NIST PFT II
Proprietary Fingerprint Template Evaluation II – PFT II
Evaluation Plan / API

May 2010

Contents

1   Overview

The NIST Proprietary Fingerprint Template Evaluation II is part of an ongoing program to measure the performance of fingerprint matching software utilizing vendor proprietary fingerprint templates. As the fingerprint capture technologies improve rapidly as well the captured fingerprint image quality, fingerprint matching algorithms are getting faster and better. NIST will continue to evaluate on two of the PFT two finger datasets; dhs2 & poebva. PFT II will also evaluate the thumb, index and middle fingerprint images from a ten print dataset. This additional fingerprint dataset, will be from either live-scan or rescan ten print cards containing both rolled and slap images. PFT II will evaluate the following combinations:

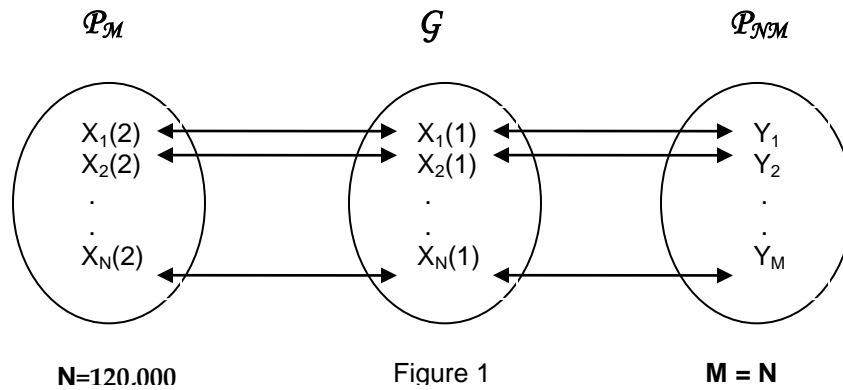- Plain image vs. plain image
- Plain image vs. rolled image

- Rolled image vs. rolled image

The dataset sample sizes will also be increasing. The evaluation protocol is similar to the earlier PFT evaluation. The probe set contains N subject ids, the gallery set contains N mates to the probe set, and a second probe set contains M (= N) non-mates (shown in Figure 1).

The evaluation is one-to-one matching. Therefore the output of the evaluation will produce N genuine scores and at least N imposter scores for analysis in the report. The non-mate probes could be matched against multiple images in the gallery to produce more non-mate scores.

$$\mathcal{P_M} \qquad\qquad \mathcal{G} \qquad\qquad \mathcal{P_{NM}}$$

$$
\begin{array}{ccc}
X_1(2) & X_1(1) & Y_1 \\
X_2(2) & X_2(1) & Y_2 \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
X_N(2) & X_N(1) & Y_M
\end{array}
$$

N=120.000   Figure 1   **M = N**

All the images used in PFT II are 500 pixels per inch. The evaluation will be an "SDK-type", in that participants will provide software, and all processing will take place on NIST hardware.

Three sub-samples of size 1K (taken from the 120K samples) with different widths and heights will be evaluated on a specific hardware and operating system to measure template size, time of template extractions, and template matching time.

Evaluation results for each SDK will be made publicly available on the PFT II website after the SDK completes the evaluation.

2   Participation

Participation in PFT II is open to anyone that has not submitted in the last 90 days (from date of last completed software SDK).

All systems must comply with the API outlined in Section 5.4. Anonymous participation will not be permitted. The Application form[1] includes details regarding application and qualification.

3   Data

3.1   Datasets

Validation Dataset

A   Validation   Dataset   is   available   at   the   NIST   website http://www.nist.gov/itl/iad/ig/pftii_validation.cfm.

---

[1] *The Application form can be found at http://www.nist.gov/itl/iad/ig/pftii_application.cfm*

The validate dataset contains a set of 40 sample images. The format of these images is identical (uncompressed 8-bit grayscale) to the images that we will use for PFT II. However, these images should not be considered representative of the images actually used for the evaluation. The images that are used for the evaluation come from a variety of sensors and have a variety of sizes (width & height dimensions).

Evaluation Dataset

The Evaluation Dataset will contain data formatted in the same manner as the Validation Dataset. The Evaluation Dataset will contain Privacy Act or FOIA Protected Information and will not be released to the participants or the public. The Evaluation Dataset will to the extent permitted by law be protected under the Freedom of Information Act (5 U.S.C 552) and the Privacy Act (5 U.S.C. 552a) as applicable.

3.2   Format

All images will be 8-bit grayscale. Raw 8-bit grayscale images are canonically encoded. The image width and height in pixels will be supplied to the SDK as supplemental information.

3.3   Resolution, Dimensions and Orientation

All images are captured at 500 pixels per inch. PFT II fingerprint images may vary from 150 to 850 pixels in width, and 150 to 1100 pixels in height. Rolled images and single finger images were captured in the upright position. Images segmented from the slaps were left in their rotated position and captured in the minimal area containing the fingerprint with white space outside the fingerprint area set to white pixels (shown in the following Figure 2).

Figure 2

3.4   Impression types

Fingerprint images used in PFT II evaluation contain two types of datasets. One type is similar to the previous PFT evaluation with only two plain index fingers. The new dataset will contain both 10 rolled and 10 plain (pre-segmented from the slap) fingerprints. The impression types are live-scan for the two index fingers and live-scan and inked for the 10 rolled/plain.

3.5   Finger positions

For the two finger datasets, right and left index finger position will be evaluated. For the ten finger dataset, right and left thumbs, index and middle fingers will be evaluated.

3.6   Dataset size

The mate and non-mate probe sets will contain 120,000 subjects for each selected finger position. The gallery set will contain the 120,000 subjects.

4   Evaluation Criteria

4.1   Performance Metrics

Performance metrics will be based on matcher score:

Matcher score metrics are evaluated in terms of DET[2] performance, by plotting False Match Rate (FMR) and False Non-match Rate (FNMR) for all score (threshold) values. Note that this approach requires that a given matcher score be comparable between genuine and imposter score.
The following will be measured:

- EER of the matcher scores from DET curve.

- 95% confidence intervals of FNMR with respect to a set of FMRs.

- Average vendor proprietary template size.

- Average extraction and matching times (on the 1K samples).

4.2   Reporting of Results

Note the application form stipulates that each participant consents to the disclosure of its performance.

Enrollment/template extraction and comparison score timing information will be reported. The report will also specify the hardware specifications used in the evaluation.

Three special subsets of size 1K from the evaluation datasets were created as the baseline for reporting the timing results for template extraction and template matching. These datasets will contain images with various widths and heights (368x368, 500x500, 800x800, and 422x1000.) These timing results will be evaluated on the same hardware for each participant's SDK.

5   Fingerprint Matching Software

5.1   Overview

Participants shall submit a SDK (Software Development Kit) that provides the interfaces defined by the PFT II API specified below. The SDKs shall be provided as dynamic libraries to run on the NIST platform specified below.

Each participant shall submit one SDK for image enrollment and comparison.

---

[2] *Detection Error Trade-off*

NIST recognizes the proprietary nature of the participant's software and will take all reasonable steps to protect the software. The software submitted will be in an executable library format, and no algorithmic details need be supplied. NIST agrees not to use the Participant's software for purposes other than indicated by the PFT II evaluation, without express permission by the Participant.

5.2   Evaluation Platform

The NIST PFT II Evaluation platform consists of an array of blade servers having a hardware configuration similar to:

Processor

- Quad 2.3 GHz/8MB Cache, AMD (4 core)
- Dual 2.8 GHz/1MB Cache, Xeon (dual-core)
- 800 MHz Front Side Bus for PE 1855

Memory

- 64bit – 192 GB RAM
- 64 bit -16 GB RAM
- 32 bit – 4 GB RAM

The operating systems available (in order of preference) are:

- CentOS Server 5.x (64-bit)
- Windows Server 32-bit & 64-bit

5.3   API

5.3.1   Evaluation Interface Description

Participants shall submit an SDK which provides the interfaces as described in the following sections.

The software undergoing evaluation will be hosted on NIST-supplied computers. The executable software under evaluation will be built up from participant-supplied SDK.

5.3.2   SDK provided Functions

5.3.2a   Get template size

```
int
get_template_size(const unsigned int width,
const unsigned int height,
unsigned int *template_size);
```

Description

This function returns the maximum number of bytes for templates given the width and height of an input image.

Parameters

width (**input**): The number of pixels indicating the width of the image.
height (**input**): The number of pixels indicating the height of the image.
template_size (**output**): The maximum number of bytes for templates.

Return Value

This function returns zero on success or a documented non-zero error code otherwise.

5.3.2b   Enroll

```
int
enroll(const unsigned char *raw_image,
const unsigned int width,
const unsigned int height,
unsigned char *template,
unsigned int *template_size);
```

Description

This function takes a raw image as input and outputs the corresponding template. The memory for the template is allocated before the call (i.e., enroll does not handle the memory allocation for the template parameter). The function returns either success (true) or failure (false). Failure indicates a failure to enroll the image. This will result in a null template that would be passed along in later comparisons.

Parameters

raw_image (**input**): The uncompressed, raw image to be used for template creation.

width (**input**):   The number of pixels indicating the width of the image.

height (**input**):   The number of pixels indicating the height of the image.

template (**output**): The processed template of size returned by get_template_size.

template_size [3](**output**): Return the actual template size.

Return Value

This function returns zero on success or a documented non-zero error code otherwise.

5.3.2c   Compare

```
int
compare(const unsigned char *probe_template,
const unsigned int probe_template_size,
const unsigned char *gallery_template,
const unsigned int gallery_template_size,
float *score);
```

Description

This function compares two templates and outputs the match score. The match score is a floating-point number.

Parameters

probe_template (**input**): The probe template returned from enroll.

---

[3] *This template size will be used to report template size statistics and as an input to SDK function* **Compare**.

probe_template_size (**input**): The size (byes) of the query template.

target_template (**input**): The target template return from enroll.

target_template_size(**input**): The size (bytes) of the target template.

Score (**output**): The score indicating the similarity between the two templates.

Return Value

This function returns zero on success or a documented non-zero error code otherwise. On failure, the function is expected to return some default "null" similarity score. Similarity scores should not be quantized. This allows for more accurate system performance analysis.

5.3.2d   Additional Functions or Parameters

When initialization and configuration functions or additional parameters are needed, supplemental documentation must be provided describing their usage.

5.3.3   Error Codes and Handling

The participant shall provide documentation of all (non-zero) errors or warning return codes. The application should include error/exception handling so that a fatal error will still return a failure code to the calling application.

All messages which convey errors, warnings or other information shall be suppressed, except where they may provide additional information not conveyable by the defined error codes alone (such as listing a specific file related to the error). During the evaluation the SDKs are executed/run on systems with limited terminal access so no GUI interfaces should be used.

At minimum the following return codes shall be used.

Return code	Explanation
0	Success
1	Image size not supported
2	Failed to extract minutiae – unspecified error
3	Failed to match templates – null probe or gallery template
4	Failed to match templates – unable to parse probe template
5	Failed to match templates – unable to parse gallery template
Others	Vendor defined – must communicate to NIST

5.3.4   SDK Library and Platform Requirements

Participants shall provide NIST with binary code only (i.e. no source code) – supporting files such as header (".h") files notwithstanding.

Note that dependencies on external dynamic/shared libraries such as compiler-specific development environment libraries are discouraged. If absolutely necessary, external libraries must be provided to NIST upon prior approval by the Evaluation Liaison.

The SDK will be evaluated in non-interactive "batch" mode (i.e. without terminal support). Thus, the library code provided shall not use any interactive functions such as graphical user interface (GUI) calls, or any other calls which require terminal interaction.

The use of multi-threading by the SDK is encouraged as the NIST evaluation platform includes dual-processor dual-core support. The SDK need not be "thread safe" as the NIST evaluation control driver itself is single threaded. If multi-threading is utilized by the SDK it shall be documented.

NIST will link the provided library file(s) to a C language control driver application (developed by NIST) using the GCC compiler *(for Windows platforms Cygwin1.7.x/GCC version 4.3.4 will be used; for CentOS 5.x/GCC 4.1.2 will be used).* For example,

```
gcc  –o pft  pft.c  -L. –lpftSDK
```

Participants are required to provide their library in a format that is linkable using GCC with the NIST control driver, which is compiled with GCC. All compilation and software execution will be performed on x86 platforms running either Windows 2003 Server or CentOS Server release 5.x (kernel 2.6.18-155 or higher) dependent upon the operating system requirements of the SDK. Thus, participants are strongly advised to verify library-level compatibility with GCC (on an equivalent platform) prior to submitting their software to NIST to avoid linkage problems later on (e.g. symbol name and calling convention mismatches, incorrect binary file formats, etc.).

5.3.5   Installation and Usage

The SDK must install easily (i.e. one installation step with no participant interaction required) to be evaluated, and shall be executable on any number of machines without requiring additional machine-specific license control procedures or activation.

The SDK's usage shall be unlimited. No usage controls or limits based on licenses, execution date/time, number of executions, etc. shall be enforced by the SDK.

It is requested that the SDK be installable using simple file copy methods, and not require the use of a separate installation program. Contact the Evaluation Liaison for prior approval if an installation program is absolutely necessary.

5.3.6   Documentation

Complete documentation of the SDK shall be provided, and shall detail any additional functionality or behavior beyond what is specified in this document. The documentation must define all error and warning codes.

5.4   Software execution process

Probe set and Gallery set template extractions.
Compare/Match Probe template against Gallery template.

5.5   Validation

As discussed in Section 3.1, a sample Validation Dataset will be provided to verify the correct operation of participants' software before and after delivery to NIST.

The images in the sample dataset are subdivided into 20 "plain" and 20 "rolled" images. In order to verify the submitted SDK is working correctly on our evaluation platform (correct operating mode,

order of comparison, etc), participants are requested to compute a 40 similarity matrix by comparing the following from your submitted SDK:

- 10 plain vs. plain

- 10 rolled vs. rolled

- 20 plain vs. rolled

Please then send us the resulting 40 scores. Using this data and the submitted SDK, identical scores must be generated by NIST to those submitted by participants in order for the submitted SDK to be accepted. Acceptance of the submitted SDK must occur prior to NIST PFT II evaluation.

5.6   Timing Requirements

The PFT II evaluation must place limits on the processing time of the major operations involving enrollment and comparison. We are recommending that the average enrollment time should not exceed <u>three</u> second per enrollment, and <u>one tenth of a second</u> per comparison.