A COMPARATIVE STUDY OF FINGERPRINT MATCHING TECHNIQUES

A Proposal

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

Table of Contents

Introduction

In 1983 when the Home Ministry Office, UK concluded that no two individuals can have the same fingerprints, it set in motion a series of events that led to the widespread use of fingerprint pattern systems, known as the AFIS (Automatic Fingerprint Identification Systems). These systems are actively used by law enforcement agencies all over the world today. In fact, these fingerprint matching systems have become so successful in criminal investigations that the term fingerprint has become synonymous with the word inherent characteristic or unique characteristic.

The success of fingerprint identification systems has spurred a wide spiral of its application beyond the forensic domain to a "second generation" geared towards civilization application such as biometric authentication. The second generation of fingerprint identification systems operate automatically and are deployed in more mainstream applications that deal with a larger cross-section of society such as unlocking mobile phones or gaining access to a secure app such as a banking or blockchain app or gaining access to homes using smart locks.

It is an undeniable fact that the current implementation of fingerprint identification systems (rephrase?) have positively reduced cases of identity theft in our society. However, the "million-dollar" question still remains; "Will these fingerprint biometric technologies work all the time? Will they work everywhere and, in all contexts, reliably identifying and authenticating a person". These are questions that have pushed researchers to develop newer and more efficient methods of automatic fingerprint identification and matching.

One of the design criteria for building such an automatic and reliable fingerprint verification system is that the underlying sensing, representation and matching technologies must be very robust. It is essential that these individual components have little to no degradation in their performance in the long term. One way to address these requirements of robust performance is to adopt a robust representation scheme that captures all discriminatory information to a high degree of accuracy.

The most popular representation of these discriminatory information is through local landmarks known as **minutiae**. This method evolved from the system of visually matching fingerprints used by forensic experts. **The minutiae-based algorithm** works by locating these local landmarks where fingerprint ridges either terminate or bifurcate and then match minutiae relative placements between a given fingerprint sample and the stored template. A good quality fingerprint contains between 25 and 100 minutiae depending on sensor resolution and finger placement on the sensor. The minutiae-based system is a well-known method for fingerprint verification.

Although the minutiae based-system is robust and the most widely used method of fingerprint verification, it struggles when given poor quality fingerprint impressions arising from dry fingers or fingers mutilated by scars and scratches. There is also anecdotal evidence that a fraction of the population may have fingerprints with relatively small number of minutiae thereby making it more vulnerable to failures.

This short-coming of the minutiae-based system meant that there was need to extend characteristic feature matching beyond minutiae points. For this, techniques such as the **Scale Invariant Feature Transformation (SIFT)** was introduced, an object matching algorithm. SIFT works by constructing a scale space from which descriptors are extracted and used in the matching process.

Using a public domain fingerprint dataset, this project will implement the two algorithms and compare their performance and useability.

Objectives

Algorithm Implementation

The minutiae-based algorithm and the scale invariant feature transform (SIFT) algorithm will be implemented using the python programming language. Both algorithms will recognize unique features in the same fingerprint sample.

Feature Recognition

Following the implementation, both algorithms would be tested against sample data to test their feature recognition capabilities.

Matching Using Features

Using the same fingerprint sample from the dataset, the matching capabilities of each algorithm will be tested along with their performance documented and represented in a graphical form. The fingerprint sample would also be distorted to further test the matching capabilities of both algorithms.

Compare Performance

Based on the values obtained from the previous objectives, the performance of both algorithms on the sample data will be tested using various performance metrics such as False Acceptance Rate (FAR) and False Rejection Rate (FRR). Other performance metrics such as time complexity, total time to process an image will be taken and analyzed.

System Implementation

A login system with fingerprint authentication will be developed to demonstrate a scenario where fingerprint authentication is necessary. Using the SIFT algorithm. An object detection system will also be built to highlight the usefulness of SIFT

Method And Design

Sample Data

The Kaggle Coventry Fingerprint dataset will be used in the undertaking of this project. The database contains over 6,000 (Six Thousand) sampled fingerprint images, as well over 14,000 (Fourteen Thousand) alterations of these fingerprint images. The sample images are a monotone set with a resolution of 96 x 103 pixels. A sample will be taken from this pool and compared with their altered version using the both algorithms.

Procedure

A theoretical approach will be taken detailing the important constituents of the both algorithms including the various techniques used in their implementation. These two algorithms will then be implemented using the python programming language. Various python libraries will be used to visualize the data in graphical form such as **PYQT5** for the user interface where the path to the sample image can be specified for the algorithm to run.

Analyses

The performance of the two algorithms will be measured, compared and represented in a graph and tabular form to visualize the differences between the two algorithms. The performance metrics to be used include

i. False Acceptance Rate
ii. False Rejection Rate
iii. Time to complete algorithm
iv. Number of minutiae extracted and successfully matched

Discussions of Findings, Conclusion and Recommendations

Results from the analyses will be discussed and conclusions drawn. Recommendations for further work will also be suggested.