# A Contemporary Implementation of Two-Class Fingerprint Indexing

Nachamai M
Associate Professor
Dept. of Computer Science,
Christ University, Bangalore,
India

Joy Paulose
Professor
Dept. Of Computer Science,
Christ University, Bangalore,
India

Samuel Marandi
M. Sc. Student
Dept. Of Computer Science
Christ University, Bangalore,
India

## ABSTRACT
Biometric is a characteristic measure of a person. Through biometrics a person can be labeled. Fingerprint is a striking biometric which supersedes all other biometrics because of the property of universality and uniqueness. Most of the organizations which are time bound rely largely on fingerprint biometric. It is a known issue that fingerprints are images and manipulating it eventually takes time to search on database and retrieve or recognize it. The method proposed and implemented in this paper is a new and concise way of indexing a fingerprint database based on the edges that come from the fingerprint. It has been lucrative in classifying them into two classes. The algorithm success rate on the UPEK database and Sfinge dataset yielded a satisfactory true positive rate of 100% and 98% respectively.

## General Terms
Soft Computing, Pattern Recognition, Biometrics.

## Keywords
Naive Bayes classifier, biometrics, fingerprint indexing, edge operator.

## 1. INTRODUCTION
Biometrics refers to metrics of human characteristics or traits. There are numerous biometrics available for identification of a person starting from iris, gait, gesture, key stroke, etc, the fingerprint happens to be the best biometric identifier. Biometrics is generally of two categories physiological and behavioural. Examples of physiological include fingerprint, palm geometry, DNA, face recognition, iris recognition, hand geometry, etc., Behavioural characteristics are related to pattern of behaviour of a person like key stroke dynamics, gait and voice. [1] Jain et al., has listed seven characteristics to be weighed when selecting a suitable feature for biometric identification. They are 1. Universality – every person using the system or machine should possess it, 2. Uniqueness – the identifier should be surely unique between all people using the application, 3. Permanence – the trait should not change over time nor due to any physiological changes, 4. Measurability – the identifier must have ease of access for collecting, storing and retrieving, 5. Performance – it computes the systems speed and accuracy using this trait, 6. Acceptability – how well the system of identification is accepted by people, and 7. Circumvention – the ease with which a substitution can be made. When a trait fits into these 7 characteristics then it happens to be the one that can be applied to any system, depending on the type of application. The fingerprint biometric happens to be the best and satisfied biometric according to the seven characteristics defined, invariably for any application [2]. Fingerprint finds its place in vast areas of sophisticated applications other than only identifying a person, like, fingerprint controlled access of mobiles and other electronic devices, electronic door locks, vehicle ignition control systems, smart cards, etc, [3].

## 2. REVIEW OF LITERATURE
Fingerprints can generally be classified into five types as Right loop, Left loop, Whorl, Arch, and Tented Arch [4] as shown in the Figure 1.

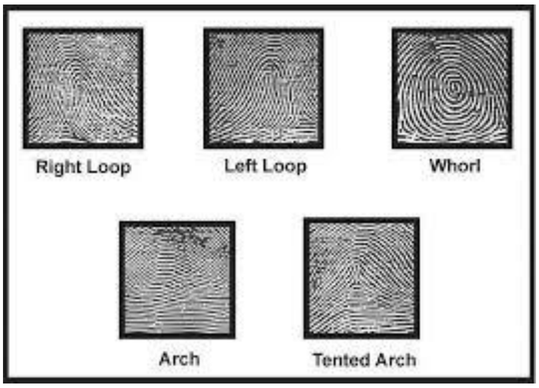


**Figure 1: Types of fingerprint**

A fingerprint recognizer generally works to do a matching on an image database. In a database of 10,000 images to identify an image, worst case the algorithm has to make 10,000 comparisons, which would eventually result in wastage of time and efforts. An indexing technique is mandatory to overcome this situation. A regular fingerprint indexing technique or a recognizer would generally try to match on a pixel by pixel method or through certain identified areas of interest. When an input fingerprint is given it must be able to identify the category and start the search on the relevant database, if database storage is made categorically. In paper [5] online verification of fingerprint systems is clearly discussed, where the standard methods adopted and applied is implemented. A basic implementation of fingerprint classification is elaborately discussed in [6]. The biometric report [7] showed the statistics of usage of fingerprint biometric used globally, which makes it an important area of research till date. All the basic methods of feature extraction and indexing are lucidly given in [8]. The most common form of indexing is using the ridge as the region of interest [9]. Indexing on other region of interest leaving out ridge, like core, delta, minutiae points are implemented in [10]. SIFT features are extensively tried by Xin [11]. Quadruplets of minutiae have been attempted and satisfactory results are given in [12]. All the methods in the literature have an exhaustive computing for indexing. The methodology implemented extorts the edge lines of the fingerprint as a feature and based on the storage size of the image it gets categorized into "Tented arch class" and "Non-Tented arch class".

## 3. METHODS AND MATERIALS

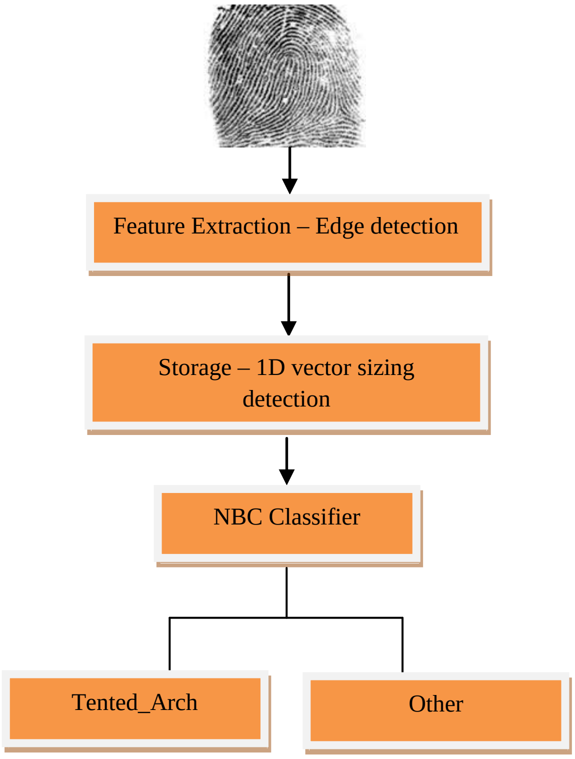


**Figure 2: Flow of the system**

The classifier used for the two class classification is Naive Bayes classifier. The classifier is a simple probabilistic classifier based on strong independence assumptions between features. This feature of Naïve Bayes classification allows categorization of a fingerprint by providing edge line features into the classifier and categorizing it into one of the classes. The step wise flow of the methodology is shown in the Figure 2. The classification into a category was implemented successfully based on the storage size of the vector after implementing edge detection on the input image. The first step towards implementing this procedure is acquiring a fingerprint image which can provide distinct edges when an edge detection operator is applied. The operator wheedles out the edges apparently. Once the image has been acquired the second step is to resize the image to a feasibly calculable size preferably to the dimensions of a square matrix and apply an edge detection operator on the image. The preferred edge detection operator for this operation used is the 'canny' operator. Once edge detection operation is performed the third step is to reshape the resultant 2D matrix to 1D vector where the number of columns is equivalent to the number of elements present in the 2D matrix. The fourth step removes the repetition of zero's and retains only one occurrence of the zero in the vector. The occurrence of one's in the vector however remains unchanged. The application of the fourth step in the resultant vector will considerably reduce the size of the vector. This is significant because, for fingerprints which are under the classification of Tented Arch, the vector size will be less than 5500 for a resized input matrix of $128 \times 128$. This means that applying the above steps to a fingerprint which is classified as Tented Arch can be separately categorized from the other four classifiers based on the storage size. This result can be used to create a training group which is based on the storage size of the resultant vector and can be used to make a classification using a Naïve Bayes Classifier.

The output generated from the implementation of the above steps is that the fingerprint which is given as input will either be categorized as Tented Arch or as Non Tented Arch. All the other four classes i.e. Arch, Left Loop, Right Loop and Whorl are grouped together as 'Other' so that when the classification is made it is evident that fingerprint is either a Tented Arch or not a Tented Arch. This attempt is a first step in identifying a simple and robust method to reduce the complex computations and avoid storage constraints faced by most of the fingerprint indexing techniques. Identifying edge lines is an uncomplicated way of feature extraction from the fingerprint image. Generally all feature extraction methods are too complex and computationally expensive. When edge lines are detected it is apparent the dimension of the feature extracted is known, which becomes an obvious hint for the first level of indexing. The dimension indicator does not need the extracted feature to be stored eventually sinking the storage requirements for such a system.

**Algorithm: Classfier_Tented_Arch()**

Input:
(i) A training group TRG to provide initial data to the classifier to make the classification.
(ii) A grouping variable Group for training. The grouping variable groups the data into two groups, Tented_Arch and Other.

(iii) A sample image which has to be classified.

Output:
The sample image is either classified as Tented_Arch or classified as other.

1: Initialize j to 1
2: Resize the image into size 128x128.
3: Perform edge detection of the resized image using the canny operator and store it in a matrix.
4: Resize the matrix to 1D vector 'imEdge' of size C (Here, C=128*128)
5: For every element i=1 to c in Arr do

/***This block will remove all the repetitive occurrence of zero with a single occurrence***/

1. if imEdge(i) is equal to 0 and i is less then C do
    1. Initialize Arr(j) to 0
    2. Increment i by 1
2. End if

/***This block will retain all the occurrence of one's both single and repetitive***/

3. if imEdge(i) is equal to 1 and i is less than C do
    1. Increment j by 1
    2. Initialize Arr(j) to 1
    3. Increment i by 1
4. End if

6: End for

7: Get the size of Arr and store it in S
/*** Naive Bayes Classification based on the size of the vector derived after removing the repetitive occurrence of 0's and retaining the 1's in the vector ***/

8: Initialize ResClass to the output of classify(S,TRG,Group)

## 4. RESULTS AND CONCLUSIONS

The implementation was done in MatLab. The work was tested in two datasets, one the UPEK database that had 128 fingerprint images in the format of portable network graphics and second the Sfinge is synthetic fingerprint generator where 500 images were generated and tested. The first dataset had only 8 samples of tented arch, so to test the robustness and vulnerability of the algorithm the attempt was made on the SFinge dataset, which manually created 234 instances of tented arch. The algorithm was extremely successful in classifying the two classes for both the datasets. The confusion matrix for both the datasets are shown in the Table 1 and 2 respectively.

**Table 1: Confusion Matrix (UPEK database)**

| | Tented Arch | Other |
|---|---|---|
| Tented Arch | 8 | 0 |
| Other | 5 | 115 |

As shown in the confusion matrix the algorithm mapped 8 tented arch into the category of tented arch giving 100% accuracy on the true positive rate. The false positive rate of classifying other into tented arch is 5 out of the entire dataset, which amounts to 3.9%. The images that were classified as tented arch from the other classes are shown in the Figure 3.

**Table 2: Confusion Matrix (SFinge dataset)**

| | Tented Arch | Other |
|---|---|---|
| Tented Arch | 234 | 0 |
| Other | 8 | 258 |

Out of the 234 tented arch images it correctly classified 234 images as tented arch giving the true positive rate as 100% the false positive rate shows approximately 2% which can be neglected. The method is a break through approach in two fold it helps as a good indexing technique and there is zero overhead involved for classification due to the simplicity of

**Figure 3: False positive images**

edge features. An attempt on making a five class classification for indexing would inadvertently open a new focus area to research further on the same techniques applied.

## 5. REFERENCES

[1] Jain, L.C. et al. (Eds.). 1999. Intelligent Biometric Techniques in Fingerprint and Face Recognition. Boca Raton, FL: CRC Press.

[2] L.C.Jain, U.Halici, I.Hayashi, Lee S. B. & S.Tsutsui. 1999, *Intelligent Biometric Techniques in Fingerprint and Face Recognition*, CRC Press LLC, United States of America.

[3] Russ, John C. 1995, *The Image processing Handbook, Second* edition. CRC Press Inc, United States of America.

[4] Edward RHenry. (1974): *Classification and uses offingerprints,* AMS Press, New York.

[5] A.Jain, L.Hong and R.Bolle, 'On-line fingerprint verification, IEEE Trans, Pattern Anal. And Machine Intell., 19(4):302-314,1997.

[6] K.Karu and A.K.Jain 'Fingerprint classification, Pattern Recognition, 29(3):389-404,1996.

[7] E.Newham, The biometric report, SJB services, Newyork, 1995.

[8] D.Maltoni, D.Maio, A.K.Jain, S.Prabhakar, 'Handbook of finger print recognition", Springer Verlang, 2003.

[9] J.Feng, A.Cai, 'Fingerprint Indexing using ridge invariants', proceedings of ICPR vol.4, pp. 433-436, 2006.

[10] J.D. Boer, A.M. Bazen and S.H. Gerez. Indexing fingerprint database based on multiple features. Proc. ProRISC, 12th Annual Workshop on Circuits, Systems and Signal Processing, 2001.

[11] Xin Shuai, Fingerprint indexing based on composite set of reduced SIFT features, Pattern Recognition, pp.1-4,19th International Conference ICPR 2008.

[12] Illoanusi O N, Gyaourova A, Ross A, Indexing fingerprints using Minutiae quadruplets, Computer Vision and Pattern Recognition IEEE workshop, june 2014.