Examples of Slap Fingerprint Segmentation Evaluation 2004

This document provides examples and illustrations of input and outputs, corresponding to the SlapSeg04 API Specification.

1.Livescan Slap Images

Below is a typical livescan image.

Below are example output segmented images from the typical livescan image above. Note the filenames (which assume the input file was named "Example.wsq"), which indicate the segmented position (A-D) and finger position (02-05).

Example_A_02.raw

Example_B_03.raw

Example_C_04.raw

Example_D_05.raw

It is permissible, but not required, to rotate the output images relative to the original, as shown below. See the Rotation section below for more information.

Example_A_02.raw Example_B_03.raw Example_C_04.raw Example_D_05.raw

Note that neighboring fingers should not be included in the segmented output, as shown by the highlights below:

2. Paper Slap Images

Below is a typical image from an inked paper source. Note the variation in background when compared to the livescan image, the handwritten and printed text, punched hole, and cropped middle finger.

Below are example output segmented images from the typical inked paper image above. Note the filenames, which indicate the segmented position (A-D) and finger position (10-07).

Example_A_10.raw Example_B_09.raw Example_C_08.raw Example_D_07.raw

3. Missing Fingers

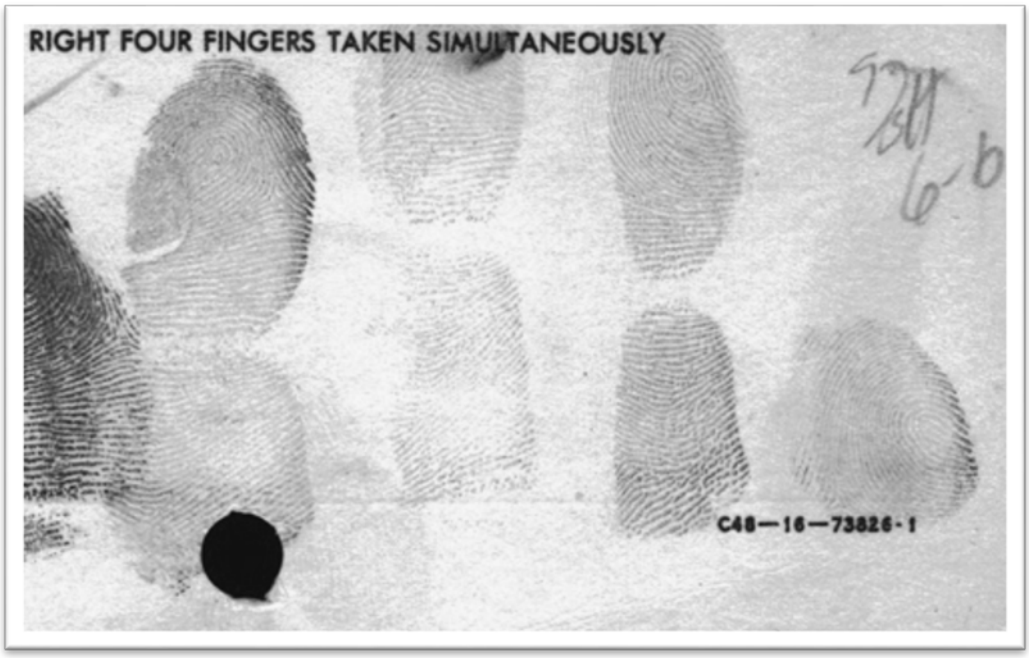
Not all images will include four fingers. In such cases, it is important that the application use the return codes (noted in section 2.3 of the SlapSeg04 API Specification) to indicate the number of fingers found:

Return code	**Explanation**
10	0 fingers could be segmented
11	Only 1 finger could be segmented
12	Only 2 fingers could be segmented
13	Only 3 fingers could be segmented

In the example below, only 3 fingers can be segmented, and the finger positions probably cannot be determined. The three resulting image files should be named (left to right) *_A_00.raw, *_B_00.raw, and *_C_00.raw. The program should return a Return Code of 13 to indicate only 3 fingers could be processed.

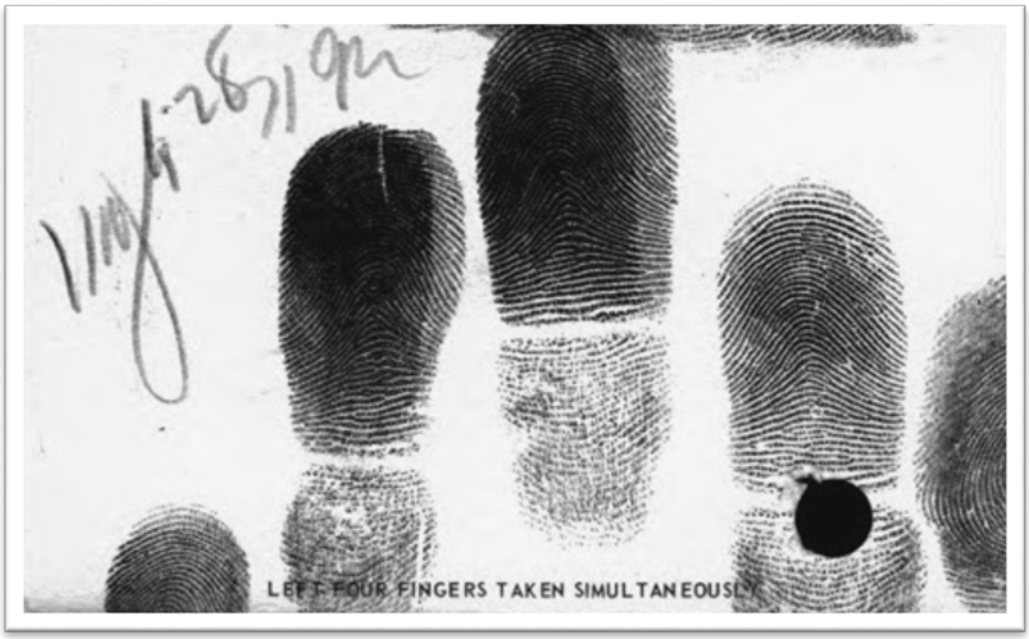
4. Extra Fingers

In some cases, images from paper cards may include fingerprints overlapped from the plain thumb impressions, which border the slap areas on the cards. Segmentation applications should never return more than 4 segmented images as output. If the finger positions cannot be determined, all fingers should be noted as undefined, with a 00 finger code in the filenames.



5. Segmentation Quality

Segmentation Quality is a user-defined numeric value. It is requested, but not required. A higher segmentation value must correspond to a higher likelihood that that image was correctly segmented. The values can be integers or decimals.
In an operational environment, a measure of segmentation quality is necessary to indicate those problem cases that may require special processing. Below are examples of some images for which it may be difficult to obtain an optimal segmentation of all fingers. In such cases, it is recommended that the segmentation application return segmentation quality values that would indicate that the segmentation was not definitive.



6. Rotation

Applications may optionally return rotation information. The study is being designed so that rotating the output does not affect the evaluation. Applications are not evaluated based on rotation information, but it greatly assists in analysis.
The Meta-information file provides for two fields, ORIGINAL_ROTATION and OUTPUT_ROTATED.

ORIGINAL_ROTATION is the amount of rotation of the original (input) finger from vertical, in degrees. Positive values are clockwise rotation, and negative values are counter-clockwise. The values can be integers or decimals. OUTPUT_ROTATED should be set to TRUE if the output images are rotated relative to the input.
In the example below, each finger is rotated about 45 degrees clockwise, so ORIGINAL_ROTATION should be set to 45. If the output segmented images are rotated to upright, OUTPUT_ROTATED should be set to TRUE.

Rotation is reported for each finger in the slap image. If rotation is only measured for the entire slap image, the rotation values should be set to that value for each finger.

7. Bounding Boxes

Applications may optionally return bounding box coordinates. These define the four corners of a rectangle bounding the individual segmented finger, measured in pixels from the top left corner of the input image. Applications are not evaluated based on bounding box information, but it greatly assists in analysis.
The example below shows bounding boxes when the output is not rotated relative to the original. Note that the bounding boxes may overlap.

The example below shows bounding boxes when the output is rotated relative to the original.