Face Detection and Smile Classification

Eric Chan Lam Choi
Northwestern CS Undergraduate
1927 Orrington Avenue
Evanston, IL 60201
1-847-648-4888
Lam-chan@northwestern.edu

Gregory Peaker
Northwestern CS Undergraduate
562 Lincoln Street
Evanston, IL 60201
1-516-225-1337
G-Peaker@northwestern.edu

Patrick Wong
Northwestern CS Undergraduate
1940 Sherman Ave
Evanston, IL 60201
1-224-628-5086
pw@ northwestern.edu

ABSTRACT

We use a combination of image analysis techniques and a neural network to design a face detector. In this paper, we apply J48, Naïve Bayes, and Artificial Neural Network Machine Learning techniques in identifying smiles within a picture. We then take the same image analysis techniques and adapt it to also quantify faces in a picture, and then use different classifiers to determine if a smile is taking place. Results indicate that identifying smiles is possible with current technology and is ready for real-world application.

Categories and Subject Descriptors

D.3.3 [**Programming Languages**]: Design, Documentation, Experimentation

General Terms

Face Recognition, Computer Vision, Digital Image Analysis, Algorithms, Measurement, Design, Experimentation, Machine Learning, J48, Naïve Bayes, Artificial Neural Network.

Keywords

Identifying Smiles, Smile Detector

1. INTRODUCTION

This paper presents our research in identifying smiles, where a collection of pictures are quantified with Digital Image Analysis techniques, processed and compared using several Machine Learning techniques, and percent correct smile identification is verified by human judgment. We discuss background research leading to this project design, a discussion on the Digital Image Analysis techniques, Machine Learning results, and concluding thoughts.

The project's goal is to correctly identify smiles in a picture. This information is useful for various researches. Several examples include modeling systems for psychological studies on human emotional responses (imagine a psychologist autonomously determining a patients happiness using our smile test), extending image search capabilities, etc.

Face and emotion classification is also useful now for various applications, such as smile detectors and auto focus in recent digital camera technologies. Also, the feature extraction techniques are useful for other fields of interest, such face recognition in security cameras. While we initially intended to pursue emotion detection, we felt that it was better to put more focus on smile detection and classification as it can be extended to classify other emotions.

2. BACKGROUND AND RELATED WORK

Our research began by taking a class at Northwestern University entitled Machine Learning taught by Professor Douglas Downey. We read on the use of Decision Tree Learning, Artificial Neural Networks, Evaluating Hypothesis, and Bayesian Network to solve problems in computer science research. [1]

We also looked at multiple face detection techniques, especially the use of eigenfaces and template matching. From going through the known algorithms, we decided to use a mixture of the strongest points of these algorithms to facilitate in designing our own, which we believe it to be computationally effective and accurate enough despite its naivety.

3. DETECTING FACES USING DIGITAL IMAGE ANALYSIS TECHNIQUES

The first part of our project is to detect human faces in a given image. Human face detection has become a major field of research over the past decade, and currently there is no deterministic algorithm that accurately detects faces for a variety of environments. On the other hand, there are several machine-learning algorithms that work very well. However, these algorithms are highly dependent on the training data that they were trained on, as well as the input image. Our approach is to utilize a mixture of these known algorithms to sufficiently detect a face and pass it further to the next part of our project for further analysis. We can define our entire face detection algorithm as a series of rejection blocks, where the input image is passed into several scripts to be filtered and is then passed into the last script for the face detection. The three main steps of this series are color segmentation, morphological transformation with size filtering and finally smile detection using neural networks. [2]

[Original Image] [Color-segmented] [Morphological Operator] [Face detection with Neural Nets]

Figure 1: Sample Image

3.1 Color Segmentation

The first task in our series of script to refine the image is skin color segmentation. Two algorithms that were considered were the histogram-based color segmentation and the adaptive Gaussian mixture model for skin color segmentation. While the Gaussian model segmentation produced better results, we decided to implement and use the histogram based segmentation algorithm because it has a much lower computation time requirement. The segmentation program was trained using 5 different samples of skin (images with people with very different skin color), which allowed us to account for faces of people from all races. While this would increase the space in which the skin values lie, we made sure to set significantly stricter thresholds per skin color such that foreign objects of similar color is less likely to be detected, thus increasing the quality of the segmentation.

The training images are collected and converted to hue/saturation/value (HSV) color space. We've decided to utilize the HSV color space because it is proven to be more effective in segmenting the colors of skin compared to the RGB color space. From the training images, we build a histogram and take the values that occur most often, labeling them as the skin color that we will be using. Using those values, the input image is scanned for those colors (with a certain maximum and minimum threshold such that the pixel values do not have to match exactly, as long as they are somewhat similar). Pixels with skin values are segmented out, and the others are simply set to 0. The end result is an image containing only pixels that we believe are of the human skin. This is then passed on to the next rejection block for further filtering.

3.2 Morphological Image Processing and Size Filtering

The resulting image that we obtained after color segmentation would still contain some noise, which is made up of scattered skin pixels and maybe some arbitrary pixels of other objects that share similar tones to that of the skin. It is also possible that some pixels are missing within regions of a face because the segmentation was too strict, thus removing some pixels which are actually real skin. This is termed as pepper noise. However, we want an image that is clean such that the neural network face detection algorithm will be able to run without difficulties at all. To accomplish this, we perform a combination of morphological operations on the color-segmented image to fill up the holes in between skin regions, as well as removing irrelevant noise. For all of these operations, we've standardized the structuring element to be a simple disk of size three.

'Closing' is performed to cover up pepper noise / holes inside and around skin regions such that the region. And then, a series of 'dilation' and 'erosion' is performed to get rid of the remaining foreground pixels. We end up with a much cleaner image after performing these operations. However, these operations will not be able to remove significantly larger objects which we think are not faces. For this, a size filter is used to filter out objects that are oversized or well below the average size of a typical human face in the image. For the calculations of the average face size, we first employ a technique called connected component labeling which scans the image for each disconnected region and label them as unique objects. The average size is then calculated by simply checking each region and then taking the average area of all the regions. The ratio of a person's face's width to height is usually 10:16, therefore if we get an extraordinary ratio for the average region size; we will adjust that number to closer match this ratio. With that box size, we rescan the entire image and filter out regions which are bigger or smaller than the box area (within a certain amount of tolerance). Ultimately, up to this step we still cannot determine if the regions are indeed faces. The subsequent step will perform various feature checks and gradient matching to finally confirm whether or not a particular region is a face.

3.3 Face Detection using Neural Networks

We decided to train our data using a neural network and have it do template matching for face detection because its formulation produces better results compared to other alternatives available such as normalized correlation. Given the region of the face and its size, we can intelligently estimate the facial features and extract them to train the neural network. A sample of about 200 face images from the Yale face database is used for training the network. The features are extracted and then quantified by their attributes (for example the angle of the eyebrow, width of the mouth, etc) and saved. We fed the neural network with block cuts of size equals to the average size of the faces in the image, and then iterated that block over the entire input image. In each of the iteration, feature areas are extracted and checked respectively with those of the training data. If they match within a certain threshold, we can safely assume that a certain region is a face given that at least 75% of the features are identified.

While the solution appears to be simple, we ran into several problems. Overlapping faces was a major problem because some of the features on the face are hidden and replaced with those of another face. We've managed to somewhat deal with this by removing regions that we've looked at before. Assuming that the faces do not overlap too much, the remaining area of the other face should be sufficient for the neural network to identify it as a face. The other problem is training the neural network with negative examples. Non-face examples span a very large space, thus making it difficult to collect solid counter-examples of faces.

To overcome this, we trained neural network by first running the algorithm on images with no faces in them, add non-face patterns that were correctly labeled as faces to the training set as negative examples, and repeat. From there, every time the algorithm is ran, the space of the negative example will grow dynamically. Finally, we ran into a problem of inconsistent lighting, where some parts of the faces are too dark. This is quickly and easily fixed by implementing a histogram normalization algorithm to combat the lighting changes.

3.4 Results of Face Detection Algorithm

Overall, after a test run with 20-30 test images, we've found that our algorithm can detect faces in an image with 82% accuracy. Some images returned false positives having detected other objects of similar color as the skin as faces. Others missed a few face here and there because some of the faces were too small compared to every other face in the image (this can be fixed if we implemented a dynamically sized block to scan the image instead of a static one calculated by the average face size). The results are promising, and we can safely use this face detection algorithm for the smile detection task as discussed further ahead in our paper.

4. DETECTING SMILES USING MACHINE LEARNING TECHNIQUES

Concurrently, we extended our face detector, by adding a second component to our project: smile detection algorithm [4]. In this part of the project, we investigate if a machine learning algorithm can deliver a smile recognition system usable in everyday life. We employ computer vision techniques (some of which were used for the face detection algorithm earlier) to accomplish this. Although current smile detection technology is deemed reliable, our goal is to be able to implement our very own detector such that it comes close to the existing technology in hopes of improving them.

4.1 Method

The algorithm utilizes faces already extracted from pictures, but for the interests of efficiency this was implemented as a separate module from the face detector. We used a collection of 134 smiling and non-smiling faces found throughout the Web. These pictures are then split this into two sets: 46 as training images and 88 as testing images. Each face is at least 250 pixels wide and 350 pixels in height so as to ensure we could extract and measure out the facial features. Ten features of the face were identified that are potentially vital in determining a person's smile, and we quantified them using image analysis algorithms, and with anomalous input, calculated the values manually. These attributes along with their values are then compiled into a spreadsheet as input for our subsequent machine learning algorithms. We've considered the following features as attributes for the trainers:

- Percent of teeth given an image of a mouth
- Normalized, average minor/major axis diameter of eyes (2 features)
- Nose height and width (2 features)
- Upper/Lower lip curvature (2 features)
- Does the cheek fold (i.e., when smiling the line near the nose and connects to the cheek is prominent)?
- Are there forehead wrinkles (e.g. from lowering eyebrows in anger)?
- Vertical distance from inner tear duct to eyebrow

The data that was collected was trained and tested with three different classifiers: Multilayer perceptrons (artificial neural network), Naïve Bayes classifier, and a pruned decision tree using the J48 algorithm. Following are the helper methods used to quantify information, and then the results and analysis of each run of the classifiers.

4.2 Feature Quantification

To figure out the percentage of teeth given a picture of just the lips portion of our face was not a trivial task. We employed the segmentation algorithm used for the face detector and modified it so that we could train on different colors.

Calculating the curvature of the upper and lower lips was done with the help of an edge detector. Here, we use the canny edge detector algorithm for the task as it gave us the most accurate results. Afterwards, we scan the results to obtain the left and right-most point that the lip starts and ends, respectively. Using the highest point of the lip, we can calculate the angle created from the three points, and do the same for the lowest point. Having the edge detected version of the image also helped us with finding out whether the given face had forehead wrinkles or cheek folding by doing another scan for lines at predicted areas of the face.

The facial detector also facilitated with finding the eyes, nose and helped us calculate the dimensions of the features in a similar manner to the curvature calculation, sans adding the third point.

4.3 Results and Analysis

Once all the information for all the picture samples has been quantified, we transferred our training and test data into Weka, open-source data-mining software with many machine learning classifiers. [5] Below are the results after running our training and running the three classifiers using the test dataset:

	J48	Naïve Bayes	ANN
RMSE	0.4366	0.2888	0.3783

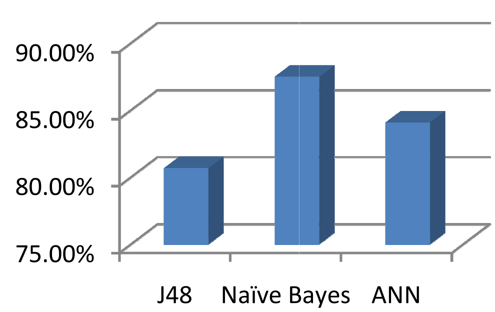

Figure 2: Percent of Correct Classification and Root Mean Squared Error (RMSE)

Overall, the most accurate classifier was the Naïve Bayes classifier which classified 77 out of 88 images correctly, while the least reliable was the decision tree with only 71 correctly classified faces. In the middle was the neural network with 74 classified correctly. The RMSE calculated represents the global error of the classifiers on the test data.

From our testing data, we find that the most influential attributes for all the classifiers is the upper lip curvature, and the percentage of white in the lips portion of the picture. Of course, they are not the sole contributors to the classification, as the below figure for the distribution of 'yes' smiles relative to the upper lip curvature and percentage of whites show:

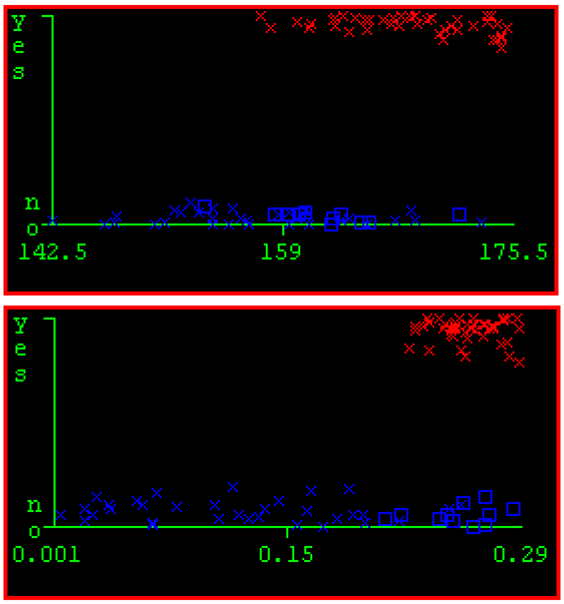

Figure 3: Upper Lip Curvature (above figure) and Percentage of White (below figure) vs. Smile Classifications with Jitter For Clarity [Data from Bayes Classifier]

This is not surprising since a cursory look through the images does indeed show a strong correlation between the two attributes and whether the person is smiling or not. It is evident, however, that the percentage of white was not as strong as there were more negative outputs overlapping with the positive ones.

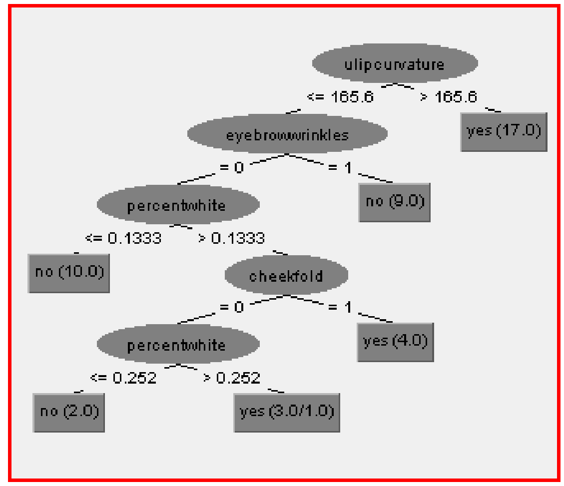

Figure 4: Pruned Decision Tree

We were quite surprised at the performance of the Bayes classifier as we thought traditionally neural networks performed best when it came to facial analysis problems. However, neural networks depend on a large dataset, which we did not have, so it could have factored into its inferior performance. Conversely, a smaller dataset is more effective when classified with Naïve Bayes.

5. CONCLUSION

This paper presents a method for identifying smiles in an image. Our method uses techniques from Digital Image Analysis to recognize and quantify facial features. We use Machine Learning techniques to train and test our smile detection.

When we were designing the attributes for the classifiers, we kept in mind that the attributes that most likely determined the smile outcome would be the lower lip curvature and the percentage of white in the mouth. From our results though, it is clear that our inductive bias had a partial influence (i.e. the percentage of white found in the mouth was a strong attribute, but not the lower lip curvature) on our classifier.

We found Digital Image Analysis techniques perform with 82% success in finding and segmenting faces. Moreover, we are successful in classifying smiles using J48 with 80.7% success, Artificial Neural Network's with 84.1%, and Naïve Bayes with 87.5%. Strengths of the Artificial Neural Network are the ability to recognize patterns using ten facial features described in Section 4.1. Decision trees are best towards a yes or no goal, and here we look for the existence of a smile.

6. REFERENCES

[1] T. Mitchell, *Machine Learning*, McGraw-Hill 1997.

[2] A. Kapoor and R.W. Picard, Real-Time, Fully Automatic Upper Facial Feature Tracking, To Appear in the Proceedings of The 5th International Conference on Automatic Face and Gesture Recognition, 2002.

[3] J. Whitehill, G. Littlewort, I. Fasel, M. Bartlett, J. Movellan, *Developing a Practical Smile Detector*.

[4] "Weka." http://www.cs.waikato.ac.nz/ml/weka/