

Novel Adaptive Auto-Correction Technique for Enhanced Fingerprint Recognition

Thejaswini P¹

Department of ECE
JSS Academy of technical education
Uttarahalli-Kengeri Road, Bangalore
560060, India

Srikantaswamy R S²

Department of ECE
Siddaganga Institute of Technology
BH Road, Tumkur 572103
India

Manjunatha A S³

Department of CSE
Siddaganga Institute of Technology
BH Road, Tumkur 572103
India

Abstract—Fingerprints are the most used biometric trait in applications where high level of security is required. Fingerprint image may vary due to various environmental conditions like temperature, humidity, weather etc. Hence, it is necessary to design a fingerprint recognition system that is robust against temperature variations. Existing techniques such as automated and non-automated techniques are not real time analysis (adaptive). In this paper, we propose an adaptive auto correction technique called Reference Auto-correction Algorithm. This proposed algorithm corrects user reference fingerprint template automatically based on captured fingerprint template and the matching score obtained on daily basis to improve the recognition rate. Analysis is carried out on 250 fingerprint templates stored in the database of 10-users captured at varying temperature from 25°C to 0°C. The experimental result shows 40% improvement in the recognition rate after applying auto correction algorithm.

Keywords—Minutiae, Euclidean distance; artificial neural network; CN- crossing number; reference auto-correction; adaptive method; ISO template; auto-correction algorithm

I. INTRODUCTION

Advancement in the technology has led biometrics to replace the conventional access control and time attendance systems [1]. Biometric systems [2] are used for recognition of user using various biometric traits like Fingerprints, IRIS, Face, Voice, DNA etc. These biometric traits are very unique features of human beings and usually remain stable throughout the life span under normal conditions. Error rate implications, accuracy and response time of fingerprint are better when compared to other biometric traits. Thus the fingerprint develops the obligatory variable in striking security and reliable empathy of the individual. Fingerprints are utilized as means of security in places such as voting, banking operations, day to day attendance system etc. The quality of fingerprint image varies on many characteristics such as humidity, weather, temperature etc., which affects the performance of the biometrics system. Even though fingerprints are considered as very stable and unique throughout the life, it has been observed that the fingerprint pattern varies due to environmental variations like temperature, humidity, dust and also due to ageing of the person [3].

Fingerprint based biometric system works on the principle of comparing live fingerprint image with the reference

fingerprint image stored in the database to find matching [4]. The variation of fingerprint image will affect the quality of captured input fingerprint image leading to poor matching with the stored reference fingerprint image in the database and hence it fails to recognize the same person's identity [5] [6]. This causes lots of inconvenience to use the fingerprint based biometric system for recognition of users especially in day to day time attendance system [7]. In spite of optimal noise reduction using filters, the fingerprint image captured may fail to match with the reference fingerprint image stored in the database due to environmental variation (like temperature), which varies the features in the captured fingerprint image. The issue of rejection of fingerprint image of some user due to variation of environmental parameter like temperature must be addressed.

An attempt is made to improve the recognition rate of fingerprint image even though the matching of fingerprint image fails after applying optimal filters under the environmental variations (like temperature variations). Hence to deal with the failure to recognize the fingerprint image of the user under the environmental variation (like temperature variation), a study and analysis of the fingerprint images are carried out. A novel auto-correction algorithm for autocorrecting the reference fingerprint template (template is the extracted minutiae features of the fingerprint image) without manually re-registering the fingerprint image of the user is proposed so that, the fingerprint recognition system is robust against the changes due to environmental parameters like temperature. Organization of sections in the paper is as follows: Section II demonstrates the proposed reference auto-correction method for fingerprint images, Section III discuss the results and experimental analysis of different datasets for users and Section IV concludes the research outcome.

II. PROPOSED FINGERPRINT RECOGNITION AND REFERENCE AUTO-CORRECTION TECHNIQUE

Due to environmental changes in the real world, fingerprint pattern may be altered. Even little variations in the fingerprint pattern leads to varied matching score in case of the authentication system, thus resulting in rejection of genuine user. The proposed reference auto-correction algorithm mainly focuses on fingerprint based time-attendance system, where the user has to place the finger daily on the fingerprint sensor to provide fingerprint image for verification. The principle behind auto-correction algorithm is, to correct

user reference fingerprint template automatically based on the everyday captured fingerprint template and the matching score obtained. Hence, recognition rate of authentication system can be improved.

The proposed Fingerprint Recognition and Reference Auto-Correction technique comprises of four stages:

- Minutiae Feature Extraction
- Euclidean distance Calculation
- Adaptive ANN
- Reference Auto-Correction

First, extract the minutiae features from the captured fingerprint image and store in the database as Reference Fingerprint Template (RFPT). For verification or authentication of user, the extracted fingerprint template from captured fingerprint image is compared with the reference fingerprint template stored in database. The Euclidean distance calculation is used for obtaining matching score based on the similarities [8] between two fingerprint templates. If the matching score is greater than the value of threshold (Th), the user is recognized (success or accepted). In such case, the captured fingerprint templates are stored in the cache database. The cache database is implemented as FIFO with the size of 30 templates (i.e. to store previous 30 days fingerprint templates). After 30 templates, the first stored template is deleted from cache. At any given point of time, previous 30 days templates are available in the cache for any user.

In the process of verification, if the matching score is less than the value of Threshold (Th), then the user is not recognized. In such cases, the user fingerprint template might have been changed beyond the threshold limit and hence failed to recognize. This situation may occur over a period of time due to environmental parameter like temperature variation, which leads to the variation of features in the captured fingerprint image due to introduction of noise.

The proposed flowchart of Fingerprint Recognition and Reference Auto-Correction system is shown in Fig. 1.

A. Minutiae Feature Extraction

Minutiae are the most widely used feature for representing the fingerprint due to its virtues compared with other features [9]. The minutiae features are extracted through alignment and pairing. Here, the essential minutiae features are extracted through the Crossing number (CN) identification approach. In CN minutiae feature extraction method, the orientation of minutiae point together with its location is obtained by using eight connected pixels for all the fingerprints.

For a pixel 'b', the crossing number can be obtained as:

$$Num_{Cr} = \frac{1}{2} \sum_{n=1}^8 |b(n) - b(n+1)| \tag{1}$$

Where, $b(9) = b(1)$

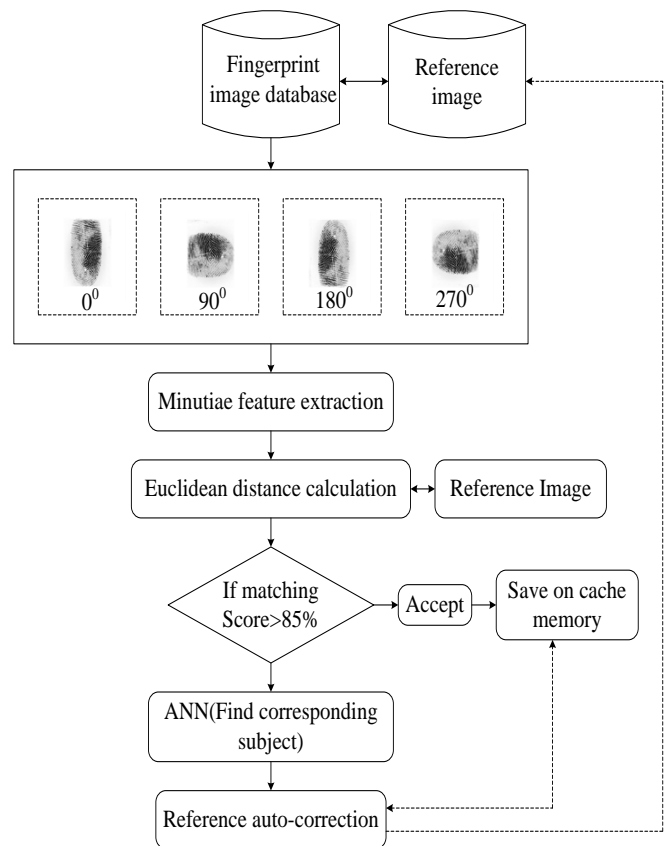


Fig. 1. Fingerprint Recognition and Reference Auto-Correction System.

In the above equation, $b(n)$ is the neighbor pixel of 'b' with $(b(n) = 0/1)$. By scanning the entire fingerprint image, we can get the minutiae points based on the characteristic value of CN. On applying this algorithm, border area may be ignored. Since the border area of the image will give more false minutiae points [10], there is no need to extract the minutiae points. Once the essential fingerprint minutiae points are extracted, the similarity between any two fingerprints could be found using Euclidean distance calculation.

B. Euclidean Distance Calculation

Euclidean Distance (ED) is a metric used for calculating the similarity between two fingerprint images [11]. In this step, matching score between the RFPT and user input fingerprint image is calculated based on Euclidean Distance calculation. The ED is a well-known distance measure, also referred as Pythagorean distance, which is normally defined as the square root of the sum of the squares of the distance between the corresponding coordinates of two images; or simply as the straight-line distance between two images in the Euclidean space.

The Euclidean distance between the fingerprint images can be computed by means of the following representation:

$$ED_{pq} = \sqrt{(m_p - m_q)^2 + (n_p - n_q)^2} \tag{2}$$

Where,

$(m_p, n_p), (m_q, n_q)$ are the pixel coordinates of two fingerprint images

In the process of verification, if the matching score is less than the value of Threshold (Th), it is considered as user is not recognized. In such cases, the user fingerprint template might have been changed beyond the threshold limit and hence failed to recognize. This situation may occur over a period of time due to environmental parameter like temperature variation, which leads to the variation of features in the captured fingerprint image due to introduction of noise.

C. Adaptive ANN

The genuineness of the rejected input fingerprint image during Euclidean distance calculation is checked for the cause of rejection through the Adaptive ANN technique [12] before applying correction on the stored fingerprint templates in the cache database. This is because; rejection can happen when some intruder places the finger instead of genuine user on the fingerprint sensor. In such case, when the rejection happens due to intruder, the correction should not be applied on the stored reference fingerprint template

D. Artificial Neural Network

ANN's are information-processing and computing method stimulated by biological neuron processing [13]. Fingerprint templates in the cache database of the user and the extracted template of input rejected fingerprint images are trained with the ANN by their specific minutiae features. The training is performed in the neural network system on known input and output data by methods of weight alteration until the system can reasonably address the input and output space. ANN contains three layers such as an input layer, hidden layer and output layer. The input layer contains the current information of the network (i.e., the minutiae features of the fingerprints) and the output layer gives the response of the given input (i.e. the class of the fingerprint). In case the obtained output does not facilitate the goal (0=Genuine and 1=Intruder) inside a predefined acceptable error, the weights and inclinations inside the system are adjusted by constraining an error function (generally error function is least of MSE) relating the simulated and the target output. The training technique will be done when the mismatch between simulated and target output is nearly small. ANN classifies the rejected input fingerprint image as genuine user or not at the time of matching.

E. Back- Propagation Algorithm

The backpropagation algorithm starts with the comparison of output pattern with the target vector [14]. The error values are calculated from the hidden units. The changes of incoming weight can begin with the output layer and pass through the hidden layer. The advantage of this algorithm is, it corrects the network weights and decreases the training error of the fingerprint recognition system.

Step 1: Set the initial value of every interconnection weight between the input to hidden and hidden to output layers as a small random number.

Step 2: Import the learning sample pair (i.e. the input minutiae points of the fingerprint samples and its corresponding target with the genuine or fake label) and operate steps 3 to 5 with each sample pair.

Step 3: For every fingerprint minutiae point sets, calculate the output (i.e. genuine or fake fingerprint) of every network layer based on below equation.

$$Y_{(m)}^{NN(out)} = \beta_m + \sum_{x=1}^X W_{xm} Y_{(x)}^{NN} \quad (3)$$

Where

$$Y_{(x)}^{NN} = \frac{1}{1 + \exp \left(- \sum_{q=1}^Q W_{qx} Y_{(q)} + \omega_x \right)} \quad (4)$$

In the above equation (3), $Y_{(m)}^{NN(out)}$ is the output of the network from m^{th} output node; β_m represents the bias of output node 'm'; W_{xm} is the interconnection weight between hidden and output nodes, and $Y_{(x)}^{NN}$ is the response from hidden layer.

Also, in eq. (4), W_{qx} is the interconnection weight between input and hidden nodes; ω_q represents the bias of hidden node 'q'; $Y_{(q)}$ represents the input minutiae points.

Step 4: Calculate training error ($\mathcal{E}_{(m)}^{BP}$) using below equation

$$\mathcal{E}_{(m)}^{BP} = Y_{(m)}^{NN(target)} - Y_{(m)}^{NN(out)} \quad (5)$$

Where $Y_{(m)}^{NN(target)}$ is the target label of the fingerprints.

Step 5: Correct weights for the next iteration ($W(t+1)$) based on the back propagation error ($\mathcal{E}_{(1)}^{BP}$) and the weights of the current iteration ($W(t)$).

$$W(t+1) = W(t) + \lambda Y_{(1)} \mathcal{E}_{(1)}^{BP} \quad (6)$$

In the above equation, λ is the learning rate, the value of the learning rate will be between 0.2 to 0.5.

Step 6: Check whether the outputs meet the accuracy requirement in the process of training every fingerprint templates stored in the cache and terminates the training process.

In the proposed Adaptive ANN, the weights are optimally selected by means of the Fruit fly optimization algorithm (FOA).

F. Fruit Fly Optimization Algorithm (FOA)

Fruit fly optimization algorithm [15] is stimulated by the characteristic fruit flies found in the environment. It is one of the optimization techniques for finding the solution for the input random population. Here, the input population is the random weights of the ANN. The smell foraging phase enables an individual to search and locate food sources around the fruit fly swarm [15]. The fruit fly is better than different species in vision and osphresis. Smell concentration is the fitness value for each food source. Here, the fitness value is the minimum error of the ANN. For each of the food sources, the smell fixation that relates to the fitness esteem is assessed next. Weight values with maximum smell concentration value are allocated in the vision foraging phase.

Accordingly, the Fruit fly goes across that food direction. The search behavior of Fruitfly for optimal weight selection is shown in Fig. 2 and the steps are explained as follows:

1. Initiate the location of fruit fly swarm randomly from X_axis and Y_axis . This is an initial population to update parameter (i.e. weights of NN).
2. To form osphresis search, random number of food sources (i.e. weights) are created around the fruit fly swarm represented by the equations,

$$X_j = X_axis + Randomvalue(M_p) \quad (7)$$

$$Y_j = Y_axis + Randomvalue(M_p) \quad (8)$$

Where, (M_p) is the fly range.

3. Calculate the distance and smell concentration judgment value (i.e. fitness value) for finding optimal weights. Here, the fitness value is the minimum error of the Neural Network.
4. Among the fruitfly swarm, find out the fruit fly with maximum smell concentration (i.e. optimal weights when an error in the NN output is minimum).

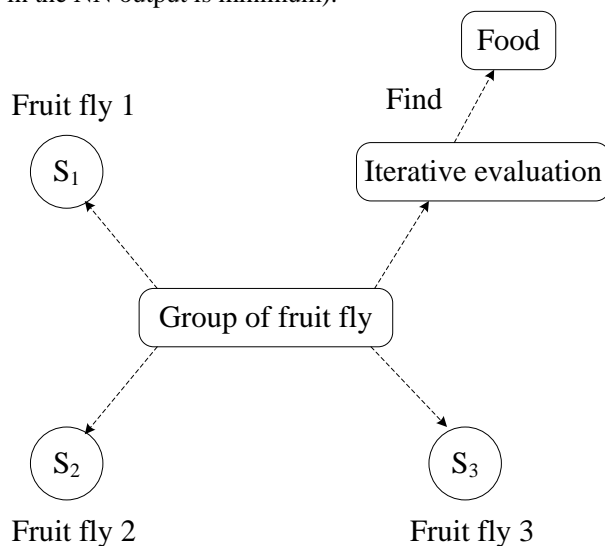


Fig. 2. Search behavior of Fruitfly for Optimal Weight Selection.

5. The swarm direct towards the location which has the best smell (optimal weights) using vision-based search.
6. Stop the algorithm when the iteration number reaches the maximum iteration number otherwise repeat the step 2

The Pseudo code of FOA is as follows (Fig. 3):

```

Input: Randomly initialized interconnection weights
Output: Best smell concentration (i.e. optimal interconnection weights)
// Initialization
Initiate iteration number represented as  $I_x$  and population size  $P_x$  equivalent to size of weight parameter
// Initiate the swarm location (SL) and fly range ( $M_p$ )
Iter = 0;
 $X\_axis = R(SL)$ ,  $Y\_axis = R(SL)$ ; where,  $R$ =random value;
Repeat
While  $i=1,2,\dots,I_x$ ;
//Osphrsis foraging phase
//Generate new weights from random flight direction and distance around current fruit fly swarm location
 $X_j = X\_axis + R(M_p)$ ;
 $Y_j = Y\_axis + R(M_p)$ ; where,  $M_p$ = fly range;
//Calculate Distance
 $Dist_j = \sqrt{(X_j^2 + Y_j^2)}$ ;
// calculate the judgment value for smell concentration
 $S_j = 1 / Dist_j$ ;
// calculate smell concentration
 $Smell_j = function(S_j)$ ;
//Identify maximum smell concentration of fruit fly among the whole swarm
 $[bestsmell \ bestindex] = \max(Smell_j)$ ;
//Vision based search
If ( $smellbest < bestsmell$ )
Return  $smellbest = bestsmell$ ;
 $X\_axis = Y\_axis = bestindex$ ;
Iter = Iter + 1;
Until Iter =  $I_x$ 
    
```

Once optimal weights are found, the same weights are used during the training and testing phases of ANN

G. Reference Auto-Correction

Adaptive Reference Auto-correction method is proposed to automatically correct the reference fingerprint template when the rejection of genuine fingerprint image occurs. Auto-correction of reference fingerprint images are carried out when the genuine fingerprint gets rejected due to environmental variations. The proposed reference auto-correction algorithm is as shown in Fig. 4.

In this technique first, consider the cache of user/subject where the corresponding fingerprint templates are stored on daily basis. By applying Euclidean distance measurement technique on the reference fingerprint template and the stored fingerprint templates in the cache obtain matched and

unmatched minutiae points of the corresponding fingerprint templates. Now, separate the matched and unmatched minutiae points of all the fingerprint templates stored in cache database and average all the separated unmatched points to obtain the new minutiae points (unmatched minutiae points are the new features obtained due to variation of temperature). Add these processed minutiae points into the last fingerprint template stored in cache. Resultant fingerprint template is a new template which is the combination of minimum good points of the reference fingerprint template required for matching plus the processed varied minutiae points. Now replace the reference fingerprint template by this new template. So, for next fingerprint image matching, this new template acts as reference template for the user; hence, reducing the percentage of false rejection of genuine subject/user. If the user is not found to be genuine, the rejected input fingerprint image is ignored and not stored in the cache database, nor used for correction, declared as intruder and ignored.

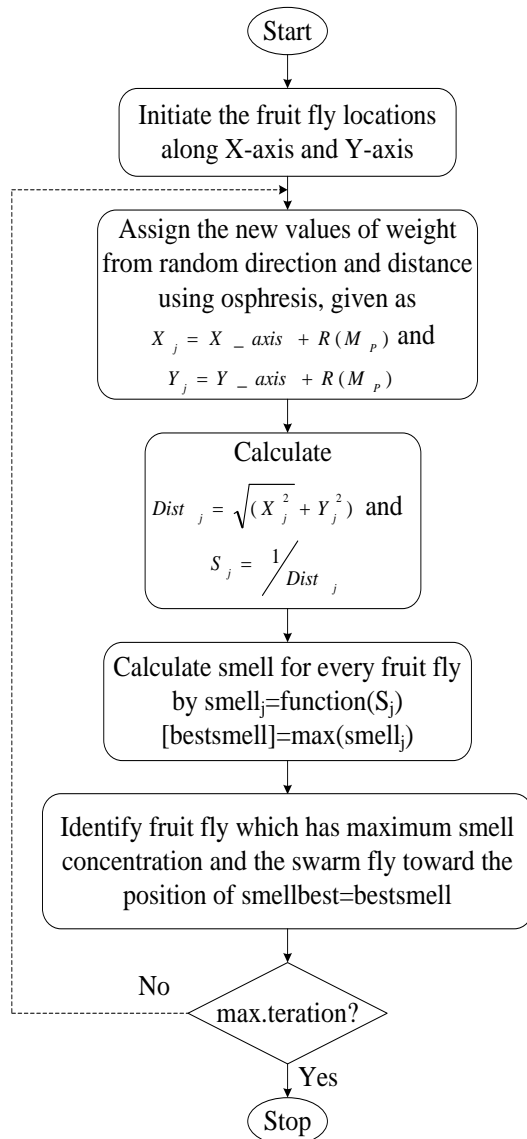


Fig. 3. Flow Diagram of Fruit fly Algorithm.

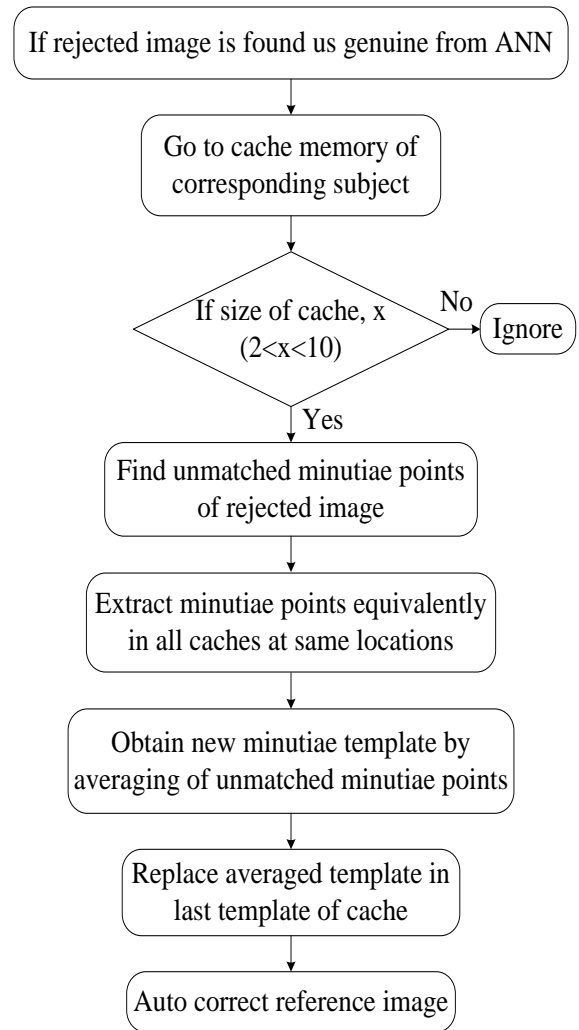


Fig. 4. Reference Auto-Correction Algorithm.

III. RESULT AND DISCUSSION

Adaptive reference auto-correction algorithm is used to correct the reference fingerprint template to match with the captured input fingerprint image when the degradation of fingerprint image occurs due to environmental variations (mainly temperature variation). The algorithm is implemented on the real-time fingerprint templates to test the improvement in the recognition of fingerprint templates. The results obtained after applying auto-correction algorithm are compared with the results obtained from without auto-correction. The values are tabulated and graphs are plotted to evaluate the performance.

The experiment is conducted with 10-user/subject fingerprint data which are captured at varying temperature from 25°C to 0°C.

First, all 10 users/subjects fingerprint images are captured at room temperature to extract minutiae features and stored as reference fingerprint template for the respective users/subjects in the database. Next, fingerprint images are captured for every user/subject by varying temperature from 25°C to 0°C using a closed chamber of controlled temperature. There are

about 26-fingerprint images captured for every user/subject with the variation of 1°C from one fingerprint image to other fingerprint image. The live finger of the user/subject is exposed to a minimum of 3-minutes every time when the temperature is set to a new value.

A. Analysis

Analysis is performed on the fingerprint data set stored in the database. Following parameters used in the process,

- 1) Type of sensor used to capture fingerprint image: Optical, with 500 DPI, 256 gray scales
- 2) Fingerprint template size- 1x996 bytes
- 3) Fingerprint image size- 256x256 bytes
- 4) Template type: ISO-1974-2 format
- 5) Threshold value set- 85% of 996 minutiae points (i.e., 847 out of 996 minutiae points are required minimum for matching between reference fingerprint template and captured input fingerprint image to be considered as recognized).

The analysis is carried out by comparing the captured fingerprint templates with reference fingerprint template for the respective user/subject using Euclidean distance matching algorithm. The matching is performed without and with using

auto-correction algorithm for all the 10-users/subjects fingerprint templates stored in database for varying temperatures from 25°C to 0°C and the matching scores are recorded and tabulated as shown in the Table I and Table II. The graphs are plotted showing the variation of matching score against the temperature for both the cases, with and without auto-correction based on the values recorded in the tabular columns are as shown in Fig. 5 and Fig. 6. Analysis of performance of all the users is tabulated in Table III. From Table III it can be observed that, without auto-correction only 2-users fingerprint images are matched the references images of their own. After applying auto-correction the result shows that, 6-users fingerprint images are matched the references images of their own. Hence there is an improvement in the success rate.

In practical scenario, the temperature variations are based on the seasons and also on the regions. The temperature is not varying always from 25°C to 0°C. Based on the regions, temperature variations may be few degrees like 20°C, 18°C, 15°C, 12°C, 10°C, 8°C, 6°C etc. Considering the regional variations in temperature, the results are analyzed for various temperature ranges and tabulated in Table IV. Fig. 7 shows the graph plotted for the values in Table IV.

TABLE. I. MATCHING SCORES OBTAINED BEFORE APPLYING AUTO-CORRECTION ALGORITHM

Temperature in °c	Matching Scores									
	User1	User2	User3	User4	User5	User6	User7	User8	User9	User10
25°	1	1	1	1	1	1	1	1	1	1
24°	0.965863	0.950803	0.944779	0.925703	0.954819	0.962851	0.899598	0.952811	0.943775	0.926707
23°	0.963855	0.937751	0.949799	0.930723	0.954819	0.963855	0.900602	0.970884	0.938755	0.924699
22°	0.943775	0.939759	0.940763	0.921687	0.940763	0.956827	0.88755	0.9749	0.934739	0.919679
21°	0.964859	0.939759	0.943775	0.925703	0.938755	0.961847	0.888554	0.965863	0.941767	0.921687
20°	0.950803	0.934739	0.944779	0.916667	0.933735	0.962851	0.863454	0.962851	0.938755	0.928715
19°	0.935743	0.921687	0.944779	0.923695	0.935743	0.958835	0.863454	0.959839	0.934739	0.908635
18°	0.938755	0.915663	0.940763	0.903614	0.929719	0.956827	0.829317	0.965863	0.929719	0.906627
17°	0.942771	0.913655	0.935743	0.893574	0.927711	0.953815	0.839357	0.943775	0.928715	0.907631
16°	0.942771	0.893574	0.932731	0.873493	0.921687	0.954819	0.818273	0.954819	0.939759	0.90261
15°	0.923695	0.878514	0.934739	0.878514	0.923695	0.951807	0.821285	0.948795	0.938755	0.911647
14°	0.927711	0.866465	0.929719	0.868473	0.925703	0.949799	0.829317	0.953815	0.925703	0.891566
13°	0.920683	0.858433	0.927711	0.858433	0.914659	0.945783	0.812249	0.941767	0.923695	0.891566
12°	0.919679	0.851405	0.917671	0.863453	0.907631	0.940763	0.811245	0.934739	0.914659	0.890562
11°	0.904618	0.843373	0.914659	0.853413	0.909639	0.947791	0.793173	0.934739	0.925703	0.859438
10°	0.918675	0.841365	0.913655	0.843373	0.899598	0.947791	0.788153	0.932731	0.911647	0.865462
9°	0.924699	0.823293	0.915663	0.838353	0.901606	0.940763	0.774096	0.932731	0.88755	0.854418
8°	0.905622	0.80522	0.908635	0.848393	0.90261	0.932731	0.778112	0.933735	0.898594	0.850402
7°	0.916667	0.799196	0.90261	0.833333	0.896586	0.935743	0.781124	0.925703	0.883534	0.839357
6°	0.90261	0.783136	0.88755	0.851405	0.891566	0.933735	0.774096	0.927711	0.896586	0.840361
5°	0.87249	0.774096	0.883534	0.798192	0.891566	0.923695	0.774096	0.911647	0.873494	0.824297
4°	0.873494	0.777108	0.874498	0.781124	0.891566	0.919679	0.78012	0.914659	0.85241	0.830321
3°	0.844378	0.763052	0.812249	0.813253	0.86747	0.918675	0.764056	0.884538	0.861446	0.825301
2°	0.832329	0.758032	0.806225	0.763052	0.862449	0.911647	0.763052	0.906627	0.838353	0.811245
1°	0.84739	0.744979	0.768072	0.742971	0.772088	0.907631	0.758032	0.87751	0.820281	0.799197
0°	0.803213	0.740963	0.75	0.758032	0.768072	0.883534	0.74498	0.866466	0.754016	0.801205

TABLE. II. MATCHING SCORES OBTAINED AFTER APPLYING AUTO-CORRECTION ALGORITHM

Temperature in °c	Matching Scores									
	User1	User2	User3	User4	User5	User6	User7	User8	User9	User10
25 ^o	1	1	1	1	1	1	1	1	1	1
24 ^o	0.965863	0.950803	0.944779	0.925703	0.954819	0.962851	0.899598	0.952811	0.943775	0.926707
23 ^o	0.963855	0.937751	0.949799	0.930723	0.954819	0.963855	0.900602	0.970884	0.938755	0.924699
22 ^o	0.943775	0.939759	0.940763	0.921687	0.940763	0.956827	0.88755	0.9749	0.934739	0.919679
21 ^o	0.964859	0.939759	0.943775	0.925703	0.938755	0.961847	0.888554	0.965863	0.941767	0.921687
20 ^o	0.950803	0.934739	0.944779	0.916667	0.933735	0.962851	0.863454	0.962851	0.938755	0.928715
19 ^o	0.935743	0.921687	0.944779	0.923695	0.935743	0.958835	0.863454	0.959839	0.934739	0.908635
18 ^o	0.938755	0.915663	0.940763	0.903614	0.929719	0.956827	0.953815	0.965863	0.929719	0.906627
17 ^o	0.942771	0.913655	0.935743	0.893574	0.927711	0.953815	0.913654	0.943775	0.928715	0.907631
16 ^o	0.942771	0.893574	0.932731	0.873493	0.921687	0.954819	0.929718	0.954819	0.939759	0.90261
15 ^o	0.923695	0.878514	0.934739	0.878514	0.923695	0.951807	0.941767	0.948795	0.938755	0.911647
14 ^o	0.927711	0.866465	0.929719	0.868473	0.925703	0.949799	0.925702	0.953815	0.925703	0.891566
13 ^o	0.920683	0.858433	0.927711	0.858433	0.914659	0.945783	0.90261	0.941767	0.923695	0.891566
12 ^o	0.919679	0.851405	0.917671	0.863453	0.907631	0.940763	0.919618	0.934739	0.914659	0.890562
11 ^o	0.904618	0.958835	0.914659	0.853413	0.909639	0.947791	0.893574	0.934739	0.925703	0.859438
10 ^o	0.918675	0.943775	0.913655	0.951807	0.899598	0.947791	0.897536	0.932731	0.911647	0.865462
9 ^o	0.924699	0.928714	0.915663	0.933734	0.901606	0.940763	0.885542	0.932731	0.88755	0.854418
8 ^o	0.905622	0.908634	0.908635	0.938755	0.90261	0.932731	0.888554	0.933735	0.898594	0.850402
7 ^o	0.916667	0.895582	0.90261	0.91465	0.896586	0.935743	0.873493	0.925703	0.883534	0.933734
6 ^o	0.90261	0.878514	0.88755	0.89257	0.891566	0.933735	0.858433	0.927711	0.896586	0.935742
5 ^o	0.87249	0.863453	0.883534	0.873493	0.891566	0.923695	0.863453	0.911647	0.873494	0.921686
4 ^o	0.873494	0.858433	0.874498	0.858433	0.891566	0.919679	0.861445	0.914659	0.85241	0.893574
3 ^o	0.943775	0.852409	0.943775	0.843373	0.86747	0.918675	0.829317	0.884538	0.861446	0.853413
2 ^o	0.935742	0.833333	0.935742	0.80522	0.862449	0.911647	0.839357	0.906627	0.963844	0.851405
1 ^o	0.913654	0.813253	0.934738	0.823293	0.963855	0.907631	0.801204	0.87751	0.952811	0.838353
0 ^o	0.89257	0.778112	0.928714	0.796184	0.955823	0.883534	0.798192	0.866466	0.93762	0.829317

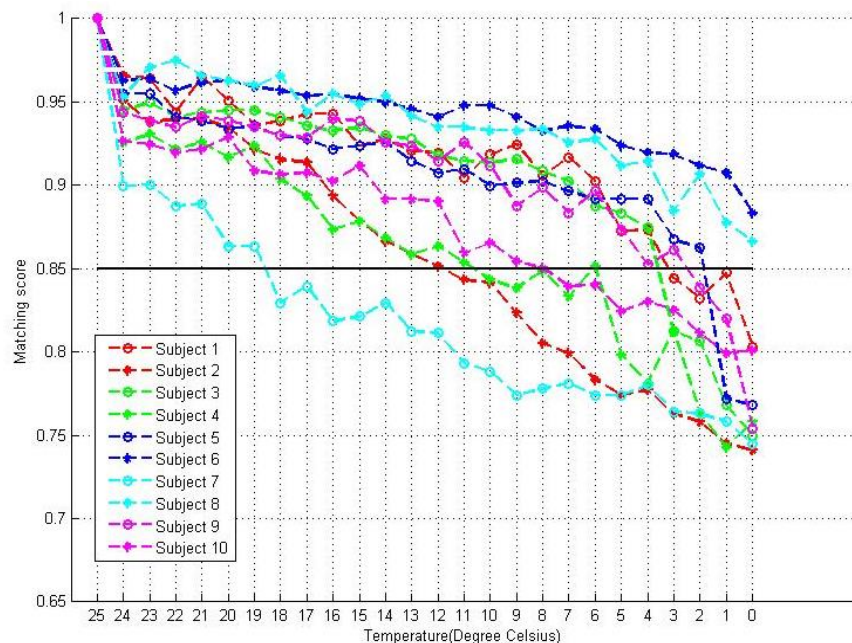


Fig. 5. Graph of Matching Scores Versus Temperature of 10-users/Subjects before Auto-Correction.

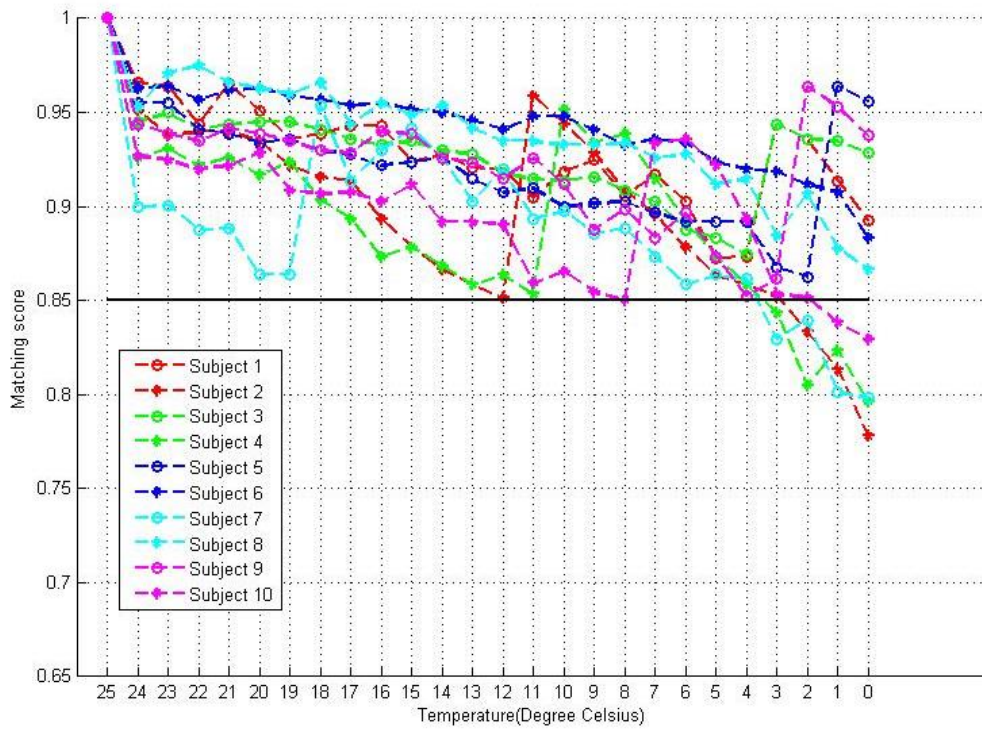


Fig. 6. Graph of Matching Scores Versus Temperature of 10-users/Subjects after Auto-Correction.

TABLE. III. SUMMARY OF SUCCESS AND FAILURE RATE

Total number of users	Without auto-correction				With auto-correction			
	Success		Failure		Success		Failure	
	Number of user	In percentage (%)	Number of user	In percentage (%)	Number of user	In percentage (%)	Number of user	In percentage (%)
10	2	20%	8	80%	6	60%	4	40%

TABLE. IV. SUCCESS AND FAILURE RATES ANALYSIS FOR DIFFERENT TEMPERATURE RANGES WITH AND WITHOUT AUTO-CORRECTION

Temperature range	Without auto-correction				With auto-correction			
	Success		Failure		Success		Failure	
	Number of user	In percentage (%)	Number of user	In percentage (%)	Number of user	In percentage (%)	Number of user	In percentage (%)
25°C to 20°C	10	100%	0	0%	10	100%	0	0%
25°C to 15°C	9	90%	1	10%	10	100%	0	0%
25°C to 10°C	8	80%	2	20%	10	100%	0	0%
25°C to 5°C	6	60%	4	40%	10	100%	0	0%
25°C to 0°C	2	20%	8	80%	6	60%	4	40%

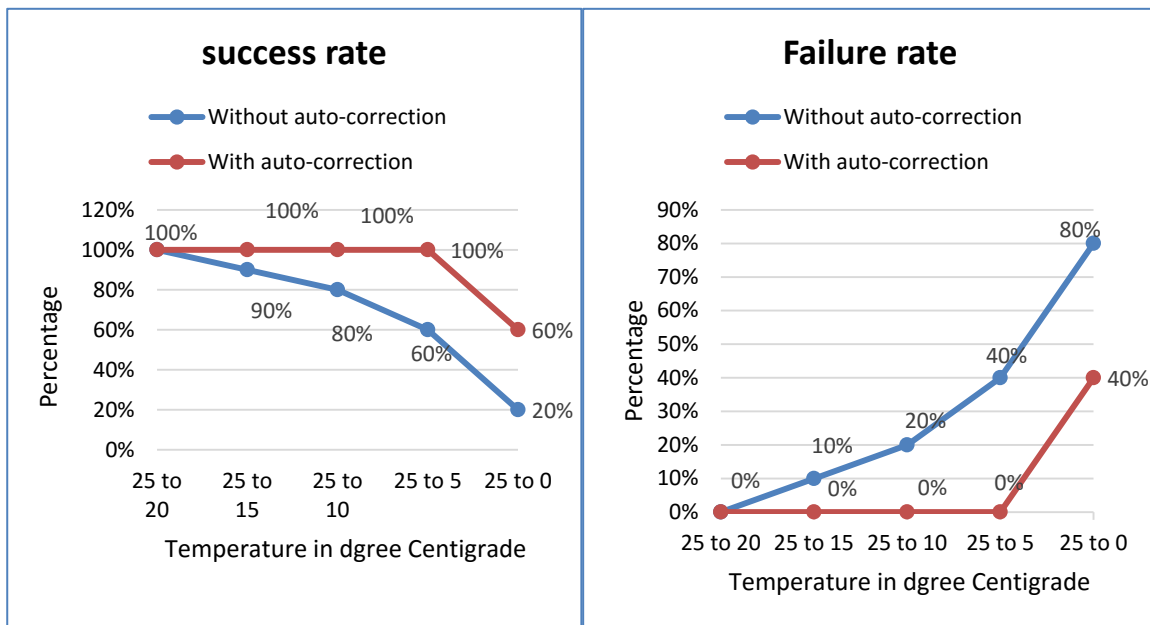


Fig. 7. Graph of Success Rates and Failure Rate with and without Correction between 25^oc to 0^oc.

IV. CONCLUSION

The proposed Reference auto-correction algorithm mainly focuses on fingerprint, based time-attendance system on daily basis for user authentication. The principle behind auto-correction algorithm is to correct user reference fingerprint template automatically whenever the rejection of genuine user occurs. Analysis is carried out on 250 fingerprint templates stored in the database of 10-users captured at varying temperature from 25°C to 0°C. Matching score is obtained by comparing the captured fingerprint templates with reference fingerprint template for the respective user without and with using auto-correction algorithm. The experimental result shows 40% improvement in the recognition rate with auto-correction algorithm. Hence, recognition rate of authentication system is improved. The proposed Reference auto-correction technique can also be applied on other Biometric traits like Face recognition, IRIS recognition, palm recognition etc., which are like to vary due aging and due to variation in temperature.

REFERENCES

- [1] Thejaswini P., Srikantaswamy R.S., Manjunatha A.S. (2018), "Impact of Fingerprint Image Quality on Matching Score", Guru D., Vasudev T., Chethan H., Kumar Y. (eds), Proceedings of International Conference on Cognition and Recognition. Lecture Notes in Networks and Systems, vol 14. Springer, Singapore.
- [2] Singh, Jagtar. "A Survey on Fingerprint Recognition Methods." Journal of Network Communications and Emerging Technologies (JNCET) www.jncet.org 7.5 (2017).
- [3] Yu, Xiaojun, et al. "Contrast Enhanced Subsurface Fingerprint Detection Using High-Speed Optical Coherence Tomography." IEEE Photonics Technology Letters 29.1 (2017): 70-73.
- [4] Peralta, Daniel, et al. "Minutiae-based fingerprint matching decomposition: Methodology for big data frameworks." Information Sciences 408 (2017): 198-212.
- [5] Li, Hai Ping. "The Application of MATLAB in Automatic Fingerprint Recognition System of Police." MATEC Web of Conferences. Vol. 63. EDP Sciences, 2016.
- [6] Yan, Haibin (2016), "Discriminative sparse projections for activity-based person recognition." Neurocomputing, 208, 183-192. doi:10.1016/j.neucom.2015.11.111.
- [7] Shanavaz, K. T., and P. Mythili. "A fingerprint-based hybrid gender classification system using genetic algorithm." International Journal of Computational Vision and Robotics 6.4 (2016): 399-413.
- [8] Kumar, Sachin, and R. LeelaVelusamy. "Kernel approach for similarity measure in latent fingerprint recognition." Emerging Trends in Electrical Electronics & Sustainable Energy Systems (ICETEESES), International Conference on. IEEE, 2016.
- [9] Peralta, Daniel, et al. "A survey on fingerprint minutiae-based local matching for verification and identification: Taxonomy and experimental evaluation." Information Sciences 315 (2015): 67-87.
- [10] Sunny Arief SUDIRO, Michel PAINDAVOINE, Tb. Maulana KUSUMA (2007), "Simple Fingerprint Minutiae Extraction Algorithm Using Crossing Number On Valley Structure", IEEE Workshop on Automatic Identification Advanced Technologies. doi: 10.1109/autoid.2007.380590.
- [11] Jing Li, Bao-Liang Lu." An adaptive image Euclidean distance", Pattern Recognition, 42(3), 349-357. Doi:10.1016/j.pactog.2008.07.017.
- [12] Kouamo, Stephane, and Claude Tangha. "Fingerprint Recognition with Artificial Neural Networks: Application to E-Learning." Journal of Intelligent Learning Systems and Applications 8.02 (2016): 39.
- [13] Werner, Gabor Á., and LászlóHanka. "Tuning an artificial neural network to increase the efficiency of a fingerprint matching algorithm." Applied Machine Intelligence and Informatics (SAMII), 2016 IEEE 14th International Symposium on. IEEE, 2016.
- [14] Gautam A., Bhateja V., Tiwari A., Satapathy S.C. (2018), "An Improved Mammogram Classification Approach Using Back Propagation Neural Network", Advances in Intelligent Systems and Computing, vol 542. Springer, Singapore.
- [15] M. Mitić, N. Vuković, M. Petrović, Z. Miljković, "Chaotic fruit fly optimization algorithm", Knowledge-Based Systems (2015), doi: http://dx.doi.org/10.1016/j.knosys.2015.08.010.