Touchless Fingerprint Capturing

Shrey Kothavade
Master of Engineering in Information Technology
SRH Hochschule Heidelberg, Germany
11018044@stud.hochschule-heidelberg.de

Abstract:

The process in which a digital computer is used to perform an algorithm on digital images is known as "digital image processing." For fingerprint detection, Image processing is widely used in the biometric field. As the user's fingertip is contacted on a solid, flat sensor, many current fingerprint sensors capture fingerprint pictures. The input pictures from the same finger can be very different as a result of this touch, and there are latent fingerprint flaws that can result in forgery. These factors have resulted in the development of a touchless fingerprint recognition device that can take a fingerprint picture without making contact. This report is focused on the Touchless Fingerprint Capturing, which takes an image of the user's hand. This image is then converted from RGB to HSV color space. We use contours in the HSV colour space to extract the Region of Interest (ROI), and we then store the resultant distinctive Fingerprints in a database that may be retrieved later.

Keywords: Image Processing, Fingertip detection, BGR (blue, green, red), HSV (Hue, saturation, value), Skin tone Segmentation, Contours detection, Thresholding, Region of Interest.

Introduction:

In government, business, and consumer applications, the usage of fingerprints for verification and identification is widespread. Human fingerprints are now used to identify people and validate their profiles. To track employee presence throughout workdays, biometrics have been placed in practically all firms and industries. Utilizing biometrics, at such places makes the process fast. Fingerprints are essential to a person's identity and could be utilized to prevent crimes in a variety of contexts, including banking, retail, and travel.

In most of the finger-print capture systems, touch between the finger and the surface of the capturing device is necessary. These systems frequently encounter problems such as low contrast brought on by dirt or dampness on the recording device plate or latent fingerprints left behind by previous users (ghost fingerprints). For more than ten years, researchers have been looking towards touchless fingerprint recognition systems as a solution to the problems with touch-based systems. In touchless capturing systems, the finger does not make physical contact with the capturing device. [1]

Skin colour segmentation is crucial to the process of taking touchless fingerprints. Since HSV (Hue, Saturation, and Value) is better suited for picture segmentation, the colour space of the fingertips captured by a camera is often RGB or BGR.The HSV image must next be transformed into a binary image, sometimes referred to as the Threshold image. Segmentation, when the backdrop is black and the skin is white, can be used to achieve this.[2]

Now you need to sketch the outlines around the palm area. Using the binary of the previously obtained Threshold image, this may be done. The objective of this project is to construct rectangular boxes as soon as the outlines are generated.

Methodology:

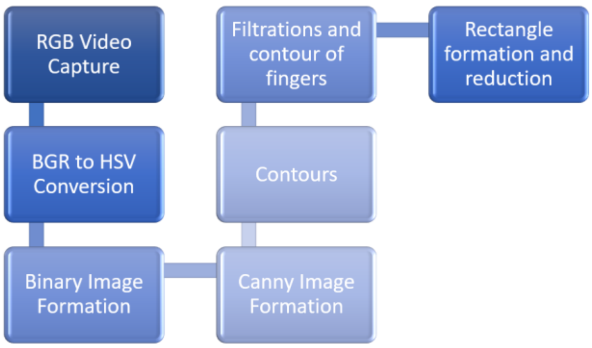

Figure 1: Block Diagram

A. BGR Video Capture

The camera has taken the picture, which is being shown in BGR (blue, green, and red) color space. This channel displays every image. The "VideoCapture" function is used in this project to capture video, and an endless loop is used to fetch or obtain photos. A video is a series of pictures that play continuously. The image container variable stores the taken picture in the BGR channel for later usage.

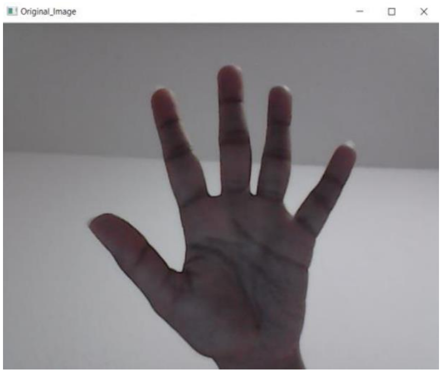

Fig 2: Video Capture

B. BGR to HSV conversion

HSV, as opposed to BGR, is the colour space that is most suited for colour segmentation. Any conversion is achievable using the OpenCv function "cvtColor". In order to segment the skin, it is changed to HSV. The human eye is incapable of distinguishing between distinct hue pixels. Color is provided by hue, saturation is represented by saturation, and value is portrayed, correspondingly, by how much colour is blended with black and white. As a result, the palm's picture stands out from the rest of the background. Segmentation is done in order to detect the palm from the complete image without any loss.

Fig 3: HSV Image

C. Binary Image Formation

It was necessary to transform the binary image in order to locate the Region of Interest (ROI). An picture in HSV colour space is transformed into a binary image using the OpenCv function "inRange". It looks that the image is in black and white. Where segmentation is identified in the picture, it is represented as white and is "1," while the remaining region is shown as black and is "0." Scalar values (0, 60, and 30) and (179, 255, 255), which correspond to the HSV low and high values, respectively, are used for skin segmentation.

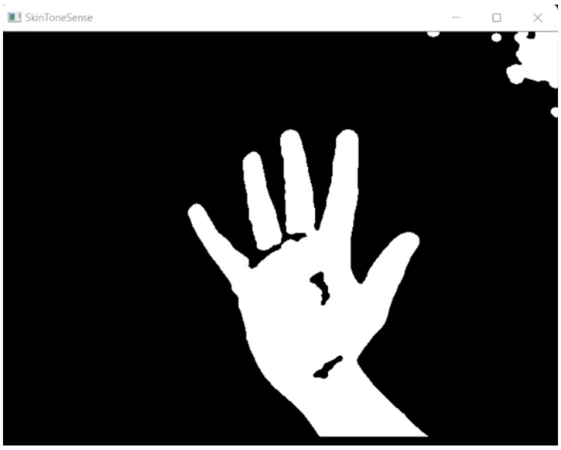

Fig 4: Binary Image

D. Threshold Binary Image Formation

The situation in this case is simple. If a pixel's value exceeds a certain threshold, it is given one value (perhaps white), otherwise it is given another value (may be white). Threshold()is the function that is utilised. [3] The source picture is the first parameter and should be a grayscale image. The second parameter is the threshold, which categorises the pixel values. The third parameter, maxVal, indicates the amount to be applied if the pixel value exceeds (or occasionally falls short of) the threshold value. Different thresholding methods are offered by OpenCV, and the fourth function argument determines which method is used.

- cv2.THRESH_BINARY
- cv2.THRESH_BINARY_INV
- cv2.THRESH_TRUNC
- cv2.THRESH_TOZERO

• cv2.THRESH_TOZERO_INV

E. Canny Image Detector

Canny edge detection is used to find edges in a picture. Low error rate, accurate localisation, and minimal response are the three major objectives that the Canny algorithm seeks to meet. To a binary picture, a Gaussian filter has been applied first, and then clever edge detection. Canny, an OpenCv function, has been employed. It is a multi-stage algorithm and its stages are as follows -

- Noise Reduction
- Finding Intensity Gradient of the Image
- Non-maximum Suppression
- Double Threshold
- Hysteresis Edge Thresholding

F. Contours

All continuous points (along the border) that are the same colour or intensity are joined by a curve known as a contour. Finding a closed form and delineating the object's boundaries is the contours' primary goal. The contours are a helpful tool for item detection and recognition as well as form analysis. The form of an item is determined by its contours. To identify and show contours, two OpenCv methods, cv.findContours () and cv.drawContours (), can be utilised.

Drawing a contour on the source picture after detecting the contour of the binary image improves accuracy. First, the source image is shown, and then finger points are found. The contours may be seen in green in this image. The contour of every object in the picture is a closed loop.

Contour filtering is required to remove tiny contours and minute interruptions while emphasising the palm and fingers of the major item that must be identified. For filtering contours, utilise the "contour area" function. In this feature, the size of the palm and fingers are adjusted to more than "500," and the area of the contour may be found. All other contours will thus only be recognised by the palm and fingers, and smaller contours will be eliminated.

Drawing a rectangle around each of the four contours or forms of the fingertip enables fingertip detection. It has been used to draw the height and breadth over four fingers using the OpenCv function "minAreaRect ()" The data is found using the OpenCv RotateRect function. If the fingertip is too close to the camera, a smaller rectangle is created at the fingers since the height is greater than the width.

Result:

The contactless fingerprint capturing has been used in this research. We shall therefore obtain the discovered fingers. The rectangular rectangles of green hue above the fingertips indicate the detected fingertips.

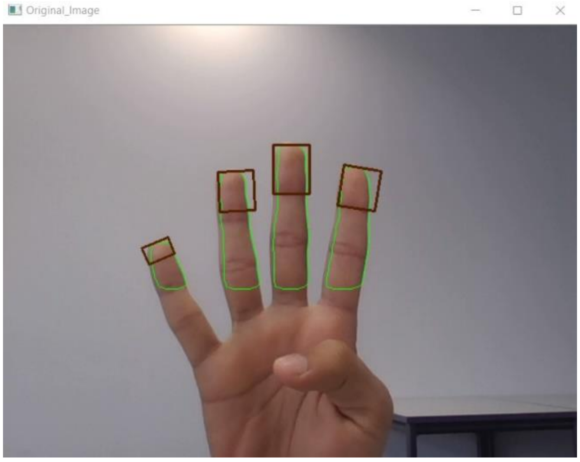

Fig 5: Fingerprint detection for 4 fingers

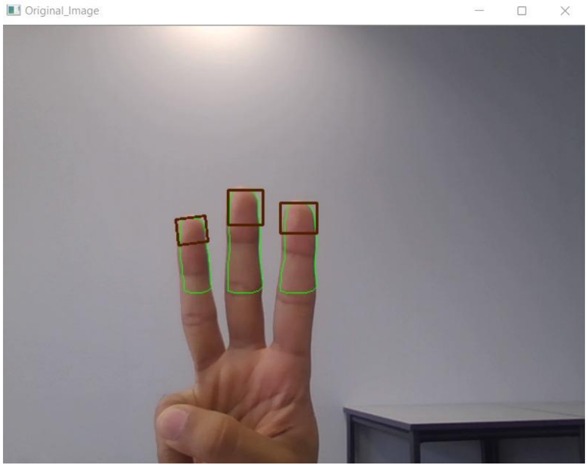

Fig 6: Fingerprint detection for 3 fingers

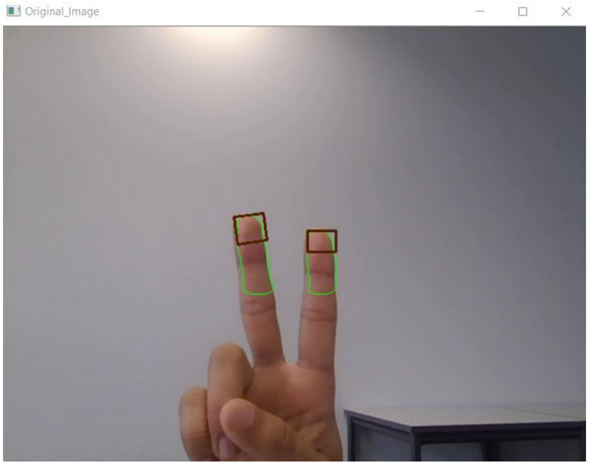

Fig 7: Fingerprint detection for 2 fingers

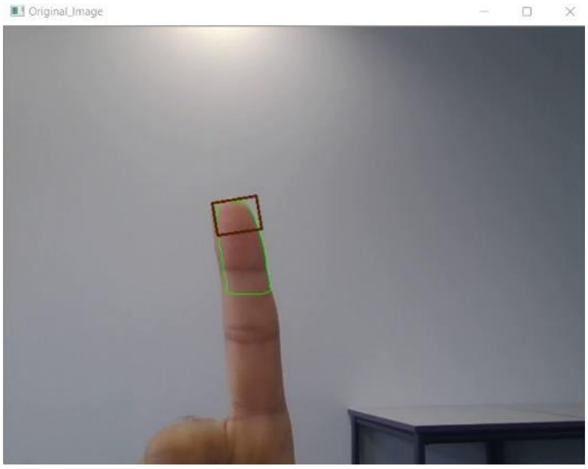

Fig 5: Fig 8: Fingerprint detection for 1 finger

Conclusion:

To obtain the fingerprint and learn about the person's profile, use the detected fingertip. High definition cameras provide faster and better detection, which in turn enhances quality. The camera is important since it will be determining effectiveness.

References:

[1] Volume V, Issue II, February 2016 IJLTEMAS ISSN 2278 – 2540 Touchless Fingerprint Recognition System by Prof. A. C. Suryawanshi1, Assistant Professor, Electronics Department, Umrer College of Engineering, Umerer, Nagpur, India. https://link.springer.com/chapter/10.1007/11815921_39

[2] Human Skin Detection Using RGB, HSV and YCbCr Color Models, S. Kolkur D. Kalbande, P. Shimpi C., Bapat J., Jatakia, Department of Computer Engineering, Thadomal Shahani Engineering College, Bandra, Mumbai, India and Department of Computer Engineering, Sardar Patel Institute of Technology, Andheri, Mumbai, India.

[3]http://opencvpython.blogspot.com/2013/05/thres hol ding.html

[4]https://www.uow.edu.au/~phung/docs/2005-TPAMI-skin segmentation.pdf

[5]https://docs.opencv.org/2.4/doc/tutorials/imgproc/shapedescriptor/find_contours/find_contours.html

[6]https://www.learnopencv.com/color-spaces- in-opencv-cpppython/

[7]https://answers.opencv.org/question/44580/how-to- resizea-contour/

[8]https://docs.opencv.org/2.4/doc/tutorials/imgproc/shapedescriptors/bounding_rotated_ellipses/bounding_rotated_ellipses.html