Assignment #1

Image Based Biometry
Faculty of Computer and Information Science
University of Ljubljana

Fingerprint Recognition

Your goal is to perform and evaluate fingerprint recognition, using NIST Biometric Image Software (NBIS) https://www.nist.gov/services-resources/software/nist-biometric-image-software-nbis. Feel free to use modern LLM helpers, search engines and websites. Just make sure you understand each line of code produced and logic behind it.

Use a programming language of your choosing (do not use assembler, Brainfuck, or other esoteric languages). Python is the preferred choice.

If you need info on what each function (program) accepts, refer to the manual available here: https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir7392.pdf or check the source files.

I. Installation and Preparation

- Download NBIS software from here: https://nigos.nist.gov/nist/nbis/nbis_v5_0_0.zip
- Select one or multiple finger datasets that are available for direct download from here: https://github.com/robertvazan/fingerprint-datasets. Images should look something similar to the image shown in Fig. 1. Class (subject) annotations you can derive from file names.
- Copy the Overleaf project from here (or download it from Eučilnica): https://www.overleaf.com/read/qdvbkmpkxbfw

Follow instructions in INSTALL_[ENV].txt to compile the software. However,

- To setup.sh supply absolute installation path and not relative.
- On Linux and Unix make sure to compile it using GCC, otherwise you will have to:
  - You need to add `#include <unistd.h>` to these files: `gzread.c`, `gzwrite.c`, and `gzlib.c` in `Rel\_5.0.0/openjp2/src/lib/openjp2/thirdparty/libz/`.
  - Change line 144 from `rindex` to `strrchr` in `Rel\_5.0.0/imgtools/src/bin/dpyimage/dpymain.c`.
- On Windows use MinGW shell to run `setup.sh`.

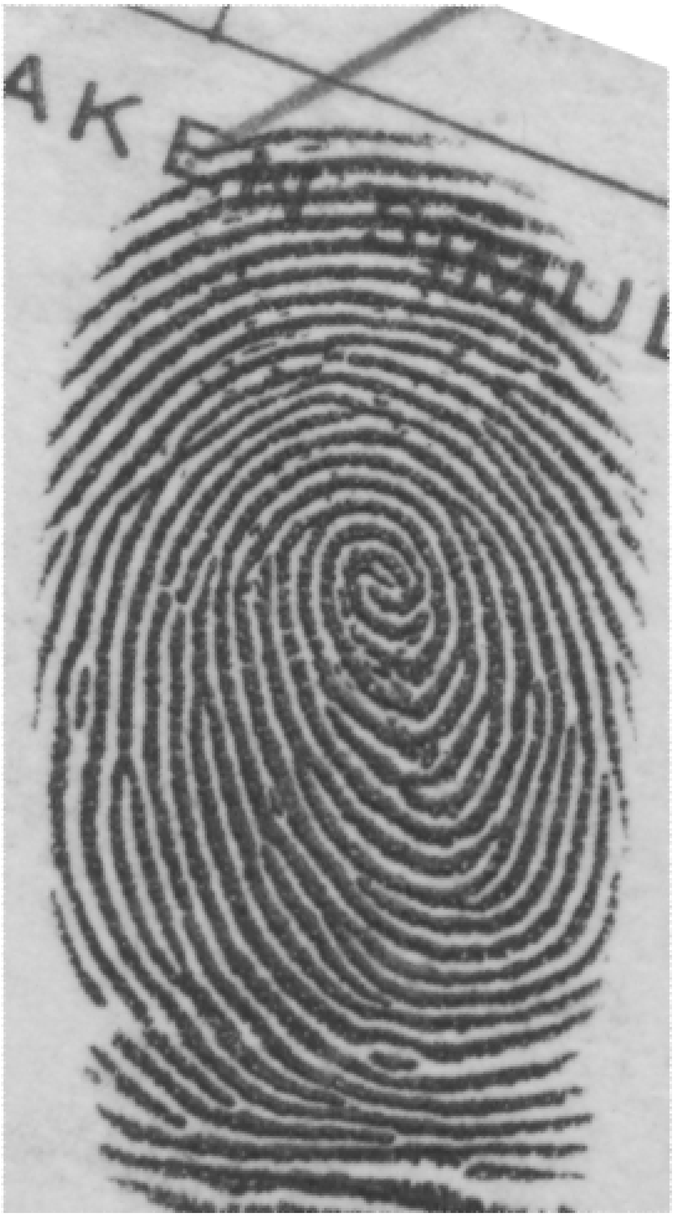

Fig. 1. Example of a fingerprint image.

- After the installation, add the `bin` folder of the target install directory to your `PATH` (`~/.bashrc`, `~/.zshrc`, or Windows `PATH`).
- Run `mindtct` to see if the installation was successful.
- Note that on Windows in order for pcasys to work, you need to go relatively all the way to the root (C) "../" and then to the location of the files.

II. Tasks and Grading

First steps:

- Transform the images from `TIFF` to `PNG` (`magick` works well: `magick img.tif img.png`).
- Compute minutiae points for all the images: `mindtct img.png out/img`.
- Compute all the `bozorth3` comparisons (all vs. all): `bozorth out/img1.xyt out/img2.xyt`.

- **0.25 pts** Plot `bozorth3` values for impostors and genuines on one graph.
- **0.25 pts** Plot the similarity matrix of all the comparisons.
- **0.25 pts** Using `nfiq` (`nfiq img.png`), which gives you an estimate of fingerprint quality (1 is best, 5 is worst), plot the quality values of the fingerprints.
- **1 pt** Determine the best threshold for `bozorth3` outputs (to distinguish same or different subjects) as accurately as possible. Use an approach of your choosing:
  - You can select it based on observations from the genuine-impostor plot.
  - You can compute FAR, FRR, and then the EER from them.
  - You can empirically set the threshold by plotting the accuracy across different thresholds.
  - Any other approach of your choosing, just describe it clearly in the Methodology section of the report.

- **1 pt** Perform classification and report classification accuracy (the number of correctly classified pairs divided by the number of all comparisons). Do this by comparing all samples only once; no comparisons with themselves or repeated comparisons (i.e., do only A vs. B, and no B vs. A, A vs. A, B vs. B), and checking whether their class annotations match or not.
- **1 pt** Modify the classification so that you first classify each fingerprint by its type (`A = Arch`, `L = Left Loop`, `R = Right Loop`, `S = Scar`, `T = Tented Arch`, `W = Whorl`), and then only match within this subgroup. To do this, call `pcasys pcasys.prs`. However:
  - You need to modify the path in `pcasys.prs` to match the relative location of your text file to `pcasys` (a text file which contains the list of images).
  - You need to edit the text file to contain the list of `WSQ` images you have. That file should be space-delimited with just some fingerprint type hardcoded (it does not matter which; you can just put `A` for each) -— see the example `paths\_to\_your\_fingerprints\_for\_pcasys.txt`.
  - To get `WSQ` format for each fingerprint, see the example `png2wsq_example.sh`. Upscale to at least 300px if needed.
  - Ignore the tabular output `pcasys` gives you (as it is an evaluation we are not interested in, and it is also incorrect since our ground truth is just a bunch of hardcoded letters). Instead, filter out the class (letter) you get after "hyp" (two in each output, but it doesn't matter which one you pick). For example: is `A`; `nn: hyp L, conf 0.94; conup n; hyp L, conf 0.94`; in this case, `"L"` is what you are looking for.
- **1.25 pt** Final report, which must follow the IMRAD structure, should fit on **one** page, and include all three plots, two performance values for both classification experiments and explanations of threshold selection. The report should have the following sections: introduction, methodology (explaining how you selected the threshold), experiments with results, and conclusion. Related work and abstract you can skip in this assignment. Everything must be extremely concise and polished to fit into one page. Use the template provided on Eučilnica.

Note: In experimental work it is crucial to split the dataset into train and test sets (while ensuring there are at least two samples/impressions per subject in each split). However, since we are dealing with very small datasets here, we will treat the whole dataset as a training dataset. All experimentation and the "final" reports will be therefore based on whole dataset.

III. Submission

Submit **your code** and **the report** on Eučilnica by the deadline. **DO NOT SUBMIT ARCHIVE FILES.** Oral defenses will be during that week.