

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 2](#_Toc115421719)

[Research Design 2](#_Toc115421720)

[Qualitative Research Design 2](#_Toc115421721)

[Research Approach 2](#_Toc115421722)

[Preliminary Research 2](#_Toc115421723)

[Dataset 2](#_Toc115421724)

[Algorithm Implementation 3](#_Toc115421725)

[Python Programming Language 3](#_Toc115421726)

[Data Analysis 3](#_Toc115421727)

[Fingerprint Matching 3](#_Toc115421728)

[Algorithm Completion Time 4](#_Toc115421729)

[Ethical Considerations 4](#_Toc115421730)

[Limitations 4](#_Toc115421731)

# 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

**Look into mixed methodology research**

Due to the nature of the research, the qualitative research design (was it only qualitative or quantitative) was chosen.

## Qualitative Research Design

This design method seeks answers to the “What’s” and “How’s” ([Research Design: Definition, Types & Characteristics | Leverage Edu](https://leverageedu.com/blog/research-design/)). (EXPAND MORE OF THIS?) In this case:

1. “**What** are fingerprint matching algorithms?”
2. “**How** do these fingerprint matching algorithms work?”
3. “**How** are they implemented?”

This type of research allows conclusions to be drawn from the findings that have been made.

# Research Approach

**(INTRO TO SECTION?)**

## Preliminary Research

To understand the science behind fingerprint identification and matching, various papers were consulted and referenced on 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 ([Sokoto Coventry Fingerprint Dataset (SOCOFing) | Kaggle](https://www.kaggle.com/datasets/ruizgara/socofing)). 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

1. NumPy
2. OpenCV2
3. Matplotlib
4. 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 ([What is NumPy? — NumPy v1.23 Manual](https://numpy.org/doc/stable/user/whatisnumpy.html)). 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: Introduction to OpenCV-Python Tutorials](https://docs.opencv.org/4.x/d0/de3/tutorial_py_intro.html#:~:text=OpenCV%2DPython%20is%20a%20library,its%20simplicity%20and%20code%20readability.)). 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 ([Matplotlib — Visualization with Python](https://matplotlib.org/)).

#### 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 ([What is PyQt? | Learn Python PyQt](https://pythonpyqt.com/what-is-pyqt/)). 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.

# Data Analysis

## 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. (MENTION ANALYSIS TECHNIQUES USED). 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

# Ethical Considerations

To ensure that accurate 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

(INTRO HERE)

* Both algorithms have not been fully optimized hence take a longer time to compute on slower computers
* The version of the Python Programming Language is not used and as such performance may be skewed