Revista Industrial Data 24(2): 293-317 (2021)
DOI: https://doi.org/10.15381/idata.v24i2.20895

ISSN: 1560-9146 (Impreso) / ISSN: 1810-9993 (Electrónico)
Facultad de Ingeniería Industrial - UNMSM

Efficient Search Method to Solve the Fingerprint Identification Problem by Applying Machine Learning

María Elena Ruiz Rivera [1]
Edgar Ruiz Lizama [2]

Received: 04/08/2021 Accepted: 04/11/2021 Published: 31/12/2021

ABSTRACT

In biometrics technology, the fingerprint identification problem has been widely studied over the last decades due to its applicability in person identification cases. In casualty cases, recognition of the victim is required, which should be done unequivocally using fingerprint identification. The aim of this research is to innovate the fingerprint identification process, developing an efficient search method in a large database that allows finding a fingerprint in less time by classifying fingerprints into segments, according to their closest characteristics, using machine learning. Then, in a given segment, a discrete linear search algorithm is applied, with which the required fingerprint is located.

Keywords: Fingerprint; biometrics; machine learning; fingerprint search.

INTRODUCTION

As a result of the increase in technology regarding communication and data transfer through networks, the use of fingerprint systems has become common in forensics and law enforcement, both in government and private institutions. One example of this is at banks, where the fingerprint is verified for cash withdrawals, which provides security to the customer against identity theft or impersonation. Another example is when a criminal enters the operating room with the purpose of undergoing several aesthetic surgeries to significantly change his/her physical appearance; the identity of this person is revealed thanks to his/her fingerprint, which is unique and unrepeatable. Similarly, in the case of a person disfigured by an accident, his or her unequivocal recognition is possible thanks to his or her fingerprints.

Investigation is based on the search for information in large databases, so the response time is reflected in very high costs; thus, the search for images further complicates the collection of information. Storing a digital image requires a large memory space, so the cost increases for a large number of images. The search for this type of information can be performed using different types of algorithms; however, the response time and cost involved in searching these large databases, where response time is critical, must be taken into account.

The efficient fingerprint search leads to the general research objective, which is to propose an efficient fingerprint search method at a lower cost and time, in which machine learning is applied for segment classification and a discrete algorithm for linear search in a given segment.

[1] Degree in Computer Science from Universidad Nacional Mayor de San Marcos (Lima, Peru). Currently working as professor at the School of Systems Engineering and Computer Science at the Universidad Nacional Mayor de San Marcos.
ORCID: https://orcid.org/0000-0003-3300-7068
Corresponding author: mruizr@unmsm.edu.pe

[2] Industrial Engineer from Universidad Nacional Mayor de San Marcos and Master in Computer Science from Pontificia Universidad Católica del Perú (Lima, Peru). Currently working as professor at the School of Industrial Engineering at Universidad Nacional Mayor de San Marcos.
ORCID: https://orcid.org/0000-0001-9403-1358
E-mail: eruizl@unmsm.edu.pe

Performing a linear search in a large database generates a delay in the search time. Fingerprints have features that allow their classification, such as arch, right loop, left loop, tentarch and whorl (Dass & Jain, 2004). The fingerprint search work presented here focuses only on fingerprints of individuals in a macro database in order to reduce the search time.

Fingerprint classification consists of systematically partitioning the database into different segments using machine learning. These segments are formed by the approximation of the characteristics of each fingerprint. The classification of fingerprints into segments significantly reduces the time spent on fingerprint identification, especially in situations where accuracy and speed are critical.

The proposed method consists of first classifying fingerprints by their characteristics using machine learning, so that it is possible to create groups by similar or closer characteristics; then, in a second step, using a discrete algorithm that allows performing a linear search and finding the required fingerprint.

THEORETICAL FRAMEWORK

Biometric Systems

A biometric system is essentially a pattern recognizer that captures biometric data from a person, extracts a set of features from that data, and compares them to other patterns previously stored in the system (Wayman, Lain, Maltoni, & Maio, 2005).

A biometric system is an automated system that performs biometric tasks. That is, its recognition decisions are based on the physical or behavioral characteristics of a person in an automated manner (Fernández, 2008).

To solve these problems, methods are being developed based on certain biometric features that guarantee the identity of individuals. These biometric traits are classified into two types (Fernández, 2008), the first is physiological biometrics, based on body parts, such as fingerprints, iris, retina, voice, hand and face; the second is behavioral biometrics, based on a person’s actions, such as a person's signature.

Characteristics of a Biometric Indicator

According to Fernández (2008), a biometric indicator is some characteristic with which biometrics can be performed, thus we have:

- Universality: the biometric trait exists for all individuals.
- Uniqueness: the biometric trait univocally identifies each person.
- Permanence: the biometric trait remains unchanged over time in the short term.
- Immutability: the biometric trait remains unchanged over time in the long term or throughout life.
- Measurability: the biometric trait is suitable for quantitative characterization.
- Performance: the biometric trait allows the individual to be recognized quickly, robustly and accurately.
- Acceptability: the biometric trait must be accepted by the majority of the population.
- Invulnerability: the biometric trait allows the system to be robust against fraudulent access methods.

Table 1 shows the various biometric technologies according to the degrees of confidence (high, medium, low) of the properties described above (Maltoni, Maio, Jain, & Prabhakar, 2003).

Biometric Identification

Biometric identification has been defined by Professor Jain Lakhmi as the process of automatically linking the identity of an individual through the use of some physical or behavioral characteristics inherent to the person (Fernández, 2008).

Fingerprint

A fingerprint is the representation of the surface morphology of the epidermis of a finger (Persto, 2020). It forms in the fetal stage of the human being and is constituted by papillary ridges; it is also immutable throughout life, unless it suffers severe injury or damage (Villamizar, 1994).

Fingerprint Characteristics

In 1892, Francis Galton published the first classification system and established the individuality and permanence of fingerprints; the “fine details” that Galton identified are used today (Persto, 2020).

Dass and Jain (2004) based their work on the fingerprint classification of Henry (1900), who proposed five main fingerprint classes based on the NIST4: (a) left loop, (b) right loop, (c) arch, (d) tentarch, and (e) whorl. This classification is shown in Figure 1.

Table 1. *Comparison of Biometric Technologies.*

Biometric Indicator	Universality	Uniqueness	Permanence	Measurability	Performance	Acceptability	Invulnerability
DNA	High	High	High	Low	High	Low	Low
Ear	Medium	Medium	High	Medium	Medium	High	Medium
Face	High	Low	Medium	High	Low	High	High
Facial thermograph	High	High	Low	High	Medium	High	Low
Fingerprint	Medium	High	High	Medium	High	Medium	Medium
Gait	Medium	Low	Low	High	Low	High	Medium
Hand geometry	Medium	Medium	Medium	High	Medium	Medium	Medium
Hand veins	Medium	Medium	Medium	Medium	Medium	Medium	Low
Iris	High	High	High	Medium	High	Low	Low
Keystroke	Low	Low	Low	Medium	Low	Medium	Medium
Smell	High	High	High	Low	Low	Medium	Low
Retina	High	High	Medium	Low	High	Low	Low
Signature	Low	Low	Low	High	Low	High	High
Voice	Medium	Low	Low	Medium	Low	High	High

Source: Data based on the perception of the authors of the book Handbook of Fingerprint (Maltoni et al., 2003).

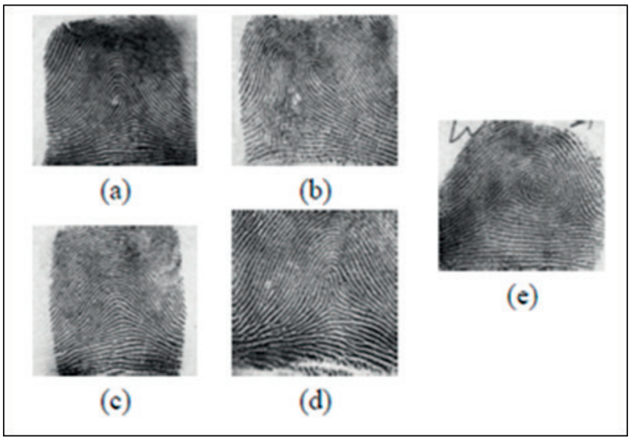

Figure 1. Fingerprint classification.

Source: National Institute of Standards and Technology (NIST4) special database (Dass & Jain, 2004).

Fingerprint Storage Module

The fingerprint storage module is responsible for the analog or digital acquisition of some biometric indicator of a person, such as the acquisition of a fingerprint image using a scanner. Once the fingerprint is obtained, it is stored in the database (patterns or templates) from a biometric device. The process is depicted in Figure 2.

Process Architecture of a Biometric System

The process starts when the fingerprint is entered into a fingerprint recognition device with an optical fingerprint scanner. This device transforms the information into digits and captures the image of the entered fingerprint. Once digitized, the fingerprint is taken to the database (template) by means of mathematical algorithms according to its particular characteristic.

The fingerprint to be searched is then live captured, extracting its characteristics and running a search in the database where the fingerprints have been previously stored. If it finds matches with the image entered, the identity of the person to whom the image corresponds can be verified, which gives a positive result in the search. If no matches are found in the digitized image of the fingerprint, it is compared and gives a negative search result (see Figure 3).

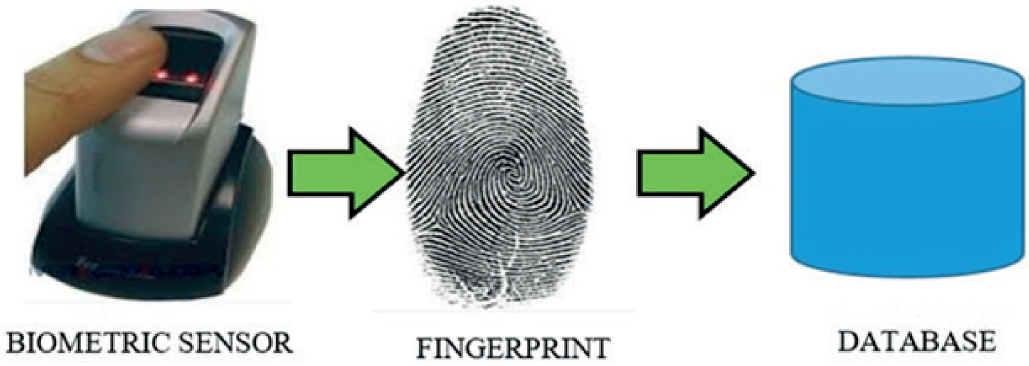

Figure 2. Fingerprint storage module of a biometric system.
Source: Prepared by the authors.

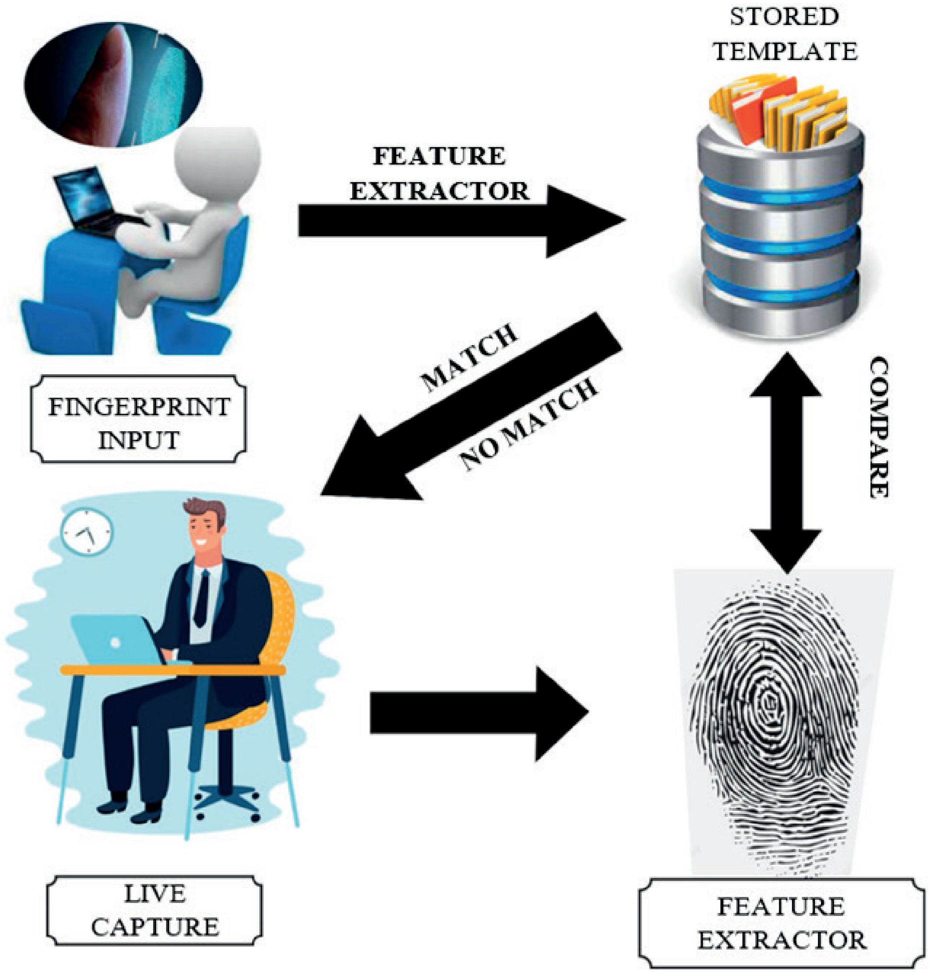

Figure 3. Process architecture of a biometric system.
Source: Prepared by the authors.

Clustering Method

Clustering has various applications in computer science, such as image compression (Scheunders, 1997) and voice digitization (Makhoul, Roucos, & Gish, 1985); retrieval of related information (Bathia & Deogun, 1998); data mining, where the search for groups with certain characteristics of interest is carried out (thus discovering new customer segments with the aim of improving the services provided)

(Fayyad, Piatetsky-Shapiro, & Padhraic, 1996); image segmentation by dividing the image into homogeneous regions (according to some characteristic of interest such as intensity, color or texture), which is especially important in medical applications (Pham & Prince, 1999); and classification of satellite images into different zones (urban, open fields, rivers, forests) (Soldberg, Taxt, & Jain, 1996).

Clustering methods differ from each other on how they compose the clusters. Those that do so according to the correspondence to a partition of the set of objects are known as hard clustering methods (Kearns, Mansour, & Ng, 1997); of these, the best known is the k-means algorithm (Forgy, 1965; MacQueen, 1967).

K-Means Algorithm

K-means algorithm (Forgy, 1965) is a heuristic commonly used to solve the clustering problem (MacQueen, 1967). The basic idea of the algorithm is to have the initial k centers and to assemble clusters by associating the objects of X to the nearest centers; then, the centers are recalculated. If the new centers do not differ from the previous centers, the algorithm terminates; otherwise, the association process is iterated with new centers until there is no variation in the centers or some new stopping criterion is established with a small number of reassignments of the objects for these methods.

Machine Learning

Machine learning is a set of computational algorithms that share a common principle: The user does not implement the evaluation function explicitly, but simply provides the computer with a way to autonomously create this function and then optimize it from experience based on learning data. In other words, the user does not enter the criteria used by the computer, the computer determines the criteria using a particular algorithm.

In this project, we specifically employ an algorithm called k-means, which is a data segmentation/clustering algorithm. This method was formally proposed for the first time in 1957 by the mathematician Stuart P. Lloyd, although it was officially published in 1982 in an article entitled Least Squares Quantization in PCM. Several optimizations of this algorithm have been carried out over the decades, leading to recent implementations. K-means consists of segmenting a set of points in a Euclidean space as follows: First, the algorithm assigns k centroids randomly (where k is an arbitrary value chosen by the user). The clusters are then segmented according to the minimum distance of each point from each of these centroids as shown in Figure 4.

Once the k clusters are formed, the centroids are recalculated using the average of the set of points belonging to each cluster, as shown in Figure 5. The algorithm is iterative, so the process is repeated until the centroids converge. In this case, convergence means that the clusters formed remain constant even in subsequent iterations. It should be noted that the convergence of this algorithm in a finite number of iterations has already been demonstrated, so it is always possible to reach an end.

METHODOLOGY

The literary review was carried out, the written information on the fingerprint search was analyzed in a large database and a good number of articles related to the topic were found in the indexed databases of recent years. Each of the articles found in the information search was reviewed, which turned out to be relevant to broaden and determine the scope of

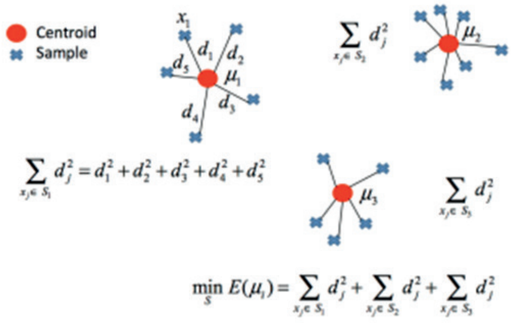
Centroid
Sample
x_1
d_1
d_2
d_3
d_4
d_5
μ_1
μ_2
μ_3
$\sum_{x_j \in S_2} d_j^2$
$\sum_{x_j \in S_1} d_j^2 = d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2$
$\sum_{x_j \in S_3} d_j^2$
$\min_S E(\mu_i) = \sum_{x_j \in S_1} d_j^2 + \sum_{x_j \in S_2} d_j^2 + \sum_{x_j \in S_3} d_j^2$

Figure 4. K-means algorithm.

Source: Zúñiga (n.d.).

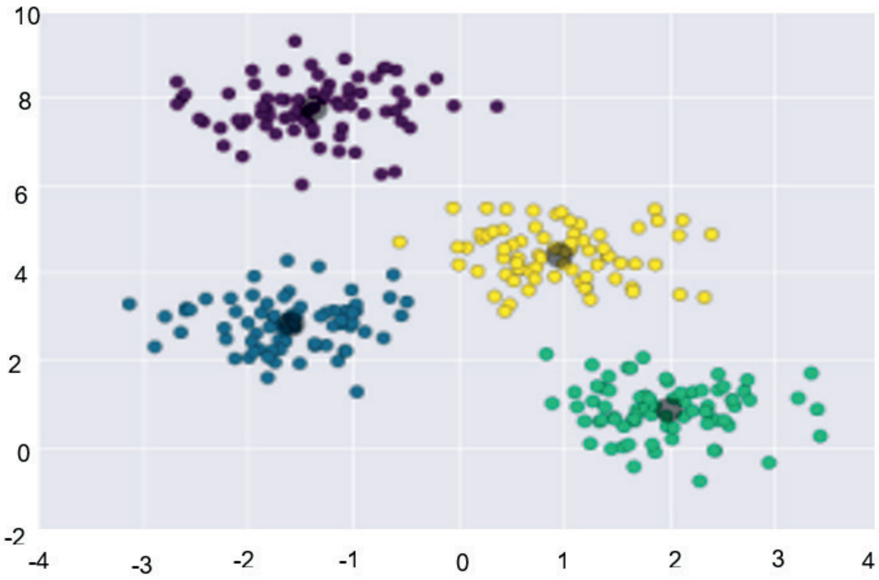

Figure 5. Data clustering under k-means algorithm criteria.
Source: VanderPlas, J. (2017).

the subject; thus, an appropriate contribution to the solution of this type of problems is provided.

Finally, the segmentation of the large-scale database in segments that group the fingerprints by their characteristics was proposed. After the segmentation, a single segment is selected and the search for the fingerprint to be identified would be performed in it. This model provides for an efficient response time when searching for a specific fingerprint in a very large database.

BACKGROUND

Fingerprint Classification Using Orientation Field Flow Curves (OFFCs)

Dass and Jain (2004) review the different sets of approaches that have been developed for fingerprint classification and verify that the hybrid methods developed have not been tested on large databases. The authors also point out that the classes used for classification are important. Other researchers have used four classes that did not prove to be effective for classification; therefore, the authors propose five classes to obtain better results.

According to Dass and Jain (2004), the procedure they used almost managed to determine the classification of the fingerprints with a percentage of 94.4% accuracy, so they propose to include the detection of the areas where the smallest loops originate in future works, in addition to continue taking as a basis the classification proposed by Henry (1900), which consists of: arch, right loop, left loop, arch, tentarch and whorl.

The procedure used is based on the NIST4 (National Institute of Standards and Technology) database, and the approach used is a combination of the structural, syntactic and purely mathematical approach (Watson & Wilson, 1992).

An Effective Method for Classification and Fingerprinting Search

The key to the task of fingerprint image classification is features. The effectiveness of feature extraction depends on the quality of the images, the representation of the image data, the image processing models, and the evaluation of the feature extraction.

Real-time evaluation of image quality greatly improves the accuracy of the identification system. Good quality images require less pretreatment and enhancement. In contrast, poor quality images require more pretreatment and, of course, enhancement. Efficient fingerprint search requires high quality fingerprint images.

Most methods classify fingerprints into four or five classes with an accuracy level of 80% to 95%. The method proposed by the authors classifies them into six classes with which an accuracy level of 97% is obtained, which shows an improvement over the previous ones (Bhuvan & Bhattacharyya, 2009).

A Real-Time Matching System for Large Fingerprint Databases

A simpler technique is provided by Ratha, Karu, Chen, and Jain (1996): it consists of placing the images in the database as a plaintext, which is a sequence of characters with the information of each pixel. The main drawback of this method is that the scene description may be different at different times for the same image, depending on the context of the query. The authors propose reducing search space using feature extraction techniques.

A fingerprint database is characterized by a large number of records (in the order of millions). The size of the FBI database has grown from over 0.8 million fingerprint cards (10 fingerprints per card) in 1924 to over 114 million fingerprint cards in 1994. The storage requirements for such a large collection of images runs into 1.140 terabytes without compression. In addition, the query type of this system is expected to differ radically from other image database application domains.

The image of the same object can vary depending on its orientation, ambient light and the sensor itself. The sensed information is of a much higher dimensionality than textual information. In a digital library there are mainly three components: data capture, storage management, and search and query techniques.

According to the proposal of Ratha et al. (1996), image information is separated into its characteristics for storage in the database. Thus, all that is needed is to take a picture of a person's fingerprint and compare it with the person's own characteristics.

Poor quality images often reduce the accuracy of the system. Therefore, according to Douglas (1993), an image quality assessment is being considered at the input stage.

Linear Search Model

As seen in Figure 6, currently the fingerprint search in a large database demands too much response time when performing a linear search, even when the database is organized, so this search is time-cost related.

Linear Fingerprint Search

When a fingerprint image is entered (see Figure 6), in bmp format, it is represented in a matrix whose elements are zeros and ones. A connection is then made to the database to search for this array, which has the BLOB (Large Object Binary) type.

Then the following code explains how SQLite performs a linear search record by record, and returns a list. If the item has been found, this list has a size greater than zero. Otherwise, the size is zero. This enables the user to determine when a search has been successful (see Figure 7).

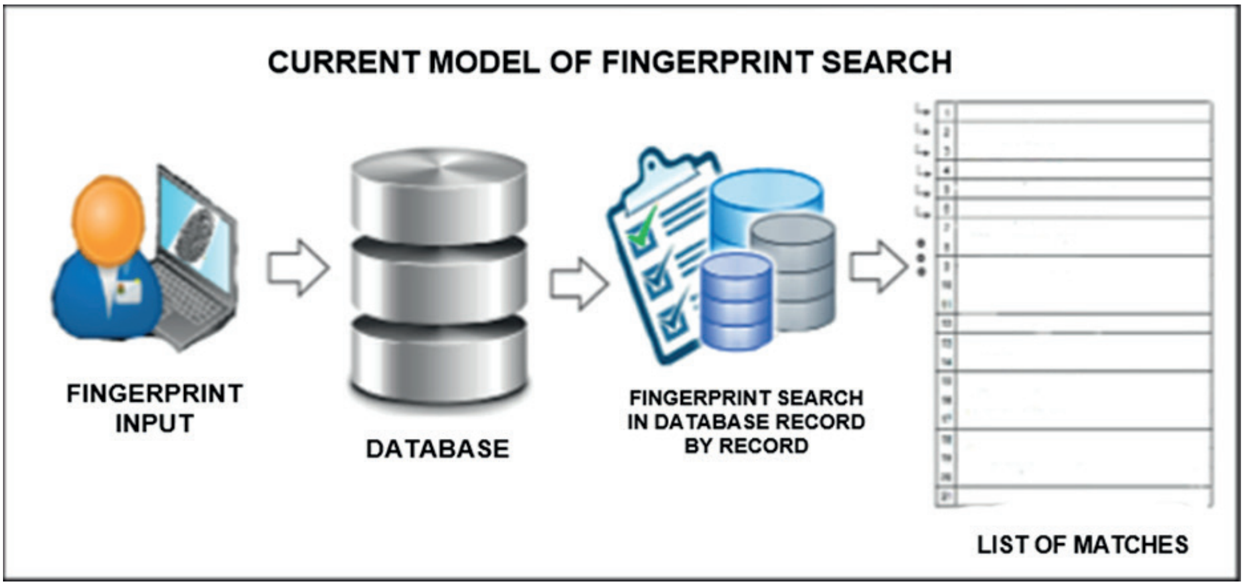
CURRENT MODEL OF FINGERPRINT SEARCH
FINGERPRINT INPUT
DATABASE
FINGERPRINT SEARCH IN DATABASE RECORD BY RECORD
LIST OF MATCHES

Figure 6. Current model of fingerprint search.

Source: Prepared by the authors.

Proposed Model

The proposed model segments the large database according to the characteristics of each of the fingerprints, so that the search is not performed in the entire database but in a certain segment, and thus an immediate result of the identification of the searched fingerprint is obtained, as shown in Figure 8.

First, the fingerprint of the person to be identified is entered. The computer runs the classification application for the fingerprint entered, which will return the corresponding segment number found in the database already trained for the respective search. This database contains all stored fingerprints. Searching the unsegmented database requires a very long response time which, in some cases, is essential for making a decision.

The database that stores the fingerprints is large, so when performing a linear search, it takes a considerable time to give a response, even when the database is indexed. These fingerprints have

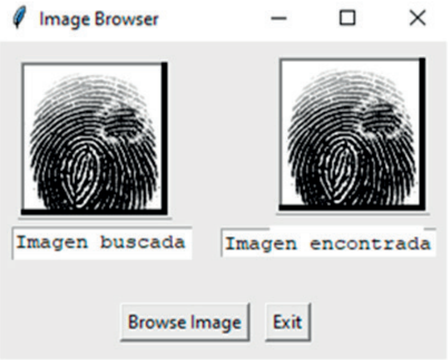

Imagen buscada

Imagen encontrada

Figure 7. Fingerprint entered and fingerprint found.

Source: Prepared by the authors.

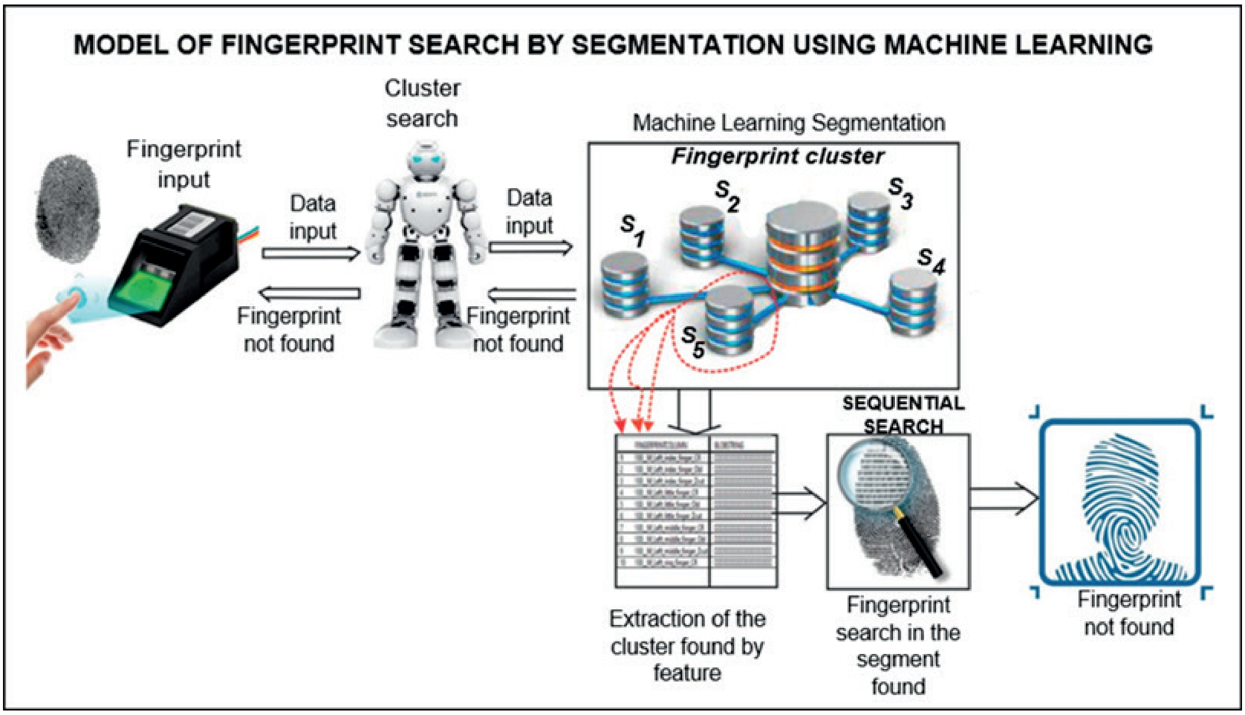
MODEL OF FINGERPRINT SEARCH BY SEGMENTATION USING MACHINE LEARNING
Cluster search
Fingerprint input
Data input
Data input
Fingerprint not found
Fingerprint not found
Machine Learning Segmentation
Fingerprint cluster
S2
S3
S1
S4
S5
SEQUENTIAL SEARCH
Extraction of the cluster found by feature
Fingerprint search in the segment found
Fingerprint not found

Figure 8. Model of fingerprint search in large databases.

Source: Prepared by the authors.

certain characteristics that allow them to be classified into segments, so the search can be limited to the selected segment containing the fingerprint to be searched. Machine learning is used for the segmentation, specifically the k-means algorithm, which allows grouping the fingerprints according to the characteristics that the algorithm identifies as the closest. Using this method, the database is segmented into five clusters (segments).

When the fingerprint to be searched is entered, it is classified according to the segment number that contains it. After being placed in the selected segment, a search is performed in that segment using a discrete algorithm. Note that the search is no longer performed in the entire database, but only in the selected segment. Therefore, the searched fingerprint can be identified in an optimal time by running the search in a single segment.

Functions and Models

- **Non-Segmented Linear Search**

  A linear search of an image in a non-segmented database is conducted in this script to obtain the time taken by the algorithm to find the searched pair and display it in the interface.

```
import time
import sqlite3
def busqueda_sec(bin):
                inicio = time.perf_counter()
                try:
                                sqliteConnection = sqlite3.
                                connect('nocluster.db') cursor =
                                sqliteConnection.cursor()
                                print("Connected to SQLite")
                 sql_fetch_blob_query =
               """SELECT * from
                        nocluster where imagen = ? LIMIT
                            1"""
                  cursor.execute(sql_fetch_blob_query,
                          (bin,))
                                record = cursor.fetchall()
                                cursor.close()
                except sqlite3.Error as error:
                                print("Failed to read blob data from
                                sqlite table", error)
                 # finally:
                 # if sqliteConnection:
                 # sqliteConnection.close()
                 # print("sqlite connection is closed")
                                   tiempo = time.perf_counter() -
                                   inicio
                                   dir_img = 'D:/Python/milhue-
                                   llas/k-images/'
                                   +record[0][0]+'.bmp'
                                   #print(tiempo)
return tiempo, dir_img
```

- **Prediction and Search**

  A previously trained model is loaded and applied to predict in which cluster the image to be searched is located, then the linear search algorithm is applied in that cluster, the time of the whole process is captured and displayed in the interface.

```
import time
from joblib import load
import sqlite3
cluster= load('2domodelo')
def busqueda_cluster(features,bin):
inicio = time.perf_counter()
clus = cluster.predict(features)[0] #clus = 0,1,2,3
try:
sqliteConnection = sqlite3.connect('cluster.db')
cursor = sqliteConnection.cursor()
print("Connected to SQLite")
sql_fetch_blob_query = """SELECT * from cluster{0} where
                                imagen = ? LIMIT 1""".format(clus+1)
cursor.execute(sql_fetch_blob_query, (bin,))
record = cursor.fetchall()
cursor.close()
except sqlite3.Error as error:
print("Failed to read blob data from sqlite table", error)

# finally:
# if sqliteConnection:
# sqliteConnection.close()
# print("sqlite connection is closed")
tiempo = time.perf_counter() - inicio
dir_img = 'D:/Python/milhuellas/k-images/' +record[0][0]+'.bmp'
#print(tiempo)
return tiempo, dir_img
```

- **Other Necessary Functions**

  Image_feature function needs InceptionV3, which is a convolutional neural network to assist in image analysis and object detection and started as a module for Googlenet.

```
def image_feature(imagen):
        model = InceptionV3(weights='imagenet', include_
        top=False)
        features = [];
        img=image.load_img(imagen, target_size=(224,224))
        x = img_to_array(img)
        x=np.expand_dims(x,axis=0)
        x=preprocess_input(x)
        feat=model.predict(x)
        feat=feat.flatten()
        features.append(feat)
        return features
        def salir():
        shutil.rmtree('Temporal')
        exit()
```

Time Tests

Finally, after the implementation of the algorithm in the program, the processing time of each search must be determined for the corresponding comparison. Fifty randomly chosen images are taken without repetition in order to analyze the evolution of the search time in relation to the position of the image in the non-clustered database. After the execution of the previously discussed algorithms, the results shown in Table 2 were obtained.

The first column describes the image position in the database, while the next two columns show the image search time in seconds using the segmented and linear algorithms, respectively.

In conclusion, we can confirm that the segmented search algorithm has advantages over the classical linear algorithm, mainly due to the following two reasons:

Table 2. *Search Time of the Proposed Algorithm.*

No.	Fingerprint No.	Segmented Time (sec.)	Linear Time (sec.)
1	531	0.004379	0.008464
2	528	0.004228	0.008589
3	517	0.004777	0.008337
4	513	0.004227	0.008667
5	506	0.003872	0.008317
6	490	0.003828	0.008067
7	486	0.005438	0.008955
8	483	0.005157	0.010635
9	480	0.004468	0.008513
10	472	0.006281	0.015658
11	444	0.002972	0.007607
12	442	0.003441	0.007469
13	437	0.003763	0.007042
14	428	0.004522	0.016173
15	408	0.004283	0.006484
16	397	0.004105	0.008492
17	392	0.004123	0.007803
18	390	0.004025	0.006753
19	377	0.003137	0.006483
20	356	0.010098	0.014196
21	337	0.003842	0.008346
22	316	0.008645	0.013955
23	312	0.003817	0.006923
24	309	0.003928	0.005397
25	272	0.003292	0.006387
26	268	0.002802	0.004809
27	264	0.009219	0.008393
28	261	0.003089	0.005612
29	259	0.003477	0.006325
30	247	0.007648	0.008603
31	237	0.007601	0.009455
32	236	0.009113	0.005527
33	233	0.003252	0.006325
34	222	0.003704	0.005572
35	211	0.008135	0.010234
36	179	0.003113	0.003579
37	173	0.002846	0.002438
38	163	0.003166	0.002587
39	156	0.002703	0.002495
40	153	0.002625	0.002485
41	144	0.002265	0.002865
42	138	0.002307	0.002889
43	121	0.002772	0.001929
44	91	0.005962	0.003999
45	80	0.002847	0.001563
46	78	0.002946	0.001216
47	66	0.002495	0.001253
48	64	0.002835	0.001167
49	42	0.002091	0.001048
50	9	0.003076	0.000588

Source: Prepared by the authors.

- Linear search in a cluster takes considerably less time on average than the search time in the complete database.
- Evaluation time of an image in the classification model is short enough not to exceed the search time of the linear algorithm.

Figure 9 shows that the intersection of the linear time and segmented time lines is at point 112.5; then, according to the regression diagrams performed, it can be deduced that for a database of 537 images, the linear search time starts to increase from the image at position 113 in the unsegmented database.

DISCUSSION

Dass and Jain's (2004) approach has four main stages: first, the extraction of the orientation field for the given fingerprint image; second, generation of orientation field flow curves (OFFCs); third, labeling of each OFFC into the four classes: left and right loops, whorl and arch; and finally, an overall classification of the fingerprint image into one of the four classes based purely on mathematics (the orientation value of the fingerprint image is a vector) and using methods from differential geometry. According to Dass and Jain (2004), the procedure they used determined the classification of the fingerprints with 94.4% of accuracy.

The purpose of this paper is to propose a highly efficient search method using machine learning, which obtains a segmentation of fingerprints from a large database and groups them according to the characteristics of each fingerprint into segments. Then, a linear search algorithm is applied to one of the selected segments, that is, the one containing the fingerprint to be identified. Using this model, search

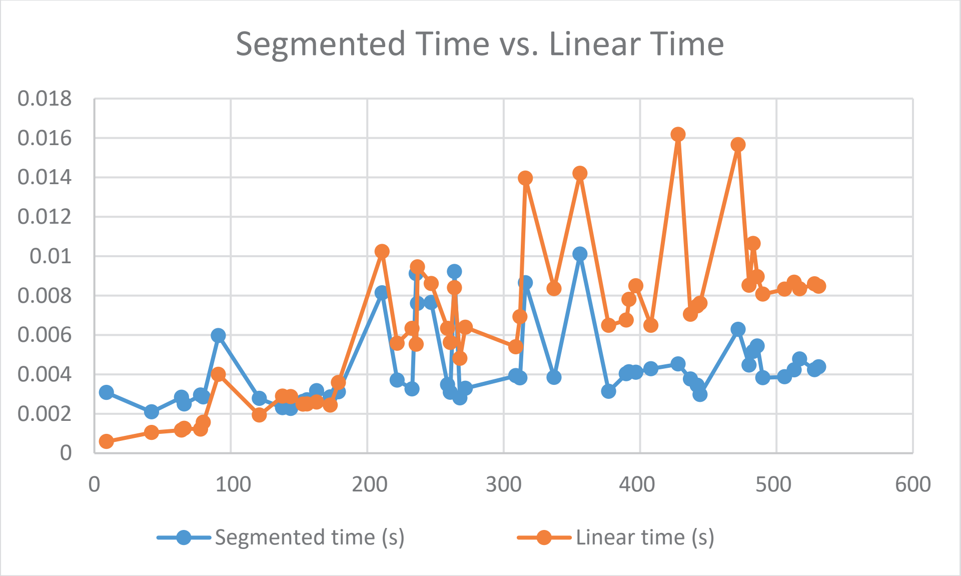

Figure 9. Time plot.

Source: Prepared by the authors.

times are minimized with a 95% confidence level. It is also demonstrated in this research that the segmented search is more efficient than performing a linear search in a large database, it is therefore more efficient than the method used by Dass and Jain (2004).

CONCLUSIONS

The main objective of this paper is to minimize search time when performing the identification of a person in a large database. After evaluating existing methods and algorithms, we conclude that machine learning allows segmenting a large database by grouping fingerprints according to their closest characteristics and then applying a linear search technique in only one selected segment to find the person's fingerprint efficiently.

This model does not search the entire database for the fingerprint, as this requires a very high response time. Instead, we propose to perform the search in only one of the segments classified by characteristics, where the searched fingerprint would be found, thus reducing search time.

RECOMMENDATIONS

For the study concerning the topic "Fingerprint search in a large database", a documentary and exhaustive methodology is suggested, using the model proposed in the topic of the research article.

Generic algorithms, which are adaptive methods that can be used to solve search and optimization problems with the sole purpose of optimizing the final results in time, are recommended when performing the segmentation.

Larger sample testing is also recommended to test the efficiency of the proposed model.

REFERENCES

[1] Bathia, S., & Deogun, J. (1998). Conceptual Clustering in Information Retrieval. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, 28*(3), 427-436.

[2] Bhuvan, M., & Bhattacharyya, D. (2009). An Effective Fingerprint Classification and Search Method. *IJCSNS International Journal of Computer Science and Network Security*, 39-48.

[3] Dass, S., & Jain, A. (2004). Fingerprint Classification Using Orientation Field Flow Curves In. *ICVGIP*. 650 - 655.

[4] Douglas H., D. (November, 1993). Enhancement and Feature Purification of Fingerprint Images. *Pattern and Recognition, 26*(11), 1661-1671.

[5] Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data Mining to Knowledge Discovery in Databases. *American Association for Artificial Intelligence*, 37-54.

[6] Fernández, F. A. (2008). *Biometric sample quality and its application to multimodad authentication systems*. (Doctoral thesis). Universidad Politécnica de Madrid, Madrid.

[7] Forgy, E. (1965). Cluster analysis of multivariate data: E_ciency vs. Interpretability of classifications. *Biometrics*, 21, 768.

[8] Henry, E. R. (1900). *Classification and Uses of Fingerprints*.

[9] Kearns, M., Mansour, Y., & Ng, A. (1997). An information-theoretic analysis of hard and soft assignment methods for clustering. (K. M. Publishers, Ed.) *Proceedings of the Thirteenth Conference on Uncertanly in Artificial Intelligence,* 282-293.

[10] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, 1*(14), 281-297.

[11] Makhoul, J., Roucos, S., & Gish, H. (1985). Vector Quantization in Speech Coding. *Proccedings of the IEEE, 73*(11), 1551 - 1588.

[12] Maltoni, D., Maio, D., Jain, A., & Prabhakar, S. (2003). Multimodal Biometric Systems. En *Handbook of fingerprint recognition* (págs. 233-255). New York, NY: Springer.

[13] Persto S.A. de C.V. (June 15, 2020). *Verificación de identidad biométrica*. Retrieved from http://www.persto.com/

[14] Pham, D., & Prince, J. (1999). An Adaptive Fuzzy C-Means Algorithm for Image Segmentation in the Presence of Intensity Inhomogeneities. *Pattern Recognition Letters, 20*(1), 57-68.

[15] Ratha, N. K., Karu, K., Chen, S., & Jain, A. (1996). A real-time matching system for large fingerprint databases. *IEEE transactions on pattern analysis and machine inteliigence, 18*(8), 793-813.

[16] Scheunders, P. (1997). A comparison of clustering algorithms applied to color image quantization. *Pattern Recognition Letters, 18*(11-13), 1379 - 1384.

[17] Solberg, A., Taxt, T., & Jain, A. (1996). A Markov Random Field Model for Classification of Multisource Satellite Imaginery. *IEEE Transactions on Geoscience and Remote Sensing, 34*(1), 100 - 113.

[18] VanderPlas, J. (2017). *Python Data Science Handbook. Essential Tools for Working wit Data*. Sebastopol, CA 95472, EE. UU.: O'Reilly Media, Inc.

[19] Villamizar, J. A. (1994). Procesamiento y clasificación de huellas dactilares. *Lecturas Matemáticas, 15*(2), 149 -165.

[20] Watson C.I., & Wilson, C. (March, 1992). *Nits 4, Special Database*. National Institute of Standars and Technology.

[21] Wayman, J., Jain, A. K., Maltoni, D., & Maio, D. (Eds.) (2005). *Biometric Systems: Technology, Design and Performance Evaluation*. Springer Science & Business Media.

[22] Zúñiga, J. (n.d.). *El algoritmo k-means aplicado a clasificación y procesamiento de imágenes*. Retrieved July 15, 2021, from https://www.unioviedo.es/compnum/laboratorios_py/new/kmeans.html