CIHSPS 2005 - IEEE International Conference on
Computational Intelligence for Homeland Security and Personal Safety
Orlando, FL, USA, 31 March – 1 April 2005

Predicting Biometric System Failure

Weiliang Li
Lehigh University
Siemens Corp. Research

Xiang Gao
Siemens Corp. Research

Terrance E. Boult
U. Colorado at Colorado Springs
Securics, Inc

Abstract –

Object recognition (or classification) systems largely emphasize improving system performance and focus on their "positive" recognition (or classification). Few papers have addressed the prediction of recognition algorithm failures, even though it directly addresses a very relevant issue and can be very important in overall system design. This is the first paper to focus on predicting the failure of a recognizer (or classifier) and verifying the correctness of the recognition (or classification) system. This research provides a unique component to the overall understanding of biometric systems.

The approach presented in the paper is the post-recognition analysis techniques (PRAT), where the similarity scores used in recognition are analyzed to predict the system failure or to verify the system correctness after a recognizer has been applied. Applying a AdaBoost learning the approach combines the features computed from the similarity measures to produce a patent pending system that predicts the failure of a biometric system. Because the approach is learning-based the PRAT is a general paradigm predicting failure of any "similarity-based" recognition (or classification) algorithm. Failure prediction, using a leading leading commercial face recognition system, is presented as an example to show how to use the approach. On outdoor weathered face data, the system demonstrated the ability to predict 90% of the underlying facial recognition system failures with a 15% false alarm rate.

I. INTRODUCTION

Recognition systems seek to correctly recognize object(s) of interest from within a large class of potentials. Most current research emphasizes improving the "accuracy" of systems, dwelling largely on the positive recognition rate[1]. However, even for a modern system, the detection or recognition rate is still less-than-perfect, [2], [3]. As papers tend to focus on the "positive" aspects of their problem, the natural focus has been on recognition. The evil twin of recognition — failure — has been generally neglected.

At an algorithm level, recognition rate and failure rate are inseparable — knowing one implies the other. Predicting failure of an algorithm does not, in general, help that algorithm perform better. However, at the system level, there are many ways to predict failure of the primary recognition algorithm and to use that information to improve the overall system performance. The simplest application is in an interactive or online system where, if we can predict failure, then we might simply re-acquire a new image and try again. This "binary" failure prediction is the primary focus of this paper as it allows us to separate the "failure prediction" from the underlying recognition algorithms.

A more advanced application would be in a system that always uses multiple sensors/images for (face) recognition, in which case it is necessary to coordinate the operations of all the sensors. The output of the fusion could be the result from the "best" sensor, or some mixture of the results. It is possible that one or more sensors may fail in recognition. But without knowledge of those sensors' reliability, such "fusion" is difficult[4]. A hybrid classifier, combining a set of classifiers, is not a new concept[1] and a special case of fusion. If PRAT is not just a binary classification, but more of an overall confidence measure (that may be thresholded for classification), then it can be very effectively used to support various approaches to fusion and hybrid classifiers. If one can predict system failure, then it simplifies the design of the combination of classifiers and should improve their reliability.

A related use of PRAT would be for a measure of system confidence, which might effect the system output. A number of the commercial face recognitions systems, when being used for verification use a process generically known as "normalization"[5] which takes the similarly scores and renormalizes them before deciding if the individual is "recognized" or verified. The important difference in normalization is that the system gets to consider the structure of the similarity scores of the entire set of people (rather than just a collection of independent measurements). While the companies that do this do not describe their ideas, the PRAT-based approach presented herein could easily be used for this normalization.

The goal of this paper is to discuss how to generalize approaches to determine or predict when a recognition system will probably fail. Although there are two categories of techniques which may be employed to estimate system failures, we

This work funded in part by DARPA HID program ONR contract N00014-00-1-0388, and by the Colorado Institute for Technology Transfer and Implementation.

are focused on post-processing techniques which analyze the information used within the recognition process itself.[1] Clearly PRAT depends on the internals of the recognition approach, and we discuss a technique useful for recognition engines that measure similarity or dissimilarity.

The discussion of this topic gets a bit tricky. There are two levels of classification, the first is the primary recognition system, e.g. the face recognition system. The second is the PRAT-based prediction/classification of the accuracy of the first system (e.g. face recognition). Throughout the paper we will use "face recognition" as the running example and the terms "recognition rate" and "correct recognition" will always apply to the primary recognition system. The term "classification" will always apply to the PRAT-based classification of the primary system recognition results. Let us now define a few key terms. To simplify the presentation we presume a simple model of PRAT-based technique, where one computes a confidence measure in the correctness of the recognition result and then threshold on our confidence to produce a binary decision.

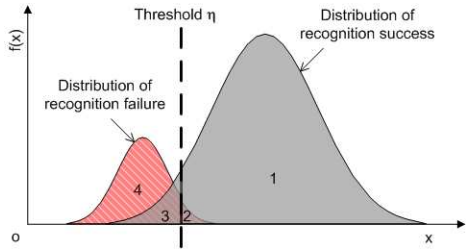

Fig. 1. Threshold discrimination on two distributions of confi dence measures.

The recognition system failures originate from the limitation of the recognition system or its inputs: for a recognition algorithm, it may classify the input image example (probe) incorrectly. We are assuming, the recognition algorithm produces a "similarity measure" for each image pair (probe vs image in the candidate image set) and reports the top N scores as its recognition result. Given PRAT produces a confidence measure in that each result one can plot, as shown in Figure 1, the distribution of the number of cases, $f(x)$ with a particular confidence measure x. Using the ground truth label for each image, we can draw two separate distributions one for the recognition successes and one for recognition failures. In general the two distributions will overlap. Using a more discriminating confidence measurement can reduce the overlap. For every possible threshold η (represented by vertical dashed line) we choose to discriminate between the two populations, resulting in four case[2]

- *Case 1* — Traditionally called "True Accept" wherein the underlining recognition algorithm was successful and PRAT predicts that it will succeed. (Note this does not mean the person was recognized, the correct operation of the recognition system could be either a recognition or a rejection.)
- *Case 2* — Conventionally called a "False Accept" is when PRAT predicts that the recognition system will succeed, but ground truth shows it does not.
- *Case 3* — Conventionally called as "False Reject", it is when the PRAT predicts that the recognition process will fail, but ground truth shows it was successful.
- *Case 4* — This region is conventionally defined as "True Reject". PRAT correctly says that the recognition system will fail.

To define false accept and miss detection rates it is also important that we normalize errors by the right items, since for different settings or algorithms the underlying recognition rate will be changing and hence changing the size of failure set.

In this paper our predictions are false alarms for items in *Case 3* (PRAT predicts they recognizer will fail but it is correct), with the Failure Prediction False Alarm Rate (FPFAR) defined as

$$FPFAR = \frac{|Case3|}{|Case3| + |Case1|}.$$

Our miss detections would be those items in *Case 2*, we predict they will be recognized but they are not, with the Failure Prediction Miss-Detection Rate (FPMDR) defines as:

$$FPMDR = \frac{|Case2|}{|Case2| + |Case4|}.$$

Because PRAT is predicting failure of a recognition system, we have two levels of "classification". To avoid confusion we eschew the terminology such as "true positive" or "true reject" throughout the paper, and will use the terms *Case 1* through *Case 4*, or FPFAR and FPMDR, instead.

While this discussion presumed a simple "confidence measure" where the classifer applied a simple threshold, this is not the best way to implement a PRAT-techniques. As we shall see, AdaBoost technique can be applied as well.

The post-recognition analysis technique is used to predict when the recognition system will likely fail. The described approach is applicable to any system that uses "similarity" (or dissimilarity) measures [6], [7], [8], [9] and does recognition based on largest (smallest) similarity values. Since, depending on the system goals, the desired tradeoff between FPFAR and FPMDR may be of varying importance, we represent our results as ROC curves showing the tradeoff between them.

II. SIMILARITY & SIMILARITY SCORE

This section provides theoretical background on features sets and why the similarity scores over many items may have interesting properties. For those more focused on what and how, rather than why, it can be skipped on first reading without loosing an understanding of the approach. Similarity measure $S(\mathbf{x}, \mathbf{y})$ between arbitrary two patterns, or images, $\mathbf{x}$

[1] The other technique is the input fi ltering technique which may estimate or predict system failures before the invocation of an classifi cation algorithm.

[2] Note that a detailed analysis might discriminate between the false positives and false negatives from the original recognition system, resulting in 8 cases to consider, but for this paper we consider only these 4 cases:

and $\mathbf{y}$ is an effective approach for classification and recognition [7]. In pattern recognition, two major models of similarity analysis are geometric model and feature model [8], [9].[3] Geometric models have been among the most influential approaches for analyzing similarity and are exemplified by multidimensional scaling (MDS) models. The similarity of $\mathbf{x}$ and $\mathbf{y}$ is taken to be inversely related to their distance $D(\mathbf{x},\mathbf{y})$, i.e. $S(\mathbf{x},\mathbf{y}) = \alpha - \beta D(\mathbf{x},\mathbf{y})$, where $\alpha, \beta > 0$. Geometric models typically assume three metric properties:

- *positivity*, $D(\mathbf{x},\mathbf{y}) \geq D(\mathbf{x},\mathbf{x}) = 0$,
- *symmetry*, $D(\mathbf{x},\mathbf{y}) = D(\mathbf{y},\mathbf{x})$, and
- *triangle inequality*, $D(\mathbf{x},\mathbf{y}) + D(\mathbf{y},\mathbf{z}) \geq D(\mathbf{x},\mathbf{z})$.

In a typical recognition system, however, if we assume that there are no identical images due to sequential data collection or noise effect, the positivity property becomes unacceptable except that the distance is positive. In this case, if we still want to use a distance measure to represent the dissimilarity of two images, there is only partial matching of any two images, that is, a part of an image matches a part of another image. Under partial matching, triangle inequality may often be violated. For example, see Figure 2 which is adapted from an example in [9].

A B C

Fig. 2. Under partial matching the triangle inequality may not hold. While A and B partially match, and B and C partially match, A and C do not match at all.

As reported in the literature [8], [9], it is empirically observed that all three properties of a geometric model are often violated. In [10], Tversky suggested an alternative approach, the feature contrast model (FCM), wherein similarity is determined by matching features of compared patterns. In the following, X, Y, and Z is used to denote the sets of binary features of compared patterns or images $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$. We also assume that $A_{XY} = X \cap Y$, $B_{XY} = X - Y$, and $C_{XY} = Y - X$. FCM is usually integrated by three properties: *Matching*, *Monotonicity*, and *Independence*.

- Matching is defined as $S(\mathbf{x}, \mathbf{y}) = \theta f(A_{XY}) - \alpha f(B_{XY}) - \beta f(C_{XY})$, where f is a non-negative function and $\theta, \alpha, \beta \geq 0$. When $\theta > 0$ and $\alpha, \beta = 0$, $S(\mathbf{x}, \mathbf{y})$ compares the common features of $\mathbf{x}, \mathbf{y}$: the more features in common, the more similar $\mathbf{x}, \mathbf{y}$ are. When $\theta, \alpha > 0$ and $\beta = 0$, we may compare the features common to $\mathbf{x}, \mathbf{y}$ with those unique to $\mathbf{x}$. The reverse is true when $\theta, \beta > 0$ and $\alpha = 0$. When $\theta = 0$, and $\alpha, \beta > 0$, we may compare $\mathbf{x}, \mathbf{y}$ only on their distinctive features.
- Monotonicity is defined as $S(\mathbf{x},\mathbf{y}) \geq S(\mathbf{x},\mathbf{z})$ whenever $A_{XY} \supseteq A_{XZ}$, $B_{XY} \subseteq B_{XZ}$, $C_{XY} \subseteq C_{XZ}$. From this property, it can easily be inferred that when $\mathbf{x}$ and $\mathbf{y}$ share more common and less distinctive features than $\mathbf{x}$ and $\mathbf{z}$, then $\mathbf{x}$ is more similar to $\mathbf{y}$ than $\mathbf{x}$ to $\mathbf{z}$.
- When $f(A_{XY}) = f(A_{XZ})$, and $B_{XY} = B_{XZ}$, $C_{XY} = C_{XZ}$, then we may approximately have $A_{XY} \approx A_{XZ}$. Similarly, when $f(B_{XY}) = f(B_{XZ})$, and $A_{XY} = A_{XZ}$, $C_{XY} = C_{XZ}$, then $B_{XY} \approx B_{XZ}$; when $f(C_{XY}) = f(C_{XZ})$, and $A_{XY} = A_{XZ}$, $B_{XY} = B_{XZ}$, then $C_{XY} \approx C_{XZ}$. The pairs of patterns $(\mathbf{x},\mathbf{y})$ and $(\mathbf{x},\mathbf{z})$ are said to agree on one, two, or three components whenever one, two, or three of approximate relations hold. A simplified expression of independence is $S(\mathbf{x},\mathbf{y}) \geq S(\mathbf{x}',\mathbf{y}') \leftrightarrow S(\mathbf{x},\mathbf{z}) \geq S(\mathbf{x}',\mathbf{z}')$ if pairs $(\mathbf{x},\mathbf{y})$ and $(\mathbf{x},\mathbf{z})$, as well as $(\mathbf{x}',\mathbf{y}')$ and $(\mathbf{x}',\mathbf{z}')$ agree on the same two components, whereas $(\mathbf{x},\mathbf{y})$ and $(\mathbf{x}',\mathbf{y}')$, as well as $(\mathbf{x},\mathbf{z})$ and $(\mathbf{x}',\mathbf{z}')$ agree on the remaining components. For detailed independence expression, see [8].

In pattern recognition, the similarity score represents the quality of match. The similarity score $s(\mathbf{x},\mathbf{y})$ is a value calculated from a set of (fuzzy) metrics of interest by a classifier implementing a set of one or more learning algorithms. This value is within a given range. Without loss of generality, we assume $S(\mathbf{x},\mathbf{y}) = s(\mathbf{x},\mathbf{y})$ and the largest similarity score is the most likely match to the subject. In the following part of the paper, we treat the term $S(\mathbf{x},\mathbf{y})$ and $s(\mathbf{x},\mathbf{y})$ same.

Performance of any classifier is determined with respect to the expected data or ground truth data [2]. It is measured as the ability to correctly identify the probe image. The performance of such a measure provides a basis for comparison of the processed image and the ground truth. Nevertheless, it is impossible to directly compare the similarity measures between different algorithms since each algorithm may adopt a different measure of similarity. Usually the similarity measure is not a metric. However, we still can use their relative ordering. In order to compare a set of similarity scores from different algorithms, it is necessary to normalize it to a common range. In our experiment, we scale the range of the similarity scores to $[0, 100]$ where 100 means *the most similar* and 0 *the least similar*.

III. POST-RECOGNITION ANALYSIS TECHNIQUE

A. Notations

Let $\mathbf{x}_i \in R^N (i = 1, 2, ..., m)$ and $\mathbf{y}_j \in R^N (j = 1, 2, ..., n)$ represent image or image feature vectors. Given a training set (*gallery*) $Y = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n\}$, a collection of images known to the classification algorithm, and a similarity measure $S(\cdot,\cdot)$ of a classifier, similarity scores of an image instance (*probe*) $\mathbf{x}_i$ of a set of unknowns, $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m\}$, can be represented by $s_1 = S(\mathbf{x}_i, \mathbf{y}_1)$, $s_2 = S(\mathbf{x}_i, \mathbf{y}_2)$, ... , $s_n = S(\mathbf{x}_i, \mathbf{y}_n)$. In the meantime, since the similarity score is a real value and any

[3] The other models include alignment-based model and transformational model.

image gallery Y is countable, the maximum similarity score exists. After sorting similarity scores, we obtain a set of monotonic similarity score $S_\Sigma = \{s_{k_1}, s_{k_2}, ..., s_{k_n} \mid s_{k_1} < s_{k_2} < ... < s_{k_n}\}$. Figure 3 illustrates a typical curve of sorted, monotonic similarity scores. When there exists a small ϵ to make $\forall s_{k_i} < \epsilon$, we may have $S_\Sigma = \{\Phi\}$. The output of the classifier on a given probe $\mathbf{x}_i$ is usually a set of *candidates* corresponding to the top p similarity scores $S_{\Sigma_p} = \{s_{k_{n-p+1}}, ..., s_{k_n}\}$.

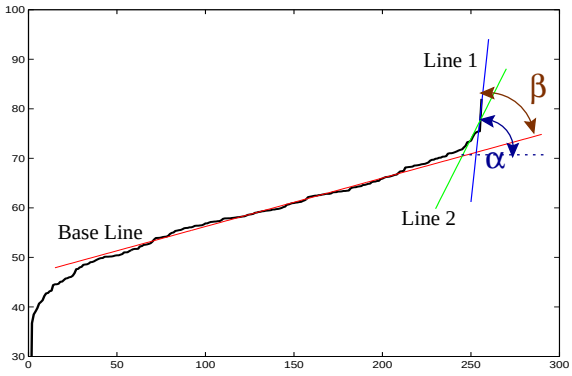

Fig. 3. Geometrical illustration of top part of sorted similarity score S_Σ of a sample: the stable and relative flat part is linearly interpolated and represented by Base Line. Line 1 is the interpolated line of sorted largest and second largest similarity scores. Line 2 denotes the interpolated line of top p similarity scores.

The prediction failures are composed of two types: *Case 2* and *Case 3* discussed in the Section I. These two types of errors (*Case 2* and *Case 3*) can only be determined with respect to the ground truth data. In this paper, we use a ranked number R_n to express the difference between the expected output and the actual output of a classifier. If $R_n = 1$, the actual output is considered to be the expected output. If $R_n = (1, \infty)$, there are $R_n - 1$ gallery images that have higher similarity scores than the probe. If $R_n = \infty$, there is no correspondence between the gallery images and the probe.

B. Feature Measures on Similarity Scores

When feature contrast model (FCM) is used on a collection of sorted, monotonic similarity scores, there are at least three forms of feature measures:

- When $\mathbf{x}_i$ is strictly more similar to $\mathbf{y}_{j_1} \in Y$ than $\mathbf{x}_i$ to $\mathbf{y}_{j_2} \in Y$, that is, $A_{X_iY_{j_2}} \subset A_{X_iY_{j_1}}$, $B_{X_iY_{j_1}} \subset B_{X_iY_{j_2}}$, $C_{X_iY_{j_1}} \subset C_{X_iY_{j_2}}$, according to monotonicity property, we have our first feature measure $F_1(Y \mid \mathbf{x}_i) = \{S(\mathbf{x}_i, \mathbf{y}_{j_1}) - S(\mathbf{x}_i, \mathbf{y}_{j_2}) > 0\}$ for $\forall \mathbf{y}_j \in Y$ and $\mathbf{x}_i$. Since this feature measure corresponds to α in Figure 3, we name it as *F-slope*.
- When $A_{X_iY_{j_1}} \approx A_{X_iY_{j_2}}$, $B_{X_iY_{j_1}} \approx B_{X_iY_{j_2}}$, and $C_{X_iY_{j_1}} \approx C_{X_iY_{j_2}}$, then $S(\mathbf{x}_i, \mathbf{y}_{j_1}) \approx S(\mathbf{x}_i, \mathbf{y}_{j_2})$. The reverse may not be true. Thus, it is necessary, but not sufficient, to agree on approximately equal components for the approximately equal similarities. Moreover, the similarity is represented by a non-negative function. When we pool a group of approximately equal similarities, we include all of the individuals who agree on $A_{X_iY_{j_1}} \approx A_{X_iY_{j_2}}$, $B_{X_iY_{j_1}} \approx B_{X_iY_{j_2}}$, and $C_{X_iY_{j_1}} \approx C_{X_iY_{j_2}}$. We call this feature measure *F-internal*. Another consideration is that when $\mathbf{x}_i$ is very similar to $\mathbf{y}_{j_1}, \mathbf{y}_{j_2} \in Y$ (which corresponds to the absolutely large similarity scores), then $A_{X_iY_{j_1}} \approx A_{X_iY_{j_2}}$ is a predominant component. In this case, when $S(\mathbf{x}_i, \mathbf{y}_{j_1}) \approx S(\mathbf{x}_i, \mathbf{y}_{j_2})$, we may have $A_{X_iY_{j_1}} \approx A_{X_iY_{j_2}}$ and $B_{X_iY_{j_1}} \approx B_{X_iY_{j_2}}$.
- As an inference from the independence property, when pairs $(\mathbf{x}_i, \mathbf{y}_{j_1})$ and $(\mathbf{x}_i, \mathbf{y}_{j_2})$ share more common features, as well as pairs $(\mathbf{x}_i, \mathbf{y}_{j_3})$ and $(\mathbf{x}_i, \mathbf{y}_{j_4})$, we are likely to have $S(\mathbf{x}_i, \mathbf{y}_{j_1}) > S(\mathbf{x}_i, \mathbf{y}_{j_3}) \leftrightarrow S(\mathbf{x}_i, \mathbf{y}_{j_2}) > S(\mathbf{x}_i, \mathbf{y}_{j_4})$. Also, it is very possible that $S(\mathbf{x}_i, \mathbf{y}_{j_1})$ and $S(\mathbf{x}_i, \mathbf{y}_{j_2})$ are in one pool, while $S(\mathbf{x}_i, \mathbf{y}_{j_2})$ and $S(\mathbf{x}_i, \mathbf{y}_{j_4})$ are in another. The interval between two consecutive pools is our third feature measure. We call it *F-external*.

In our approach, we adopt the AdaBoost method to use the above feature measures to predict recognition failure with large collections of sorted, monotonic similarity scores.

C. Algorithm Description

The recognition prediction algorithm uses the boosting framework. Boosting algorithms have been reported to be successful in improving the performance of classifiers [11], [12], [13], [14], [15], [16]. Equation 1 is the representation for the final strong classifier after T rounds boosting:

$$H(x) = sign\Big(\sum_{t=1}^{T} \alpha_t h_t(x) - \lambda \sum_{t=1}^{T} \alpha_t\Big) \qquad (1)$$

where stronger classifier $H(x)$ is an additive combination of a sequence of weak classifier $h_t(x_i)$ with its weight parameter α_t through majority voting (t is boosting trial variable). $\lambda \in [0, 1]$ is a parameter to adjust the performance of the classifier and control the FPFAR versus FPMDR.

The ranked number R_n (section A) is an expressive parameter to differentiate the expected output and the actual output. During the experiment, as a benchmark, the ranked number R_n is used to partition the training examples into two classes, i.e., the output of similarity scores from a recognition system is labeled by a given R_n range through "greater" or "not greater" operations.

All of the three features discussed in section B have been used for boosting. *F-slope* is an intuitive measure. We find that sorted similarity scores keep linear in a specific range which have relative lower values denoted by the Base Line, but change abruptly for a few top similarity values such as Line 1 or Line 2 (an example shown in Figure 3). In the following section, we explain how to use measures *F-internal* and *F-external* in detail.

C.1 Clustering Similarity Score

Clustering similarity scores (CSS) is an operation to quickly determine whether a probe has been correctly classified or

needs further analysis. When the difference between the sorted consecutive similarity scores is less than a threshold τ , they are clustered into the same pool. During the boosting, different values of τ have been tested. In our experiments, we find that the optimal τ should be between 0.5 and 1.5. Given a non-empty set of ranked similarity scores $S_\Sigma = \{S_1, S_2, ..., S_n\}$, we take the top p ($p > 1$) similarity scores $S_{\Sigma_p} = \{S_{n-p+1}, ..., S_n\}$ for analysis since the user of the recognition system only cares about the top p candidates. Our experiments show the prediction of recognition failures also strongly depends on these values. During the boosting procedure, we keep increasing p from 2 to 25.

In the following analysis, we assume that PRAT is provided a collection of top p similarity scores S_{Σ_p}. An *F-internal* distance measure di is used to denote internal maximum distance of a cluster interval which is the largest difference among a set of clustered similarity scores. If there is only one similarity score in a cluster, di is equal to 0. An *F-external* distance measure de is used to denote the difference between the end points of clustered consecutive interval. After the clustering operation, we obtained a sequence $D = \{di : de\} = \{di_1, di_2, ..., di_m : de_1, de_2, ..., de_{m-1}\}$, where m is the cluster number we obtained during the clustering procedure. For example, suppose we have a set of top $p = 10$ similarity scores which can be clustered into four sets of clustered similarity scores $\{80.64\}$, $\{78.02\}$, $\{74.66, 74.65\}$, and $\{73.07, 72.88, 72.36, 72.28, 71.85, 71.50\}$ with $\tau = 1.0$. After clustering operation, we have $D = \{0, 0, 0.01, 1.57 : 2.62, 3.36, 1.58\}$.

The geometrical explanation of CSS is shown in Figure 3. In most case, de_1 is the largest difference between the largest similarity score and the second largest score when $di_1 = 0$, i.e., there is only one similarity score in the first cluster. This is illustrated by the slope of Line 1 in Figure 3. Please note that, in this case, de_1 also corresponds to *F-slope* measure, but they are different in concept. When we limit R_n in a given range, most correct classification results fall in this case. However, when de_1 is very close to $de_2, de_3, ...$, or even smaller than these values, we find that we need to calculate the cumulative distance and its distribution for further analysis.

C.2 Cumulative Distance

Cumulative distance is a measure to conglomerate previously defined internal distance di and external distance de into a whole entity. The purpose is to determine the ratio of *F-internal* and *F-external* measures. It is composed of two parts: total internal distance $Di = \sum_{j=1}^{m} di_j$ and total external distance $De = \sum_{j=1}^{m-1} de_j$. We further define $F - slope_p = Di + De$. As illustrated in Figure 3 and discussed in the previous section, when the angle intersection between Line 1 and Line 2 is over a threshold value, we might predict the output as top p. However, when Line 1 and 2 are very close, it is difficult to predict the output.

Cumulative distance can be adopted to effectively overcome this difficulty. Through our experiment, we find that, in the first few distributions of $F - slope_p$, a large amount of the samples' $F - slope_p$ fall into the range of $[\ 0, 0^+)$ which means the first cluster of the samples have only one or several very close similarity scores. With the increase of p, the distributions of $F - slope_p$ will shift toward the positive direction and become stable. Two exemplary distributions of $F - slope_p$ and De are shown in Figure 4. The experimental dataset is composed of face images of 256 subjects. Each subject has four similar images. The algorithm for recognition, in all experiments, is the FaceIt product from the Identix (originally Visionics) SDK, version 4. (This is not the current release version.) All of images are subject to the variations of JPEG images with JPEG quality varying from 0 to 100 (100 means the best quality). Each image is used as either a *probe* or an element of *gallery*. From the statistical point of view, it is possible to discriminate the two populations using the distribution models of $F - slope_p$ or De and Di with different R_n settings as shown in Figure 4, especially the right graph. In left graph of Figure 4, most of samples' $F - slope_p$ are fall into recognized population when $R_n \leq 10$, while some of samples' $F - slope_p$ in unrecognized range when $R_n > 10$. The right graph of Figure 4 shows the distributions of both of internal distance Di and external distance De with two R_n settings: $R_n \leq 10$ and $R_n > 10$. We expect the second distribution should have a better performance in discriminating the two populations since it seems more separable. In fact, as shown in Figure 5, external distance De is a predominant measure in a strong classifier.

D. Algorithm Diagram

- Input : Sorted top p similarity scores $S_{\Sigma_p} = \{S_{n-p+1}, ..., S_n\}$, and $F - internal$ thresholded interval $[\tau_1, \tau_2]$ and step length $\Delta\tau$.
- LOOP:
  1. $\tau = \tau_1$.
  2. Let similarity score index $i = 1$ and clustering index $k = 1$. Put S_n to cluster C_1.
  3. $i = i + 1$. Save $S_n - S_i$ in $\{d_{F-slope}\}_i$.
     if $S_n - S_i < \tau$ Put S_i to cluster C_k;
     else $k = k + 1$; Put S_i to cluster C_k.
  4. if all $S_i \in S_{\Sigma_p}$ are considered break;
     else goto 3.
  5. for $j = 1$ to k
     Save max$\{C_j\}$ - min$\{C_j\}$ to $\{d_{F-internal}\}_j$.
     if $j > 1$ Save max$\{C_j\}$ - min$\{C_{j-1}\}$ to $\{d_{F-external}\}_j$.
  6. Update $\tau = \tau + \Delta\tau$.
     if $\tau \geq \tau_2$ stop;
     else goto 2.
- Boost on $\{d_{F-slope}\}$, $\{d_{F-internal}\}$ and $\{d_{F-external}\}$ to generate the strong classifier using Equation 1.

Currently, the setting for the parameters τ_1, τ_2, $\Delta\tau$, and p is done based on our experience. The research on how the parameters will affect the boosting system and the prediction accuracy just starts.

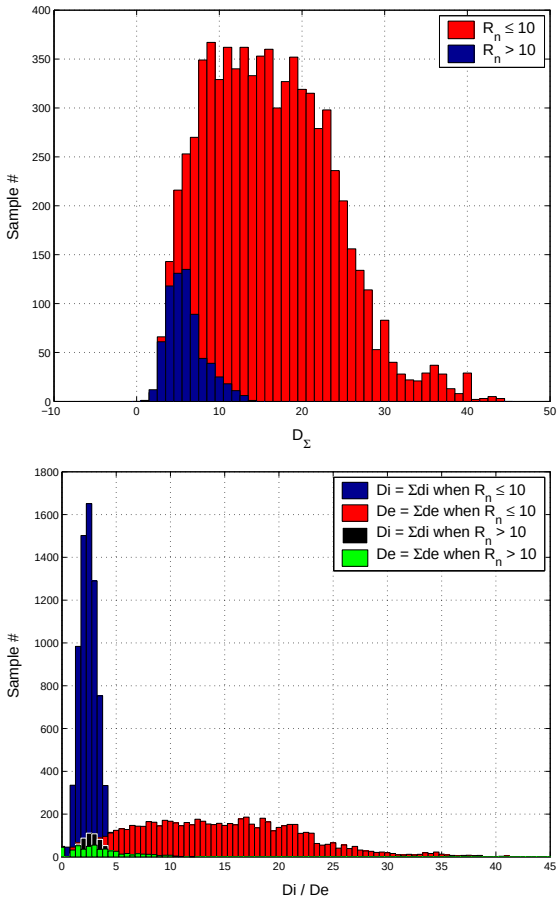

Fig. 4. Top: Distribution of $F - slope_p$ with 8243 samples using FaceIt. The experimental data is from FERET databases. Experiment I is for $R_n \leq 10$ and the range of $\{F - slope\}$ is $[2.40, 43.53]$ in red color (lighter in BW version). Experiment II is for $R_n > 10$ and the range of $\{F - slope\}$ is $[1.70, 13.65]$ in blue color (darker in BW version). Bottom: Distribution of Di and De using the same data set and classifi er of top graph.

IV. EXPERIMENTS

In the experiments after boosting, ROC plot is adopted to denote the tradeoff between the fraction of false alarmed and miss detected examples over the total population for every possible value of λ in Equation 1. In all the experiments, λ is varied from 0 to 1. This shows the overall performance choices and for a particular installation can determine how one might set the parameters to achieve a particular FPFAR vs FPMDR tradeoff. The variation of λ along/within the curve cannot be seen in the ROC curve and is discussed in the next section.

In one of our experiments using different JPEG qualities on training examples (top of Figure 5), the same training data has been partitioned three times with different ranges of R_n. In the first partition, the examples of $R_n = 1$ and $R_n > 1$ are labeled into two classes, respectively. Similarly, the other two partitions are delimited by $R_n = 5$ and 10. The upper graph of Figure 5 shows the mean error rate where $121,308 \times 4$ examples are used for the boosting. It is observed that the performance of the strong classifier will be improved with the increase of the R_n range. This is what we expected since it is well-accepted that in an ordinary computer vision information retrieval system, the query result might not be in the first, but the first few output images. In principle, this is due to the fuzzy property of pattern matching and similarity measurements. As explained in section C.2, we find that the distributions of the feature measures of similarity scores become stable with the increase of p values. Therefore, it is more reliable to take a relatively large p value to predict failure or verify the output result. However, for a security application, which requires near zero FPMDR and small FPFAR, increasing p will not improve the prediction performance.

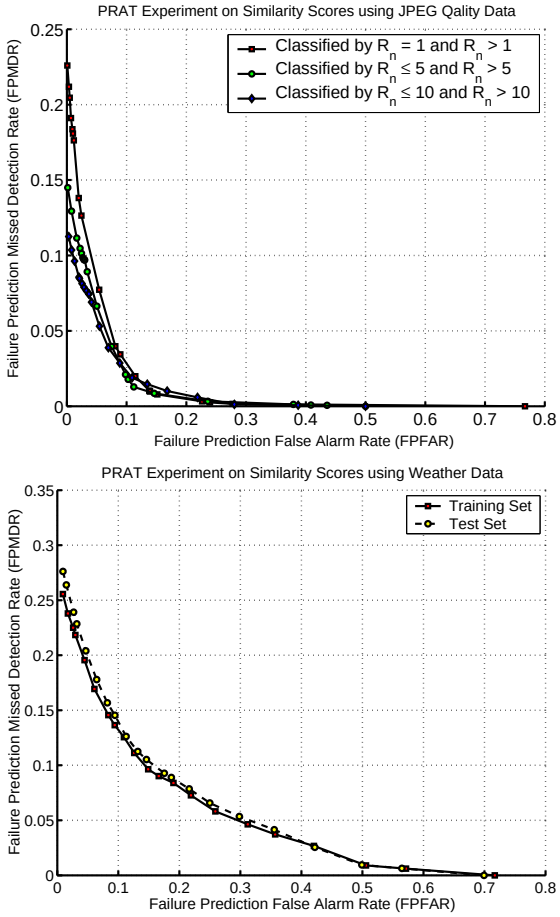

Fig. 5. ROC curve using three feature measures for boosting. Left graph shows experiment results using varying JPEG qualities on training data which has been partitioned by three ranges of R_n. Right: image data are obtained under different weather conditions. FPFAR and FPMDR coordinates are all in their percentage to the total population.

The second and more important experiment was to test the training sets obtained under different weather conditions, as shown in the bottom of Figure 5. The "weather" data, also known as the photohead dataset, was collected as part of the DARPA HID program for inclusion in their HBASE collection. The data re-images FERET images, displayed on a LCD monitor. The cameras were at a distance of approximately 100ft

and 200ft, and zoomed such that the facial images had approximately 200-240 pixels between the eyes.

Since the training data are real images collected in an outdoor environment, it is much closer to the real world outdoor environment. However the use of a known dataset (FERET) and the photoheads controls for enrollment variations by using fixed images and consistent display. During the boosting, 21,535 training examples are used. In the experiment, we use a cross-validation approach for error estimation. Possibly because all the data is from the same environment and there was no severe weather change, the test error is comparable to the training error. The bottom graph of Figure 5 shows the training errors from two different training sets. In comparison with other experiment results, such as the top graph of Figure 5 and Figure 6, it is obvious that in an outdoor environment, the weather is an important factor which may greatly affect the accuracy of recognition, verification, and prediction.

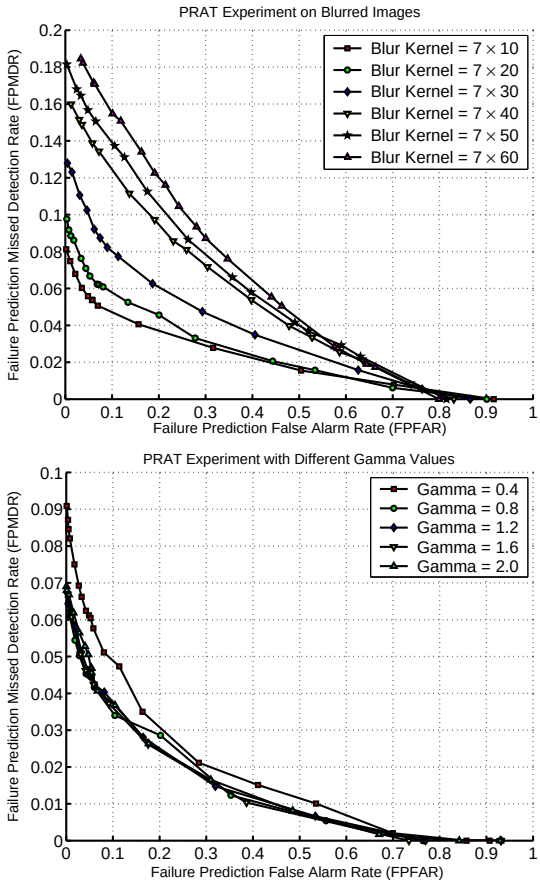

Fig. 6. Upper graph shows the experiment results using different Gaussian blurred kernel functions. Lower graph is from varying γ values.

Two more experiments are shown in Figure 6. The upper graph shows the blur effect to the prediction of the system failure and the lower graph shows the effect of Gamma correction to the prediction. In the experiment on the blur effect, only the probe is blurred. From the left graph, it is clear that when we decrease the size of blur kernel, using PRAT can get a better prediction result. This is not surpriseing since a much clearer image (kernel size is 7×10) keeps most of the details, and there is no overprocessing. From the right graph, we can see that in a normal condition, the Gamma effect doesn't change the prediction accuracy. The main reason is that histogram equalization and other normalization algorithm is a standard procedure [17], [18]. After we do the Gamma transform to the data, the normalization done inside the recognizer largely restores the images' contrast. As long as our Gamma correction is "reasonable" (γ changes from 0.8 to 1.6), we can't change anything significantly to the image to effect the system performance.

V. DISCUSSION ON PARAMETER SETTING

As mentioned in Section I, *Case 2* / *Case 3* tradeoff has impact on the overall system design, especially when we consider the reliability of a system. Different tradeoffs between *Case 2* and *Case 3*, which depends on design requirement, may have different effects on future system(s). We can use the ROC curves to select the parameters λ (from strong classifier), p (the cardinality of S_{Σ_p}), and R_n for the failure prediction system, to achieve a particular choice of FPFAR or FPMDR.

The following is a brief summary of our suggestions on parameter settings:

1. λ — Theoretically, the expected optimal setting λ is 0.5. Across the experiments, that the best performance has been achieved with λ value in $[0, 0.6]$ (see Fig. 7 and Fig. 8.
2. p — p will greatly influence the efficiency of the learning procedure. To improve the learning procedure, we suggest using use $p \leq 25$.
3. R_n — This is a function of end-user application. If doing pure identification, then $R_n = 1$ is the appropriate choice. If being used for a "watch-list", then $R_n = 10$ is a reliable setting.

In the above sections, the experimental results on varying p and R_n have already been shown and discussed. Now, we briefly discuss the impact of λ on the experimental results of the strong classifier $H(x)$. In binary classification, a typical output of a strong classifier is either 1 or 0 to denote the class belonging to the input example x. When the examples of two classes have equal probability, the λ value in Equation 1 is usually set to 0.5. However, when the examples are from unbalanced distributions, such as the examples in Figure 4, we need to adjust λ value to determine an optimal setting for achieving the best performance of the strong classifier. The experimental results of varying λ are illustrated in Figure 7 and Figure 8. It is obvious that the varying λ domain will greatly affect the performance of $H(x)$. This is in correspondence with the discussion in section IV. In practice, the optimal λ value can be empirically chosen in this way.

Themost important aspect of these graphs is that, except for the extreme blur examples, the graphs have very large and flat sections around the minimal error.

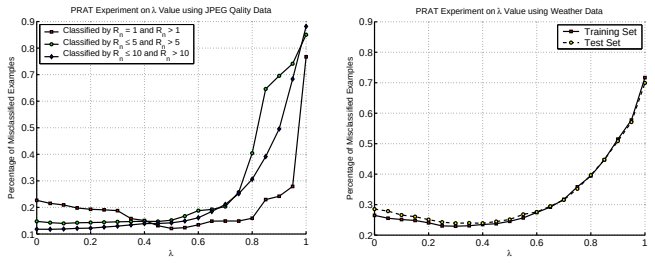

Fig. 7. Varying λ values for Jpeg and Weather data.

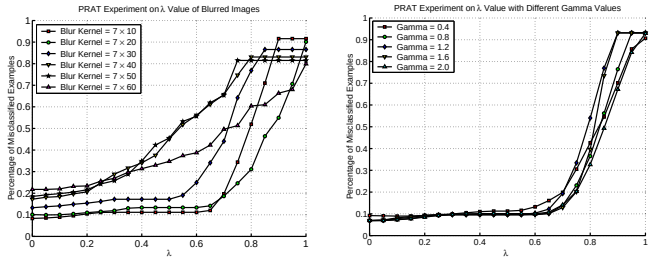

Fig. 8. Varying λ values for Blur and Gamma experiments.

VI. CONCLUSION

This paper introduces the approach of recognition failure prediction, briefly introduces its potential as a systems-level tool and explores an algorithm for such prediction. The Post-Recognition Analysis Technique is based on analysis of similarity scores resulting from a detection or recognition system. This technique provides a reliable and feasible way for predicting recognition failure. It is based on the observation that if the similarity scores considered "recognized" are distant from the "unrecognized" class, it is probably correctly recognized. However when there is little separation between the classes, failure is more likely. The paper explored an effective approach to formalize this intuitive clustering of similarities. The experimental results, on both simulated degradations and real data, show clearly that at an individual image prediction level, this technique is effective, with its prediction ability continues across varying pose and illumination. The paper presented ROC curves showing the wide range of False Alarm / Miss Detection tradeoffs that can be achieved with this approach as well as studying the impact of AdaBoost parameters.

Future work will explore using the PRAT for multi-sensor fusion, predicting which of the inputs have more value and then using that for decision-level fusion.

Since recognition using "similarity measures" is a very widely adopted technique, even though our test results are from face recognition, PRAT should be applicable in a broad context.

REFERENCES

[1] A. Jain, R. Duin, and J. Mao, "Statistical pattern recognition: a review," *IEEE PAMI*, vol. 22, no. 1, pp. 4–37, 2000.

[2] P. J. Phillips, P. Rauss, and S. Der, "FERET (Face Recognition Technology) recognition algorithms development and test report," Tech. Rep., U.S. Army Research Laboratory, 1997.

[3] M. Yang, D. Kriegman, and N. Ahuja, "Detected faces in images: a survey," *IEEE PAMI*, vol. 24, no. 1, pp. 34–58, 2002.

[4] Patrick Verlinde, Gerard Chollet, and Marc Acheroy, "Multi-modal identity verification using expert fusion," *Information Fusion*, vol. 1, no. 1, pp. 17–33, 2000.

[5] P.J. Phillips, P. Grother, R.J Micheals, D.M. Blackburn, E Tabassi, and J.M. Bone, "FRVT 2002: Evaluation report," Tech. Rep., Face Recognition Vendor Tests, 2002.

[6] L. Shapiro and G. Stockman, *Computer Vision*, Prentice Hall, 2001.

[7] L. Wenyin and D. Dori, *Performance Characterization in Computer Vision*, chapter Principles of Constructing a Performance Evaluation Protocol for Graphics Recognition Algorithms, Kluwer, 2000.

[8] S. Santini and R. Jain, "Similarity measures," *IEEE PAMI*, vol. 21, no. 9, pp. 871–883, 1999.

[9] R. Veltkamp, "Shape matching: similarity measures and algorithms," Tech. Rep., Utrecht University, 2001.

[10] A. Tversky, "Features of similarity," *Psychological Review*, vol. 84, no. 4, pp. 327–352, 1977.

[11] R. Kohavi, "A study of cross-validation and boostrap for accuracy estimation and model selection," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 1995.

[12] T. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, no. 7, pp. 1895–1924, 1998.

[13] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transitions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, August 1998.

[14] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: bagging, boosting, and variants," *Machine Learning*, vol. 1-2, no. 36, pp. 105–139, 1999.

[15] P. Viola and M. Jones, "Robust real time object detection," in *ICCV Workshop on Statistical and Computational Theories of Vision*, Vancouver, Canada, 2001.

[16] S. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, and H. Shum, "Statistical learning of multi-view face detection," in *The 7th European Conference on Computer Vision*, Copenhagen, Denmark, May 2002.

[17] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[18] H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," in *IEEE CVPR*, 1996.