A COMPARATIVE STUDY OF FINGERPRINT MATCHING TECHNIQUES

CHAPTER 3 - Methodology

ID	FULL NAME
10736694	Terence Ugo Nacciarone Quashie
10729461	Abdul-Aziz Abubakar Saddick

Contents

Introduction

The aim of the research is to effectively compare two fingerprint matching algorithms; minutiae matching and Scale Invariant Feature Transformation (SIFT) matching. Implementation of both algorithms were done using the Python Programming language accompanied by various libraries such as PYQT5 for the Graphical User Interface (GUI), NumPy for various math functions, OpenCV2 library for image processing as well as matplotlib for data visualization. Detailed in the next sections are the various processes and ways that all these tools were synthesized together to collect data.

Research Design

This design method of the research combines various techniques to seek answers to "What's" and "How's" (Team Leverage Edu, 2021). In this case:

i. "**What** are fingerprint matching algorithms?"
ii. "**How** do these fingerprint matching algorithms work?"
iii. "**How** are they implemented?"
iv. "**Which** algorithm is more accurate?"
v. "**What** are the times each algorithm takes to finish?"

From these questions, the results collected were analyzed to find patterns and make interpretations.

Data/Information Collection

To understand the science behind fingerprint identification and matching, various papers were consulted and referenced on the two methods chosen. Implementation of the two algorithms varied and as such multiple sources were combined to obtain a general implementation of both algorithms.

Dataset

For this research, the Sokoto Coventry Fingerprint Dataset (SOCOFing) was used. This dataset is a biometric fingerprint database designed for academic purposes. It consists of over six thousand (6,000) fingerprint images from over six hundred (600) subjects with unique attributes such as labels for gender, finger and hand name. Also included with the dataset, are three different levels of altercations for obliteration, z-cuts and central rotations (Ruizagara, 2018). A summary of the nature of the dataset is given below:

Fingerprint Sets	**Real**	**Altered - Easy**	**Altered - Medium**	**Altered – Hard**
Dimension	96 x 103	96 x 103	96 x 103	96 x 103
Image Type	Bitmap (.BMP)	Bitmap (.BMP)	Bitmap (.BMP)	Bitmap (.BMP)
Image Size (KB)	38.7	10.7	10.7	10.7
Number of Images	6,000	17,931	17,067	14,272

Algorithm Implementation

To realize these two algorithms, the Python Programming language was used accompanied with various libraries.

Python Programming Language

Python version **3.9.13 MSC v.1929 64bit (AMD64)** was used throughout the implementation of both algorithms. This version was chosen for its stability, ease of use as well as the volume of computer vision and mathematical computation libraries supported by major backers. The libraries which were used include the following

i. NumPy
ii. OpenCV2
iii. Matplotlib
iv. PyQt5

NumPy

NumPy is a library for scientific computing in Python, it provides a variety of routines for fast operations on arrays including mathematical, logical, selecting, basic linear algebra, I/O as well as discrete Fourier Transforms (NumPy Developers., 2022). The version being used for this project is ***version 1.23.1***

OpenCV2

OpenCV2 is a Python library started by intel's Gary Bradsky in 1999, it a set of Python bindings designed to solve computer vision problems (OpenCV, 2022). It is originally a C++ library that uses Python Wrappers to create modules, making code run as fast as its original implementation in C++. ***Version 4.6.0.66*** is used for this project.

Matplotlib

Matplotlib is a comprehensive library for creating static, animated and interactive visualizations in Python. Matplotlib allows a diagrammatic representation of all unique points on a fingerprint, as well as its corresponding matching points on the sample image (Hunter, 2007).

PyQt5

PyQt5 is a Graphical User Interface (GUI) framework that wraps around the C++ library, Qt. It allows the construction of GUI that hide the abstraction of code (pythonpyqt, 2020). Using PyQt5, an algorithm comparison program has been built which allows users to load images and run both comparisons, results are displayed side by side in tabs in such a way that the various phases of each algorithm can be analyzed directly.

Fingerprint Matching

Each image was processed with both algorithms two times to reduce false acceptancy rate (FAR) and false rejection rate (FRR) and compared to its three (3) respective altered versions with their matching scores. Graphs were drawn to visually represent the match score between the two algorithms

Algorithm Completion Time

Both algorithms were tested on two separate computer systems with the following specifications

Computer A	
Operating System	Windows 10 Pro 64-bit (10.0, Build 19044)
Processor	Intel(R) Core™ i5-3210M CPU @ 2.50Ghz (4 CPUs)
Memory	8192MB RAM

Computer B	
Operating System	Windows 10 Pro 64-bit (10.0, Build 19044)
Processor	Intel(R) Core™ i7-10870H CPU @ 2.20Ghz (16 CPUs)
Memory	16384MB RAM

Computer specifications were taken by running "dxdiag" on both systems

The time taken to complete both algorithms on the same set of data was recorded and represented in graphs

Validity of Data

To ensure that consistent results were obtained, the following procedures were put in place:

- All algorithms were run on the same set of images in succession
- The same versions of the Python Programming Language with its various libraries were used on both computer systems
- Both algorithms were run using the same configurations
- All match scores were taken on the first instance of running the algorithm to simulate a real-world experience

Limitations

- Both algorithms could not be fully optimized due to time limitation (hence each take a longer time to compute on slower computers)
- The latest version of the Python Programming Language was not available

References

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering, 9*, 90-95. doi:10.1109/MCSE.2007.55

NumPy Developers. (2022). *What is NumPy?* Retrieved from NumPy: https://numpy.org/doc/stable/user/whatisnumpy.html

OpenCV. (2022, June 5). *Introduction to OpenCV-Python Tutorials*. Retrieved from OpenCV: https://docs.opencv.org/4.6.0/d0/de3/tutorial_py_intro.html

pythonpyqt. (2020). *What is PyQt?* Retrieved from LEARN PYTHON PYQT: https://pythonpyqt.com/what-is-pyqt/

Ruizagara. (2018). *Sokoto Coventry Fingerprint Dataset (SOCOFing)*. Retrieved from Kaggle: https://www.kaggle.com/datasets/ruizgara/socofing

Team Leverage Edu. (2021, November 17). *Research Desi*. Retrieved from Leverageedu: https://leverageedu.com/blog/research-design/