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 baseline model 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 have currently chosen our baseline model and implemented this model for getting a benchmark of the predictions. The details of our baseline model 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

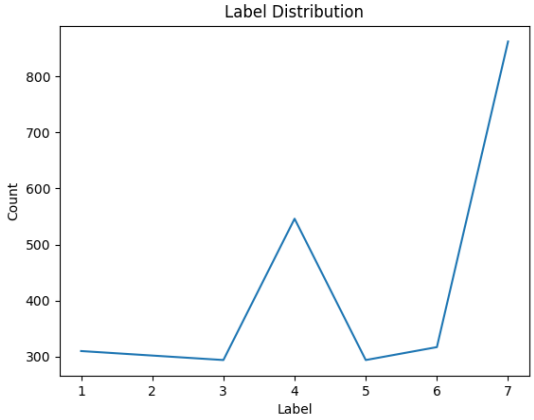
We have two data files: labeled_images.mat and unlabeled_images.mat. For labeled_images.mat, there are 3 components including tr_identity, tr_labels and tr_images.

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.

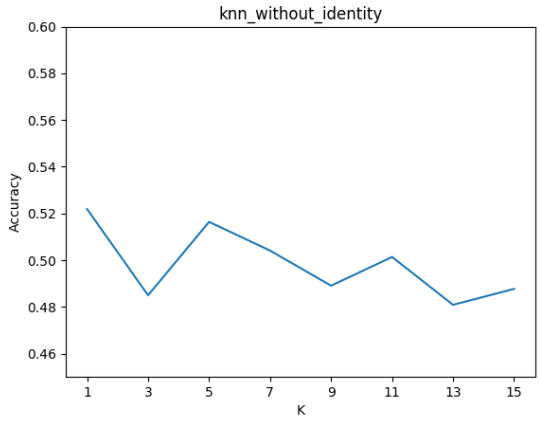
In unlabeled_images.mat, there is only unlabeled_images which contains 98,058 32 by 32 pixel matrices.

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. We would take this into consideration in future implementation.

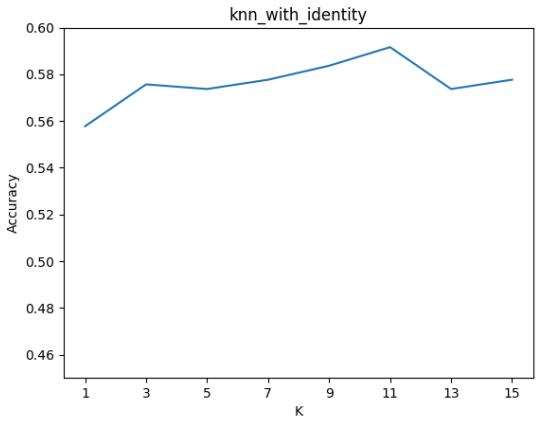


5 Experimental Setup

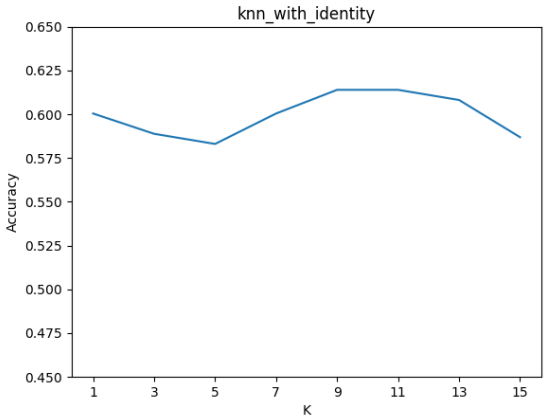
We first randomly split the data into the train and validation sets, and ran kNN with k = 1,3,5,7,9,11,13 and 15. The highest accuracy is around 52%.



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.

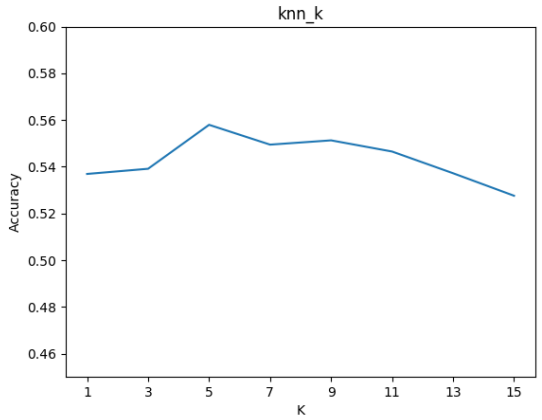


We then started adjusting the parameters for kNN in order to further increase the accuracy. We found that setting algorithm to auto can improve the accuracy over 0.6, and the highest we could get was around 0.61, which is a fairly good number for a baseline model.

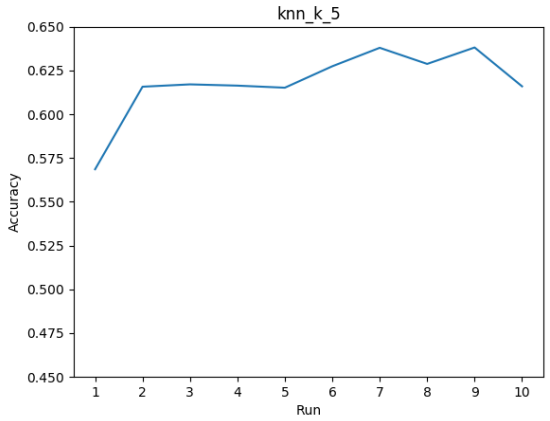


6 Results and any Analysis

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%.



However, we could never improve the accuracy over 65% by trying different parameters. One major reason is that the labeled dataset has only 2925 images, while the unlabeled dataset has 98,058 images which could not be used by kNN. In addition, we found that different values of K could effectively affect the model's performance. As a result, a more complex model and unsupervised learning will definitely improve the performance. Based on the current baseline model, we are looking for achieving an accuracy over 80% in the future.

7 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