorithms on two different data sets. One data set is experimental face images with different face features, another one is Labeled Faces in the Wild, a database of face photographs collected from the web. Each face has been labeled with the name of the person pictured.

The goal of the project is to compare the experimental results of those algorithms on different data sets in order to select better algorithm for different cases.

2 Facial Feature Recognition

The facial feature analysis system contains two major parts. First part is data processing, it reads the image file from data sets and processes them to matrix data. Second part is facial features recognition. The project implements two major pipelines, one is to reduce the features number first then apply the learning method, another one is classifying the features on the raw data. In the first pipeline there are couple different learning algorithms implemented. More details about the pipelines and algorithms will be described in section 2.2.

2.1 Data Processing

The input to the system is a sequence of images. The project uses two different data sets, one is from the experimental face images provided by CMU. The images contain different facial expressions, facing different positions and some are wearing sunglasses. Figure 3 is one example of the input images of this data set.

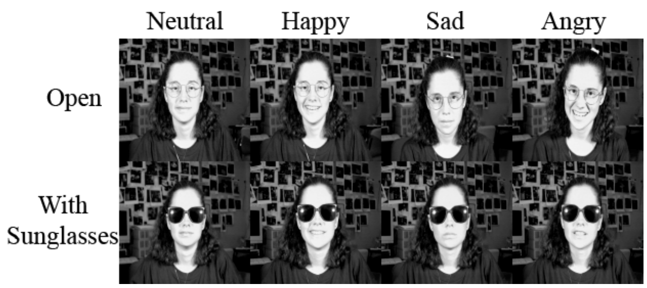

Figure 1: Images facing straight with different facial features.

Another data set is Labeled Faces in the Wild, a database of face photographs contains more than 13,000 images of faces collected from the web. Each face has been labeled with the name of the person pictured. Figure 4 is one example of this data set.

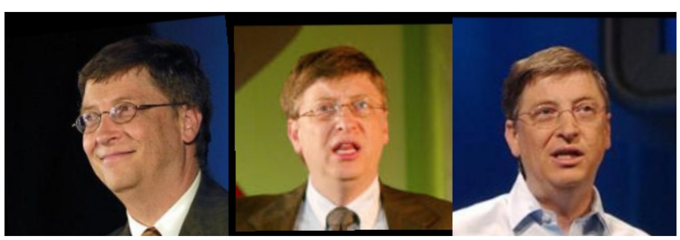

Figure 2: Images facing different positions of one lable people.

The image data goes into the data processing system. It reads the labels of images and based on different features, the images are classified as an input vector $\mathbf{y}$. The images are composted into an input matrix $\mathbf{X}$ = image matrix - mean of all the image matrices.

2.2 Facial Feature Recognition Pipeline 1

The facial feature recognition system contains two different pipelines. First one is to reduce the features number first then apply the learning method. It contains a data processing system that reading the image and processing them into matrices. Principal Component Analysis (PCA) to decompose the image matrix into eigenvector. Applying Training methods on the eigenvector to learn the data and get the classification results.. There are two learning methods to choose that are logistic regression with 5-fold cross validation or SVM with different kernel types.

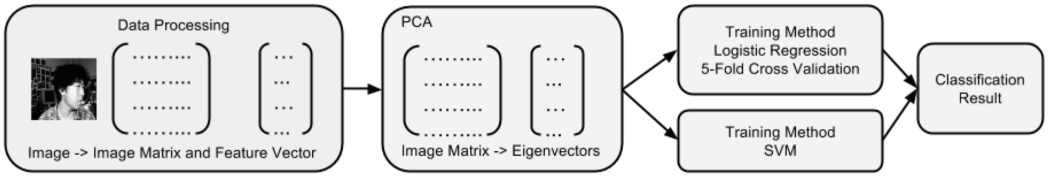

Figure 3: Pipeline1 of facial feature recognition system.

2.2.1 PCA

The image data goes into the principle component analysis to be decomposed into eigenvectors. First, it calculates the covariance matrix of the input $\mathbf{X}$:

$$\mathbf{S} = \mathbf{X}\mathbf{X}^T$$

Then find the eigenvector decomposition $\mathbf{v}_i$ of $\mathbf{S}$:

$$\mathbf{S}\mathbf{v}_i = \mathbf{T}\mathbf{T}^T\mathbf{v}_i = \lambda_i\mathbf{v}_i$$

$$\mathbf{T}\mathbf{T}^T\mathbf{T}\mathbf{u}_i = \lambda_i\mathbf{T}\mathbf{u}_i$$

So, eigenvector $\mathbf{v}_i = \mathbf{T}\mathbf{u}_i$

2.2.2 Logistic Regression

To get the traning data by decompositing on the eigenspace, we have $\mathbf{X}' = \mathbf{X}\mathbf{V}$. Use logistic regression to find weight $\mathbf{w}$:

$$\mathbf{w} = \mathbf{w} + \eta * (-\lambda * (\mathbf{w}) + \frac{1}{N} * \mathbf{X'}^T \cdot (\mathbf{Y} - (1 - \frac{1}{1 + e^{w_0 + \mathbf{X'} \cdot \mathbf{w}}})))$$

$$\mathbf{w_0} = \mathbf{w_0} + \eta * \frac{1}{N} * (\mathbf{Y} - (1 - \frac{1}{1 + e^{w_0 + \mathbf{X'} \cdot \mathbf{w}}}))$$

2.2.3 Cross Validation

The system uses 5-fold method to calculate the λ. Bases on the $\mathbf{w}$ we divided data into 5 groups, using 4 groups as training data and calculates the error of the fifth group. Apply the same calculation on all the 5 groups to get 5 error data. Take the average on the error data and find the λ corresponding to the minimum error. λ starts from 100 and divided by 1.4 at each step. Based on Figure 3, $\lambda = 0.2$.

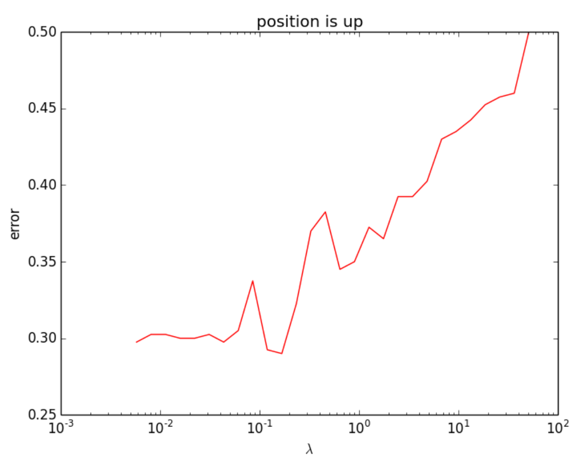

Figure 4: Cross validation error.

2.2.4 SVM

SVM is a widely-used learning algorithm for classification. It finds support vectors to decide the decision hyper plane, and it is able to deal with non-linear-separable data with different kernels. We use the implementation provided by sklearn based on LibSVM.

2.2.5 Classification

The first pipeline are implemented into two types classifications: uni-variate classification and multi-classification. In the uni-variate classification we classify the feature into 0 or 1 class.

Compared with 0-1 classification, multi-classification deals with multiple classes. It is more complicated and better characterized the real world. It can be implemented through 1 vs. rest method or 1 vs. 1 method. The project extends both logistic regression and SVM to perform multi-classification.

2.3 Facial Feature Recognition Pipeline 2

Second pipeline is to apply K-NN learning method directly on the processed raw image data without reducing the feature numbers.

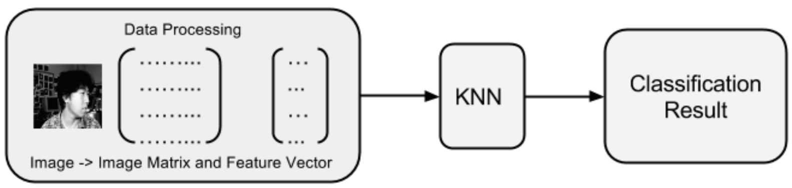

Figure 5: Pipeline2 of facial feature recognition system.

2.3.1 K-NN

As a parametric-free classification algorithm, we also implement K-nearest-neighbor(KNN) as a baseline of comparison. We regard every picture as an element in high-dimensional space, where the dimension is the number of pixels. In order to classify an incoming picture, we find the nearest k pictures in the training dataset, and then decide the feature of the input as the majority of these k neighbors.

3 Experiment Result and Analysis

3.1 Uni-variate Classification

The system applies the first facial features analysis pipeline on uni-variate data. The algorithm uses PCA to reduce the features number, then apply logistic regression with 5-fold cross validation. It classifies the images into two classes: whether the people in the image wearing sunglasses or not.

3.1.1 result

Table 1: Sunglasses Classification Result with Image Size 32*30 pixels

Facing Position	Class 0 Precision	Class 0 Recall	Class 1 Precision	Class 1 Recall
Straight	1.0	0.92857	0.93333	1.0
Up	1.0	0.75862	0.79412	1.0
Right	0.61765	0.77778	0.714286	0.53571
Left	0.84	0.75	0.78125	0.86207
Mix	0.64539	0.80531	0.73494	0.54955

Table 2: Sunglasses Classification Result with Image Size 64*60 pixels

Facing Position	Class 0 Precision	Class 0 Recall	Class 1 Precision	Class 1 Recall
Straight	1.0	0.92857	0.93333	1.0
Up	0.92857	0.89655	0.89286	0.92593
Right	0.61765	0.77778	0.714286	0.53571
Left	0.67857	0.67857	0.67857	0.67857
Mix	0.70635	0.79464	0.76289	0.66667

3.1.2 Result Analysis

The project learns four data sets separately, including facing straight, facing up, facing left and facing right. It also learn a mix data set with different facing directions. Class 0 is open face without sunglasses. Class 0 precision is the number of people predicted correctly not wearing sunglasses over the number of all the people predicted as not wearing sunglasses. Class 0 recall is number of people predicted correctly as not wearing sunglasses over the number of people actually not wearing sunglasses. Class 1 is wearing sunglasses. The projects did the experiment on different image size groups. The experiment result is shown in Table 1 and 2.

Facing positions: The results of recognizing whether wearing sunglasses of the faces facing straight are near to 1, which means the prediction result is very good and we have enough confident to classify and prediction whether people wearing sunglasses when they facing straight.

The precision and recall decreases when faces facing to different directions. The precision and recalls are still larger than 0.5, which means the prediction learning based on the method is better than random guess. The error may be caused by the bad alignments of the faces and sunglasses when facing to different directions and the area of the sunglasses region reduced, so it becomes harder to detect.

Image sizes: The results of recognition of the faces facing straight and top are better on larger images. This may causes by larger sunglasses areas shown on larger images, to the learning algorithm is easier to find.

From the table we can see, the precision and recall of facing right and left of larger images goes down to 0.7 - 0.8 which is worse than facing straight and up. This may be caused by the worse alignments of the faces and sunglasses on larger images.

3.2 Multi-Classification - Name

The system applies different learning algorithm on name classification for two data sets. After processing the data, it applies PCA to reduce features number then run logistic regression or SVM to learn the features, or run 1NN algorithm without reducing and feature numbers. The results are shown in Table 3 and 4.

3.2.1 result

Table 3: Name Classification Result Data Set 1

Image Size (pixels)	Logistic Regression	SVM	1NN
128 * 120	1.0	0.992	1.0
64 * 60	0.992	0.992	0.992
32 * 30	0.992	0.992	1.0

Table 4: Name Classification Result Data Set 2

Logistic Regression	SVM	1NN
0.76744	0.86047	0.59690

3.2.2 Result Analysis

The system learns two data sets with different algorithms: logistic regression, SVM and 1NN. It also applies algorithms on different sizes images from data set 1. The results record the precision which is the fraction of events where we correctly predicted labeled name i out of all instances where the algorithm declared labeled name i.

The prediction on both data sets are good that around 0.8 - 1.0, especially the first data sets. Because when learning the name, the algorithms can learn the features from the whole faces, even those faces have different expressions or positions, the overall parts of the face remain same. It is easy to recognize and classify.

Data sets: The results of data set 1 are better than data set 2. Because data set 1 is experimental images, all the faces on aligned to the center of the image and the backgrounds are same, while the labeled faces in wild have different background and face sizes and positions.

1-NN shows incredibly high accuracy with the first data set. This is because if two picture are of the same person with the same position, then their distance will be very small, even if their expression or sunglasses may be different. However, we can see that 1-NN sacrifice other features to achieve this high accuracy.

But 1NN algorithm does not show the advantages when using data set 2, this may cause be different face alignment of the faces, the nearest neighbor the algorithm find is no longer as stable as the experimental face images.

Actually, when predicting expressions, 1-NN performs extremely bad, since it always tends to return the same ID instead of expression.

Image sizes: From the data we can see the precision on classify the names in data set 1 are all very high on name classification. The results are near 1, the small error may caused by the typical image difference. So image size does not affect lots in name classification of data set 1.

3.3 Multi-Classification - Facing Position

The system applies different learning algorithm on facing positions classification for data set 1. After processing the image data, it apply PCA to reduce features number of the images, then runs logistic regression or SVM to learn, or run 1NN algorithm without reducing and feature numbers. In this case the algorithms learn to classify which direction the person faces, including straight, up, left and right. The results are shown in Table 5 and 6.

3.3.1 result

Table 5: Facing Positions Classification Result 1

Image Size (pixels)	Logistic Regression	SVM	1NN
128 * 120	0.90476	0.93651	0.88889
64 * 60	0.93651	0.98127	0.88889
32 * 30	0.92063	0.95238	1.0

Table 6: Facing Positions Classification Result 2 with Image Size 32 * 30 pixels

Methods	Precision
Logistic Regression	0.90476
SVM [Linear]	0.93651
SVM [poly, deg=3]	0.88889
SVM[sigmoid]	0.52381
SVM[rbf]	0.25397
1NN	0.88889
2NN	0.77778
3NN	0.71429

3.3.2 Result Analysis

The system learns varied size images from first data sets with different algorithms: logistic regression, SVM and 1NN. It also applies different kernels for SVM and different nearest neighbor number for KNN. The results record the precision which is the fraction of events where we correctly predicted facing position i out of all instances where the algorithm declared facing position i.

The prediction of facing position with logistic regression, SVM with linear kernel and 1NN are good. The precision are all around 90%.

Kernels: The system applies different kernel types for SVM algorithm. Linear kernel is the best. Followed by it is polynomial kernel with degree 3. Sigmoid and rbf kernels do not work well in this case.

Nearest neighbor number: The system tried different number of nearest neighbors. Less nearest neighbors works better than more neighbors. This is because training set usually contain one picture that only differs from the test on expression, and this train data can give perfect prediction on position as the nearest neighbor.

Image sizes: The precision of different images sizes are all around 0.9, so the size does not affect lots.

However, we can see that pictures with low resolution tends to predict better. This may come from the fact that the original picture contains some noise, and then after taking the average, the noise cancels with each other, hence it may be easier to predict.

Of course, the image cannot be too small, otherwise we may lose too much information.

3.4 Multi-Classification - Facial Expression

The system applies different learning algorithms on facial expression classification for data set 1. After processing the image data, it apply PCA to reduce features number of the images, then runs logistic regression or SVM to learn, or run 1NN algorithm without reducing and feature numbers. In this case the algorithms learn to classify the facial expressions the person expressed in the image, including sad, happy, angry and normal. The results are shown in Table 7.

3.4.1 result

Table 7: Facial Expressions Classification Resuls

Methods	Precision
Logistic Regression	0.19048
SVM	0.23810
1NN	0.03175
3NN	0.14286

3.4.2 Result Analysis

The results of facial expression are very bad. This is because the algorithm learns the feature from the whole images, it learns all other parts from the face. Based on face features it will select a similar, in this case it will probably select the face with the same labeled name, because the major parts of the face will be same since they are same people. But the image in the training data will not show up in the testing data, so it will labeled current image with an expression of same people but different facial expression. Thus the precision is very low. To improve this, we can learn each parts of the face. To classify facial expression based on the eyes and mouths shapes, but not the whole face.

4 Conclusion

The project implements different facial features analysis algorithms on different sets, including logistic regression with 5-fold cross validation and SVM with different kernels after running PCA to reduce the number of features. K-NN without reducing the features number. The project applies those methods on two different data sets, experimental face images and labeled faces in the wild to learn different features, including whether wearing sunglasses, name classification, facing position classification and facial expression classification.

The results of uni-variate classification depends on how well the feature is aligned on each image and the size of the feature. If the size of the feature is larger and alignment is better, the learning result is better. Logistic regression with 5-fold cross validation works well on uni-variate classification in our experimental face images. To improve the uni-variate classification results, the system can pre-align the feature objects in the images first.

For multi-classification, the results varies based on the different feature it learns. Logistic regression, SVM and 1NN all work well on name and facing positions classification. This is because name the facing position is a relative large feature on the image. 1NN performs slightly better on name classification. For SVM algorithm, linear kernel works the best rather than other kernels. For KNN algorithm, 1NN performs better than other number of nearest neighbors. But all the algorithms perform bad on facial expression classification, this it due to the facial expression is a relative small changes based on the whole face, if the algorithm learns whole face it will classify the image to that person, but not a similar expression on others' faces. To select the feature region and only learn that parts may help to improve in this case.

The project also runs the algorithm on different data set. The results of experiment images are better than labeled wild images. This is because the features are well aligned in experiment images.

The project is success, it shows us the differences of different algorithms learning different features. It can help to choose better algorithm for different features. Also, by comparing the results on different data sets, it help us to know how to pre-process the images to improve the classification accuracy.

5 Future Work

The system can be improved in couple ways. Since the predicted results depend on the feature alignments, in data processing step the system can pre-aligned the feature objects on the images.

To improve the facial expression classification, the system can divide the images into different parts, like mouths and eyes, then learn the features from those parts to classify the expression by the changes on them instead of learning the whole face. The similar technique can be applied on classifying the glasses or other facial features by select particular parts to improve the precision.

As the machine learning are developed rapidly recently, there are many algorithms with higher accuracy and faster speed, for instance deep learning. The system can implement more algorithms to compare.

References

[1] Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. Journal of cognitive neuroscience, 3(1), 71-86.

[2] Hyvarinen, A. (1999). Survey on independent component analysis. Neural computing surveys, 2(4), 94-128.

[3] Chang, C. and Lin, C. (2011). LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2(3), 27:1–27:27.