HAND GESTURE BASED RECOGNITION SYSTEM FOR HUMAN COMPUTER INTERACTION

Rajarshi Das
Electronics and Telecommunication Department, Assam Engineering College, Guwahati, India
rajarshidas24@gmail.com

Roshan Kumar Ojha
Electronics and Telecommunication Department, Assam Engineering College, Guwahati, India
its.roshanojha@gmail.com

Dipam Tamuli
Electronics and Telecommunication Department, Assam Engineering College, Guwahati, India
dipamtamuli97@gmail.com

Swapnil Bhattacharjee
Electronics and Telecommunication Department, Assam Engineering College, Guwahati, India
nil.neel4@gmail.com

Niranjan Jyoti Borah
Electronics and Telecommunication Department, Assam Engineering College, Guwahati, India
nborah@aec.ac.in

Ruchira Mazumdar
Electronics and Telecommunication Department, Assam Engineering College, Guwahati, India
Ruchira.ete@aec.ac.in

***Abstract* - Gesture provides a form of interaction that reflects the expression of users in the actual world. For human computer interaction, visual interpretation of hand gesture can help in achieving the ease and naturalness. Recently, the demand for interaction by gesture has become widely used and, in the future, it may replace the concept of mouse and keyboard by vision-based approach. This motivates a very active research area concerned with computer vision-based interpretation of hand gesture. This paper focuses on a project where we aim to develop a system which recognises a set of hand gestures and thus in turn produce commands for human computer interaction based on image processing and extraction techniques.**

***Keywords* – Image processing, Hand gesture recognition, Human computer interaction (HCI), ATM, COVID**

I. INTRODUCTION

A. *Brief introduction*

In the current time, interaction with computers has enhanced to such an extent that as a human being it has become very essential and further, we also cannot live without it. Technology seems to be so incorporated in our regular lifestyle that we're using it to work, shop, communicate and even entertain ourselves. Computing, communication and display technologies are being commonly acknowledged to keep advancing, but existing techniques may become a barrier in the effective use of the available information flow.

Recent decades have witnessed a drastically accelerated speed in the creation and acceptance of new technologies, even while major disparities exist in terms of adoption in different areas of the world, notably in the least developed nations. These rapid changing involves new technologies like: machine-learning, IOT, big data, AI, image processing, nano technology etc. In this project we have used image processing to detect some set of gestures for better human computer interaction.

B. *Introduction to area of work and present-day scenario*

Most computer applications require more and more interaction to be used efficiently. Human-computer interaction (HCI) has therefore been a thriving line of research over the last few years. First of all, based in the past on punched cards, reserved for experts, interaction has progressed to a graphical user interface. The interaction consists of a direct manipulation of graphic objects, such as icons and windows, by means of a pointing device. Even if the invention of the keyboard and the mouse is a great step forward, there are still situations in which these devices are incompatible for HCI. This is the case, in particular, for the interaction with 3D objects. The mouse's two degrees of freedom (DOFs) are insufficient to accurately represent the three dimensions of space. Hand movements are a natural and appealing alternative to these inconvenient control systems for human-computer interaction. Using the hands as a tool can make it easier for people to communicate with computers. When we associate with other people, our hand gestures are important to other people, and the knowledge they transmit is important in a variety of forms.

Image processing and artificial intelligence, gesture recognition is a rapidly expanding area. Gesture recognition is a technique in which human body parts movements or postures are recognised and utilised to operate computers and other electronic devices. Our main area of work is on Image Processing through which we have used to recognise some set of gestures mainly numbers then these numbers and are mapped with some co-ordinate on the screen for some operations like clicking. So, basically through these gestures we were able to use system for better human-computer interaction which can be a better alternative for mouse’s two degrees of freedom.

C. *Motivation to do the project*

Gestures are not only decorative elements of spoken language, but also integral parts of the language generation process. A gesture is a physical movement of the hands, arms, face, and body made with the intention of conveying a message. Recognizing hand movements for contact, in particular, will assist in achieving the ease and convenience that many people desire.

For human-computer interactions, naturalness is needed. Hand gestures are often used by users. Their emotions are expressed and their opinions are communicated. According to reports, compared to other body sections, the hand has been commonly used for the purpose of gesturing as it is a natural means of human-to-human communication, thus it is best suited for human computer interaction.

D. *Objective of the work*

In order to achieve human computer interaction, many systems can be developed. As we have mentioned above, we are looking to build a hand gesture recognition system to achieve interaction between human and computers. There are many approaches which can be adopted in order to build a hand gesture recognition system. These are mentioned in the following chapter of literature review. We have used vision-based approach for the recognition of hand gestures. Like we Firstly tried to detect hand through skin colour range, then we have used some mathematical operations to calculate the angle, area of convex hull, area of contour formed by hand and used these to calculate the area ratio. We mainly used these parameters to recognise some set of gestures. Furthermore, we mapped those hand gestures into specific commands to our computer through various methods of image processing and extraction. The detailed methodology will be explained in the following chapters.

During the first wave of COVID people were afraid about getting COVID-19 if they use the public ATMs. Further it was a bit unhygienic and risky also to operate the machine. So, we come up with this idea of making a prototype for touchless ATM which might be a good alternative for traditional touchpad or mouse’s two degree of freedom.

II. LITERATURE REVIEW

Based on above surveys and reviews we get a clear understanding how hand gesture recognition system works. Like after reading the Kanniche’s (2009) we get an idea that feature of hand gesture recognition schemes are psychological aspects of movements dependent on hand gesture taxonomy and representation. Several gesture recognition taxonomies were suggested in the literature from individual to individual. Some of the Vision based hand gesture taxonomies given by Kanniche are shown below in the Fig. 1.

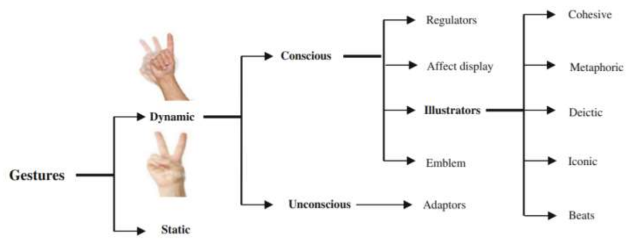

Fig. 1 Vision based hand gesture taxonomies

We also have to decide on which basis we want to do hand recognition. Basically, there are two methods as mentioned by *Bourke et al.*and these methods are- 3D model-based detection and appearance based as shown in Fig. 2.

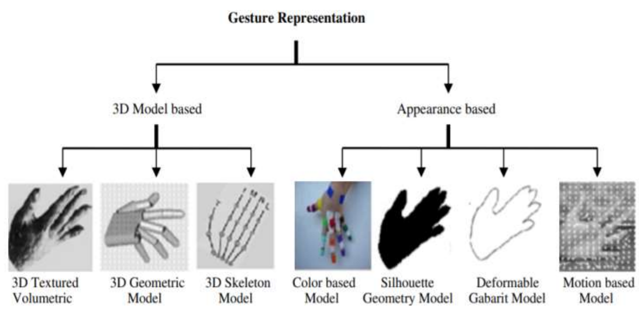

Fig. 2 Vision based hand gesture representations

From the above researches we can conclude that hand gesture recognition system comprises of three fundamental phases- detection, tracking and recognition. This portion of the survey examines some of the popular strategies for hand gesture identification used by most researchers by grouping the three main phases of the detection and recognition in the verticals as seen in the Fig. 3

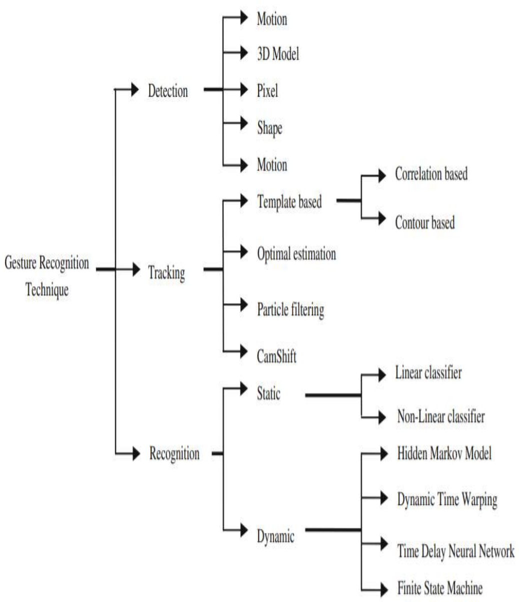

Fig. 3 Vision based hand gesture recognition techniques

III. METHODOLOGY

A. *Basic Overview*

Digital image processing is the use of a digital computer to process digital images through an algorithm. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and distortion during processing.

Image processing basically includes the following three steps: Importing the image via image acquisition tools; Analysing and manipulating the image to reduce the level of noise; Output in which result can be altered image or report that is based on image analysis.

After acquiring the desired results, we use those results to control the buttons of a web page interface by noting down each and every coordinate of the interactive buttons and mapping it with our image processing model so that it works in the user friendly and interactive way.

B. *Step by step process*

```
def click(x,y):
    win32api.SetCursorPos((x,y))
    win32api.mouse_event(win32con.MOUSEEVENTF_LEFTDOWN,x,y,0,0)
    win32api.mouse_event(win32con.MOUSEEVENTF_LEFTUP,x,y,0,0)
```

Fig. 4 Click function defined

This is a function defined to perform a click on a certain coordinate of our screen. Firstly, we will set the cursor to a position in a screen and then we do an event of clicking by a mouse using Windows API.

```
# define range of skin color in HSV
lower_skin = np.array([0,20,70], dtype=np.uint8)
upper_skin = np.array([150,255,255], dtype=np.uint8)
```

Fig. 5 Range of skin colour is defined

Here we are defining the range of our skin colour in Hue Saturation Value using Numpy which is a Python Library.

```
#Morphological Approximation
erode=cv2.erode(mask,kernel,iterations=4);
mask = cv2.dilate(mask,kernel,iterations = 4)
```

Fig. 6 Morphological Transformations

Dilation and erosion are basic morphological processing operations that produce contrasting results when applied to either gray-scale or binary images.

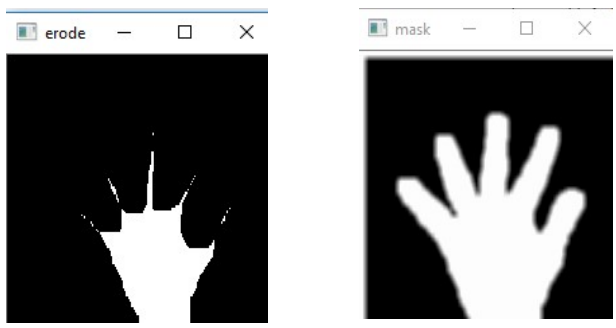

Fig. 7 Erosion and Dilation

After that we determine the contours and the convex hull of our hand and then calculate the area ratio of the contour and the convex hull.

```
arearatio=((areahull-areacnt)/areacnt)*100
```

Fig. 8 Area ratio is defined

Then we find the defects of a hand or in simple words the space between the two fingers. Geometrically when we split two fingers a triangle is formed, so using Heron's Formula we determine the area of that triangle and use the Cosine rule to determine the angle between two fingers.

```
for i in range(defects.shape[0]):
    s,e,f,d = defects[i,0]
    start = tuple(approx[s][0])
    end = tuple(approx[e][0])
    far = tuple(approx[f][0])

    # find length of all sides of triangle
    a = math.sqrt((end[0] - start[0])**2 + (end[1] - start[1])**2)
    b = math.sqrt((far[0] - start[0])**2 + (far[1] - start[1])**2)
    c = math.sqrt((end[0] - far[0])**2 + (end[1] - far[1])**2)
    s = (a+b+c)/2
    ar = math.sqrt(s*(s-a)*(s-b)*(s-c))

    #distance between point and convex hull
    d=(2*ar)/a


    # apply cosine rule here
    angle = math.acos((b**2 + c**2 - a**2)/(2*b*c)) * 57
```

Fig. 9 Heron's formula and cosine rule are applied

Comparing these different factors like Area Ratio and angles we determine how many fingers are being shown by the user. Currently, if the area ratio is less than 10 or in other words if we are showing the 'Clear' gesture by showing our palm, we will manipulate a parameter to change its value to 200. Later in program when we will check for the value of that parameter, and if it is 200 then we will perform a click in the clear button four times in a row using a for loop. Similarly, if the ratio is less than 19 and greater than 10 or in other words if we are showing the 'Done' gesture by showing thumbs up, we will manipulate a parameter to change its value to 100. Later in program when we will check for the value of that parameter, and if it is 100 then we will perform a click in the done button.

```
elif arearatio<10:

    cv2.putText(frame,'Clear',(0,50), font, 2, (0,0,255), 3, cv2.LINE_AA)
    l=200

elif arearatio<19: #15

    cv2.putText(frame,'Done',(0,50), font, 2, (0,0,255), 3, cv2.LINE_AA)
    l=100
```

Fig. 10 Logic for the clear and done gestures

A 5-second timer will run continuously i.e., once the timer reaches 0 it will start counting from 5 again. Once the timer reaches 0, at that instance it captures the current gesture that was being shown and stores the value shown in the first 5 seconds. In the next session of 5 seconds when the timer reaches 0, it will capture the gesture again in the same way.

In this 10 second slot if the two gestures are numbers, then they will be added together and will be stored in a variable called finalTerm. So, in this way we can get the outputs ranging from 0-9 using one hand and it also increases the flexibility of our project. For example, for getting an output of 7 we can use different set of gestures like (Gesture 2, Gesture 5), (Gesture 5, Gesture 2), (Gesture 4, Gesture 3), (Gesture 3, Gesture 4).

In any of the 5 second slot if the gesture shown is 'Done' or 'Clear' a click will be triggered directly to the corresponding buttons.

```
if TIMER==-1:
    TIMER=5
    if counter==0:
        if l==100:
            click(969,671)
            print("Done")
            counter=0
            continue
        if l==200:
            for k in range(4):
                click(742,663)
            print("Clear")
            counter=0
            continue
        term1=l
        print('First Term is :'+str(term1))
        counter=1
```

Fig. 11 Timer set for storing simultaneous gestures

Once we get a finalTerm after the addition of the numbers shown by the two gestures, we perform a click in a predefined location on the screen. For example, let the two gestures corresponds 2,4. After addition finalTerm will be 6. So, a click will be performed above the button corresponding to 6 in the keypad.

```
if term1 >-1 and term2 >-1:
#print('running')
    finalTerm=term1+term2
    term1=-1
    term2=-1
    print('Final Term is :'+str(finalTerm)+'\n')


    if finalTerm==1:
        click(768,359)
    elif finalTerm==2:
        click(861,337)
    elif finalTerm==3:
        click(971,343)
    elif finalTerm==4:
        click(750,451)
    elif finalTerm==5:
        click(860,449)
    elif finalTerm==6:
        click(971,451)
    elif finalTerm==7:
        click(749,555)
    elif finalTerm==8:
        click(861,556)
    elif finalTerm==9:
        click(970,557)
    elif finalTerm==0:
        click(862,670)
```

Fig. 12 Finding out the final term

In this way we can increase the range of outputs using one hand, and mapping all the combinations of gestures to the corresponding actions as per our need, in our case we mapped all the buttons of the numpad using 8 gestures.

The 8 gestures are as follows:

- 0 Gesture (Fist)
- 1 Gesture (Raising one finger)
- 2 Gesture (Raising two adjacent fingers)
- 3 Gesture (Raising three adjacent fingers)
- 4 Gesture (Raising four adjacent fingers)
- 5 Gesture (Raising five fingers)
- Done Gesture (Thumbs up)
- Clear Gesture (Upright palm and keeping fingers attached)

C. *Tools used*

1) *Anaconda*: Anaconda is a distribution of python and R languages for scientific computing. The Distribution includes data science packages suitable for Windows, Linux and MacOS.

2) *Spyder*: Spyder is an open-source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with a number of prominent packages in the scientific Python stack, including NumPy, SciPy, Matplotlib, IPython, SymPy and Cython, as well as other open-source software.

3) *OpenCV:* OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

4) *Numpy:* Numpy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

5) *Windows API:* The Windows API, informally WinAPI, is Microsoft's core set of application programming interfaces available in the Microsoft Windows operating systems. The name Windows API collectively refers to several different platform implementations that are often referred to by their own names.

6) *XAMPP:* XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible

IV. RESULT ANALYSIS

The hand gesture recognition system has been tested with hand images. In this section, we see various components of the output which contribute to the final result of the system. After acquiring the result, we map the hand gestures into specific commands. Fig. 13 shows the key components in the output of the code.

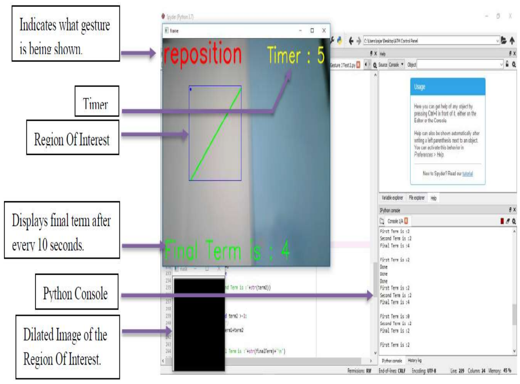

Fig. 13 Key components of output

As we can see in the output above, the frame window indicates all the information relevant to the user. The timer is shown in the top right section whereas the top left portion of the frame indicates which gesture is perceived by the system. The region of interest is the portion where the user needs to put his hand in for the system to be able to detect the gesture. The final term is shown in the bottom section of the frame. Until now our system has been able to recognise 8 gestures as stated in methodology. Fig. 14 and Fig. 15 show the gestures for ‘Done’ and ‘Clear’ respectively.

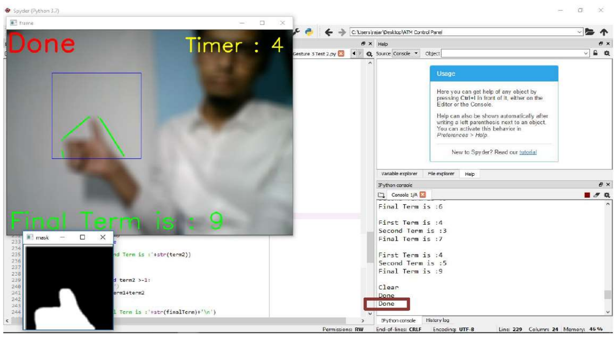

Fig. 14 Gesture 'Done' being shown

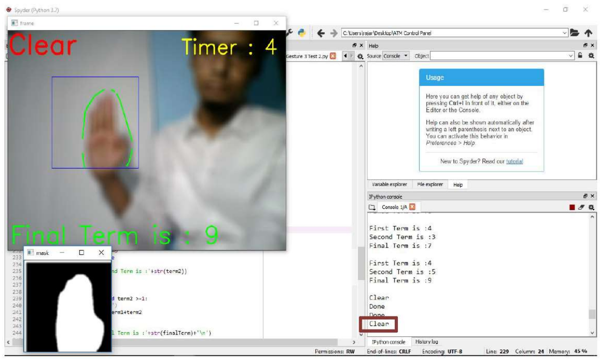

Fig. 15 Gesture 'Clear' being shown

After recognising the gestures after every 5/10 seconds, we were able to generate the corresponding command that is associated with each gesture and mapping them with all the buttons of a web page through their specific coordinates.

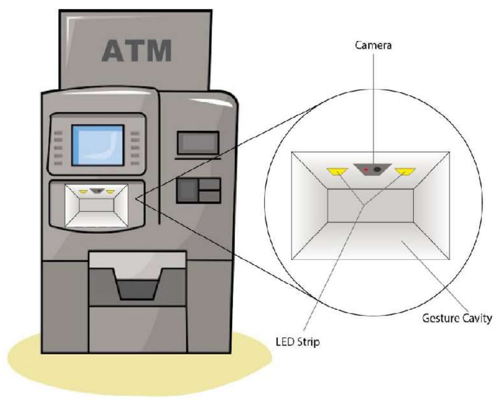

Fig. 16 Pictorial representation of practical implementation of the project

If we implement our whole project in a real time scenario, we will just need to replace the ATM keypad with a gesture cavity where a user will insert their respective hand and show the gestures as per their requirement.

In this gesture cavity three components are must and they are as follows:

- LED strips for proper illumination
- Camera
- Enough space to manoeuvre hand

V. CONCLUSION AND FUTURE SCOPE

A. *Brief summary of work*

The domain of application of vision-based hand gesture recognition system can be very large. But in this project, we are mainly focusing on the problem at hand i.e., the demand for touchless ATM keypad which arose because of the existing COVID-19 pandemic. So, the work done for this project is restricted to a single application as the cognitive mapping of the gesture to commands are specific to ATM system only. We have adopted vision-based approach for the recognition of hand gestures. At first, we analyse and manipulate the image of our hand gesture to reduce noise. Then we map those hand gestures into specific commands through various methods of image processing and extraction.

B. *Conclusion*

The purpose from any hand gesture recognition system is to recognise the hand gesture and use it to transfer a certain meaning for computer control or any other device. This vision-based approach works well under constrain situation.

This project outlines a hand gesture recognition system that is designed to be capable of recognising the hand gesture in real time. Our system does not require the use to wear any specific equipment or attach any device to the body. The hands a read by a camera instead of sensors attached to the device. Until now, we have been able to achieve natural interaction between human and computer up to a decent extent. We are able to map hand gestures onto the keypad inputs of the ATM.

C. *Future scope*

There are some aspects of projects which can be improved in future. For now, we have been able to map all the gestures necessary for an ATM transaction. But there are many areas to work on where it can be made more user-friendly. For example, currently in the worst-case scenario it takes a total of 100 seconds (20 gestures each of 5 seconds) for a transaction. We aim to improve such cases to make the system user adaptive with high penetration and low stress and fatigue.

Several real time challenges may occur during implementation of the hand gesture recognition system into the ATMs. Different surveys need to be done to generate a system which satisfies all the real time challenges.

For now, our system is developed only to be used in ATMs. We won't limit our hand gesture recognition system to just ATM. and try to implement it in various industrial machineries. We will evolve a hand gesture recognition system which is capable of switching the cognitive mapping of same gestures to different sets of commands.

Using the concept of gesture recognition, it is possible to point the finger at the computer screen so that the cursor will move accordingly. This will break the two-dimensional wall associated with the conventional mouse used today and will enable us to perform cursor operations in three dimensions. Click operation of mouse also seems possible through hand gesture recognition. These can potentially have numerous applications in the industrial, educational, medical and many other fields.

ACKNOWLEDGEMENTS

We express our deep sense of gratitude and thanks to those people without whose guidance, suggestions, insights and assistance it would not have been possible for us to carry on our project work and to prepare this report.

We would like to thank Prof. D.S Pegu (HOD), Electronics and Telecommunication Engineering Department, Assam Engineering College, for his constant guide and encouragement.

REFERENCES

[1] Pavlovic VI, Sharma R, Huang TS, "Visual interpretation of hand gestures for human–computer interaction: a review", 1997, page no. 677–695

[2] Moeslund T, Granum E, "A survey of computer vision based human motion capture. Comput Vis Image Underst", 2001, vol-81,231–268

[3] Derpanis KG, "Mean shift clustering, Lecture Notes. http://www.cse.yorku.ca/~kosta/CompVis_Notes/mean_shift.pdf" , 2005.

[4] Kanniche MB, "Gesture recognition from video sequences",PhD Thesis, University of Nice, 2009.

[5] Chaudhary A, Raheja JL, Das K, Raheja S, "intelligent approaches to interact with machines using hand gesture recognition in a natural way: a survey", 2011, Int J Comput Sci Eng Survey (IJCSES) 2(1), page no.122–133.

[6] Bourke A, O'Brien J, Lyons G, "Evaluation of a threshold-based tri-axial accelerometer falls", 2007.

[7] Detection-algorithm, "http://www.sciencedirect.com/science/article/B6T6Y-4MBCJHV1/2/f87e4f1c82f3f93a3a5692357e3fe00c"

[8] Terrillon J, Shirazi M, Fukamachi H, Akamatsu S, "Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in colour images" 2000, In: Proceedings of the international conference on automatic face and gesture recognition (FG), page no.54–61