THESIS

EVALUATING THE PERFORMANCE OF IPHOTO FACIAL RECOGNITION AT THE BIOMETRIC VERIFICATION TASK

Submitted by

Keegan P. Patmore

Department of Computer Science

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Spring 2012

Master's Committee:

Advisor: Ross Beveridge

Bruce Draper
Geoff Givens

ABSTRACT

EVALUATING THE PERFORMANCE OF IPHOTO FACIAL RECOGNITION AT THE BIOMETRIC VERIFICATION TASK

The Faces feature of Apple's iPhoto '09 software uses facial recognition techniques to help people organize their digital photographs. This work seeks to measure the facial recognition performance of iPhoto Faces in order to gain insight into the progress of facial recognition systems in commercial software. A common performance evaluation protocol is explained and performance values are presented. The protocol is based on performance measurements of academic and biometric facial recognition systems performed at the National Institute of Standards and Technology. It uses the data set developed for the Good, the Bad, & the Ugly Face Recognition Challenge Problem which contains images with varying levels of facial recognition difficulty. Results show high performance on the hardest faces to recognize, less than peak performance on the easier faces, and overall less variation in performance across varying levels of difficulty than is observed for alternative baseline algorithms.

TABLE OF CONTENTS

1 Introduction

This work focuses on software sold by Apple Inc. called iPhoto[2], and more specifically on one feature of iPhoto called “Faces”[1]. The purpose of iPhoto is to help consumers organize and manage their digital photographs. iPhoto’s Faces feature uses facial recognition techniques to identify people in photos, so users can easily find all the photos of a particular person in their collections. The goal of this project is to quantify the performance of iPhoto’s Faces feature (called Faces from here on) at the task of traditional biometric verification on data for which there are known performance standards. The task of biometric verification [5] is to use biometric data, faces in this case, to verify the identity of a person.

iPhoto is a commercial software product not instrumented for testing, and is intended for non-scientific consumer use. In order to begin investigation of iPhoto’s performance, a couple of assumptions are necessary. The first assumption is that it even makes sense to measure iPhoto’s performance at the task of facial recognition. This measurement would give some insight into how much progress has been made at introducing facial recognition to the mass market consumer space. The second assumption is that iPhoto is trying to solve a problem similar to what is commonly called “Facial Recognition”. This is a necessary assumption if the measurement is to be reasonably compared to performance measurements on other facial recognition software and algorithms.

iPhoto was not designed to do the task of verification on biometric data. This is a very important realization to keep in mind with regards to this experiment. The task iPhoto was designed for is to help consumers organize their photo collections. More specifically, iPhoto Faces is designed to recognize people’s friends within their photo libraries so that users can easily find pictures of people they know. To do this well, Faces is probably optimized to recognize a relatively small number of people. It is also apparent from the user interface that Faces is designed to refine it’s recognition techniques as more pictures are presented to it. Faces learns over time from the user how to better identify the faces of that user’s friends. Faces does this by getting feedback from the user about whether it has correctly recognized someone in an image. The tests presented here short circuit much of that mechanism, by asking Faces to recognize a large number of people and not providing feedback to Faces about the correctness of matches returned. It should be said again that the tasks presented to iPhoto in this experiment are not the tasks it was designed to do.

2 Background

2.1 Facial Recognition Overview

The term Facial Recognition has different meanings to different groups of people. To those people who are not Machine Vision researchers it means recognizing a person's identity by seeing their face, in an image, a video, or in person. To Machine Vision researchers it means treating a face in an image as a pattern of data, and matching that pattern to other known patterns to establish the identity of the person.

A complete, or fully automatic, facial recognition system has four major components [6]. The first component is face detection or tracking. In the face detection stage the task is to locate the faces in the input image, if there are indeed faces in the image in the first place. Locating a face in an image means being able to identify the subset of pixels in the image which are part of the face pattern. This step is necessary because later steps may use techniques specific to faces. The second component is the normalization, or alignment, component. Once a face has been located it is necessary to adjust the data (the face's pixels) to better match the design of and assumptions made by later components. The orientation, or pose, and size of the face may be changed to put the face within a 'normal' pose and size range. The color data recorded in the face may also be adjusted, to normalize the face's lighting or even shift it into a more convenient color space.

The third component in a facial recognition system is feature extraction. A feature is any piece of salient information, useful for matching the face to a known face. Much work has gone into development of feature detectors for use during this process; the question of what information to extract from a face pattern is central to the design of a face recognition system. There are two different broad approaches used to extract features, namely feature based and appearance based methods [5]. In feature based methods specific salient points are chosen as a way to represent the face. These are commonly geometric information about the location of and relationship between interesting locations on the face, such as the eyes, nose, and mouth.

Appearance based methods treat the whole face as a pattern and work with properties of that pattern. In an appearance based approach the face is converted into a different representation (which may not look anything like a face) where the pattern is projected onto a series of basis vectors chosen to reduce the complexity of the representation, removing (hopefully) irrelevant information in the process.

The fourth and final component of the system is feature matching. This component matches the features extracted from an input face to those features that were extracted from other faces the system has already processed. If the features of the input face are a close enough match to the features of a known face, the system can say that the input face belongs to the same person as the known face thus establishing the identity of the input face.

3 Performance Measurement

The work described herein is based on the procedures developed by Johnathon Philips and others [8], for measuring the performance of face recognition algorithms. Projects such as the Face Recognition Technology (FERET) program [11], and the more recent Face Recognition Vendor Test (FRVT) [12] implemented similar techniques to measure the state of the art in facial recognition at different points in time.

3.1 Face Recognition Technology Program

The Face Recognition Technology program (FERET) [11] was the initial work on measuring the performance of face recognition systems. FERET developed a testing procedure and a large database of images on which to apply that procedure. This procedure and database were used in a series of tests to measure the performance of face recognition software submitted by external researchers. The stated goals of these trials were threefold: to assess the state of the art, identify future research areas, and measure algorithm performance.

Along with the overall goals the FERET program set out four rules for conducting such evaluations, summarized in [10]. The first rule was that evaluations be designed and conducted by a group which is independent of the group whose algorithm is being tested. The second rule was that groups working on an algorithm not have access to the test data that will be used in the evaluation. The third rule was that the design and techniques used to administer the evaluation be published. And the fourth rule was that results be reported in a way which revealed meaningful differences between the systems tested. These rules were designed to ensure that the evaluation was fair, that it produced useful results, and that facial recognition systems were general purpose systems not built just to pass the test.

The FERET program conducted three rounds of testing[11]: August 1994, March 1995, and September 1996. The image database grew from 316 individuals in 1994 to 14,126 images of 1,199 individuals in the 1996 test. The FERET tests provided algorithm researchers with a standard and general way to test their systems and measure progress. As the performance of the tested systems increased, the data set used to test them needed to change as well. This is because of what Phillips calls the *three bears problem*: If the test data set is too easy good performance can be achieved by simply tweaking existing systems, but if the test data set is too hard no system will succeed; the difficulty of the testing data must be just right. Difficulty of test data sets was adjusted by choosing a subset of the data set's images which was judged to have an appropriate level of difficulty to create a meaningful evaluation.

3.2 Face Recognition Vendor Test

The Face Recognition Vendor Test (FRVT) built on the evaluation methodology established in the FERET series of evaluations [10, 8]. By this time facial recognition systems were starting to leave the laboratory and enter the commercial market to be used in real world applications. The FRVT tests had the same goals as the FERET program, and were designed to have the same role in facial recognition research, namely to provide reliable and fair evaluations of the current state of the art and to point to future research areas.

Similar to the FERET evaluations, there were three rounds of FRVT testing. The first round was in 2000, the second in 2002, and the third in 2006. Over this time the test data set evolved to keep pace with the capabilities of the systems being tested, and the *three bears problem* motivated continual expansion and updating of the testing data sets. By the time the FRVT 2006 evaluation was done, the test data had expanded to several subsets designed to test permutations of expression and control over lighting conditions. One of the subsets contained 3D face measurements as there was a need to test systems which use such data. The largest subset was 108,000 images of 36,000 individual people. This increase in test set complexity reflects the advances made in facial recognition systems in the twelve years since the first FERET test in 1994.

Throughout both the FERET and FRVT evaluations, the same overall technique was used to test systems at the verification task. First, images in the target set were enrolled into the system along with their associated identity. Second, images from the query set were given to the system along with a claim of identity. The system's response was recorded and used to measure performance. This method ensures that images from the query set are unknown to the system before they are presented in the test. Both the target and query sets used for testing were not released to anyone prior to testing, further ensuring that the images presented from the query set were indeed novel to the systems in the test.

3.3 Sample sets

The comprehensive approach to performance testing done by Phillips [9], requires three sets of biometric samples. These samples can be any sort of biometric data, but the same three sets of samples are needed for each type of data. One set is called the gallery (or target) and the other two are called probe (or query) sets. The gallery is the set of samples the system is initially made aware of. The first probe set contains novel samples from people who are in the gallery set. The system should be able to identify or match these people. The second probe set contains samples from people who are not in the gallery set; the system is expected to not be able to match these samples. The result of not being able to match a sample to a person will depend on the system's design goals.

3.4 Facial Recognition Tasks

There are two distinct tasks a facial recognition system can engage in [5]. These tasks are the identification task and the verification task. The goal of the identification task is to find out which person the sample is from, assuming that the system has a method for representing identity. The system is presented with a sample from a probe set and is expected to determine if that sample is from someone in the gallery, and if so, identify the person. This process requires the system to search it's database of known samples and try to match the novel sample to a sample in the database. This is a one-to-many comparison. The goal of the verification task is to verify a person's identity. The system is presented with a sample from the probe set and a claimed identity. The system is then expected to confirm or deny the claimed identity of the sample. The verification task does not require the same search, since the claimed identity is already known. The system can instead perform a one-to-one comparison between the presented sample, and the samples(s) it has already stored for that identity.

4 Motivation

Commercial facial recognition software is already available to the broad consumer audience. A relatively direct comparison between iPhoto Faces and traditional biometric verification systems is valuable for three main reasons. The first reason is to gain information about what progress has been made on facial recognition in consumer products. The second reason is to gain some insight into the level of performance necessary to make facial recognition useful to a broad audience, in their daily lives. The final reason is to provide some information about how the problem of organizing a photo album differs from the more strictly defined problem of biometric verification.

A quick internet search will find product reviews of iPhoto's Faces feature, such as [13]. These reviews provide useful information about the product, especially about the features that matter to consumers. Reviewers focus on things such as interface design, and how well Faces seems to work on the reviewer's personal photos. They also note situations where the reviewer encountered unexpected or incorrect results. It was noted, that with enough prodding, iPhoto Faces will recognize cats [3]. While this sort of information is useful and interesting to people thinking of purchasing iPhoto, it is not a quantitative performance measurement. Such a measurement would say something like "On a given data set, the software correctly identifies the person in an image 75% of the time". This sort of measurement would provide insight into how far facial recognition software has progressed in the consumer space.

It must be clearly and emphatically stated that *iPhoto was not designed to be tested as a biometric verification system.* The investigators are mindful of this point, despite the fact that drawing as direct a comparison as possible between iPhoto and traditional biometric face recognition systems is the goal of this work. Since it is acknowledged that this test is outside the apparent design goals of iPhoto, the reader is urged not to take the results as an absolute measure of performance but rather as a rough indication.

5 Data Set

To investigate the performance of iPhoto's Faces feature at the task of biometric verification using facial recognition, a data set of face images is needed. This experiment was conducted using a data set called: the Good, the Bad & the Ugly (GBU) [7]. The GBU data set was built from the same Notre Dame data set used in the Face Recognition Vendor Test of 2006 [12]. The GBU data set is composed of images of people in common place settings (outdoors and hallways) looking at the camera [7]. The data is partitioned into three subsets based on how difficult the faces in a particular partition are for machines to match. The Good partition has faces that are pretty easy to match, the Bad partition is harder and the Ugly is hardest. An amalgam of the top performing algorithms from the Face Recognition Vendor Test 2006 was used to classify the difficulty of the images used to create the GBU data set. By definition the images in the Bad partition were hard for those algorithms to recognize, but not as hard as the images in the Ugly partition. The data set also comes with some meta data files, which contain the ground truth of which person is in which images. Hence it is possible to randomly select a set of *people*, and get all the images of those people from the entire data set or from a particular partition.

6 Methods

The experiments conducted were designed to measure the performance of the underlying facial recognition algorithm, and not performance of other user convenience or fine-tuning features. At a high level, the design of the experiments was as follows. First, randomly select *people* from the data set. Second, retrieve two images of each person, and make the first image part of the query set, and the second part of the target set. Third, for each image in the query set make iPhoto find the matching image of the same person in the target set.

Table 1: A contrived similarity matrix.

	1_filename	2_filename	3_filename
face1	.5840	0	0
face2	0	.4822	.3502
face3	0	0	1.0

To support later analysis, the data extracted from iPhoto was in the form of a similarity matrix [12]. Table 1 is a contrived example of a similarity matrix in the form used for this experiment. Each row of the matrix represents an image in the query set containing a known and labeled person. Each column represents an image in the target set, containing an unlabeled face. Each cell in the matrix represents a matching, or pairing, of one query image (the row) and one target image (the column). The value actually in the cell is the similarity score for that cell's particular pair of images, rating how similar the two faces are. Higher similarity scores indicate that two faces are more similar. The cells along the diagonal of the matrix represent matches between a target and query image containing the same person, these are correct matches.

Note that these similarity matrices are fairly sparse, with most cells containing a zero similarity score. This arises because iPhoto only makes scores available for some pairs of images, not all possible pairwise combinations of one target and one query image. If iPhoto does not report a score for a pair of images, the corresponding similarity score in the similarity matrix is set to zero. Figure 1 is a screen shot of part of the Faces interface that

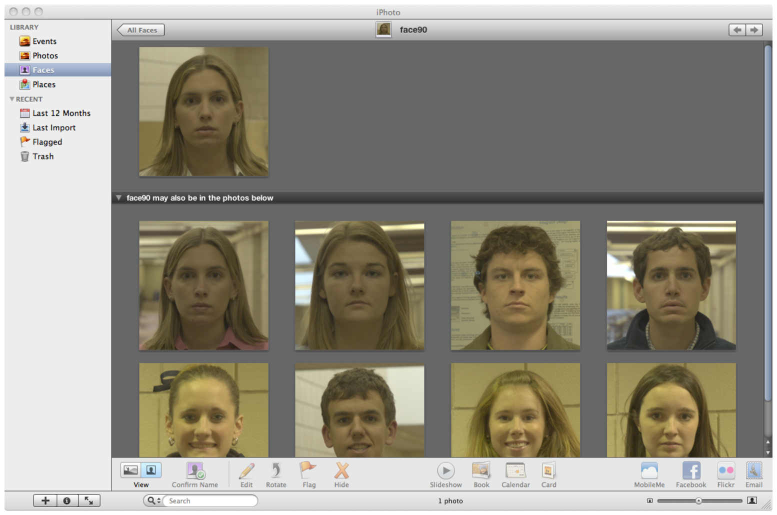


Figure 1: Example of iPhoto's Faces interface. The image above the dividing line is the query image. Images below the divider are the potential matches.

illustrates this situation. The image above the horizontal divider is the query image, those below are images in the target set which iPhoto thinks might be the same person. For this query image, these are the only target images for which a similarity score is available.

The machine used to perform the experiment was an Apple iMac with a 2.8 GHz Intel Core i7 processor and 16 gigabytes of memory, running version 10.6.8 of Mac OS X Snow Leopard. The software used was iPhoto '09, version 8.1.2. The images themselves are 3008 pixels wide by 2000 pixels tall and encoded in JPEG format. The scripts used for interfacing with iPhoto's database files and building the similarity matrix were written in Python. The R programming language was used to compute similarity scores and produce ROC curves.

Much of the experimental design depended on the specifics of how iPhoto stores data on the hard disk. It appears that all of iPhoto's data is stored in an archive file called a Library. More than one Library can be created, but iPhoto only uses one Library at any given time. The Libraries are independent from each other, so actions taken while one Library is loaded have no effect on other Libraries. Users can switch between Libraries upon starting iPhoto. This experiment was run multiple times with different sets of images, each run was done using a separate Library file. Library archives contain images that the user imports to iPhoto, meta data about the images, SQLite database files maintained by iPhoto, and anything else iPhoto needs to store. By default, imported images are copied into the Library. Several of the database files in each Library seem to be maintained by iPhoto for the purpose of providing data about the stored photo collections to outside programs. These database files and the data they expose are the basis for the design of this experiment.

6.1 Steps of the experiment

The first step in the experiment was to build the set of images for a particular run of the experiment. One of the Good, Bad or Ugly partitions was chosen to be the source of images for a particular Library. Meta data files included in the data set were read to get a list of all the people who had images in that particular partition. One hundred people were chosen at random, using Python's built in random library, from this list to make a smaller list called the experiment list. Two images were retrieved, from the appropriate partition, of each person on the experiment list. This resulted in a set of two hundred images, which contained two images of each person in the experiment list. The ground truth of which person is in each image was encoded in that image's file name, since the file name is easily accessible in iPhoto's interface and in databases within the Library which were later used to generate the similarity matrix. From this set of 200 images, 100 were used as the target set and the other 100 were used as the query set. Each person had one image in the target set and one image in the query set.

Once the above set of 200 images had been built, the images were imported into iPhoto. Since iPhoto Libraries store the images and related data that iPhoto is working with, a new Library archive was created using iPhoto's interface. This new Library stored the images and iPhoto's files for each particular run of the experiment. Multiple Libraries were used to distinguish different experimental runs. Once the new Library had been created, the set of 200 images was imported to that Library using iPhoto's user interface. Creation of the Library file and importation of images to that Library were separate steps. During this process, iPhoto copies each image into the internal storage of the Library archive and creates many other files within the Library which it appears to use for it's own book keeping purposes. Once the import is complete, the Faces feature automatically starts processing the new images. It appears that most, if not all, of the facial recognition work done by Faces is done at this stage of the import process. An animation in the main iPhoto interface lets the user know that Faces is still working.

Once Faces had finished processing the new images in the Library, the next step of the experiment was to enroll one image of each person into the target set for the experiment. This was done using iPhoto's user interface to label, or "name" the person present in a particular image. For each person in the data set, one image of them was labeled in this manner and the other image was left unlabeled. Oddly, the labeled images were actually the query set and the unlabeled ones were the target set. This is because iPhoto's interface is designed to do searches by identity. When the interface is asked to get all the images of a particular person, it is actually starting with the labeled image(s) and looking for unlabeled ones containing the same face. This is discussed in more detail in the description of how similarity matrices are built.

The next step was to use iPhoto's interface to search for each labeled person in the set of images. This was done by using the Faces interface to get all the for images one of the people in the Library, and was repeated for all one hundred people. Figure 1 above is an example of this process, Faces retrieves all images which contain faces potentially matching the face(s) which have a particular label. The results of this searching were not directly recorded. Rather, the purpose of these searches was instead to populate the database files which were read during later steps of the experiment. More specifically, this step caused similarity scores to be exposed in those databases. Those similarity scores were needed to make a similarity matrix. Note that it is likely that these searches are not actually doing face recognition, but are simply searching the results of the face recognition computations done when the images were imported into the Library.

At this point in the experiment, a set of images has been created and iPhoto Faces has been exposed to them and allowed to do it's work. The next step is to extract the results of that work from the Library archive and build the similarity matrix. Three SQLite databases were extracted from each Library, using tools built into the operating system. Between these three databases, there is enough information to build the similarity matrix. There is a field called "similarity" within the databases which ranges from zero to one, and it was possible to recover which pair of images this similarity field related. A script ran SQL queries on the three databases and was used to arrange the data into a similarity matrix like the one in Table 1. This similarity matrix is the data that was collected from iPhoto. This similarity matrix is also more sparse than traditional similarity matrices because similarity scores are not available for every pair of images. Scores are only available for image pairs which the interface has previously presented, during the previous search by identity step, as potentially containing the same person. For image pairs with unavailable scores, I set the score to zero in the similarity matrix. Note that the above mentioned similarity field was treated as a similarity score which describes how similar two faces are. This was a necessary assumption in order to build similarity matrices and compute verification rates using these methods.

Once similarity matrices had been built, they were analyzed to measure performance at the task of face recognition. For each experiment (one library, one similarity matrix), the *verification rate* was computed at a given *false accept rate* (FAR) [5]. The verification rate measures how many faces were correctly recognized, out of the total number of faces in the Library. A verification rate of 78 out of 100 means that there were 100 distinct faces in the set, and the system matched the query image to the correct target image for 78 of them. The verification rate is related to the False Reject Rate (FRR), which measures how often correct matches are rejected. The sum of the verification rate and the FRR equals one, since between them they account for all correct matches. The FAR is the rate at which false matches (claiming two images are of the same person when they are not) are acceptable. A FAR of 1 in 1000 means that for 1000 possible target-query image pairs, only one pair is reported as a match when it is in fact not a match.

Computing the verification rate has four steps. The first step is to compute how many non-match scores represent the desired false accept rate. To do this the total number of non-matching target-query pairs is multiplied by the false accept rate. The second step is to extract the match and non-match scores as two sorted lists, sorted with higher scores first. The third step is to get the Nth non-match score from the list of non-match scores, where N is the number of non-match scores representing the desired false accept rate (from step one). This score is the verification threshold. The fourth step is to count the number of match scores higher than the verification threshold, this is the verification rate (out of the total possible correct matches). This verification rate is the measure of performance produced by this experiment.

6.2 Significant Differences with Phillips' method

This process departs from the techniques used by Johnathon Phillips [8]. Phillips' technique is to enroll images in the target set without exposing the system to the images which will later be in the the query set. So when the system is presented with the query images, it has not seen them before. Under the technique used here, the system has seen both the target and query sets at the start. Labels are then used in an attempt to define the target set, with the remaining unlabeled images effectively becoming the query set. It is hard to tell if this process produces the expected two distinct sets of images, since the internal workings of iPhoto are unknown. But it does seem to have the desired effect since images in the target set are associated with an identity, and images in the query set are not.

Another significant difference from the technique employed by Johnathon Phillips is that the target and query sets are, in a way, backwards. Normally the procedure would be to take an image in the query set and see which image(s) in the target set match that query image. However in iPhoto's interface, we have to do the opposite. There is not a good way to select an unlabeled image (effectively a query image) and have it find the matching labeled image (in the target set). Instead one must *choose a label*, not a labeled image, but the label itself to search by. So it is a search by identity. The results of that search are first the labeled images from the target set, then unlabeled images from the query set which are a close enough match. So we are reversing the roles of the sets, and searching the query set for in image in the target set.

These differences are part of the reason for extracting similarity scores and building a similarity matrix instead of simply recording the results of each identity search in the iPhoto GUI. It does not avoid exposing iPhoto to the query set prematurely, but it does allow for more direct analysis without having to work backwards from a search by identity. The final downside to extracting similarity scores and building a similarity matrix is that iPhoto does not provide similarity scores for all image pairs. Similarity scores are only available for image pairs which would have been returned from the 'choose a label' search by identity mentioned above. There is effectively a lower threshold on returned similarity scores. It is the investigator's intuition that all such scores are computed and then a threshold is applied to remove unlikely matches from the search results. The methods used by Phillips assume that a similarity score is available for all image pairs, so this is another departure from his technique. This departure should not adversely affect the experiment, since the focus is on the higher similarity scores above the threshold.

7 Results

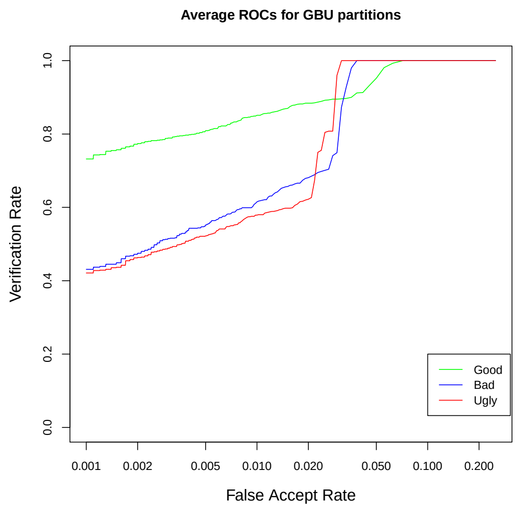
The experiment was run on thirty distinct sets of images. Each set was constructed from 100 randomly selected people, with two images of each person for a total of 200 images per set. These image sets were evenly divided between the Good, Bad, and Ugly partitions so there are ten sets of images drawn from each partition. After extracting similarity matrices, verification rates were computed for the range of False Accept Rates between 1 in 4 and 1 in 1000. This was used to generate the ROC curves in Figure 2, as well as to examine the verification rate at specific False Accept Rates to generate a quick view of the data. Table 2 contains a quick examination of average False Accept Rates. Even from this quick view it is apparent that performance is similar on the Bad and Ugly partitions

Table 2: Average verification rates at FARs of 1 in 100, 1 in 500, and 1 in 1000

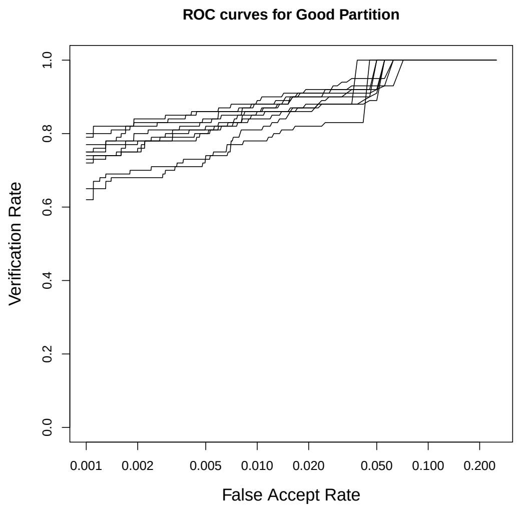
		Partition		
		Good	Bad	Ugly
FAR	1 in 100	85%	60%	58%
	1 in 500	77%	46%	46%
	1 in 1000	73%	42%	42%

Figure 2 shows the ROC curves generated by the experiments in detail. Figure 2a compares the average ROC curves from each partition. Each curve here is simply the average of the ten curves generated for a particular partition. Figure 2b shows the ten ROC curves generated from ten experiments on the Good partition. Similarly, figures 2c and 2d show the ten ROC curves from the Bad and Ugly partitions respectively.

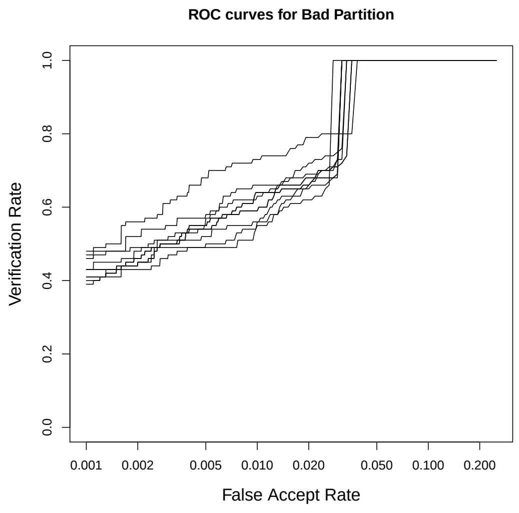
These ROC curves look different from traditional examples of such curves in two ways: the verification rate does not change smoothly but rather changes in a stair-step pattern as the False Accept Rate increases, and there is a point where the verification rate seems to jump to 100%. The stair-step pattern arises directly from the data, where a range of False Accept Rates all have the same Verification Rate. The sudden jump in Verification Rate arises because similarity scores are not available for all possible image pairs within an experiment. This causes a significant number of zeros in the non-match scores list. When one of those



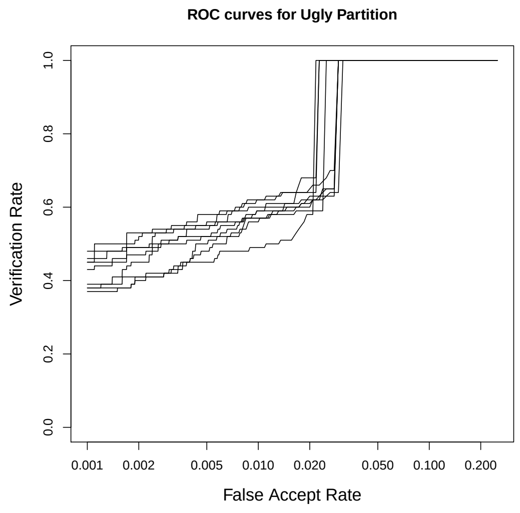
(a) Average ROC for each partition



(b) ROCs for Good partition



(c) ROCs for Bad Partition



(d) ROCs for Ugly Partition.

Figure 2: ROC curves: (a) average curves for each partition, (b) all curves from Good partition, (c) all curves from Bad partition, (d) all curves from Ugly partition.

zeros is chosen as a verification threshold, all match scores are greater than the threshold resulting in a verification rate of 100%. There are zeros in the non-match scores list because iPhoto does not expose similarity scores for all possible image pairs, as previously discussed in the Methods section.

7.1 Results in Context

The motivation for conducting these experiments was to attempt to understand via measurement the performance of iPhoto's Faces feature. Since the purpose of the GBU data set is performance evaluation and the data set provides performance benchmarks, it is reasonable to attempt a comparison between iPhoto and the GBU benchmarks. However, direct comparison is impossible. This is because the iPhoto experiments have a different sample size than the GBU benchmarks, and because the target and query sets used for iPhoto are different than those used for GBU. While both the GBU benchmarks and the iPhoto experiments drew their target and query sets from the same partition in the data set, the exact image pairs are different. What this means is that not every exact pair of images scored by iPhoto has a score from GBU, and vice versa.

The GBU challenge provides two algorithms as performance benchmarks, the Baseline and Fusion algorithms [7]. The Baseline algorithm represents a standard facial recognition algorithm, and is intended as a starting point for researchers working on the GBU challenge. The Fusion algorithm is an amalgam of the top performing algorithms tested during the Face Recognition Vendor Test of 2006, and represents the state of the art at that time.

Since the GBU Fusion algorithm is the one used for comparisons in the GBU challenge, Figure 3 shows both the performance of iPhoto and the performance of the Fusion algorithm using ROC curves. The ROC curves presented for iPhoto's performance are the same average ROC curves from the previous section. The red vertical bar on the ROC curves for iPhoto represents where the curve runs out of non-match scores as discussed above. Verification rates for False Accept Rates to the right of the red bar are misleading. It is critical to note

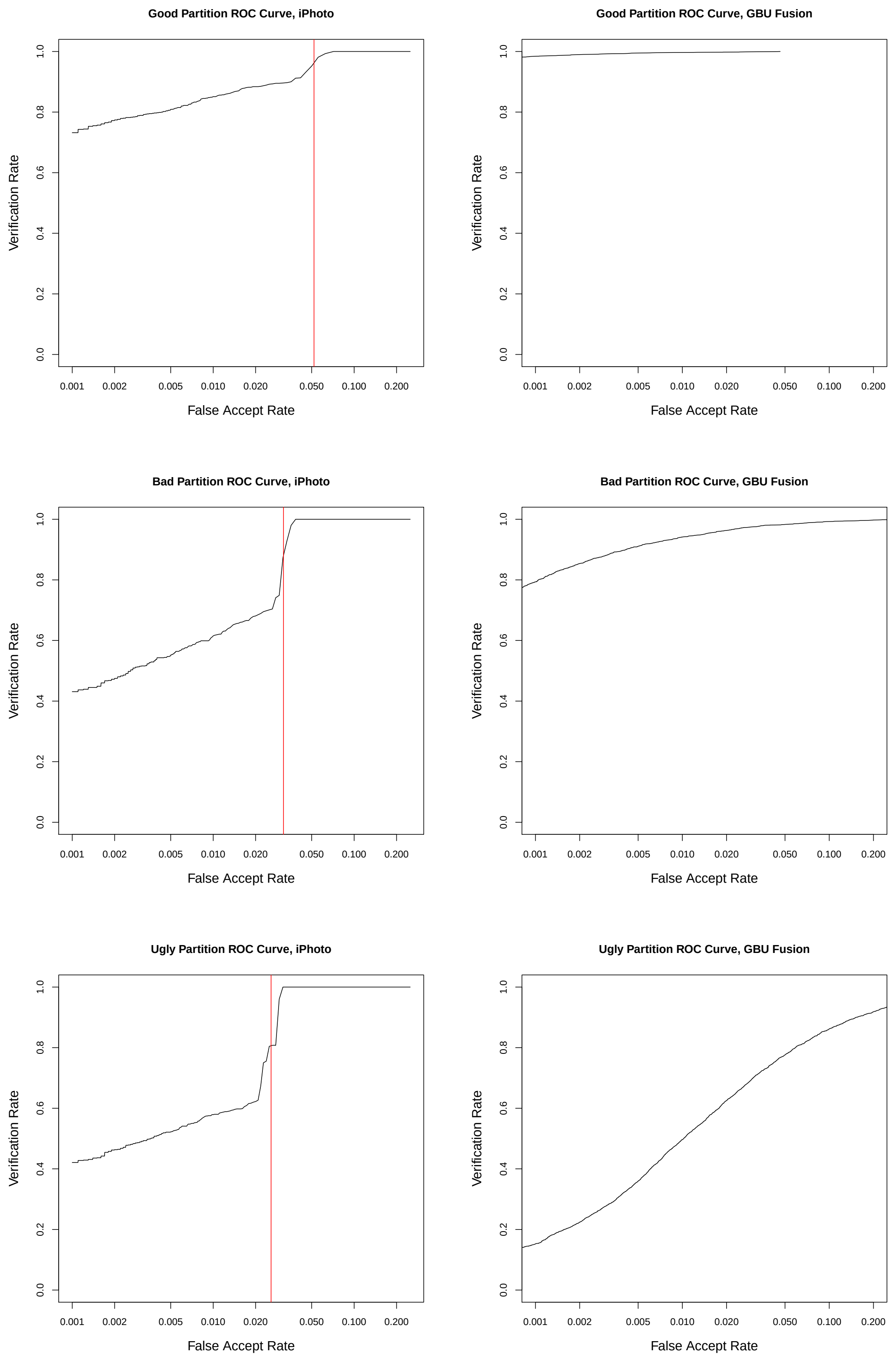


Figure 3: ROC Curves for both iPhoto and GBU Fusion on the Good, Bad and Ugly Partitions. Note that the target and query sets, as well as the sample size, are different between iPhoto and Fusion. The red vertical lines on iPhoto ROCs depict where there is a loss of precision caused by zeros in the non-match scores list.

that the sample sizes are different for the left verses the rigtht columns in Figure 3, and fruther that the image pairs involved are not always the same between iPhoto and GBU Fusion. The sample size for the iPhoto experiments was 100 people, while the sample size for the GBU Fusion algorithm was the entire relevant partition of the data set.

Since a precise quantitative comparison between the performance of iPhoto and GBU Fusion is problematic, we must draw other insights. Recall that the GBU data set was designed to generate a significant separation in performance between the Good, Bad and Ugly partitions when used with an algorithm designed to perform biometric verification. The GBU Fusion algorithm shows this effect quite nicely, there is a dramatic difference in it's performance between the three partitions.

On the other hand, iPhoto has less performance variation between the partitions. This effect is most pronounced with the Bad and Ugly partitions, where iPhoto only has a small difference in performance between the two. Overall, iPhoto seems to be have more stable performance under conditions of varying recognition difficulty than was expected.

I suspect that this stability is related to iPhoto's design goals. Consumers are likely to present iPhoto with images that have a wide variety of facial recognition difficulty. iPhoto also encourages the user to provide feedback about the correctness of a face match, such information is clearly used to fine tune the facial recognition system to better match a user's specific collection of images.

8 Conclusion and Future Work

A technique for measuring the performance of iPhoto's Faces feature has been proposed and implemented. It was assumed that such a measurement was reasonable and worthwhile, and that iPhoto Faces falls into the broad category of facial recognition systems. Further, it was acknowledged that the biometric verification task is outside of iPhoto's apparent design goals. Specific methods for accomplishing the measurement based upon well established techniques were described. The results of those measurements have shown that iPhoto Faces has better than expected performance on the most difficult faces to match, and that its performance is more stable than expected across variation in recognition difficulty.

The next steps in the investigation would be to increase the sample size to include entire partitions, use the exact same target and query sets used for the GBU challenge problem, and use the latest version of iPhoto. Using the same sample size, target, and query sets as the GBU challenge problem would allow a much more precise comparison between the Fusion algorithm and iPhoto. However, it is worth restating that iPhoto Faces is designed to cluster personal photo collections, not to perform biometric verification on single image pairs. Further, the task of helping a user organize their photo collection based on facial information is different from that of biometric verification and it is not even clear in principle that an algorithms well suited to one should necessarily excel at the other.

REFERENCES

[1] Apple - iPhoto - faces, places, and other new features. http://www.apple.com/ilife/iphoto/whats-new.html#faces.

[2] Apple - iPhoto - organize, edit, and share photos on the web or in a book. http://www.apple.com/ilife/iphoto/.

[3] Roberto Baldwin. iPhoto's faces recognizes cats | Mac|Life. http://www.maclife.com/article/news/iphotos_faces_recognizes_cats, January 2009.

[4] Tom Fawcett. ROC graphs: Notes and practical considerations for researchers. http://home.comcast.net/~tom.fawcett/public_html/papers/ROC101.pdf, 2004.

[5] Anil K. Jain, Patrick J. Flynn, and Arun A. Ross. *Handbook of biometrics*. Springer, New York, N.Y., 2008.

[6] S. Z. Li and Anil K. Jain. Introduction. In S. Z. Li and Anil K. Jain, editors, *Handbook of face recognition*, pages 1–11. Springer, New York, 2005.

[7] P. J Phillips, J. R Beveridge, B. A Draper, G. Givens, A. J O'Toole, D. S Bolme, J. Dunlop, Yui Man Lui, H. Sahibzada, and S. Weimer. An introduction to the good, the bad, & the ugly face recognition challenge problem. In *2011 IEEE International Conference on Automatic Face & Gesture Recognition and Workshops (FG 2011)*, pages 346–353. IEEE, March 2011.

[8] P. J. Phillips, W. T. Scruggs, A. J. O'Toole, P. J. Flynn, K. W. Bowyer, C. L. Schott, and M. Sharpe. FRVT 2006 and ICE 2006 large-scale results. Technical Report NISTIR 7408, National Institute of Standards and Technology, March 2007.

[9] P. Jonathon Phillips, Patrick Grother, and Ross Micheals. Evaluation methods in face recognition. In S. Z. Li and Anil K. Jain, editors, *Handbook of face recognition*, pages 329–348. Springer, New York, 2005.

[10] P.J. Phillips, P.J. Grother, R.J. Micheals, D.M. Blackburn, E. Tabassi, and J.M. Bone. Face recognition vendor test 2002: Evaluation report. Technical Report NISTIR 6965, Nat'l Inst. of Standards and Technology, March 2003.

[11] P.J. Phillips, Hyeonjoon Moon, S.A. Rizvi, and P.J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(10):1090–1104, 2000.

[12] P.J. Phillips, W.T. Scruggs, A.J. O'Toole, P.J. Flynn, K.W. Bowyer, C.L. Schott, and M. Sharpe. FRVT 2006 and ICE 2006 Large-Scale experimental results. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(5):831–846, 2010.

[13] Wilson Rothman. What to know about iPhoto '09 face detection and recognition. http://gizmodo.com/5141741/what-to-know-about-iphoto-09-face-detection-and-recognition, January 2009.

Appendix 1: The ROC Curve

The ROC curve plots the Receiver Operating Characteristic as a graph, and is designed to provide a comparison between accuracy and error rate [4]. This comparison is useful because there is often a trade off between error rates and accuracy, with higher accuracy coming at the expense of higher error rates. ROC curves are commonly used to illustrate this trade off for a particular system.

The operation of an ROC curve is usually described in terms of data produced by a binary classifier. This is a system that only makes a binary positive or negative decision about its data. For a binary classifier there are four possible outcomes to making a decision: true positives, false positives, false negatives and true negatives. A true positive outcome is when the system's decision was positive, and positive was the correct decision. A false positive outcome is when the system's decision was positive, but negative was the correct decision. A false negative outcome happens when the systems decides in the negative when positive was actually the correct decision. And finally a true negative happens when the system correctly makes a negative decision. The rate of false positives is computed by dividing the number of false positive outcomes by the total number of negatives. Similarly the true positive rate is computed by dividing the number of true positive outcomes by the total number of positives. Common ROC curves plot the false positive rate on the x-axis and the true positive rate on the y-axis. In the case of this work, the true positive rate is called the verification rate and the false positive rate is called the FAR. This nomenclature was adopted from the work in evaluating facial recognition systems upon which this work is based.

In short, when looking at an ROC curve: points which are higher and more to the left represent better performance than points which are lower and more to the right. They represent higher true positive rates achieved at lower false positive rates. Note also that the ROC curves used herein have logarithmic x-axes, this is because most of the data we are interested in is compressed in to the lower false positive rates. Plotting the x-axis in a logarithmic scale makes this data easier to see without distorting the basic rules of ROC curves.