A High Speed Algorithm for Identifying Hand Gestures for an ATM Input System for the Blind

Sudhir Rao Rupanagudi, Ranjani B. S., Varsha G. Bhat, K. Surabhi, P.R. Reshma, Shruthi G., Sarayu K. P, Sangeetha R
WorldServe Education, Bengaluru, India
sudhir@worldserve.in

Rajesh Rao B
Department of Electronics & Communication
Jyothy Institute of Technology
Bengaluru, India

Vasanti S
Department of Telecommunication
Atria Institute of Technology
Bengaluru, India

Abstract— **With the evolution in science and technology, a lot has been done over the past few years to make the lives of the differently-abled more comfortable and easy. This paper concentrates on a novel methodology to ease the use of an ATM machine for the blind. It describes an approach wherein both the username and PIN for the ATM machine can be input using British Sign Language. A cost effective setup and also a high speed algorithm for hand gesture recognition has been elaborated. In comparison with previous algorithms, the method explained in this paper is 1.65 times faster thus proving its efficacy and efficiency. All algorithms were first designed and developed in MATLAB 2011b and then later deployed as software using the Java programming language.**

Keywords— ATM; hand gesture recognition; image processing; British Sign Language; JAVA; security; Sign language recognition; video processing

I. INTRODUCTION

The dawn of the 21st century has heralded immense growth and progress in the field of science and technology. From automated cars [1] to eye gaze recognition systems for MND patients [2] - the advancements and innovations in this field has led to a major improvisation in human lifestyle. Along with this evolution, there has also been a great need to safeguard the security of the users utilizing these contraptions.

Security forms a very important and integral part of our daily life. Daily activities including checking email, entering certain areas of a secure installation, online shopping and also withdrawing money from an ATM, all involve a certain type of secrecy in order to prevent unauthorized users from accessing private data. The simplest form of security which is used quite popularly till date is entering a password [3] – a combination of alphanumeric and special characters known only to the user. Though secure, it is found through many surveys, that the password is the most forgotten or most divulged secret [4].

In order to overcome the disadvantages of the use of a traditional password, many other types of user authentication methodologies have come into light. One of the most popular methods nowadays, is face recognition [5]. The user stands in front of a camera and software in the background captures an image of the face, pinpoints strategic locations on the face and compares the same with an already provided photograph of the user. Another way of providing access is the retinal scan [6]. The user stands in front of a biometric sensor which captures the picture of the eye. Based on comparison of the pattern of rods and cones present in the retina with a database, authorization is provided. Apart from these methodologies, fingerprint recognition is also widely used [7]. In this method, a biometric fingerprint sensor is provided wherein the user swipes his finger and the fingerprint is recorded. Once again like other methodologies, this is compared with an already predefined database of various users' fingerprints. Over the past few years, voice recognition is also coming into the limelight [8], where the user speaks a particular sentence and this is analysed for various parameters such as pitch, tone, frequency and baritone of the user and access is granted if matched.

Though all the above methodologies are quite popular in various applications, they come along with their own set of disadvantages as well, especially in a perspective of a differently abled person who is blind. In most public access systems such as ATM's, keyboards available to key-in passwords do not contain keys in Braille. This makes it a very arduous task for the blind to input their password and would require assistance annulling the idea of privacy. Though voice activated systems exist, they would require the user to speak out the password, once again compromising security. Retinal scans are once again ineffective for the blind, due to deformities existing in the eye anatomy. Though fingerprint scanners could be used, reproducing fingerprints or forcefully obtaining the same could be a major security concern.

In this paper, we present a novel methodology based on the research carried out in [9], to utilize sign language for the blind to access ATM systems. In continuation with the ideologies presented in [9], this paper proposes a much more faster algorithm to identify various numbers involved in showing the pin at the ATM. It also proposes certain modifications in the existing setup along with a new added security feature which involves the user to input a username followed by the pin, which was a major flaw in the method elaborated in [9]. Section II briefly describes various methodologies to identify hand gestures. The modifications to the setup have been explained in Section III. The novel high speed algorithm for gesture identification has been described in Section IV. Section V deals with the results obtained along

978-1-4673-9542-7/15/$31.00 ©2015 IEEE

with speed comparisons and Section VI forecasts the future scope of this research.

II. Existing Methodology

As mentioned in the previous section, the current work is a continuation of the research carried out in [9]. In [9], the authors present a novel methodology to utilize sign language hand gestures to input a pin for an ATM machine. The main idea of the project is to ease the use for blind to access such systems. An innovative cost effective setup made of thermocole is shown. [9] also presents an algorithm which utilizes a combination of calculating the centroid of the hand and also bounding boxes to locate and identify different fingers shown. The method described though robust, involves a setup which lacks security. The algorithm though better than a previously used methodology (which uses Otsu's algorithm), can still be improvised and made faster.

Over the years there have been other notable approaches in identifying hand gestures but most of them involve the use of a glove for giving inputs. Electronic components such as accelerometers and other MEMS architectures are utilized for the same [10]. This method though accurate, involves the user to always require an expensive specialized glove for depicting gestures in turn leading to discomfort.

Other papers such as [11], discuss an algorithm to identify hand gestures for a number guessing robot. The algorithm currently works only for the numbers 1 to 5 based on the requirement. In [12], the authors describe a method to identify hand gestures based on the Haar transform. [13] uses the SOM-hebb classifier for the same purpose and [14] utilizes the mean shift and Kalman filter. [15] and [16] involve the Sift algorithm and Artificial Neural Networks for classifying the gesture identified respectively. All the aforementioned approaches though highly accurate involve several complex arithmetical calculations slowing down the process of recognition. Also, such level of complexity is not a necessity for our current requirement.

With regards to setup, all the above mentioned papers perform the hand gesture recognition in an open environment which can pose to be a major security issue in an ATM environment. The camera to be selected for the experiments to be conducted also plays a major role. In [17], mention is made of using a high end Kinect camera for performing experiments, but since cost-effectiveness is also a need of the hour, utilizing a low cost low resolution camera is a major necessity. The next section describes in detail our modified setup with all major specifications.

III. Setup

As mentioned in the previous section, a lot of importance has to be given for both cost-effectiveness and also security with respect to the setup, since it shall be used in a critical application i.e. showing the pin number in the form of gestures to access an ATM.

Keeping the cost factor in mind, the setup is made in the form of a box made up of a cheap thermacole sheet. The sheet depth of the material chosen is one inch. The length and width of the box is 37 cm and 25 cm respectively and the height is 24 cm. At one end of the box, a web camera with image resolution of 5500 x 3640 pixels and speed of 30 frames per second is placed. The image obtained from the camera is resized to 640x480 to ease processing. This is fixed at the center of the far end, with a height of 12 cm, such that it can easily capture the hand gestures that are shown at the opposite end of the box. The distance between the hand and the camera is around 28 cm. The box is blackened inside and on the outer surface with cardboard in order to prevent any excess light from entering. This ensures constant lighting conditions at all points of time. LED's of the camera can be used when need required.

As mentioned in Section I, the setup described in [9] had a major security flaw. The gestures shown at the open end of the box was completely visible to the outer world and was not enclosed. In order to overcome this issue, we utilize a black-coloured cloth of thickness 2mm to cover the entire open area. The black cloth is fixed to the top end of the box and left free at the bottom end, so that the user can easily insert his hand and show the gestures. In order to make sure the user keeps his hand at the correct position at all times, instead of using a protruding horizontal column as shown in [9], a semicircular groove is now provided wherein the wrist of the user can be placed. This proves to be more comfortable to use unlike the previous design.

The setup diagram has been shown in Fig. 1, with the major parts of the same identified and labelled.

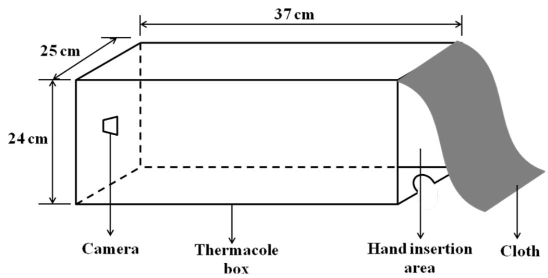

Fig. 1. Setup used for the experiments

In the next section we describe the novel high speed algorithm to identify various gestures in sign language and in turn identify the pin.

IV. Our Methodology

The complete algorithm mentioned in this section was designed and developed first using MATLAB programming language and further implemented and deployed as software using the JAVA programming language. For all experiments conducted, the British variant of sign language was utilized. Since we utilize the same, only for showing the pin number in an ATM machine, we concentrate on numbers alone and not alphabets. Numbers depicted using the British Sign Language (BSL) can be seen in Fig. 2.

As mentioned apriori, in order to make the algorithm more robust and secure than the one explained in [9], the algorithm is now divided into two stages. The first stage involves the user to first show his username in the form of a four digit number and then on authentication provide the password corresponding to his username. The previous variant [9] had no provision to provide both and hence would work only for a single user. The steps involved in the same can be seen in the flowchart in Fig. 3.

Fig. 2. Numbers (delimiter and 1-9) depicted using BSL

```
Step 1: Run algorithm for Sign Language Detection.
Step 2: Compare filtered array with username.
        BEGIN LOOP
            IF username is matching database
                Load password for corresponding username
                and run Sign Language Detection algorithm.
                Also compare filtered array with password.
                    IF password is matching
                        Proceed to further transactions.
                    ELSE
                        Alert Security.
                    END IF
            ELSE
                Return to Step 1.
            END IF
        END LOOP
```

Fig. 3. Pseudocode for a complete ATM authentication using BSL

It can be seen from Fig. 3, that the first step in the algorithm is to identify the gestures shown and store the numbers in an array. Once this is complete, the numbers stored in the array are compared with a database of various usernames. In the case the username does not match, the user is asked to retry again. In case the username does match, the password associated with this username is fetched. The user is then asked to once again show gestures, this time the numbers corresponding to the password. In case the numbers shown match with the password fetched, the remaining splash screens to withdraw money, are shown to the user. In case it does not, then the security is alerted or a certain number of retrials are given.

The algorithm to identify the number shown, in the form of gestures, has been depicted in the form of a flowchart in Fig. 4 and has been explained below in detail.

A. Capturing the live video feed

Using the setup mentioned in Section III, a live video feed of the user showing various gestures is captured. Since the gestures shown are captured at a rate of 30 frames per second, there is a high possibility of the same number being stored repeatedly and hence there is a major requirement to filter the results obtained. To ease this process, the user is asked to show a delimiter after showing each number. Usually this delimiter could be a blank gesture (not showing the hand at all) in order to differentiate between the various gestures shown.

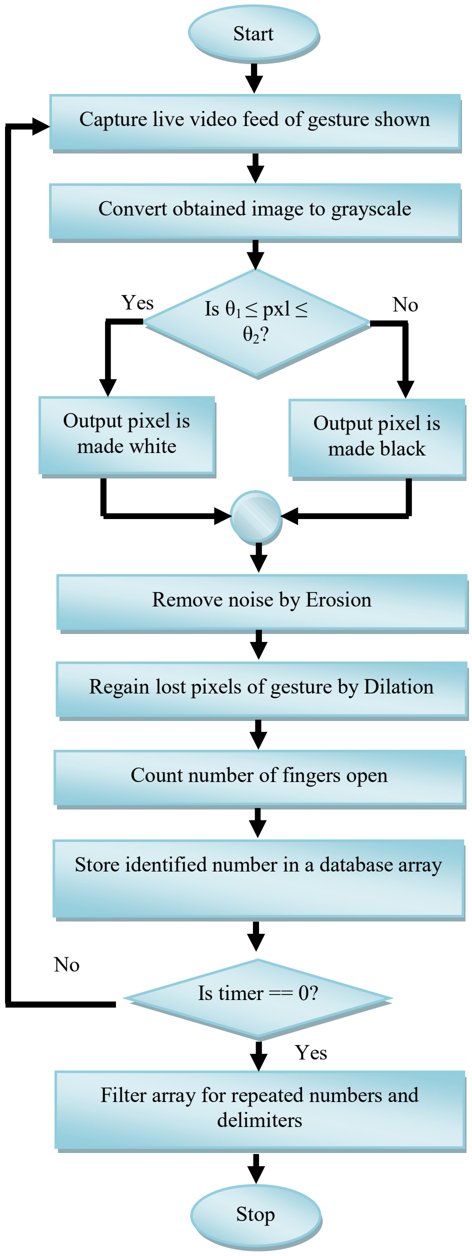

Fig. 4. Algorithm to identify BSL numbers

B. Color Conversion and check for security breach

The color image captured, as mentioned in Section II, is of size 640x480. It is a well known fact that a color image is represented by 3 planes of color – Red, Green and Blue. Since 8 bits are required to represent each color pixel, the total number of bits required to represent an image would be 640*480*8 which would compute to a memory requirement of around 73,72,800 bits per picture! Processing the same amount of data would require several CPU clocks and this would slow down the process drastically. Hence in order to reduce the time taken, the images obtained are first color converted to a predefined color model. In these experiments, the picture was converted to grayscale. This conversion comes with a great set of advantages. The first advantage is that the number of bits required per pixel now reduces by 1/3rd. The second major advantage is that the grayscale color space model is dependent on brightness. This though disadvantageous in many other projects proves to be favourable in this experiment. Since the setup chosen has a constant lighting condition as explained in Section III, the ranges of the hand would not change irrespective of any user, hence making it very easy to extract the hand from the background. The third advantage comes in view as a future scope. In case an unauthorized person tries to lift the black cloth whilst the system is being used by the blind (in order to view the gestures being shown), apart from feeling the motion of the cloth by the user, the algorithm itself would immediately cease to work due to changes in light conditions and in turn alert the user of a security breach. This could further enhance the confidence level of the blind users utilizing this device.

C. Segmentation

The color converted image obtained above is now subjected to a segmentation based on thresholding algorithm. Through several sets of experiments on various users, the common range in grayscale for any user was obtained. A change in hand color does not affect the range selected since it is the palmar part of the hand visible – which is usually brighter. Also, since the hands of any user is illuminated by the same amount of light from the camera and is not affected from any external light source, the ranges thus would remain the same.

The segmentation algorithm is guarded by (1).

$$S_{(i,j)} = \begin{cases} 255, & \theta_{y1} > y > \theta_{y2} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The picture is scanned from the top left hand corner to the bottom right and each and every pixel is identified as to whether it belongs to the hand or not. This is performed by checking whether the pixel value falls in the predefined range and converting the same to white if it does. The pixels are converted to black, if not. In this way, from an 8 bit per pixel picture, it is now converted to a binary image represented by just a single bit per pixel. This brings down the memory requirement by 1/8th and improves processing time by the same amount.

D. Noise removal

Due to imperfections in the color of the black cloth used in the background and also due to unforeseen transparency in certain regions of the cloth, minute portions of the same might be considered as the hand region & made white. In order to eliminate the same, a morphological operation by name erosion is utilized. This is applied on the binary converted image, several sets of times (depending on the amount of noise) and the noise is in turn removed. Erosion works on the principle that if there is any white pixel surrounded on any side by any black pixel, then that white pixel is converted to black in a resultant image.

The disadvantage of erosion is that it shrinks the edges of the object of interest as well – in this case the hand.

E. Regaining lost pixels of the hand

This process is the absolute opposite of the erosion process explained above. In this process, a complete scan of the eroded image is performed and if in case any black pixel is now surrounded by white, it is made white in a resultant image. This ways all the lost pixels towards the edges of the hand are now regained. Upon completion of this step, counting and identifying the number of fingers open can begin.

F. Identifying fingers and counting the number of fingers open

As mentioned in Section I, the whole idea of the novel algorithm is to speed up the execution time in comparison with the existing methodologies. Instead of using the centroid and the bounding box approach mentioned in [9], a much simpler approach can be followed. This is possible due to following reasons.

Firstly, due to the modifications made to the setup, majorly the groove, the hand position of the user remains constant making it unnecessary to find the centroid of the hand, as performed in [9]. Another major improvisation is the use of BSL, which in turn makes the necessity to find which finger is open totally redundant as opposed to American Sign Language used in [9]. The need for bounding boxes used in [9] therefore disappears which were leading to erroneous results due to movements in the hand in the previous method.

In order to understand the algorithm better, we take into consideration Fig. 5 wherein the user is showing the number 7 using BSL.

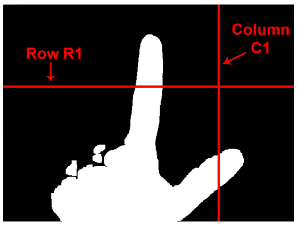

Fig. 5. The number 7 shown in BSL with Row and Column Marked

Usually, the most common way to count is to use an edge detection algorithm along with contour tracing, but this is commonly practiced when the objects to be counted are distinct blobs [18]. In the case of fingers, since they are conjoined after dilation, this approach cannot be followed.

Since the hand is always in the same position due to the groove provided, it can be clearly seen that the little, ring, middle and index finger would always lie in the same row. Due to the anatomy of the hand, the thumb would always lie lower than the rest of the fingers. It lies slightly perpendicular to the rest of the hand when opened, but at the same time would exist in the same column irrespective of the gesture shown. Taking these properties into account, counting the number of fingers open becomes a very easy task and this can be done by fixing a single row for finger counting and a single column for the thumb.

Let R1 be the row selected for counting fingers and C1 the column to identify the thumb. It can be clearly seen that a finger can be considered open whenever there is a transition from black to white. Hence by counting the total number of these transitions on R1, the total number of fingers open can be easily obtained.

Fig. 6a shows the area of the transition near the finger open. If the row and column co-ordinate under consideration (R1,C) is black and the next subsequent pixel (R1,C+1) is white, a counter to count fingers (countf) is incremented. Similarly, for the thumb region, instead of checking for a transition on a row, a transition is checked for on a column. This is zoomed and shown in Fig. 6b where (R,C1) is the pixel under consideration and is black, and (R+1,C1) is the white pixel belonging to the thumb. In case a transition does occur, a separate counter for the thumb (countt) is incremented.

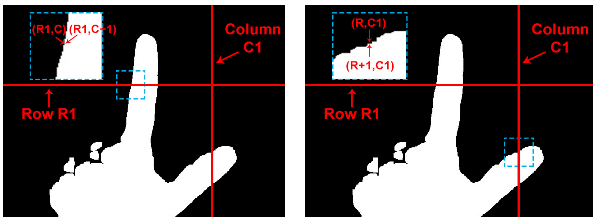

Fig. 6. Transitions near (a) finger area and (b) thumb area

Finally, in order to obtain the final number gestured, the two counts – countf and countt have to be combined together based on the rules of BSL and this represented in the form of a pseudocode in Fig. 7. In case no number has been shown, a delimiter - in this case the number 24 is passed on to the array.

```
Step 1: Obtain 'countt' an 'countf' values.
Step 2: BEGIN LOOP
            IF 'countt' is 0 and 'countf' is 0
                'res' is 24.
            ELSE IF 'countt'is 0 and 0 < 'countf' ≤ 4
                'res' is 'countf'
            ELSE IF 'countt'is 1 and 0 < 'countf' ≤ 3
                'res' is 'countf' incremented by 6.
            ELSE
                'res' is 5.
            END IF
        END LOOP
```

Fig. 7. Pseudo-code for identifying the number based on count

The final number thus obtained is then stored in an array and the process continues till a timer runs out for showing either the username or the pin.

G. *Filter array for repeated numbers and delimiters*

As mentioned earlier, since the camera captures at 30 frames per second, showing a particular gesture for a duration of say 1 second, would store it in the array repeatedly 30 times. Utilizing this array, directly for comparing with the username or pin in the database, would prove to be erroneous. Hence in order to rectify this problem, the array must be filtered. This step is performed once the completely filled array is obtained at timeout.

A number present in the array is filtered if and only if it has repeated at least more than 3 times, is the last number of the repeated set (to prevent further repetitions) and is not the number 24. This filtered array thus obtained is now compared with the username or password in the database and the authentication is then verified.

The results obtained at every stage of the aforementioned algorithm and also the speed tests performed have been elaborated in the next section.

V. Results

As mentioned in the previous section, all algorithms were designed and developed first using MATLAB 2011b and then later deployed as software using the Java programming language along with the Java Development Kit (JDK) 1.7, OpenCV 2.4 and JavaCV 0.7 [19]. The outputs obtained at each every stage of the algorithm have been shown below.

Fig. 8a shows the number 7 shown by a user in BSL.

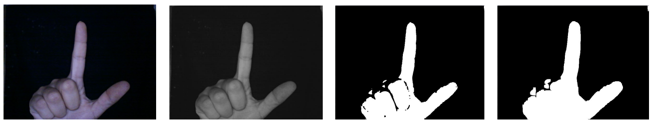

Fig. 8. The number 7 (a) as seen by the camera in RGB (b) in grayscale (c) after erosion and (d) after dilation

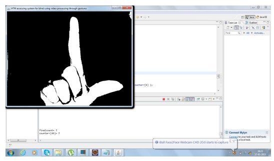

Fig. 9. The final count obtained

It can be clearly seen, since the box is completely closed on all sides and the interiors are dark, the hand is perfectly illuminated whilst the background remains dark. This assists in further processing. Fig. 8b shows the grayscale converted image of the same number. The output obtained after thresholding using segmentation was already seen Fig. 5. The region of the hand is made totally white and the background black. It can also be seen, the presence of noise in the form of extra white pixels towards the left and top right of the photograph. This can be removed using morphological processing. Fig. 8c shows the resultant of the erosion operation. It can be seen that the noise present in Fig. 5 has totally disappeared but along with that the edges of the hand have diminished thanks to erosion. In order to regain back the edges, dilation is performed and this can be seen in Fig. 8d. The final result obtained after counting can be seen in Fig. 9, displayed in the Java console window.

The efficiency of the algorithm presented in this paper can be evaluated by comparing its speed of execution with two other popular methods – one being the method using bounding boxes[8] and the other using a combination of Otsu algorithm and Method of Moments (MoM) [20], [21]. The speed in execution was calculated in all 3 cases using the same version of MATLAB and can be seen tabulated in Table I.

It can be easily seen that the method explained in this paper wins hands down in terms of speed when compared with the two other popular methodologies. By being 1.2 times faster than the bounding box method and 1.65 times faster than the Otsu + MoM method, it clearly shows that the methodology described in this paper could be deployed for ATM input systems using sign language technologies.

TABLE I. COMPARISON OF SPEED OF VAROIUS ALGORITHMS

Steps in the algorithm	Various algorithms		
	Our algorithm	***Bounding box method [9]***	***Based on Otsu and MoM [20],[21]***
Segmentation & thresholding	377 ms	377 ms	539 ms
Finding reference point ρ	N/A	2.5 ms	14 ms
Identifying bounding box filled or not	N/A	75 ms	75 ms
Identifying count	2.3 ms	15 µs	15 µs
Total execution time	379.3 ms	454.515 ms	628.015 ms

It is also important to note, that an ASIC implementation of our method proves to be area efficient as well. The reason being, the on-chip architecture for identifying gestures, both for the username and pin identification is one and the same. This adds to the advantage of the design.

VI. CONCLUSION AND FUTURE SCOPE

In this paper, a novel high speed algorithm to detect British Sign Language numbers for an ATM input system has been described. The algorithm proves to be speed efficient in comparison with two other popular approaches. In order to improvise the ease of use for the blind, a voice system was also implemented as an extension, in conjunction with this setup. This aided users to be able to hear the number shown. Another major highlight of the project was the cost effectiveness of the setup. Installing and setting up the same takes a matter of minutes and this would in turn be a great boon especially in developing countries such as India. As a future scope, major technology firms which deal with building and developing ATM machines could use this technology and therefore simplify greatly, the ATM user experience, for millions of blind users across the world.

REFERENCES

[1] Prashanth, C.R.; Sagar, T.; Bhat, N.; Naveen, D.; Rupanagudi, S.R.; Kumar, R.A., "Obstacle detection & elimination of shadows for an image processing based automated vehicle," Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on , vol., no., pp.367,372, 22-25 Aug. 2013

[2] Rupanagudi, S.R.; Bhat, V.G.; Karthik, R.; Roopa, P.; Manjunath, M.; Glenn, E.; Shashank, S.; Pandith, H.; Nitesh, R.; Shandilya, A.; Ravithej, P., "Design and Implementation of a Novel Eye Gaze Recognition System Based on Scleral Area for MND Patients Using Video Processing", Advances in Computing, Communications and Informatics (ICACCI), 2014 International Conference on , vol., no. 320, pp. 569-579, 24-27 Sept. 2014

[3] Conklin, A.; et.al, "Password-based authentication: a system perspective," System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on , vol., no., pp.10 pp.,, 5-8 Jan. 2004

[4] Bonneau, J.; et.al., "The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes," Security and Privacy (SP), 2012 IEEE Symposium on , pp.553,567, 20-23 May 2012

[5] Peter, K.J.; et. al, "Improving ATM security via face recognition," Electronics Computer Technology (ICECT), 2011 3rd International Conference on , vol.6, no., pp.373,376, 8-10 April 2011

[6] Meenakshi, V.S.; Padmavathi, G., "Security Analysis of Hardened Retina Based Fuzzy Vault," Advances in Recent Technologies in Communication and Computing, 2009. ARTCom '09. International Conference on , vol., no., pp.926,930, 27-28 Oct. 2009

[7] Tariq, A.; Akram, M.U.; Khan, S.A., "An automated system for fingerprint classification using singular points for biometric security," Internet Technology and Secured Transactions (ICITST), 2011 International Conference for , vol., no., pp.170,175, 11-14 Dec. 2011

[8] Khadilkar, S.U.; Wagdarikar, N., "Android phone controlled voice, gesture and touch screen operated smart wheelchair," Pervasive Computing (ICPC), 2015 International Conf on , pp.1,4, 8-10 Jan. 2015

[9] Dhruva, N.; Rupanagudi, S.R.; Kashyap, H.N.N., "Novel Algorithm for Image Processing Based Hand Gesture Recognition and Its Application in Security", Advances in Computing, Communication and Control (ICAC3), 2013 International Conference on , vol., no. 361, pp. 537-547, 18-19 Jan. 2013

[10] Dekate, A.; et. al, "Magic Glove - wireless hand gesture hardware controller," Electronics and Communication Systems (ICECS), 2014 International Conference on , vol., no., pp.1,4, 13-14 Feb. 2014

[11] Chyi-Yeu Lin; et. al, "An interactive finger-gaming robot with real-time emotion feedback," Automation, Robotics and Applications (ICARA), 2015 6th International Conference on , vol., no., pp.513,518, 17-19 Feb. 2015

[12] Tripathy, A.K.; et. al, "Voice for the mute," Technologies for Sustainable Development (ICTSD), 2015 International Conference on , vol., no., pp.1,6, 4-6 Feb. 2015

[13] Hikawa, H.; Kaida, K., "Novel FPGA Implementation of Hand Sign Recognition System With SOM–Hebb Classifier," Circuits and Systems for Video Technology, IEEE Transactions on , vol.25, no.1, pp.153,166, Jan. 2015

[14] Gupta, P.; Joshi, G.; Dutta, M., "Comparative Analysis of Movement and Tracking Techniques for Indian Sign Language Recognition," Advanced Computing & Communication Technologies (ACCT), 2015 Fifth International Conference on , vol., no., pp.90,95, 21-22 Feb. 2015

[15] Agarwal, R.; Raman, B.; Mittal, A., "Hand gesture recognition using discrete wavelet transform and support vector machine," Signal Processing and Integrated Networks (SPIN), 2015 2nd International Conference on , vol., no., pp.489,493, 19-20 Feb. 2015

[16] Ullah, S.; Saman, G.; Khan, F., "Hand gesture recognition for automatic tap system," Intelligent Systems and Computer Vision (ISCV), 2015 , vol., no., pp.1,5, 25-26 March 2015

[17] Chong Wang; Zhong Liu; Shing-Chow Chan, "Superpixel-Based Hand Gesture Recognition With Kinect Depth Camera," Multimedia, IEEE Transactions on , vol.17, no.1, pp.29,39, Jan. 2015

[18] Hedberg, H.; Kristensen, F.; Owall, V., "Implementation of a Labeling Algorithm based on Contour Tracing with Feature Extraction," Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on , vol., no., pp.1101,1104, 27-30 May 2007

[19] Rupanagudi, S.R.; Ranjani, B.S.; Nagaraj, P.; Bhat, V.G.; Thippeswamy, G., "A novel cloud computing based smart farming system for early detection of borer insects in tomatoes," Communication, Information & Computing Technology (ICCICT), 2015 International Conference on , vol., no., pp.1,6, 15-17 Jan. 2015

[20] Qiuyu, Z., Fan, C., Liu, X., "Hand Gesture Detection and Segmentation Based on Difference Background Image with Complex Background", Embedded Software and Systems (ICESS), 2008 International Conference on, pp. 338 – 343, 2008

[21] J-Apiraksa, et. al, "A Simple Shape-Based Approach to Hand Gesture Recognition", ECTI-CON, 2010 International Conference on, pp. 851-855, 2010