Facial Expression Prediction

I Huang
Cornell Tech
ih265@cornell.edu

Tao Yuan
Cornell Tech
ty353@cornell.edu

Abstract

This paper explores how we may predict people's emotion based on facial expression. There is an overview of the datasets we selected, as well as the demonstration and analysis of the four models we implemented.

1 Introduction

Communication efficiency can be boosted by accurately telling the audience's emotions. With the development of machine learning, people nowadays are expecting to better tell other's emotions. We found this topic very interesting since it can be a useful tool implemented on, for example, Google Glass, to help users identify the emotions of the people they are talking to. By using this technology, for example, sales will be able to tell their customers' mood and interest, teachers can learn if the students are actually paying attention during class.

We are interested in learning the correlation between the image and emotion in order to be able to predict the emotion of any image into its closest category. We first chose our baseline model and implemented this model for getting a benchmark of the predictions. Afterwards, we explored different models to improve the performance. The details of our model implementation can be found at https://github.com/bayernstar/5304project.

2 Related work

We built our models based on the Toronto Faces Dataset[1], which provides 2925 labeled images and 98,058 unlabeled images. Each image is a 32 by 32 grayscale that contains a facial expression. Each labeled image is associated with one of the seven emotions: 1-Anger, 2-Disgust, 3-Fear, 4-Happy, 5-Sad, 6-Surprise, 7-Neutral.

There has been a lot of research going on in Computer Science for analyzing facial expression. For example, Facial Action Coding System (FACS) Action Unit (AU) detection and classification[2] is used to determine the emotion states based on images.

In addition, people have developed deep neural networks[3] on Toronto Faces Dataset to implement multi-task learning and reached an accuracy of 87%.

3 Baseline Model Description

Since we have both labeled and unlabeled images, we decided to start with supervised learning first with the labeled data as our baseline model, then move on to the unsupervised learning utilizing the unlabeled images.

We chose k-nearest neighbors algorithm (k-NN)[4] to build the baseline model, since it is one of the most common and fastest supervised learning algorithms. We split the training and validation data into 5 folds, and used cross-validation to get the validation accuracy.

4 Dataset Description

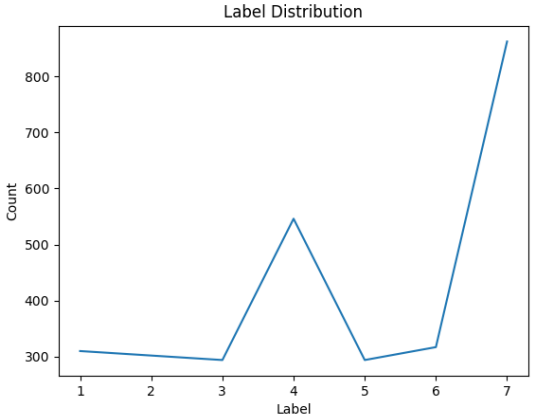
We have two data files: labeled_images.mat and unlabeled_images.mat. For labeled_images.mat, there are 3 components including tr_identity, tr_labels and tr_images.

Tr_identity contains a 2925 * 1 matrix, each row has an anonymous identifier unique to a

given individual. Tr_labels contains a 2925 * 1 matrix, and each label is one of the seven emotions. Tr_images contains 2925 images given by pixel matrices (32 pixels by 32 pixels by 2925 images). It is worth to notice that it is possible for one person to have multiple expressions in the labeled set.

In unlabeled_images.mat, there is only unlabeled_images which contains 98,058 32 by 32 pixel matrices.

We plotted the distribution of all seven labels and observed that there are way more number of label 4 and 7 compared to other labels, who are evenly distributed.



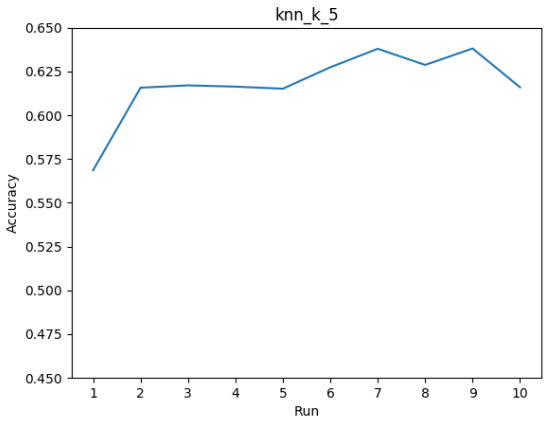
5 Experimental Setup

We first randomly split the data into the train and validation sets, and ran kNN with k = 1,3,5,7,9,11,13 and 15. The highest accuracy is around 52%.

To improve the accuracy, we thought about taking advantage of the identity data. We decided to avoid putting expressions of the same person into both test and validation sets. So we changed the way of splitting the data and ran kNN again. The accuracy improved from 0.52 to 0.59.

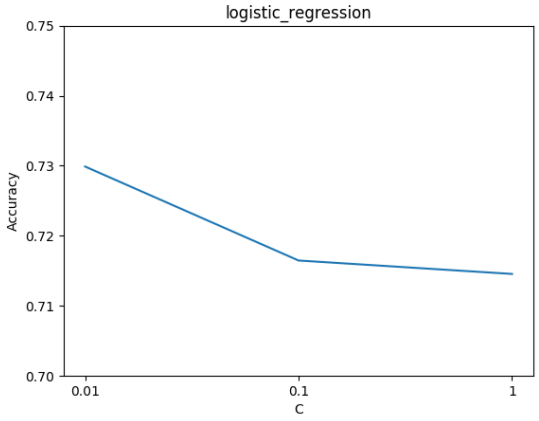
After changing the cross validation method, we tested each value of K for ten times to obtain the average accuracy. Finally, we reached an average accuracy of 56% with K equals to 5. Compared to random guessing whose accuracy is 14%, kNN is obviously a good choice for the baseline model because of its simplicity, speed and accuracy.

By setting K to 5, we tested the model with our selected parameters and the best performance was 63.8%.

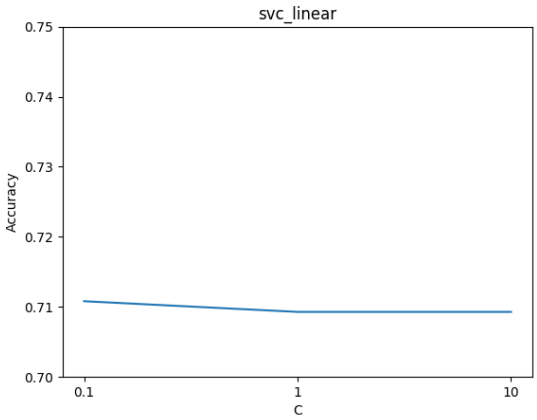


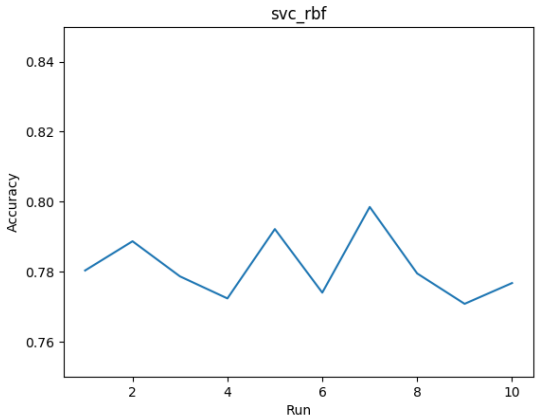
6 Part2 Models

To improve the performance, we started from using Logistic Regression[5], which is one of the most common and fastest classification method. We tried different values of regularization strength(C), and found the model reached the accuracy of 73% with a regularization strength of 0.01.

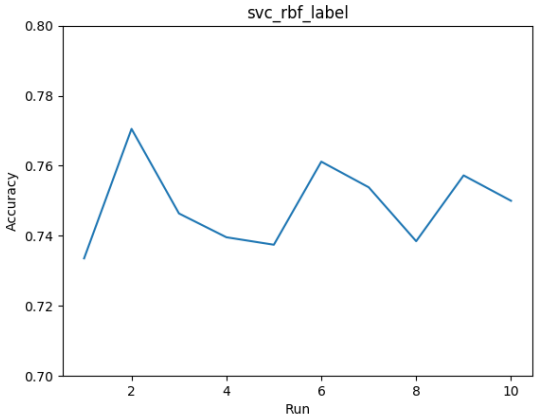


Since Logistic Regression performed much better than kNN, we concluded that the dataset might be fairly linearly separable. So we decided to continue with linear classification by using Linear SVM[6], which is more effective in high dimensional spaces. As the labels of the dataset are not binary, we used OneVsRestClassifier and tested different values of penalty parameter(C). The accuracy of Linear SVM is consistent around 71%.





We realized that there was no improvement for Linear SVM compared to Logistic Regression, which indicated that 73% was close to the best accuracy for linear classification. We looked into different SVM-Kernels, and found out that the RBF Kernel would work better for classifying images[7]. We used GridSearchCV[8] for searching over specified parameter values and found the best parameters of C = [100] and gamma = [0.01]. The RBF SVM reached an accuracy of 77%.



It is clear that RBF SVM performed much better than Linear SVM. However, training so many images appeared to be slow on RBF SVM, so we decided to perform dimension reduction by applying PCA. The accuracy increased up to 80%

7 Results and any Analysis

kNN	64%
Logistic Regression	73%
Linear SVM	71%
RBF SVM	77%
RBF SVM with PCA	80%

At this stage, we have made a big progress from the baseline model. We found that nonlinear classification worked better than linear, and meanwhile it is much slower. Therefore, it is a tradeoff between speed and performance. To achieve a balance, we tried to reduce the dimension and obtained the best accuracy so far. For the next step, we are looking for methods to fully utilize the unlabeled images and try ensemble modeling for getting a better result.

8 Citations

[1] http://www.aclab.ca/
[2] Michel Valstar. 2002. *Meta-Analysis of the First Facial Expression Recognition Challenge*. http://www.cs.nott.ac.uk/~pszmv/Documents/fera_smcb.pdf
[3] Terrance Devries. 2014. *Multi-Task Learning of Facial Landmarks and Expression*. http://www.uoguelph.ca/~gwtaylor/publications/gwtaylor_crv2014.pdf
[4] https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
[5] https://en.wikipedia.org/wiki/Logistic_regression
[6] http://scikit-learn.org/

stable/modules/generated/
sklearn.svm.LinearSVC.html#
sklearn.svm.LinearSVC
[7] Charles Martin. 2012. *KERNELS PART 1: WHAT IS AN RBF KERNEL? REALLY?*.https://calculatedcontent.com/2012/02/06/kernels_part_1/
[8] http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html