Greg Crow, Corey Grief
EECS 395/495: Biometrics
Prof. Xin Chen
12/1/2015

Twin Face Recognition using multiple-feature detectors and FLANN

Introduction

In this paper we propose a combination of different feature detection algorithms in conjunction with a k-Nearest Neighbors to distinguish between a pair of twins. By doing this we are able to successfully classify between many pairs of twins with a high successful classification rate.

Related Work

Twin facial recognition is a difficult problem in the current Biometrics landscape, as presence of two users with two incredibly similar templates can lead to many misclassifications. Identical twins represent the worst case scenario for current facial recognition systems, and most current solutions cannot differentiate between twins compared to systems without twins in the dataset. However, some work has been done to improve existing system performance on twin data.

Context plays a large role in improving most biometric systems. Consistent context improves feature detection and analysis, which leads to more accurate comparisons over the templates in the system. This is especially true of twin data, where the differing features are minor, and can easily be obfuscated by differing contexts. According to “Double Trouble: Differentiating Identical Twins by Face Recognition” (Paone, 2014) and “Distinguishing Identical Twins by Face Recognition” (Phillips, 2011), consistent lighting, facial expression, and time of image have significant impact on the error rates of systems when a twin dataset is used. Error rate improvements of 5-10% were cited for each of the various contexts.

Identical twins often vary their appearances in very minute ways, such as scars, blemishes, or wrinkles. According to “Analysis of Facial Features in Identical Twins” (Klare, 2011), these features are not generally used to determine differences in standard individuals, but could provide features with which to distinguish identical twins. This work suggests Level 2 features and blemish detection to accomplish this.

Idea and Implementation

Based on our survey of prior work, we aimed to create a system for differentiating twins in two steps. 1. Determine which set of twins an input template corresponds to, and 2. Classify the image between the two individuals using alternative features such as blemishes and level 2 features. We focused on step 2 for the purposes of this project, as we assume that a modern biometric system can accurately determine which set of twins an input belongs to. We accomplished this step by first detecting features using a combination of SIFT, SURF, and ORB feature detection. These features were then used to create a biometric template for the training data, acting as an enrollment for that user in our system.

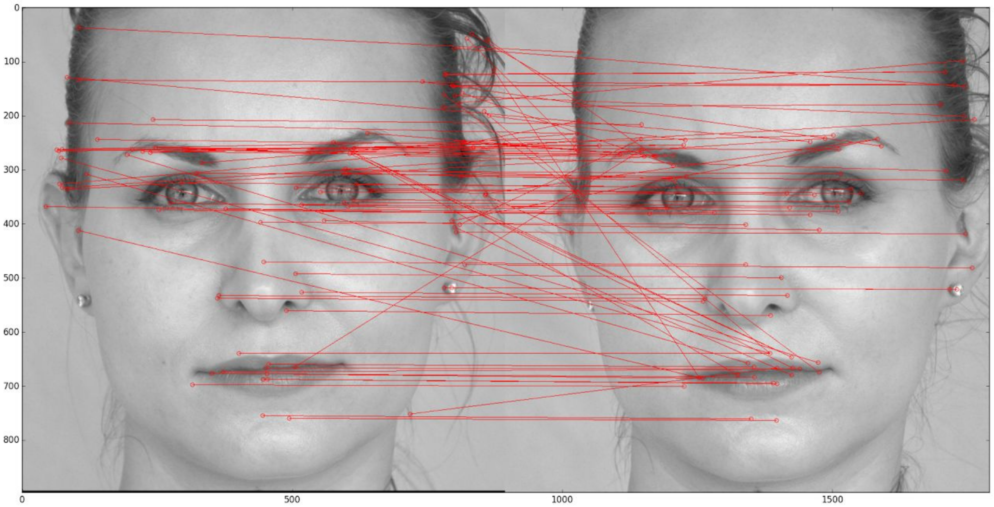
SIFT is a feature detection algorithm that extracts keypoints from an image as follows: Scale-space filtering is used by using difference of gaussians for the image with various sigmas. This acts as a blob detector to find potential key points It also creates an 8 bin orientation histogram with several measure for robustness against illumination, rotation, etc. as a descriptor for each keypoint.

SURF improves upon SIFT using wavelet responses in the horizontal and vertical neighborhoods. SURF is good at handling blurring and rotation, but not as good at handling viewpoint and illumination variances.

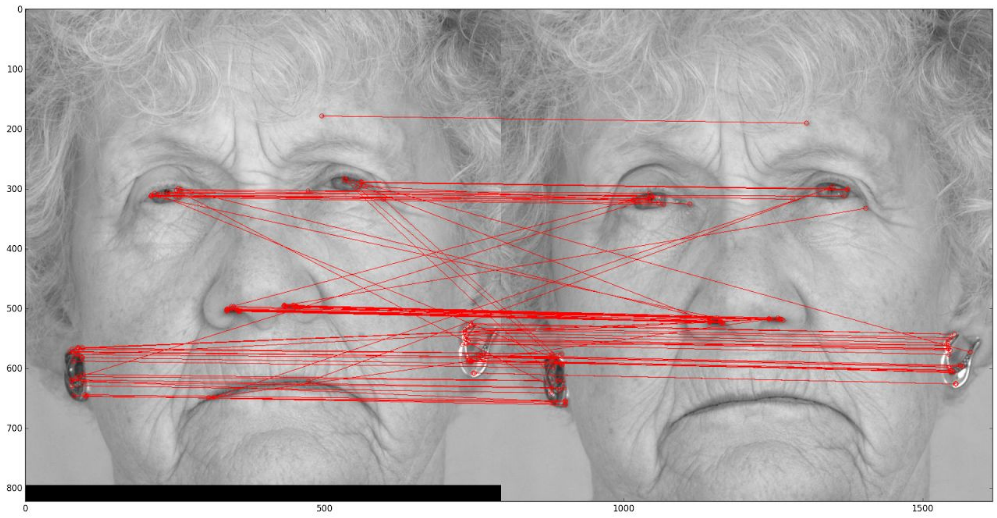
ORB is a corner-detection algorithm using the ID-3 Decision tree algorithm. It is very fast but is not extremely robust to a lot of noise. It is robust to rotation.

These features combined do an excellent job at gathering the major key points for traditional classification, but with a reduced threshold they also are excellent at detecting imperfections in a face for the lower-impact keypoints. These are the key points of interest for us. Even though the majority of the major points will be almost-perfectly matching between the twins, the lower-impact keypoints are not.

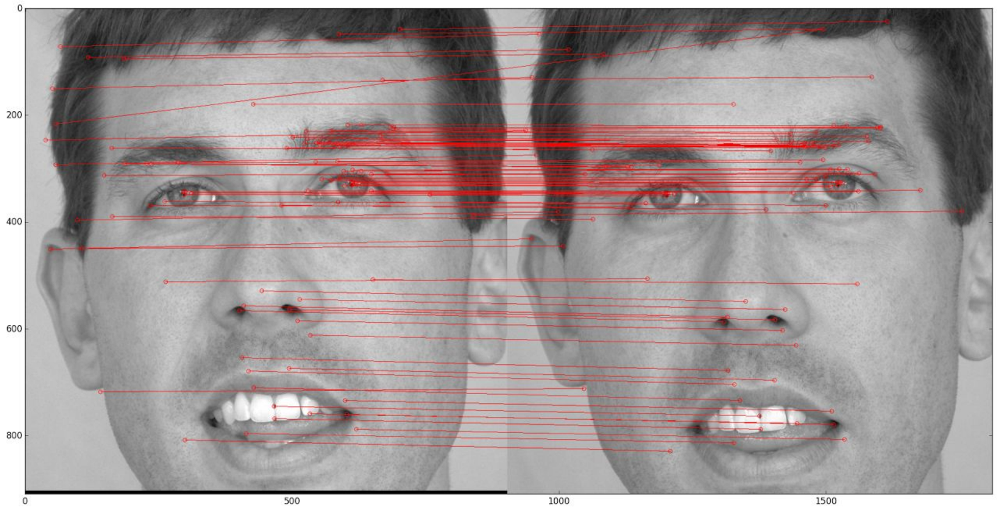
In order to classify an input image, we first extract all of the features in the same way as the “enrollment” step. We then input the features into a classifier. We first attempted using a deep-neural net for learning the weight of the features, but the nature of the data led to severe overfitting. We instead, use a k-Nearest Neighbor classifier in order to determine which template the input image most closely matches. Our distance measure for this classifier uses FLANN, or fast library for approximate nearest neighbors. This optimizes the nearest neighbor search for high-dimensional data. Each FLANN match produces a distance, which is a similarity score for a pair of feature points. This is normalized to a value between 0 and 1. We also introduce a penalty score for features that have no match. The distance measure is used to classify the input template. Examples of FLANN matching templates for the 100 most important features can be seen in the following figures. Lines that go between roughly the same point on each face have a low distance score, while lines that do not have a high distance score.



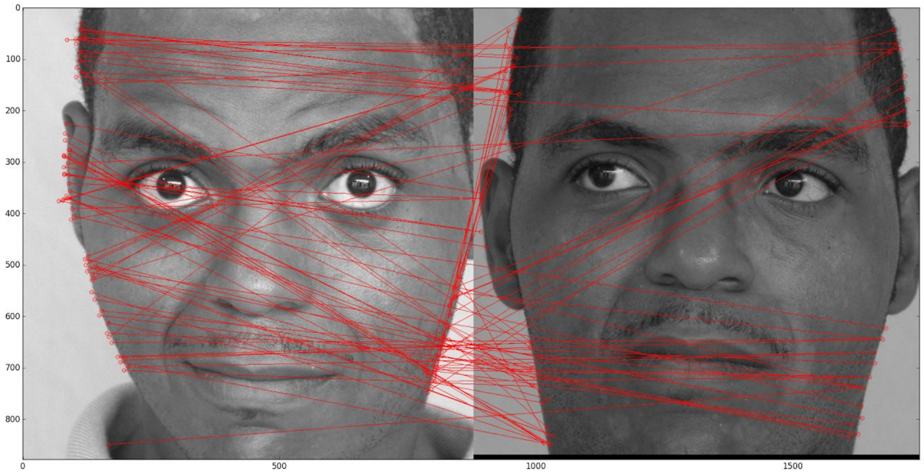
SURF: same individual



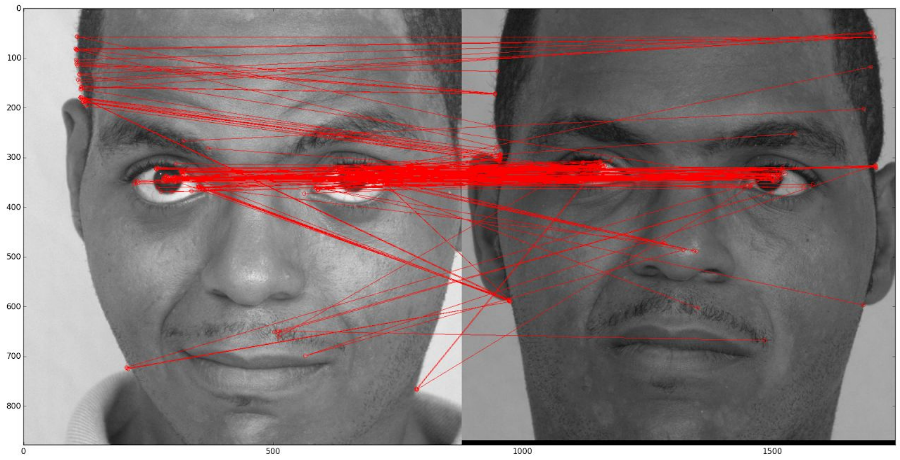
ORB: same individual



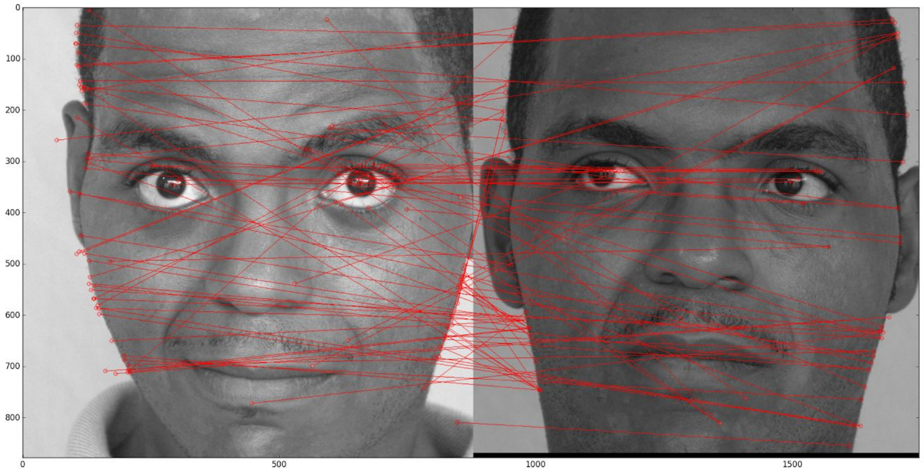
SIFT: same individual



SURF: different twins



ORB: different twins



SIFT: different twins

As you can see from these figures, distances are relatively low for key features in all 3 feature extraction techniques for the same individual as evidenced by the primarily horizontal lines matching features on the faces. For different twins, SIFT and SURF display major distance increases evidenced by the large variance in line slopes and locations of matched features. ORB remains roughly accurate with different twins, though some increase in distances does occur.

Experimental design

For our experiment, we took 25 pairs of twins and separated their images into training and testing sets. Our training set included one or more images of each twin and our testing set included a different image of each twin. Based on the best practices found in prior work, we used the front-facing, neutral facial expression, controlled lighting images for each individual. We then trained our classifier on the images in our training set, and tested its performance on each image in our testing set.

Our experiment was designed primarily to classify an image given the twin group it belongs to. For each image in our testing set, we provide the true label for which individual it is. We then input the image into our classifier, which provides a distance between the input image and the template images. We report the winner and collect the scores for each template in order to aggregate the error in the system. Our final correct classification rate is 1-((# of errors)/(# of trials)).

We performed this experiment on two of the feature extractors described above (SURF, SIFT), as well as a combination of the three in order to assess the best strategy and the relative error rates between the methods. We also briefly examined the impact of glasses being present in both the input image and the training set through a few users.

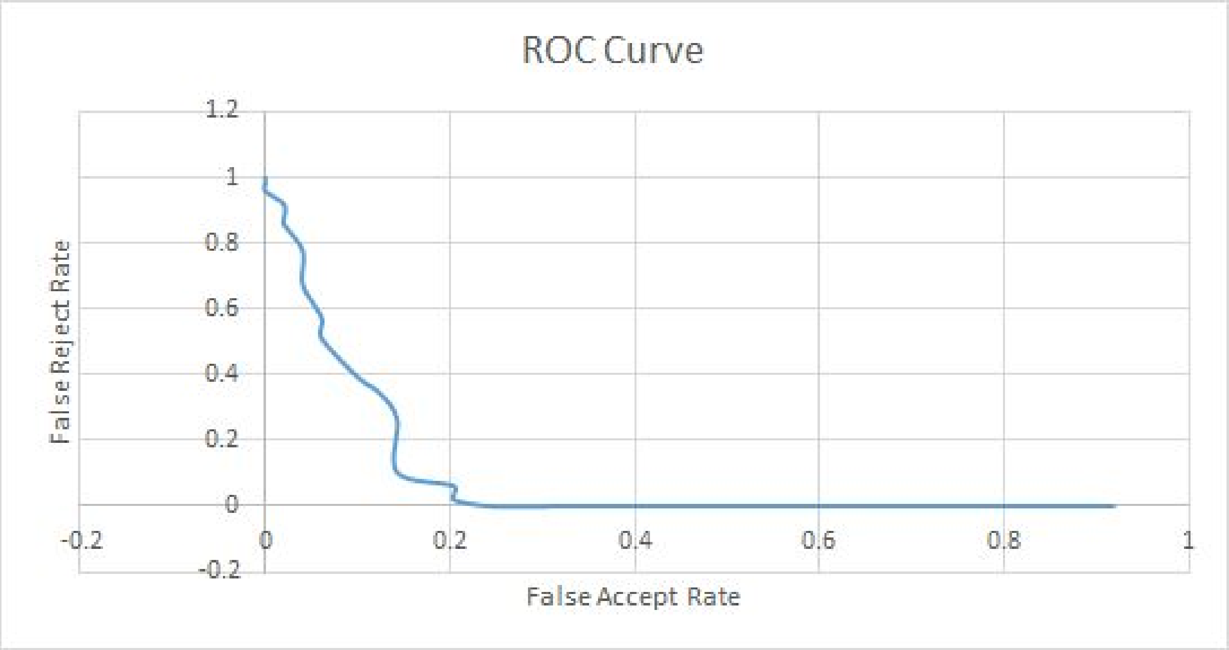
Testing results:

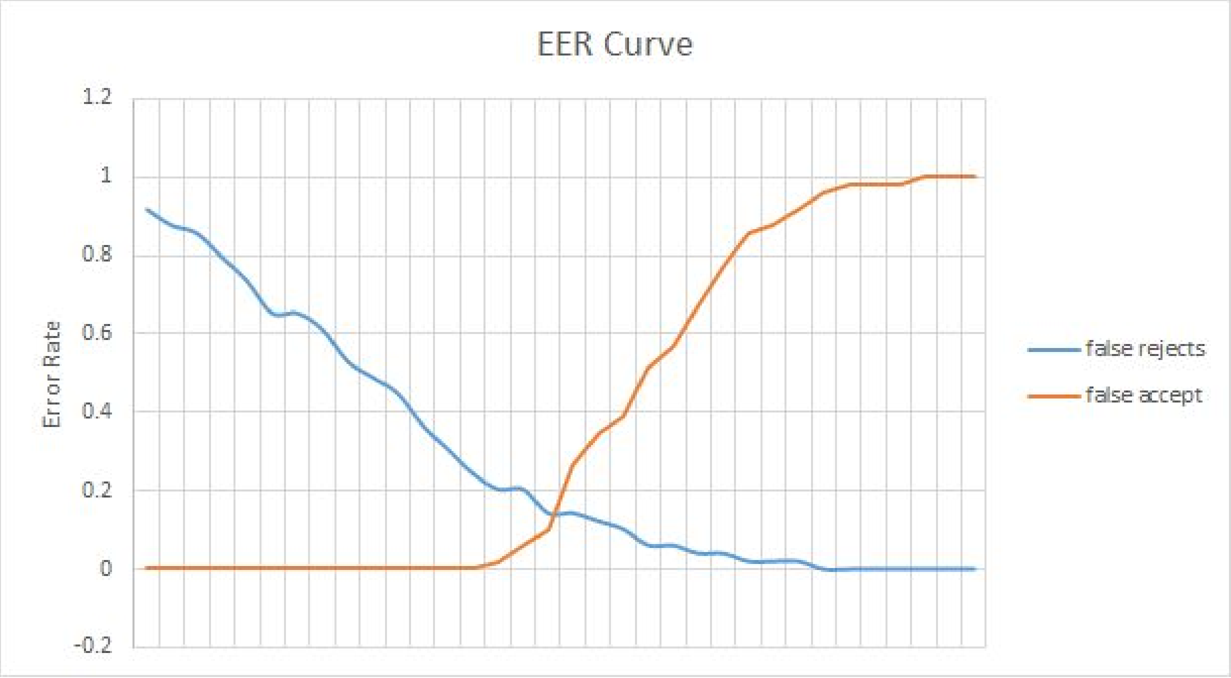
After running our experiment on the feature extractors we measured the following results.
Classification error rate was calculated as percent of incorrect classifications using the k-NN classifier with FLANN. The results were:

Sift Error rate: 0.06122
Surf Error rate: 0.4694
Combo Error rate: 0.0408

As you can see, the Sift algorithm outperformed the surf algorithm by roughly 40%. In conjunction, however, these algorithms performed extremely well, obtaining a classification rate of 95.92%

Presence or absence of glasses in either the testing or training sets did not seem to have an effect on the classification. This could be promising for future work, as glasses can often pose problems in face-recognition systems.





The equal error rates for the combo algorithm was approximately 15% This is very modest despite the high successful classification rate exhibited. We will discuss some potential reasons for this in the following section.

We also performed a T-test on the mean differences between the distance measures for each of the twins in the pair for the combo distance metric. We operated under a null hypothesis of 0 difference between the distributions at a significance level of 0.05. The t test produced a T value of -13.66, with a P value of < 0.00001, which is significant at $p < 0.05$. This shows that the distance for the correct twin is statistically significantly less than the distance measure for the incorrect twin. This suggests that our distance metric is a useful one in classifying individual twins between the two in their pair.

Conclusion:

The system performed remarkably on classification, correctly classifying almost 96% of input images to the correct. There was also a statistically significant difference in the distance-metric between pairs of twins. This is incredibly promising for the second phase of our proposed system, in which an individual is identified between a pair of twins.This classification rate is very high and should allow for more robust twin differentiation in the future.

We did observe a relatively high equal error rate of ~15%. This can be attributed to a few reasons. 1. The distance measure is not constant between sets of twins. The equal error rate cited is for all sets of twins, and the method by which we extracted features puts weight on the small nuances that differ between twins. These vary greatly between each twin, let alone sets of twins, so the distance measure can fluctuate based on these nuances. This means that imposing a thresholding function like classic biometric systems is not necessarily appropriate. A more appropriate solution would be to have a threshold per pair of twins, as this would move our EER towards the classification rate. However, this was not possible given the size of our dataset.

We had a few limitations to this work, the greatest of which was the dataset. The high quality of the images led to massive execution times in spite of FLANN due to the lazy nature of Nearest Neighbor classifiers. Furthermore, based on many of the previous work recommendations, we did not want to use the majority of our dataset for testing or training. Gathering additional images that are consistent with one another would improve the robustness of our classifier and allow for cross validation and more thorough testing. It would additionally help prevent overfitting for use with a deep neural net. DNN technology has shown promising advances in image recognition and processing, and could be valuable in this space.

Citations

https://www3.nd.edu/~kwb/PaoneEtAlTIFS_2014.pdf
https://www3.nd.edu/~kwb/PhillipsEtAlFG_2011.pdf
http://www.cse.msu.edu/biometrics/Publications/Face/KlarePaulinoJain_AnalysisFacialFeaturesIdenticalTwins_IJCB11.pdf