IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

Noble Model of Detecting Facial Parts Using Optimization techniques

Sudha Rani. J and G Nagendra
Vidya Jyothi Institute of Technology, Hyderabad, Telangana, India
sudharanijece@vjit.ac.in and nagendrag@vjit.ac.in

Abstract: *Face recognition is one of the most popular applications for automatically identifying or verifying a person. It requires high-performance image processing systems and therefore, the implementation of processing algorithms in hardware emerges as a good viable solution. Reconfigurable devices, as field programmable gate array, and system level hardware programming languages (HDL) have further accelerated the design of image processing in hardware. In this work, an algorithm for eye and mouth detection in face detection is implemented using Simulink (Math Works). This algorithm included three stages: image capture and pre-processing, face detection and eyes and mouth and nose detection. Eyes ,nose and mouth were mapped and detected considering different characteristics of chrominance and luminance of the eyes, mouth and skin. The purpose of creating an algorithm in Simulink is the posterior use of the software tool, the DSP Builder (Altera), which allows generating digital signal processing algorithms in HDL, directly from the Simulink environment. As results, the proposed Simulink algorithm was able to detect the eyes, nose and the mouth in human face images. However, factors as lighting, distance and focus, act as noise sources, affecting the algorithm performance and false regions also were detected. Iris recognition is an automated method of biometric identification that uses mathematical pattern-recognition techniques on video images of one or both of the irises of an individual's eyes, whose complex patterns are unique, stable, and can be seen from some distance. This iris scanner detects the pupil in the image of eyes using Simulink.*

Keywords: chrominance, Iris, Simulink, luminance, Biometric

I. INTRODUCTION

Face recognition is a form of biometric identification based on physiological characteristics of the face and it is totally non-intrusive. One of the main critical point in the face recognition applications is the execution time required by the image processing algorithms. The size of images necessary for recognizing and the demand for real-time operation increases the exigencies in the processing time and, therefore, the need to design efficient and high performance image processing systems.

Hardware description languages (HDLs) are system level hardware programming languages that facilitate the conception, design, analysis/simulation, documentation, and manufacturing of digital computer systems [4]. By means of HDL, a system can be described as a network of elements of varying degrees of complexity, from the system level to the logic gate level. The purpose of creating a design in Simulink was the posterior use of a software tool, the DSP Builder (Altera), which allows generating digital signal processing (DSP) algorithms in HDL directly from the Simulink environment. Thus, DSP Builder can enable rapid prototyping with Altera DSP development board.

Simulink is linked with the Matlab toolbox suite, which includes arithmetic libraries as well as powerful numerical and algebraic manipulation, thus providing Simulink with methods for efficiently integrating complex sequential algorithms into pipeline models and allowing synchronization through explicit clock connections .

In this work, an algorithm for eyes and mouth detection in face recognition is implemented using Simulink (Math Works), as a part of a system intended to extract PCA-based features of face images (PCA – principal component analysis). This work is developed into the framework of the project "Multiple Biometric Recognition System" to be developed and implemented in FPGAs by our group. The algorithm for eyes and mouth detection included a pre-processing stage in order to reduce the noise and to improve the contrast in the images. Then, the image is segmented

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

and the region with the face is detected by means of a skin detector. Finally, eyes and mouth are mapped and detected considering different characteristics of chrominance and luminance of the eyes, mouth and skin.

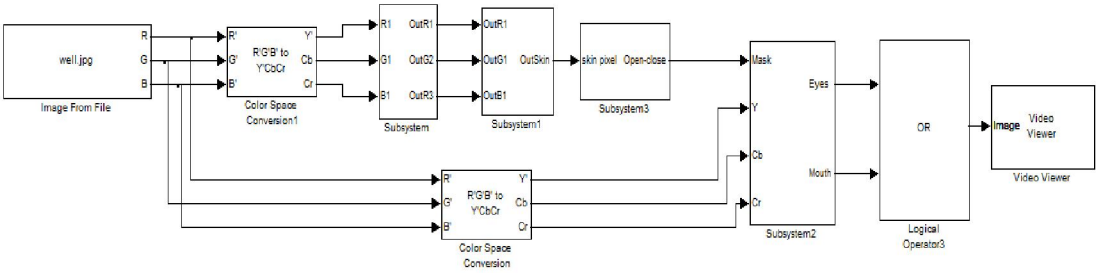


Fig. 1. General block diagram implemented in Simulink for eyes and mouth detection in face recognition

II. METHODOLOGY

This section describes the phases and main features considered in this study for the implementation of an algorithm for eyes and mouth detection in face recognition using Simulink. The algorithm is analysed considering three stages: image capture and pre-processing; face detection and; eyes and mouth detection.

2.1 Image Capture and Pre-processing

In this stage, the aim is to highlight the image features in order to make easier the detection and extraction of the image characteristics. Contrast intensification, smoothing or noise cancelation, and edge detection were included in the pre-processing stage. In this work, images were obtained from a database which was recorded in RGB format.

2.1.1 Noise cancellation

Noise can be cancelled using low-pass filters. However, these filters produce objects with unclear edges in the image. Therefore, a median filter was preferred, which change the gray level of the pixel by the median of the gray level of the neighborhood pixels.

2.1.2 Contrast intensification

It was included to normalize the varying lighting conditions between different images by adjusting the pixel values in a linear scale between 0 and 1.

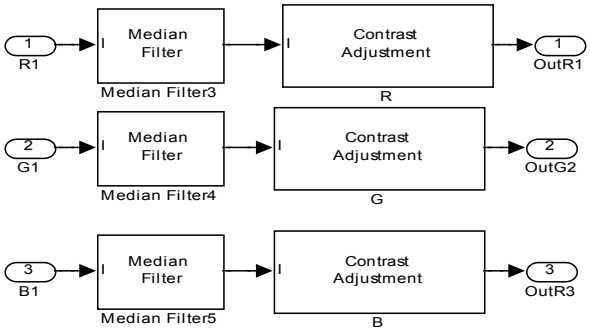


Fig. 2. Simulink blocks used to implement the noise cancellation and the contrast intensification

2.2 Face Detection

Images were segmented in this stage following a homogeneity criterion. In this work, techniques based on the skin color were used. However, it is possible that image regions, different from face but with similar color as skin, can be classified as face skin. In order to avoid that, it was necessary to decide which part of the images has face features, e.g. the eyes and the mouth, as reference points.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

2.2.1 Skin Pixel Detection

The RGB colour space was used to facilitate the chrominance measurement without dependency of the image luminance. Although it is not the best method, it offers acceptable results with a simple implementation. A pixel was considered as human skin if it satisfies some fixed conditions. The followings were the parameters used in this work, considering that pixels are in the range [0,255] for each RGB space. Consider the following operations and execute them

$$\text{Max}\{R,G,B\}-\text{Min}\{R,G,B\}>15 \quad \text{and} \quad |R-G|>15 \text{ , } R>B \text{ and } R>G$$

where R, G and B are red, green and blue colours, respectively. *max* and *min* represent the maximum and minimum functions. This procedure generates a binary image where "1" represents pixels fulfilling all the conditions and "0" the other ones.

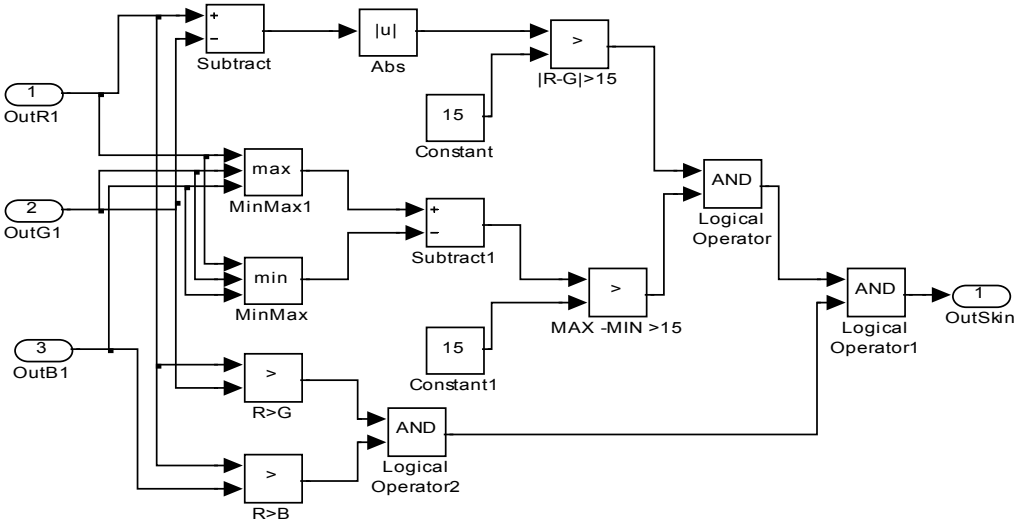


Fig. 3. Part of the Simulink blocks used to detect the skin pixels

2.2.2 Process of Filtering and Grouping

The next step was to choose which regions could be part of the face and to eliminate the rest. This function was based on binary morphology. The first operation is an opening of the image, which applies erosion followed by dilation. The erosion is used to remove small groups of pixels with a minimum probability to be part of the face. The dilation allows components that were not eliminated to be reconstructed by adding points in the neighbourhood of the object edges. The second operation is a closing of the image, which applies dilation followed by erosion, which is implemented in order to close holes generated during the image opening.

A disk was selected as the structuring element during the implementation of morphological operations, which allows isotropic behaviour, necessary when the face orientation is unknown. Several tests, both during the opening and the closing operations, were applied in order to determine the disk radius with the best performance. The size was fixed in relation to the percentage occupied by the face into the image.

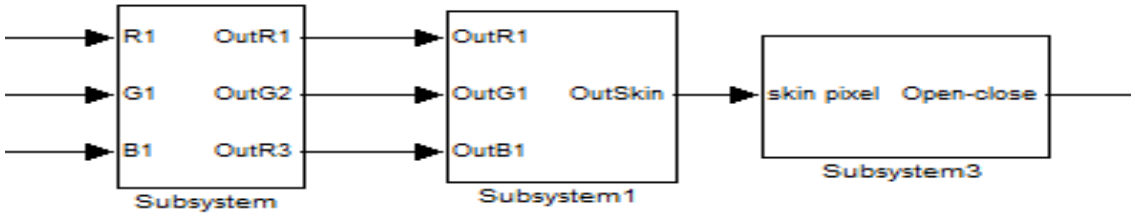


Fig. 4. Simulink blocks used during the "process of filtering and grouping"

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

2.3 Eyes and Mouth Detection

2.3.1 Eyes Map:

It was based on two concepts: low red intensity (low Cr) and high blue intensity (high Cb) values are found around the eyes; eyes show both very light pixels and very dark pixels. Therefore, two eye-maps were computed: one uses the chrominance information, while the other uses the luminance component of the pixels.

Luminance map was generated using morphological operators as dilation and erosion, highlighting the bright and the dark pixels in the luminance component.

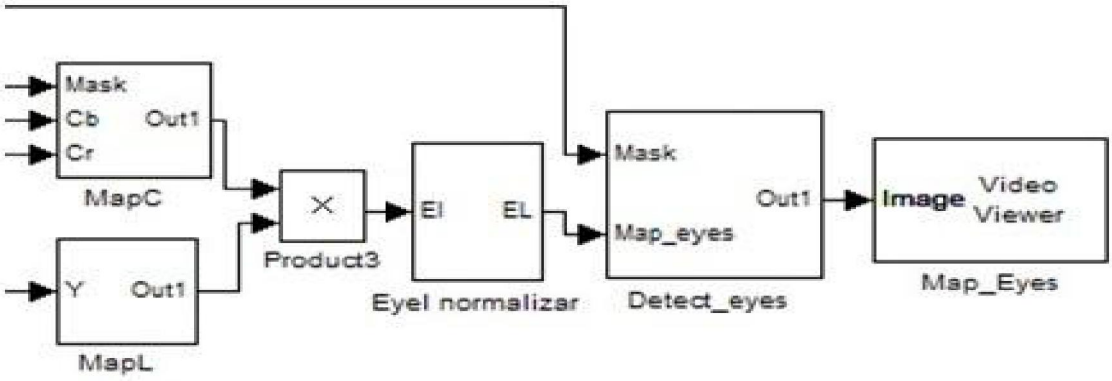


Figure 5: Simulink model for Eyes detection

Chrominance and luminance maps were normalized in the range [0,255] before being combined using the OR operation. The result was masked and normalized to the range [0,255]. Eyes regions were detected by applying a threshold, giving as result a binary image where pixels higher than the threshold are assigned a value "1" and the rest of the pixels are assigned to "0". Then, regions without minimum size were eliminated using a morphological filter, which applies opening and closing of the binary image.

IRIS Detection Methodology

a. Image from file
b. Color space conversion
c. Median filter
d. Autothreshold
e. Blob Analysis
f. Draw Shapes
g. Video Viewer

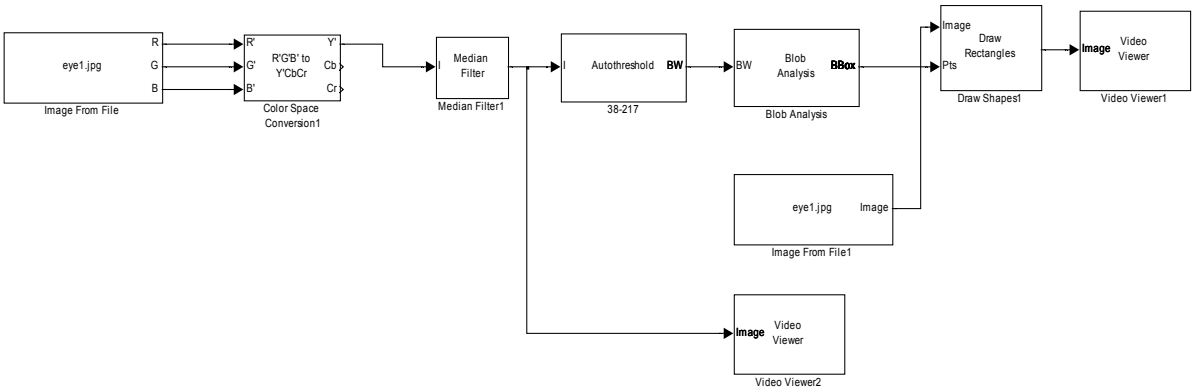


Fig.6 Block diagram for iris detection using Simulink

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

a. Image from File

We use the Image from File block to import an image from a supported image file. If the image is a M-by-N array, the block outputs a binary or intensity image, where M and N are the number of rows and columns in the image. If the image is a M-by-N-by-P array, the block outputs a color image, where M and N are the number of rows and columns in each color plane, P.

For the Video and Image Processing Blockset blocks to display video data properly, double- and single-precision floating-point pixel values must be between 0 and 1. If the input pixel values have a different data type than the one you select using the Output data type parameter, the block scales the pixel values, adds an offset to the pixel values so that they are within the dynamic range of their new data type, or both.

b. Color Space Conversion:

The Color Space Conversion block converts color information between color spaces. Use the Conversion parameter to specify the color spaces you are converting between. Choose R'G'B' to Y'CbCr.

a. Median Filter

The Median Filter block replaces the central value of an M-by-N neighbourhood with its median value. If the neighbourhood has a center element, the block places the median value there.

The block pads the edge of the input image so, pixels within [M/2 N/2] of the edges may appear distorted. Because the median value is less sensitive than the mean to extreme values, the Median Filter block can remove salt and pepper noise from an image without significantly reducing the sharpness of the image.

d. Autothreshold

The Autothreshold block converts an intensity image to a binary image using a threshold value computed using Otsu's method.

This block computes this threshold value by splitting the histogram of the input image such that the variance of each pixel group is minimized.

Use the Thresholding operator parameter to specify the condition the block places on the input values. If you select > and the input value is greater than the threshold value, the block outputs 1 at the BW port; otherwise, it outputs 0. If you select <= and the input value is less than or equal to the threshold value, the block outputs 1; otherwise, it outputs 0.

Select the Output threshold check box to output the calculated threshold values at the Th port.

Select the Output effectiveness metric check box to output values that represent the effectiveness of the thresholding at the EMetric port. This metric ranges from 0 to 1. If every pixel has the same value, the effectiveness metric is 0. If the image has two pixel values or the histogram of the image pixels is symmetric, the effectiveness metric is 1.

If you clear the Specify data range check box, the block assumes that floating-point input values range from 0 to 1. To specify a different data range, select this check box. The Minimum value of input and Maximum value of input parameters appear in the dialog box. Use these parameters to enter the minimum and maximum values of your input signal.

Use the When data range is exceeded parameter to specify the block's behavior when the input values are outside the expected range. The following options are available:

Ignore — Proceed with the computation and do not issue a warning message. If you choose this option, the block performs the most efficient computation. However, if the input values exceed the expected range, the block produces incorrect results.

Saturate — Change any input values outside the range to the minimum or maximum value of the range and proceed with the computation.

Warn and saturate — Display a warning message in the MATLAB Command Window, saturate values, and proceed with the computation.

Error — Display an error dialog box and terminate the simulation.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

e. BLOB Analysis:

Use the Blob Analysis block to calculate statistics for labeled regions in a binary image. The block returns quantities such as the Centroid, label matrix, and blob count.

The Blob Analysis block supports variable size signals at the input and output.

For information about how pixel and spatial coordinate systems are defined in the Video and Image Processing Blockset documentation, see Coordinate Systems in the Video and Image Processing Blockset software User's Guide.

Use the Variable Selector block to select certain blobs based on their statistics. For more information about this block, see the Variable Selector block reference page in the Signal Processing Blockset documentation.

f. DRAW SHAPES:

The Draw Shapes block draws multiple rectangles, lines, polygons, or circles on images by overwriting pixel values. As a result, the shapes are embedded on the output image.

This block uses Bresenham's line drawing algorithm to draw lines, polygons, and rectangles. It uses Bresenham's circle drawing algorithm to draw circles.

g. VIDEO VIEWER:

The Video and Image Processing Blockset product extends Simulink software with a rich, customizable framework for the rapid design, simulation, implementation, and verification of video and image processing algorithms and systems. It includes basic primitives and advanced algorithms for designing embedded imaging systems in a wide range of applications in aerospace and defense, automotive, communications, consumer electronics, education, and medical electronics industries.

2.3.2 Mouth map

Mouth is characterized by a stronger red component and a weaker blue component than the other facial regions.Mouth regions were detected in a similar way as detecting eye regions: a threshold is applied to obtain a binary image (pixels higher than the threshold are assigned a value “1” and the rest of the pixels are assigned to “0”); a morphological filtering of the binary image is applied in order to remove regions without minimum size and; region detected as mouth is labelled as shown in figure 11. Finally, eyes map and mouth map were combined.

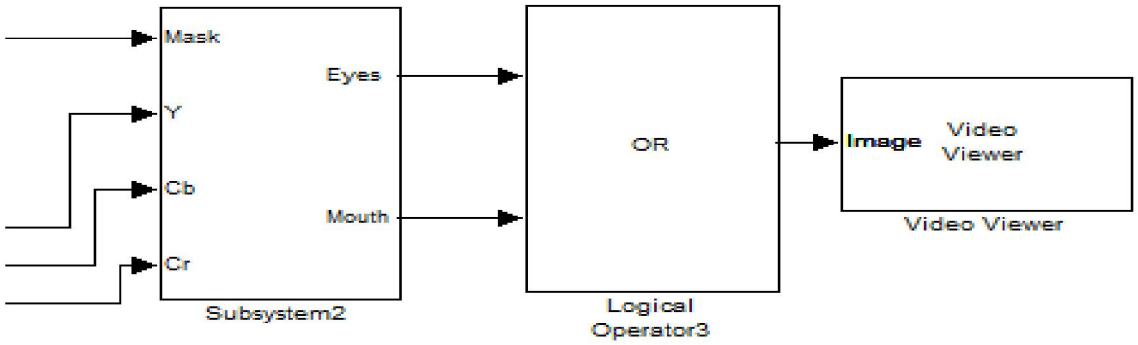


Fig. 7. Combination of chrominance and luminance maps

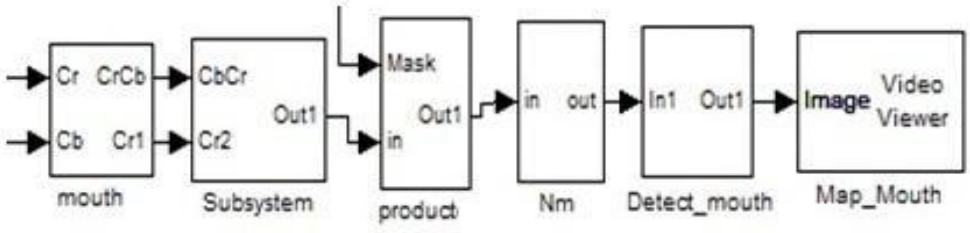


Fig. 8. Simulink blocks used to create the mouth map

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

III. RESULTS

In order to limit the length of this manuscript, only few Simulink blocks are described.. The first stage (image capture and preprocessing) was implemented with "Capture" and "preprocessing" blocks, while the second stage (face detection) with "skin detection" and "filtering and grouping" blocks. "RGB to YCbCr", "eyes map and mouth map", "combination" and "Eyes and mouth" blocks were included in third stage (eyes and mouth detection).

3.1 Image Pre-processing

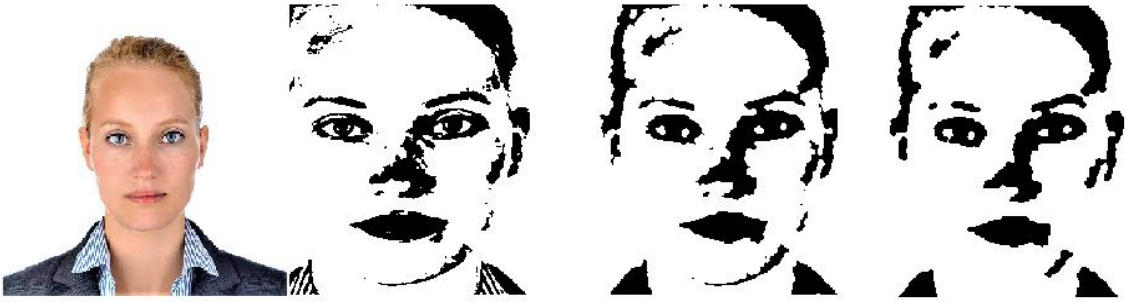
The Simulink blocks used to implement the noise cancellation and the contrast intensification. First, a median filter was applied to each color component (R,G, and B). This filter was implemented using the Simulink block "Median Filter" (in the library "Video and Image Processing Blockset"– Filtering category), which allows the neighborhood matrix size (Neighborhood size) and others parameters to be changed. Then, the contrast intensification was implemented using the Simulink block "Contrast Adjustment" (in the library "Video and Image Processing Blockset"

3.2 Face Detection

Skin pixel detection was implemented using basic Simulink blocks, while the process of filtering and grouping was implemented using the Opening and Closing blocks (in the library "Video and Image Processing Blockset" - Morphological Operations category), the Simulink blocks are used to implement equations during the skin detection. Some images obtained during the face detection

3.3 Eyes, Nose and Mouth Detection

Chrominance and luminance maps for eye detection were implemented in the Simulink blocks "MapC" and "MapL". The regions detected as eyes nose and mouth are shown . It is observed that the algorithm was able to detect the eyes. Simulink blocks used to create the mouth map are shown in Fig. 8 the algorithm correctly detected the mouth in the human face. Finally, the whole algorithm for eyes and mouth detection was applied to a face image.



(a) (b) (c) (d)

Fig. 9. Images obtained during the face detection: a) pre-processed image; b) binary image after applying the skin detection algorithm; c) binary image after the opening operation with structuring element of disk radius 4; d) binary image after the closing operation with structuring element of disk radius 4.

(a) (b)

Fig.10 (a) pre-processed image (b)image after applying colour conversion and median filter

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

Fig.11. detection of eyes, nose and mouth

IV. CONCLUSION AND FUTURE SCOPE

In this work, an algorithm for eye,nose and mouth detection in face recognition is implemented using Simulink (MathWorks). This is a preliminary study as part of the research project "Multiple Biometric Recognition System", which will be developed and implemented in FPGAs. Simulink is a versatile tool integrated with the Matlab toolbox suite, which is very useful in the development of complex sequential algorithms. Besides, Simulink can interact with the DSP Builder software tool (Altera) in order to generate digital signal processing algorithms in hardware description language directly from the Simulink environment.

The implemented algorithm was able to detect the eye and the mouth in human face images. However, different factors as lighting conditions, distance and focus of the camera, etc., act as noise sources affecting the algorithm performance, which could detect some regions as false eyes or as false mouth. Therefore, it is necessary to control the environmental conditions during the image capture, ensuring adequate lighting and uniformity in the distance and the focus for objects that are different from the face. Also, it is convenient to implement additional rules for avoiding to include false regions during the eye and mouth detection.

This work will be validated in a future study by using an extended database in order to quantify the performance of the algorithm during the detection of the eye and the mouth in a human face. This validation will include, among others, images with different resolutions and controlled environmental conditions.

REFERENCES

[1]. D. Bhattacharyya, R. Ranjan, F. Alisherov, and M. Choi, “Biometric authentication: a review,” International Journal of u- and e- Service, Science and Technology, vol. 2, pp. 13-28, 2009.

[2]. R. Jafri, and H. Arabnia, “A survey of face recognition techniques,” Journal of Information Processing Systems, vol. 5, pp. 41-68, 2009.

[3]. Hardware Description Languages,Sarah L. Harris, David Money Harris, in Digital Design and Computer Architecture, 2016

[4]. Altera Corporation, "DSP Builder Handbook," San Jose: Altera, vol. 2, 2013.

[5]. M. A. González, P. N. Posso, E. Sarria, J. C. Cruz, J. F. Valencia”Algorithm for eyes and mouth detection in face recognition using Simulink”

[6]. D. V. Rao, S. Patil, N. A. Babu, and V. Muthukumar, “Implementation and evaluation of image processing algorithms on reconfigurable architecture using C-based hardware descriptive languages,” International Journal of Theoretical and Applied Computer Sciences, vol. 1, pp. 9-34, 2006.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

[7]. R. Waxman, and M. Israel, “Hardware description languages,” Encyclopedia of Computer Science, pp. 768-773, 2003.

[8]. Amir, L. Zimet, A. Sangiovanni-Vincentelli, and S. Kao, "An embedded system for an eye-detection sensor," Computer Vision and Image Understanding, vol. 98, pp. 104-123, 2005.

[9]. J. Kovac, P. Peer, and F. Solina, "Human skin colour clustering for face detection," EUROCON 2003. Computer as a Tool. The IEEE Region 8, vol.2, pp. 144-148, September 2003.

[10]. C. Harshith, S. Karthik, R. Manoj, S. M.V.V.N.S, and L. Naveen, "Survey on various gesture recognition techniques for interfacing machines based on ambient intelligence," International Journal of Computer Science & Engineering Survey (IJCSES), vol. 1, pp. 31-42, 2010.

[11]. H. Rahman, and J. Afrin, "Human face detection in color images with complex background using triangular approach," Global Journal of Computer Science and Technology, vol. 13, pp. 45-5