SEMI-AUTOMATED EXTRACTION OF PERINASAL REGION OF INTEREST (ROI) FROM THREMAL FACIAL IMAGES

by

Nourah Fheed F Alqahtani, BTech

THESIS

Presented to the Faculty of

The University of Houston-Clear Lake

In Partial Fulfillment

Of the Requirements

For the Degree

MASTER OF SCIENCE

in Computer Science

THE UNIVERSITY OF HOUSTON-CLEAR LAKE

DECEMBER, 2018

SEMI-AUTOMATED EXTRACTION OF PERINASAL REGION OF INTEREST (ROI) FROM THREMAL FACIAL IMAGES

by

Nourah Fheed F Alqahtani

APPROVED BY

Pradeep Buddharaju, Ph.D., Chair

Hisham Al-Mubaid, Ph.D., Committee Member

Alfredo Perez-Davilla, Ph.D., Committee Member

APPROVED/RECEIVED BY THE COLLEGE OF SCIENCE AND ENGINEERING

Said Bettayeb, Ph.D., Associate Dean

Ju H. Kim, Ph.D., Dean

Dedication

To my beloved parents, who have always been my role in life to accomplish the best and never give up.

To my husband Badr and my lovely son Humoud for their unconditional love and support.

Acknowledgements

I would like to express my special thanks and gratitude to my advisor Dr. Pradeep V Buddharaju, for his monumental support and guidance the entire time to complete this work. It has been an honor and pleasure working under his supervision and benefiting from his experience and knowledge.

Sincere gratitude is extended to the distinguished committee members, Dr. Hisham Al-Mubaid and Dr. Alfredo Perez-Davilla for being in my committee and providing their constructive feedback after reviewing my thesis document.

ABSTRACT

SEMI-AUTOMATED EXTRACTION OF PERINASAL REGION OF INTEREST (ROI) FROM THREMAL FACIAL IMAGES

Nourah Fheed F Alqahtani
University of Houston-Clear Lake, 2018

Thesis Chair: Pradeep Buddharaju, Ph.D.

S-Interface is a C# WinForms software developed by Computational Physiology Lab (CPL) at University of Houston for studying physiological variables such as perinasal perspiration, breathing rate etc. These physiological variables, particularly perspiration, account for the amount of stress faced by a human being in strenuous situations. The noticeable and frequent area of perspiration of a human body is the perinasal (around mustache) region. The accuracy of physiological algorithms running in the current S-Interface software depends on the operator selection of the measurement region of interest (MROI), which may vary from one operator to another. The main goal of this thesis is to semi-automate the MROI detection process by requiring the operator to click on the nose tip and then automatically extracting the perinasal region consistently.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I: INTRODUCTION

1.1.Introduction

Stress is one of the mechanisms the human body uses to cope with any threats or challenges, and it has a strong effect on human health. Various methods have been used in the process of stress assessment. The measurement of emotional and physiological variables, stress included, has gained a lot of interest in the past few years.

Thermal imaging is one of the most promising non-invasive methods probing the physiological status of human [1]. Stress is usually expressed by the body in the form of perspiration, and one of the noticeable and frequent areas of perspiration of a human body is the perinasal (around mustache) region. S-Interface is one of the systems that use this particular region to extract a perspiration signal as part of its physiological studies [2].

S-Interface (previously OTACS) is a C# WinForms software developed by Computational Physiology Lab (CPL) of University of Houston – Main Campus. CPL developed the S-Interface software to measure physiological variables, such as perinasal perspiration and breathing rate using thermal imaging. These physiological variables account for the amount of stress faced by a human being in strenuous situations. The main goal of CPL's research is contact-free detection of stress in its early stages of manifestation so that necessary steps can be taken to intervene and ameliorate its effects.

1.2.Current Challenges

Current S-Interface software requires the operator to manually select the measurement region of interest before running its physiological algorithms. Hence the accuracy of perspiration measurement depends on the operator's selection of the ROI in the perinasal region.

1.3.Contribution

In this thesis, I developed an algorithm and implemented a module in the .NET C# WinForms that extracts the perinasal area from facial thermal images. The implementation will be used by the Computational Physiology Lab (CPL) of University of Houston – Main Campus as a plug-in in the S-interface software. The main purpose of the plug-in is to semi-automate the process of extracting a perinasal Region of Interest (ROI) substituting the current manual approach currently used. An operator is required to click on the nose tip only, rather than selecting the region entirely. After the mouse click, the desired region is automatically extracted using the proposed algorithm. The experimental results show that our algorithm extracts the perinasal ROI consistently with high accuracy as long as the user clicks somewhere on the nose tip.

CHAPTER II: LITERATURE REVIEW

2.1. Thermal Images

2.1.1. Facial thermography

Determining the temperature distribution on an object surface using thermal imaging is referenced as thermography [3]. In recent years, the studies on thermal imagery gained a lot of attention. Different studies, using facial thermal imagery in particular, are reported in the literature [4-7]. The advantages that the facial thermal imagery have over the visual facial images are that the facial temperatures captured by the thermal cameras are not affected by the color and lighting conditions [8]. In addition, this type of images gives insights on the emotional and physiological levels, which can be easily concealed in visual images facial expression.

2.1.2. Stress measurement using facial thermal images

The human body expresses stressful emotions in different ways such as rapid or shortness of breathing, temperature fluctuations, or sudden manifestation of perspiration. The forehead, cheeks, nose, and perinasal (around the mustache) regions of the human face are some of the areas that can be used to detect different emotional and physiological variables. Studying the facial temperature variant distribution on thermal images provides a noninvasive approach of stress detection and measurement [9]. The perspiration in the perinasal area is considered a physiological variable that accounts for the amount of stress faced by a human being in tensed situations [6, 10].

2.2. Computational Physiology Lab

The Computational Physiology Lab (CPL) is a lab established by Dr.Loannis Pavlidis in 2002. The lab’s projects have been mainly focusing on sustainably monitoring different physiological variables. One of its featured projects is the S-Interface system [11].

2.2.1. S-Interface

The S-interface, a modular system in C#, consists of multiple measurement algorithms. The algorithms use thermal images to measure different physiological variables such as perspiration and breathing rate. These variables reveal the extent of stress the human body encounters during strenuous situations. The system consists of multiple modules, called plug-ins, where each module is focusing on a specific functionality. Measuring the perspiration using thermal facial images is achieved using a perspiration signal extraction algorithm plug-in [2].

Perspiration signal extraction algorithm

This plug-in main functionality is extracting perspiration signals at a distance from facial thermal imagery that is equivalent to EDA signals [12], working collaboratively with the tissue tracking plug-in reported in [13]. The plug-in extracts an energy signal E(k;j) to quantify the spatial frequency pattern resulting from the appearance of small ‘cold’ (dark) spots amongst substantial background clutter. The signal E(k;j) represents the perspiration activity in the upper orbicularis oris of the perinasal area of subject k, for experimental session j [12].

Currently the algorithm was tested extensively in the following two applications:

- **Contact free measurement of surgeon dexterity and stress [6].**

  In this study CPL has monitored stress and performance patterns among surgeons during training in an inanimate laparoscopic skills lab (Figures 1 and 2). Traditionally galvanic skin response (GSR) sensing on the fingers has been the standard method used to quantify stress in real-time. However, this method is not applicable in surgical training assessment for obvious reasons; the surgeons' fingers are engaged, a limitation that would apply to all dexterous task scenarios. To solve the problem, they have developed a novel stress quantification methodology where the targeted physiological response is transient perspiration on the perinasal area.

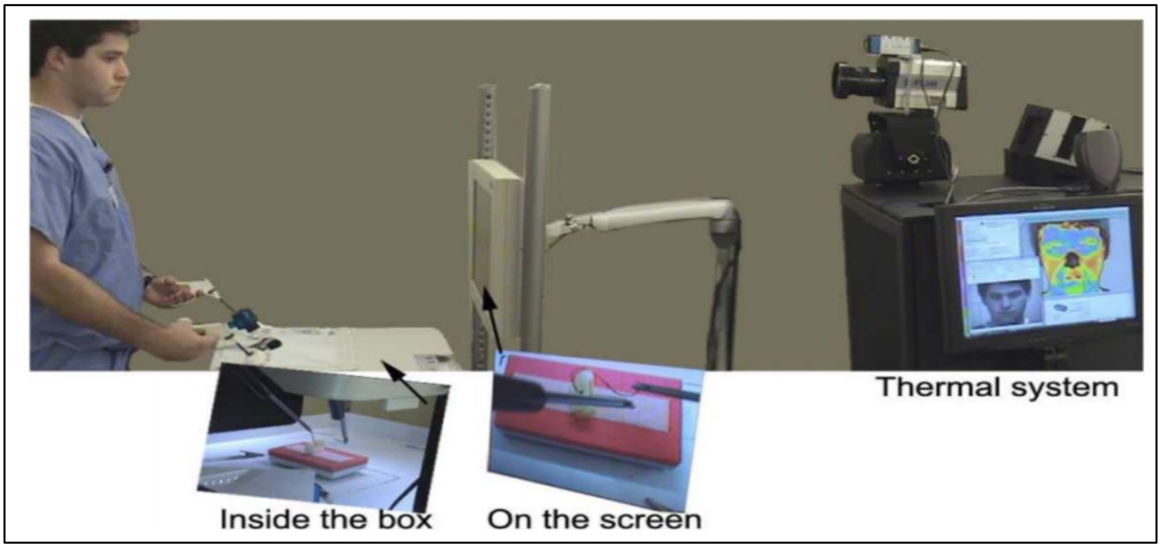


Figure 1
Experiment lab setup in [6]

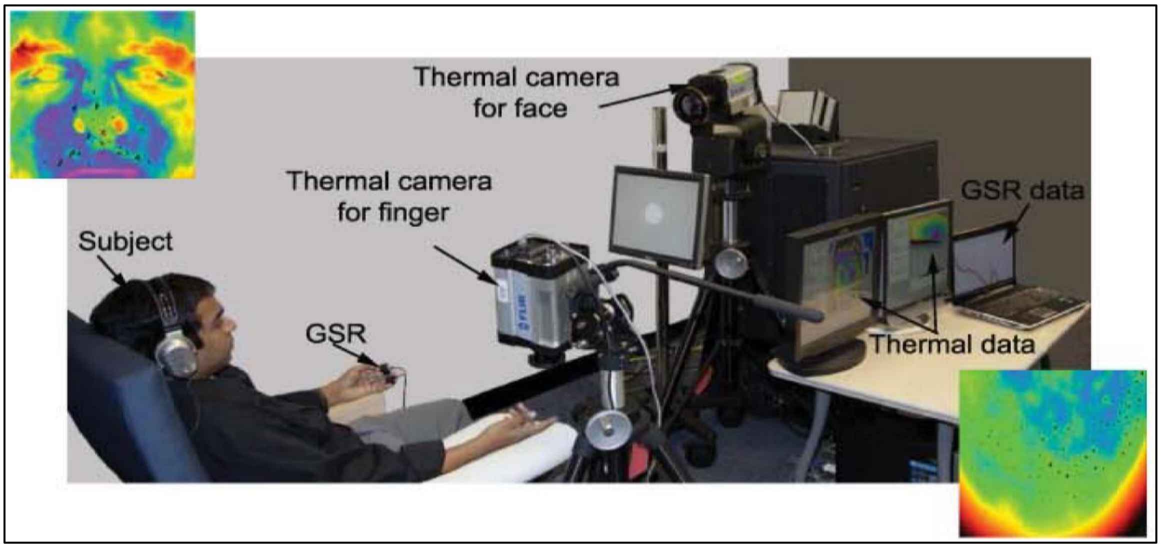


Figure 2
Experiment lab setup in [6]

- **Contact free measurement of driver distraction and stress [10].**

  In this study CPL reported results on a driving simulator experiment where subjects operated under normal and stressful conditions (Figure 3); the stressful conditions featured four types of stressors - cognitive, emotional, sensorimotor, and mixed. They delivered the cognitive and emotional stressors through appropriate oral questionnaires, and the sensorimotor stressor in the form of texting while driving. The found from the results of this study that all stressors incurred significant increases in mean sympathetic arousal accompanied by significant increases in mean absolute steering.

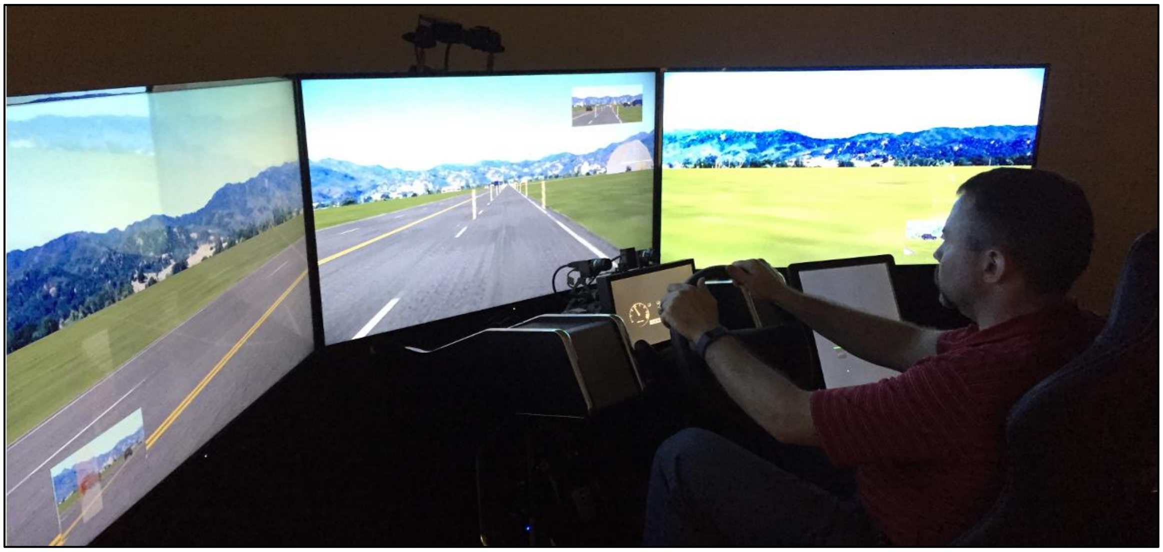


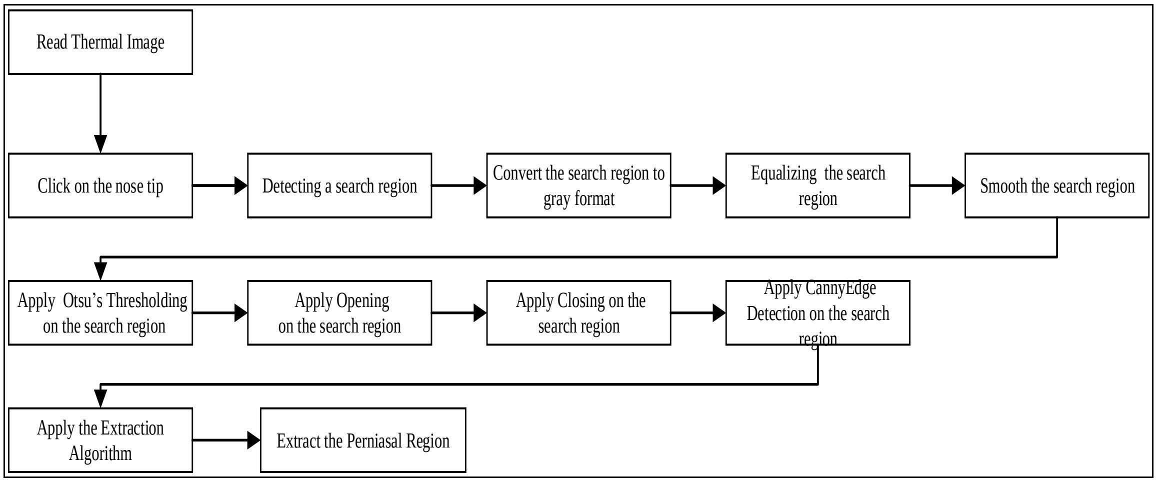
Figure 3
Experiment lab setup in [10]

2.2.2. Perinasal region extraction from facial thermal images

Automated extraction of the perinasal region from thermal images has not yet been reported in the literature to the best of my knowledge. The studies in [6, 7, 10]are based on manually extracting the perinasal region.

CHAPTER III: METHODOLOGY

The main objective of the developed algorithm is to semi- automate the process of extracting the Measurement Region of Interest ROI. The current S-Interface software requires the operator to manually select the measurement region of interest (MROI) before running its physiological algorithms. Hence the accuracy of perspiration measurement depends on the operator's selection of the correct ROI in the perinasal region. Minimizing the operator involvement to only click on the nose tip, and then automatically detecting the perinasal region, eliminates any error during the region selection. Since the S-Interface supports only C# Winform plugins, the plug-in is implemented in C#. The algorithm flow of the proposed perinasal ROI extraction algorithm is shown in Figure 4.



Read Thermal Image
Click on the nose tip
Detecting a search region
Convert the search region to gray format
Equalizing the search region
Smooth the search region
Apply Otsu's Thresholding on the search region
Apply Opening on the search region
Apply Closing on the search region
Apply CannyEdge Detection on the search region
Apply the Extraction Algorithm
Extract the Perniasal Region

Figure 4
ROI extraction algorithm work flow

3.1. ROI Extraction Algorithm

The algorithm consists of three main phases: image acquisition, image processing, and perinasal region extraction. The Open Source Computer Vision (OpenCV) library provides an optimized version of several image processing algorithms used in the proposed algorithm. As the plug-in is developed in C# Windows forms in .Net framework, the OpenCV functionality is used through EmguCV. EmguCv is a cross-platform .Net wrapper to the OpenCV enabling OpenCV functions to be called during the image processing phases.

3.1.1. Image acquisition

The dataset is collected using a Tau 640 long-wave infrared (LWIR) camera (FLIR Commercial Systems, Goleta, CA); it features a small size (44×44×30 mm) and adequate thermal (<50 mK) and spatial resolution (640×512 pixels). The Tau 640 camera was outfitted with a LWIR 35 mm lens f/1.2. Thermal data was collected at a frame rate of 7.5 fps. A color mapping is applied to the temperature data extracted from the thermal images for visualization purposes upon which thermal image processing can be treated similarly to traditional visual image processing.

3.1.2 Image processing

The main goal of image pre-processing is to prepare a search region for further processing steps. After reading the thermal image, a click on the nose tip is required. Then, a search region around the mouse click is marked (140 pixels width, 50 pixels height), Where the mouse click location is centered in the search region. The search region includes the lower part of the nose which will be used to detect the nose tip.

3.1.2.1 Image pre-processing and enhancement

The selected region is pre-processed by eliminating the unwanted data and enhancing the region intensity and quality. EmguCV library provides multiple functions for image pre-processing and enhancement such as color conversion, contrast enhancement, and noise reduction.

A. GrayScale conversion

The acquired facial images are in RGB format. Converting the search region format into gray scale is to prepare the region for edge detection in the following stages. The EmguCV Convert function (more details in Appendix A) is used to generate the grayscale image as shown in Figure 6 from Figure 5.

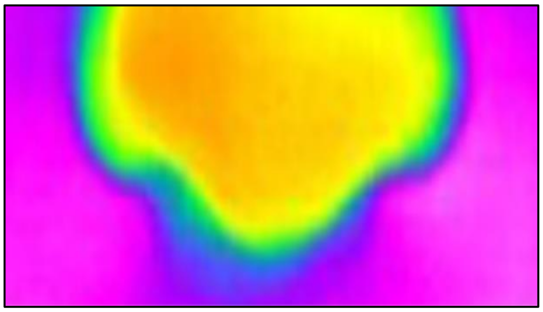


Figure 5
Search region in RGB format

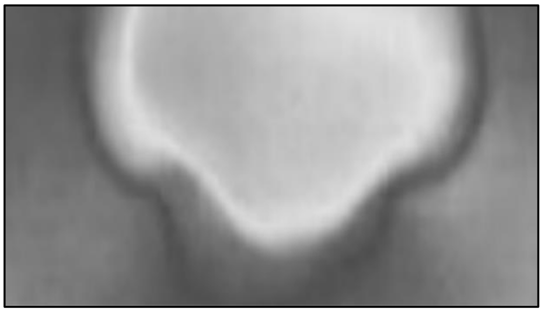


Figure 6
Search region in grayscale format

B. Contrast inhancement

After converting the nose region into grayscale, the contrast, sharpness, and structure of the region may be compromised. Applying an equalization enhancement technique improves the gradient variation in the area. The following formulae represent the equalization process:

Equalization $s = T(r) = (L-1)\int_0^r p_r(w)dw$

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

First the probability mass function is calculated of all the pixels for the search region then the cumulative distributive function is computed. Where the L is the number of gray levels in the image. Figure 7 is the result of applying equalization on the previous Figure 6.

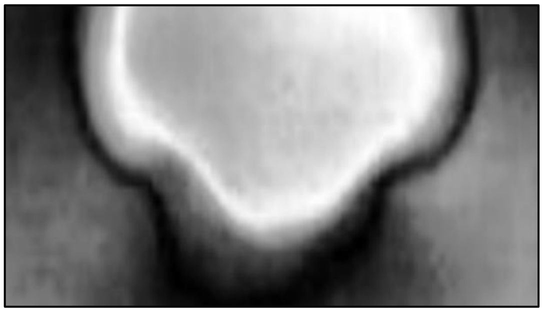


Figure 7
Search region after applying equalization

C. Noise reduction

Smoothing filtering is used in this step to reduce the noise present in the search region. The EmguCv Gaussian Blurring function (more details in Appendix A) is applied with a 5× 5 kernel size. Figure 8 is the result of applying a gaussian blur on the previous Figure 7.

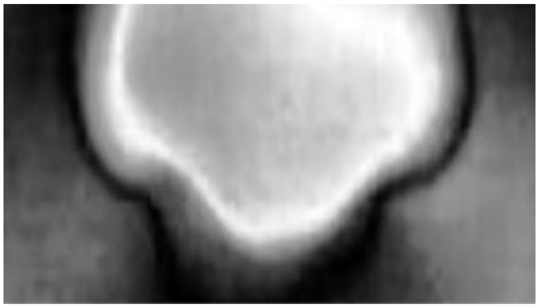


Figure 8
Search region after applying gaussian blur

3.1.2.2. Nose segmentation

Segmentation process uses pixel-level or object-level properties of an object to demarcating an object on an image. Texture, edges and pixel intensity variation, are the properties that can be used in the segmentation process [3]. Decomposing an image into regions for further analysis and performing a change of the representation of an image for faster analysis are the two main goals of image segmentation [14]. To detect the nose tip effectively a combination of segmentation techniques is used. Thresholding segmentation and edge detection segmentation are the techniques used for nose segmentation.

A. Thresholding segmentation

This technique uses a grayscale image to output a binary image. The binary image marks the nose shape on the search region. Many different thresholding algorithms are available, and for the optimal threshold value, Otsu's thresholding method [15] is used to segment the nose area from the background using the following formulae:

$$\sigma_{\omega}^{2}(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

Where:

$$q_1(t) = \sum_{i=1}^{t} P(i) \quad \& \quad q_2(t) = \sum_{i=t+1}^{I} P(i)$$

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)} \quad \& \quad \mu_2(t) = \sum_{i=t+1}^{I} \frac{iP(i)}{q_2(t)}$$

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{iP(i)}{q_1(t)} \quad \& \quad \sigma_2^2(t) = \sum_{i=t+1}^{I} [i - \mu_2(t)]^2 \frac{iP(i)}{q_2(t)}$$

Due to the effects of breathing, i.e. inhaling oxygen (cold air) and exhaling carbon dioxide (hot air), the nose always appears at a gradient compared to the surrounding areas of the face. The nose shape must be segmented to detect the edge pixels of the nose present in the search region. The segmented nose shape in Figure 9 resulted after applying Otsu's thresholding in Figure 8.

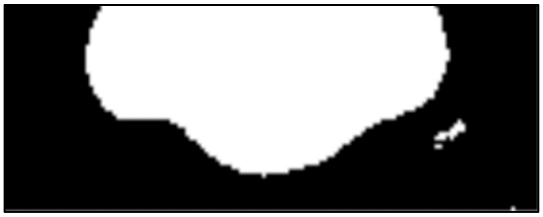


Figure 9
Segmented search area using Otsu's thresholding

B. Opening and closing

After the nose area is segmented using Otsu’s algorithm, we noticed that some of the regions contained gaps form of both foreground and background patches as shown in figures (Figure 11 (a) and Figure 12 (a)). These noise pixels affect the extraction performance if they are counted as part of the nose shape edges, and removing or minimizing their presence allow more accurate results.

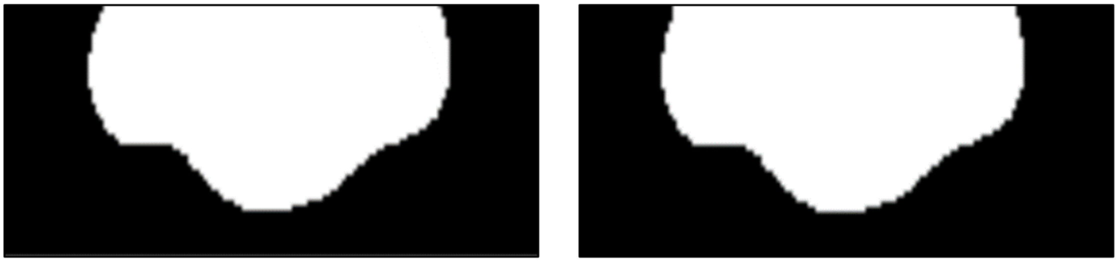
A morphological opening operation followed by a closing operation is used to fill the gaps and remove the noise pixels from the background. The opening and closing operations are applied using the EmguCv (more details in Appendix A), and the formulae used are as following:

$$Open(src,kernel)=dilate(erode(src,kernel)) \quad (1)$$

$$Close(src,kernel)=erode\ (dilate\ (src,kernel)) \quad (2)$$

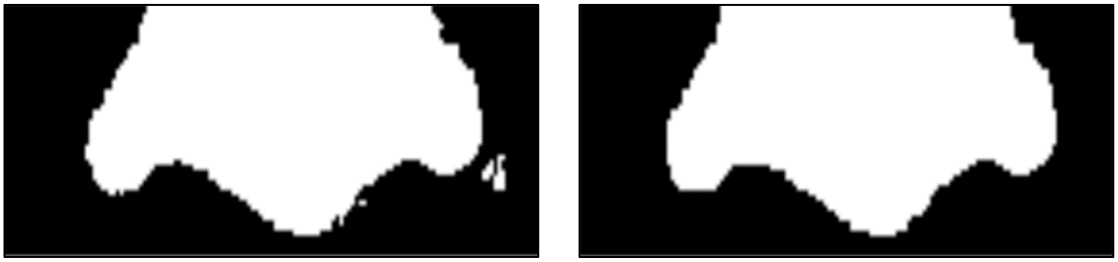
The open (1) operation is an erosion followed by a dilation applied on the search region (src) using a 3×3 structure element (kernel) to eliminate the gabs in the search region, whereas the closing (2) is a dilation followed by an erosion with the same preceding structure element to fill in any missing linking pixels.

Figures 10 To 12.((a) before and (b) after) shows different results of applying the opening (1) and closing(2) functions respectively using a 3×3 kernel .



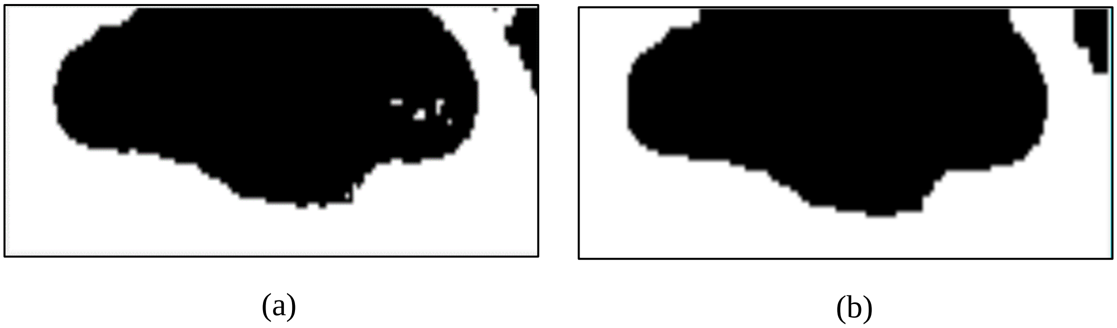
(a) (b)

Figure 10
(a) before (b) after applying an opening followed by closing operations on a region with no noise



(a) (b)

Figure 11
(a) before (b) after applying an opening followed by closing operations on a region containing a noise



(a) (b)

Figure 12
(a) before (b) after applying an opening followed by closing operations on a region containing a noise

C. Edge detection

Canny edge detector is used to detect the nose edges. The Canny edge detection is a multilevel algorithm used to finds edges by locating a local maximum of the gradient of f (x, y) [16]. The first stage after smoothing the region is computing the gradient magnitude and direction as following:

$$M(x,y) = \sqrt{g_x^2 + g_y^2}$$

$$\alpha(x,y) = \arctan(g_x/g_y)$$

$$where \qquad g_x = \partial f_s/\partial x \text{ and } g_y = \partial f_s/\partial y$$

The following stages are to thin the wide ridge around local maxima by applying nonmaxima suppression, and then use a double thresholding and connectivity analysis to detect and link edges. These edges are used to position the nose tip in the next stage of the algorithm.

Figures 13 and 14 are the result of applying EmguCV Canny function (using thresholds 0.1 and 0.4) on the segemented images shown in Figures 9 and 11 respectively.

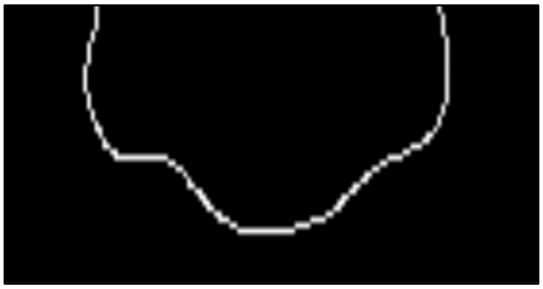


Figure 13
Edges detected after applying canny edge detection on figure 9 (b)

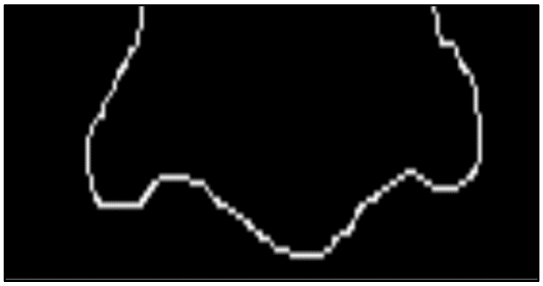
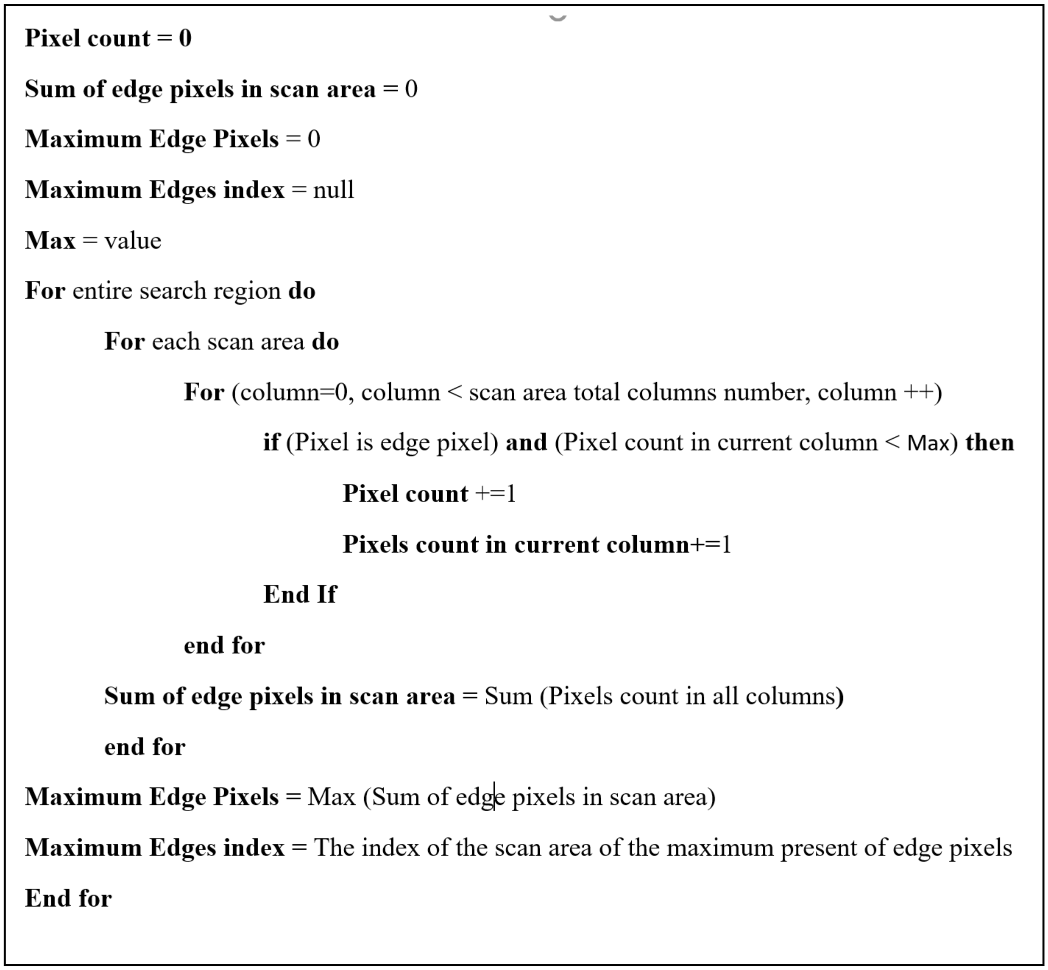


Figure 14
Edges detected after applying canny edge detection on figure 11 (b)

3.1.3 Perinasal region extraction

3.1.3.1 Nose tip detection algorithm

We have developed a new algorithm, as shown in Figure 15, that runs on the nose edges, detected in the previous step, and detects the location of the bottom tip of the nose. It scans the search region to determine the area with the maximal presence of the edge pixels, which is more dominant around the nose tip.

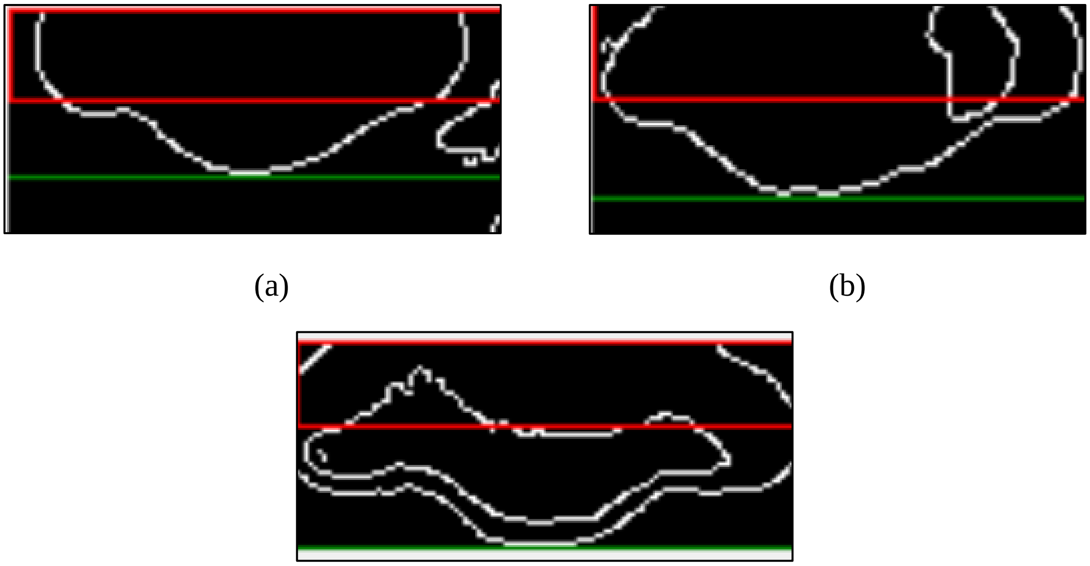


```
Pixel count = 0
Sum of edge pixels in scan area = 0
Maximum Edge Pixels = 0
Maximum Edges index = null
Max = value
For entire search region do
          For each scan area do
                    For (column=0, column < scan area total columns number, column ++)
                              if (Pixel is edge pixel) and (Pixel count in current column < Max) then
                                        Pixel count +=1
                                        Pixels count in current column+=1
                              End If
                    end for
          Sum of edge pixels in scan area = Sum (Pixels count in all columns)
          end for
Maximum Edge Pixels = Max (Sum of edge pixels in scan area)
Maximum Edges index = The index of the scan area of the maximum present of edge pixels
End for
```

Figure 15
Pseudocode of ROI extraction algorithm

The developed algorithm uses the edges extracted on the preceding step using the Canny edge detector to locate the maximal presence of edge pixels in the search region. The algorithm starts with a scanning window (20 pixels height) represented as the red box in Figure 16. The scanning window is moving vertically on the search region by one pixel at a time. The area contained in the scanning window is searched column by column for edge pixels, and the maximum edge pixels counted in each column is 3 pixels, after computing all the columns in the scanning region the sum of the counted edge pixels is calculated.

After computing the sum of the counted edge pixels for each scanned region, the maximum value of the sum is considered the area containing the nose tip. The bottom border of the scanning window, that has the maximum number of the edge pixels, is then marked to be the nose tip. The green lines shown in Figures 16 (a), (b), and (c) are the result of the nose tip detection algorithm on different nose shapes



(a)

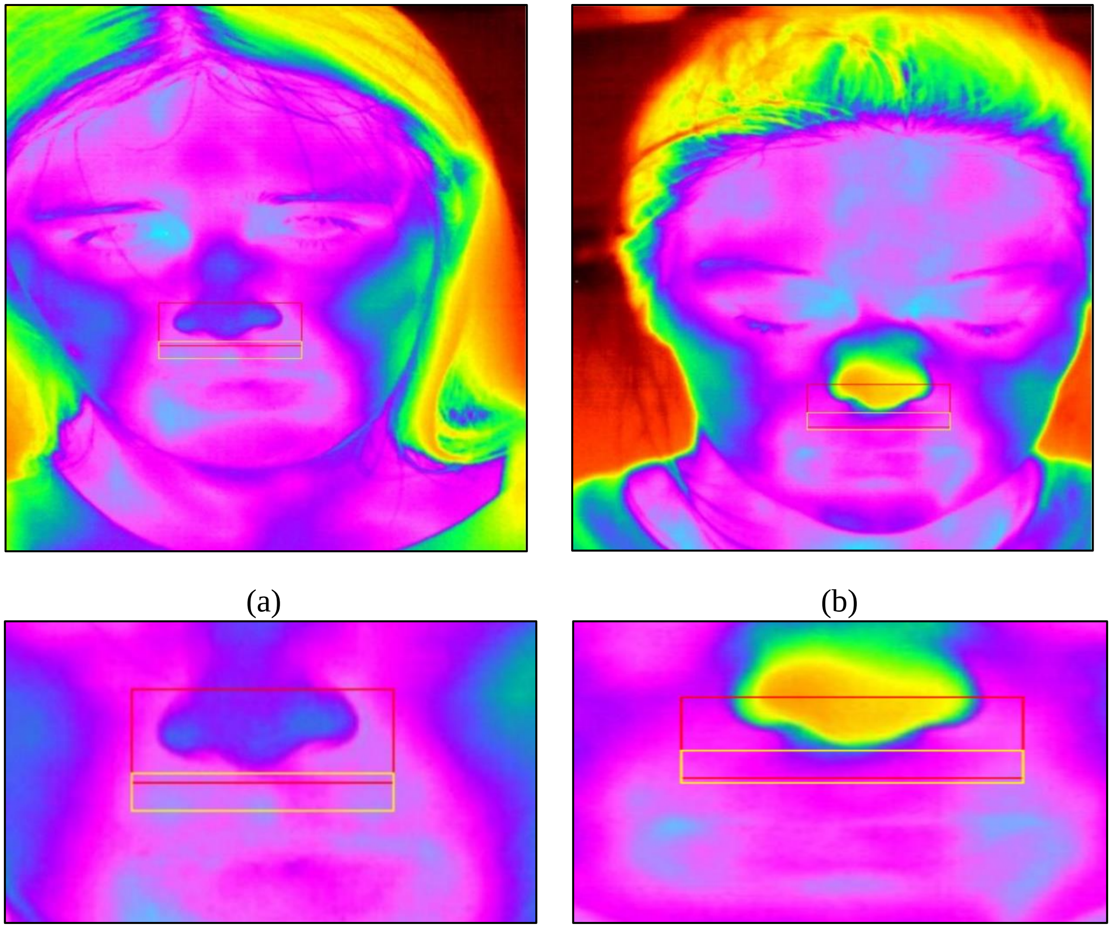
(b)

(c)

Figure 16
Nose tip detection

3.1.3.2 ROI extraction

Using the line, that resulted after detecting the nose tip in the previous step, as the upper bound, the perinasal ROI is extracted with the dimensions of 140 pixels width and 20 pixels height. Figures 17 (a) and (b) show samples of final ROI extractions (marked with yellow rectangles) on different thermal facial images.



(a) (b)

(c) (d)

Figure 17
(a)&(b)Extracted region of interest from facial thermal images

CHAPTER IV: EXPERIMENTS AND RESULTS

The proposed algorithm was tested on twenty-five thermal facial images (Eleven male, fourteen female) of a different age range.

The experiment was to validate the accuracy and consistency of the extracted ROI. The accuracy of the algorithm was measured using the extracted ROI of these images. The consistency of the ROI extractions was tested by multiple mouse clicks on the nose tip of every image. Specifically, we conducted 5 test runs on each of the 25 facial images. During each run, we clicked randomly somewhere on the nose and recorded the result of ROI extraction using the proposed algorithm as shown in Table 1 and 2. Since we are measuring consistency here, we used the ROI output of the first run as the baseline and compared it against the remaining 4 runs. This way we measured the error rate, precision, and recall as shown in Table 3. The error rate was calculated as following:

$$\text{Error rate} = \frac{True\ positive\ pixels}{Total\ number\ of\ pixels\ in\ the\ \ entire\ ROI}$$

The precision and recall were calculated using the following formulae:

$$\text{Precision} = \frac{True\ positive\ pixels}{True\ positive\ pixels + False\ positive\ pixels}$$

$$\text{Recall} = \frac{True\ positive\ pixels}{True\ positive\ pixels + False\ nigative\ pixels}$$

Results

The algorithm successfully extracted the ROI of 24 facial images out of the 25 subjects. A successful ROI extraction is in Figure 18, while unsuccessful extraction of only one of the subjects is shown in Figure 19. Regardless where the mouse click occurs on the nose tip the desired perinasal region was extracted.

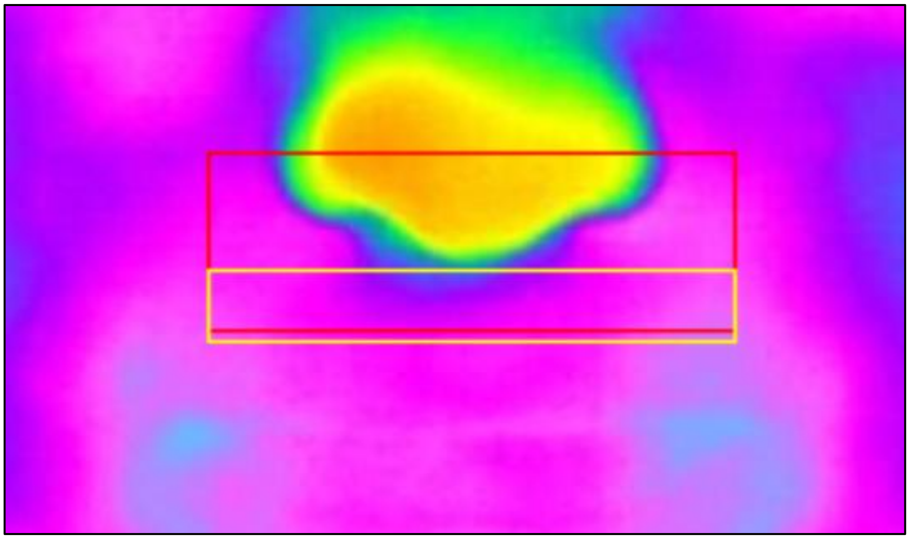


Figure 18
Successful ROI extraction

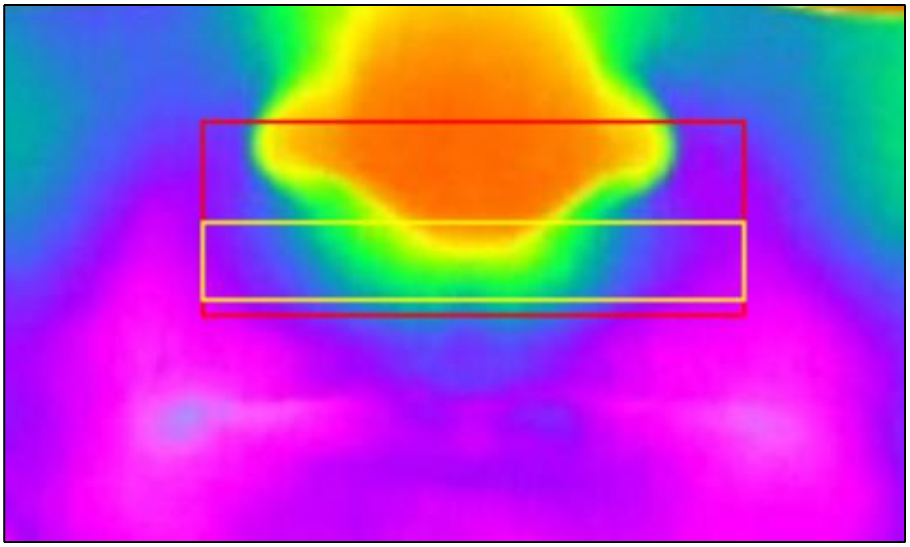


Figure 19
Unsuccessful ROI extraction

For the conducted 5 test runs on each of the 25 facial images. The coordinates of the random click on the nose, and the top left corner and the bottom right corner of the extracted region of interest were recorded as shown in Table 1 and Table 2.

subject	Try 1			Try 2			Try 3		
	Click	TopLeft	BottomRight	Click	TopLeft	BottomRight	Click	TopLeft	BottomRight
1	304,464	234,474	374,494	303,463	233,473	373,493	304,460	234,473	374,493
2	217,373	147,388	287,408	217,377	147,394	287,414	216,376	146,393	286,413
3	269,440	199,462	339,482	267,437	197,460	337,480	269,442	199,457	339,477
4	249,484	179,502	319,522	249,482	179,501	319,521	248,470	178,494	318,514
5	261,462	191,477	331,497	257,469	187,485	327,505	259,467	189,478	329,498
6	262,361	192,376	332,396	262,355	192,373	332,393	260,358	190,374	330,394
7	293,417	223,436	363,456	298,420	228,438	368,458	292,413	222,435	362,455
8	248,385	178,406	318,426	248,396	178,416	318,436	248,402	178,415	318,435
9	249,446	179,458	319,478	251,441	181,462	321,482	250,434	180,458	320,478
10	277,378	207,396	347,416	276,381	206,396	346,416	277,383	207,397	347,417
11	253,271	183,291	323,311	256,276	186,291	326,311	255,274	185,291	325,311
12	241,387	171,411	311,431	240,401	170,417	310,437	240,393	170,417	310,437
13	269,384	199,400	339,420	268,392	198,401	338,421	268,383	198,399	338,419
14	251,490	181,505	321,525	254,492	184,506	324,526	253,490	183,505	323,525
15	219,393	149,409	289,429	221,394	151,410	291,430	217,390	147,410	287,430
16	271,474	201,486	341,506	272,476	202,486	342,506	270,475	200,484	340,504
17	296,405	226,426	366,446	297,414	227,426	367,446	297,410	227,426	367,446
18	213,357	143,380	283,400	214,358	144,382	284,402	215,355	145,379	285,399
19	237,356	167,379	307,399	233,354	163,377	303,397	234,357	164,379	304,399
20	221,440	151,447	291,467	220,438	150,446	290,466	222,441	152,447	292,467
21	259,450	189,465	329,485	258,454	188,465	328,485	258,448	188,464	328,484
22	Unsuccessful Extraction of ROI								
23	228,504	158,527	298,547	226,511	156,530	296,550	227,505	157,527	297,547
24	262,419	192,440	332,460	260,427	190,438	330,458	259,423	189,438	329,458
25	257,430	187,445	327,473	256,425	186,448	326,468	257,427	187,448	327,468

Table 1

Recorded coordinates of the click on the nose and the top left and bottom right corners of the extracted ROI for try 1,try 2 and try 3

subject	Try 4			Try 5		
	Click	TopLeft	BottomRight	Click	TopLeft	BottomRight
1	304,469	234,475	374,495	300,457	230,473	370,493
2	216,377	146,394	286,414	219,385	149,398	289,418
3	269,449	199,459	339,479	271,447	201,459	341,479
4	251,474	181,497	321,517	249,471	179,495	319,515
5	253,468	183,478	323,498	261,472	191,490	331,510
6	263,360	193,375	333,395	295,385	189,374	329,394
7	291,417	221,435	361,455	239,414	223,435	363,455
8	255,400	185,406	325,426	251,402	181,416	321,436
9	250,446	180,458	320,478	250,438	180,459	320,479
10	278,383	208,397	348,417	288,382	218,396	358,416
11	254,269	184,291	324,311	251,274	181,291	321,311
12	241,402	171,417	311,437	239,398	169,417	309,437
13	272,382	202,399	342,419	268,386	198,401	338,421
14	252,487	182,505	322,525	253,489	183,505	323,525
15	221,395	151,411	291,431	220,394	150,410	290,430
16	268,479	198,484	338,504	269,471	199,484	339,504
17	279,402	227,426	367,446	295,417	225,426	365,446
18	212,360	142,383	282,403	216,358	146,382	286,402
19	233,351	163,373	303,393	235,354	165,377	305,397
20	223,442	153,447	293,467	221,441	151,447	291,467
21	259,453	189,465	329,485	255,447	185,464	325,484
22	Unsuccessful Extraction of ROI					
23	226,509	156,530	296,550	226,498	156,522	156,542
24	260,427	190,438	330,458	259,432	189,442	329,426
25	258,428	188,448	328,468	256,421	186,445	326,465

Table 2

Recorded coordinates of the click on the nose and the top left and bottom right corners of the extracted ROI for try 4and try 5

As we can see from Table 3 the high precision and recall, and low error rates indicate that the proposed algorithm is effective and can reliably extract the perinasal ROI with just a mouse click on the nose.

Subjects	Try 1			Try 2			Try 3			Try 4		
	Error Rate	Precision	Recall	Error Rate	Precision	Recall	Error Rate	Precision	Recall	Error Rate	Precision	Recall
1	0.11	0.94	0.94	0.1	0.95	0.95	0.1	0.95	0.95	0.15	0.92	0.92
2	0.6	0.7	0.7	0.11	0.94	0.94	0.01	0.99	0.99	0.45	0.77	0.77
3	0.22	0.88	0.88	0.5	0.75	0.75	0.3	0.85	0.85	0.32	0.83	0.83
4	0.11	0.94	0.94	0.8	0.59	0.59	0.52	0.73	0.73	0.7	0.65	0.65
5	0.85	0.57	0.57	0.1	0.93	0.93	0.21	0.89	0.89	1.3	0.35	0.35
6	0.3	0.85	0.85	0.2	0.88	0.88	0.11	0.94	0.94	0.21	0.89	0.89
7	0.27	0.86	0.86	0.11	0.94	0.94	0.12	0.93	0.93	0.1	0.95	0.95
8	1	0.5	0.5	0.9	0.55	0.55	0.1	0.95	0.95	1	0.47	0.47
9	0.42	0.78	0.78	0.01	0.99	0.99	0.01	0.99	0.99	0.11	0.94	0.94
10	0.01	0.99	0.99	0.1	0.95	0.95	0.11	0.94	0.94	0.15	0.92	0.92
11	0.04	0.97	0.97	0.02	0.98	0.98	0.01	0.99	0.99	0.02	0.98	0.98
12	0.61	0.69	0.69	0.61	0.69	0.69	0.6	0.7	0.7	0.62	0.69	0.69
13	0.11	0.94	0.94	0.11	0.94	0.94	0.14	0.92	0.92	0.11	0.94	0.94
14	0.14	0.92	0.92	0.02	0.98	0.98	0.01	0.99	0.99	0.02	0.98	0.98
15	0.12	0.93	0.93	0.12	0.93	0.93	0.22	0.88	0.88	0.11	0.94	0.94
16	0.01	0.99	0.99	0.21	0.89	0.89	0.23	0.88	0.88	0.22	0.88	0.88
17	0.01	0.99	0.99	0.01	0.99	0.99	0.01	0.99	0.99	0.01	0.99	0.99
18	0.21	0.89	0.89	0.12	0.93	0.93	0.31	0.84	0.84	0.23	0.88	0.88
19	0.25	0.87	0.87	0.04	0.97	0.97	0.64	0.68	0.68	0.22	0.88	0.88
20	0.11	0.94	0.94	0.01	0.99	0.99	0.02	0.98	0.98	0	1	1
21	0.01	0.99	0.99	0.11	0.94	0.94	0	1	1	0.15	0.92	0.92
22	Unsuccessful Extraction of ROI											
23	0.32	0.83	0.83	0.01	0.99	0.99	0.32	0.83	0.83	0.52	0.73	0.73
24	0.22	0.88	0.88	0.23	0.88	0.88	0.22	0.88	0.88	0.23	0.88	0.88
25	0.31	0.84	0.84	0.29	0.85	0.85	0.31	0.84	0.84	0.01	0.99	0.99

Table 3
The error rate, precision and recall results

CHAPTER V: CONCLUSION

Conclusion

In this thesis, a ROI extraction algorithm is developed to extract the perinasal area from thermal facial images. A C# WinForms plug-in is implemented to be deployed to the S-Interface software that is used by the Computational Physiology Lab (CPL) at University of Houston for studying physiological variables. The plug-in is semi-automated as it requires a mouse click on the nose tip by an operator as the first step before extracting the ROI. The benefit of the plug-in is it minimizes the chances of wrongly selecting the required region by an inexperienced operator. The algorithm extracts a maximal region of the perinasal area which leads to more accurate results of perspiration measurements using the S- Interface software's physiological algorithms.

Future Work

A further optimization of the algorithm is to include the mouth as a factor determining the lower bound of the extracted region.

REFERENCES

[1] K. Hong and S. Hong, "Real-time stress assessment using thermal imaging," *VIS Computer*, vol. 32, no. 11, pp. 1369–1377, October 26 2015.

[2] (2016, October 15). *S-Interface Software*. Available: http://cpl.uh.edu/software/s-interface/

[3] A. Duarte *et al.*, "Segmentation Algorithms for Thermal Images," *Procedia Technology*, vol. 16, pp. 1560-1569, 2014/01/01/ 2014.

[4] V. Kosonogov *et al.*, "Facial thermal variations: A new marker of emotional arousal," *PloS one*, vol. 12, no. 9, pp. e0183592-e0183592, 2017.

[5] J.-S. Eom and J.-H. Sohn, *Emotion Recognition using Facial Thermal Images*. 2012.

[6] I. Pavlidis *et al.*, "Fast by Nature - How Stress Patterns Define Human Experience and Performance in Dexterous Tasks," *Scientific Reports*, Article vol. 2, p. 305, 03/06/online 2012.

[7] M. Dcosta, D. Shastri, R. Vilalta, J. K. Burgoon, and I. Pavlidis, "Perinasal indicators of deceptive behavior," in *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 2015, vol. 1, pp. 1-8.

[8] C. Ma, N. T. Trung, H. Uchiyama, H. Nagahara, A. Shimada, and R.-I. Taniguchi, "Adapting Local Features for Face Detection in Thermal Image," *Sensors (Basel, Switzerland)*, vol. 17, no. 12, p. 2741, 2017.

[9] A. Prochazka, H. Charvatova, O. Vysata, J. Kopal, and J. Chambers, "Breathing Analysis Using Thermal and Depth Imaging Camera Video Records," (in eng), *Sensors (Basel)*, vol. 17, no. 6, Jun 16 2017.

[10] I. Pavlidis *et al.*, "Dissecting Driver Behaviors Under Cognitive, Emotional, Sensorimotor, and Mixed Stressors," *Scientific Reports*, Article vol. 6, p. 25651, 05/12/online 2016.

[11] (2016, October 15). *Computational Physiology Lab*. Available: http://cpl.uh.edu/

[12] D. Shastri, M. Papadakis, P. Tsiamyrtzis, B. Bass, and I. Pavlidis, "Perinasal Imaging of Physiological Stress and Its Affective Potential," *IEEE Transactions on Affective Computing*, vol. 3, no. 3, pp. 366-378, 2012.

[13] Y. Zhou, P. Tsiamyrtzis, P. Lindner, I. Timofeyev, and I. Pavlidis, "Spatiotemporal Smoothing as a Basis for Facial Tissue Tracking in Thermal Imaging," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 5, pp. 1280-1289, 2013.

[14] G. E. Sujji, Y. V. S. Lakshmi, and G. W. Jiji, "MRI Brain Image Segmentation based on Thresholding," *International Journal of Advanced Computer Research*, vol. 3, no. 1, pp. 97-101, 2013.

[15] J. Liu, W. Li, and Y. Tian, "Automatic thresholding of gray-level pictures using two-dimension Otsu method," in *China., 1991 International Conference on Circuits and Systems*, 1991, pp. 325-327 vol.1.

[16] P. P. Acharjya, R. Das, and D. Ghoshal, "Study and Comparison of Different Edge Detectors for Image Segmentation," *Global Journal of Computer Science and Technology,* vol. 12, no. 13, 2012.

APPENDIX A

EmguCV functions used in the proposed algorithm

a. Convert Method

Convert the current image to the specific color and depth

Syntax:

Image<TOtherColor, TOtherDepth> Convert<TOtherColor, TOtherDepth>()

Type Parameters:

- **TOtherColor** The color type of the source image.
- **TOtherDepth** The color depth of the source image.

Return value:

Image of the specific color and depth.

b. ToBitmap Method

Convert an image into Bitmap, the pixel values are copied over to the Bitmap.

Syntax:

Image<TColor, TDepth>.ToBitmap ()

Return value:

This image in Bitmap format, the pixel data are copied over to the Bitmap.

c. EqualizeHist Method

The algorithm inplace normalizes brightness and increases contrast of the image. For color images, a HSV representation of the image is first obtained and the V (value) channel is histogram normalized

Syntax:

Image<TColor, TDepth>._EqualizeHist ()

d. Smooth Gaussian

Perform Gaussian Smoothing in the current image and return the result.

Syntax:

SmoothGaussian (**Int** ***kernelwidth,***
Int ***kernelHeight,***
Double ***sigma1,***
Double ***sigma2***
)

Parameters:

- **KernelWidth** The width of the Gaussian kernel.
- **KernelHeight** The height of the Gaussian kernel.
- **Sigma1** The standard deviation of the Gaussian kernel in the horizontal dimwnsion.
- **Sigma2** The standard deviation of the Gaussian kernel in the vertical dimwnsion.

Return Value:

The smoothed image.

e. CvInvoke.Threshold Method

Applies fixed-level thresholding to single-channel array. The function is typically used to get bi-level (binary) image out of grayscale image (cvCmpS could be also used for this purpose) or for removing a noise, i.e. filtering out pixels with too small or too large values. There are several types of thresholding the function supports that are determined by threshold_type.

Syntax:

Threshold (

IInputArray ***src,***

IOutputArray ***dst,***

Double ***threshold,***

Double ***maxValue,***

ThresholdType ***thresholdType***

)

Parameters:

- **src** The source array (single-channel, 8-bit of 32-bit floating point).
- **dst** Destination array; must be either the same type as *src* or 8-bit.
- **threshold** Threshold value.
- **maxvalue** Maximum value to use with CV_THRESH_BINARY and CV_THRESH_BINARY_INV thresholding types.
- **thresholdtype** Thresholding type.

f. CvInvoke.MorphologyEx Method

Performs advanced morphological transformations.

Syntax:

MorphologyEx(

IInputArray ***src,***
IOutputArray ***dst,***
MorphOp ***operation,***
IInputArray ***kernel,***
Point ***anchor,***
int ***iterations,***
BorderType ***borderType,***
MCvScalar ***borderValue***

)

Parameters:

- **src** Source image.
- **dst** Destination image.
- **operation** Type of morphological operation.
- **kernel** Structuring element.
- **anchor** Anchor position with the kernel. Negative values mean that the anchor is at the kernel center.
- **iterations** Number of times erosion and dilation are applied.
- **borderType** Pixel extrapolation method.
- **borderValue** Border value in case of a constant border.

g. Canny Method

Find the edges on this image and marked them in the returned image.

Syntax:

Image<TColor, TDepth>.Canny Method (Double, Double)

Image<Gray, byte> Canny (

double ***thresh*****,**

double ***threshLinking***

)

Parameters:

- **thresh** The threshold to find initial segments of strong edges.
- **threshLinking** The threshold used for edge Linking.

Return value:

The edges found by the Canny edge detector