

SMART MIRROR

Nicole Lemieux
Austin Pursley
James Quinn
Amanda Wolken

SYSTEM OVERVIEW

Austin Pursley

04/28/2018

Contents

Table of contents	Error! Bookmark not defined.
1) Overview	3
2) Design.....	3
2.1. Thermal Image Processing, Skin Temperature	3
2.2. Color Image Processing.....	6
2.2.1. Face and Face Feature Detection	6
2.2.2. Skin Detection	7
2.2.3. Skin Lesion Detection.....	7
2.2.4. Skin Lesion Clasification	10
3) Validation and Testing	11
3.1. Skin Temperature	11
3.2. Skin Color	112
3.3. Skin Lesion Detection	12
3.4. Skin Lesion Classification	13
4) Conclusion	14
5) Sources.....	14

1) Image Processing: Overview

This subsystem uses image processing on the images of the user to extract health metrics. There is a process for each type of image, thermal and color. Thermal image process finds the skin temperature of the user and color image processing finds skin color and characteristics about skin lesions such as moles or acne.

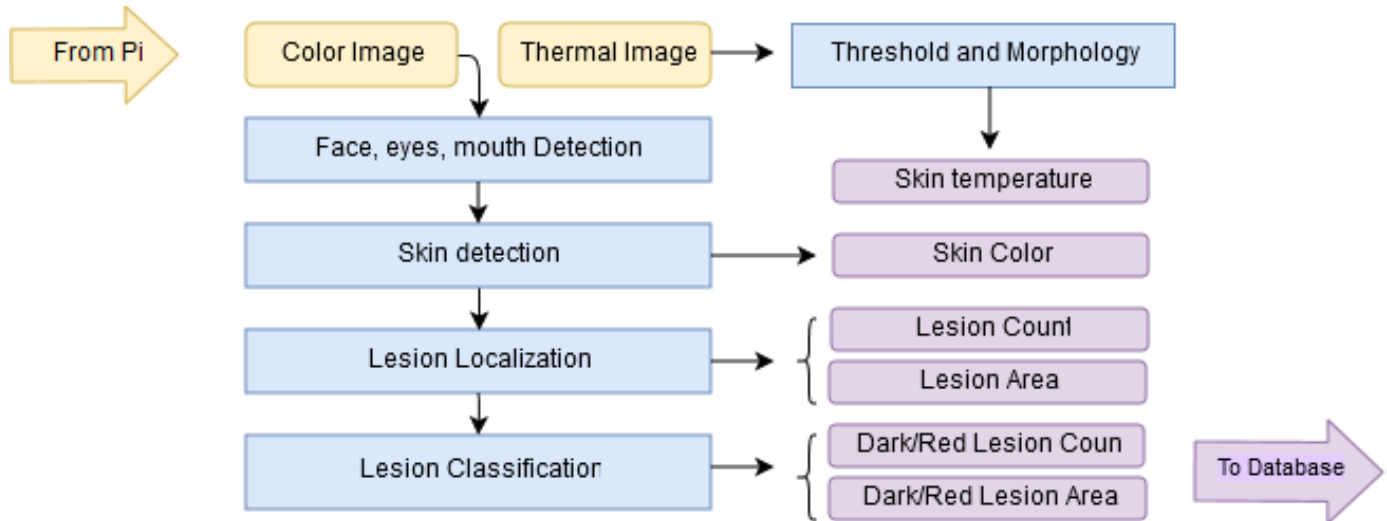


Figure 1: Overview diagram of image processing subsystem.

2) Design

2.1. Thermal Image Processing

Our object is to accurately and consistently read the skin temperature intensity. The overall plan is to isolate the skin area and take the highest intensity value from that reading. The FLIR Lepton thermal camera we are using captures 80x69 images and the optimal operating temperature range is 14 °F to +149 °F (1). In general it is expected that the coldest area will be a surface of an object in the background and that the hottest area will be the user's skin. However there will be processing in place to deal with extreme values. Our analysis will be based on relative changes in temperatures so we are not concerned with calibrated readings in Fahrenheit or Celsius.

Step 1: Threshold

The first image processing function we perform is a threshold to exclude the colder section of the image. Pixels with an intensity higher than the threshold value will be white and those lower will be black. This is the “mask”, the area of the image we are focusing on to find the skin temperature. This threshold value must be high enough to exclude the background but not so high that it could exclude the face. This also removes any very cold background objects such as ice drinks in copper mugs or a window on a cold day.

Our input starts as a text file that contains the matrix of thermal values that are integers around 8000-8500. I used C++ and OpenCV to convert this into a Mat object to perform image processing on. We tested with three types of threshold techniques: Otsu's method, a hard-coded threshold values, and

a median threshold value. Otsu's method is an automatic way to calculate a threshold value that results in the lowest variance of the background (black) and foreground (white). However OpenCV Otsu's function only works when the image values are normalized between 0-255. This works if the range of values is always the same, however it breaks with the simulated hot spot in the background that raises the range (Figure 2, the black square). Another method I tried was a hard-coded threshold value found with experimenting. This works without normalizing the values and so it will work with the simulated hot spot and also gives a more focused mask of the skin. However a hardcoded value is rigid and will not have correct results if the expected range of skin temperature changes. The final method I arrived at was using the median pixel value of the image as the threshold value. This has the advantage of being immune to extreme hot or cold values.



Figure 2: Simulated "hot spot" in the left corner. Otsu threshold only works with the normalized image and the increases in range makes it not work. All that was left after Otsu was the hot spot and then it was eroded, leaving nothing.

Step 2: Morphology open and close

If the user is wearing glasses or has a beard then the threshold will leave holes in the mask that can ruin the consistency of the mask. So we perform a morphology close operation which will fill in those gaps. This will leave a general area of the user but may contain objects in the background that are hot such as light bulbs, hot drinks, or warm windows. To deal with this we apply a morphology opening to remove isolated hot spots in the background.

Step 3: Bitwise AND, Finding Max Skin Temperature

What's left of the image is a 'mask' that outlines the user's skin. An "AND" operation can be performed with this and the original image to isolate the thermal values of the skin area in the original image. From this we can measure the skin temperature by finding the maximum pixel value.

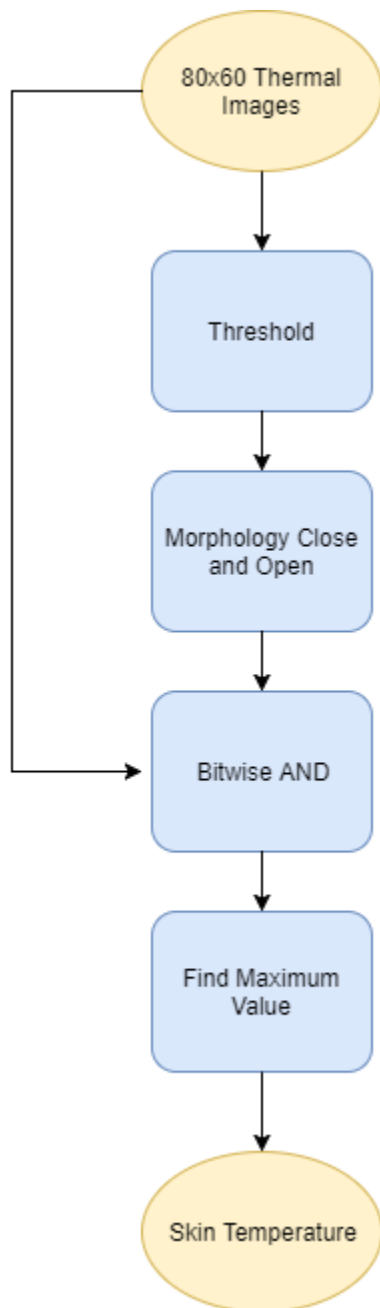


Figure 3: Plan for thermal image skin temperature

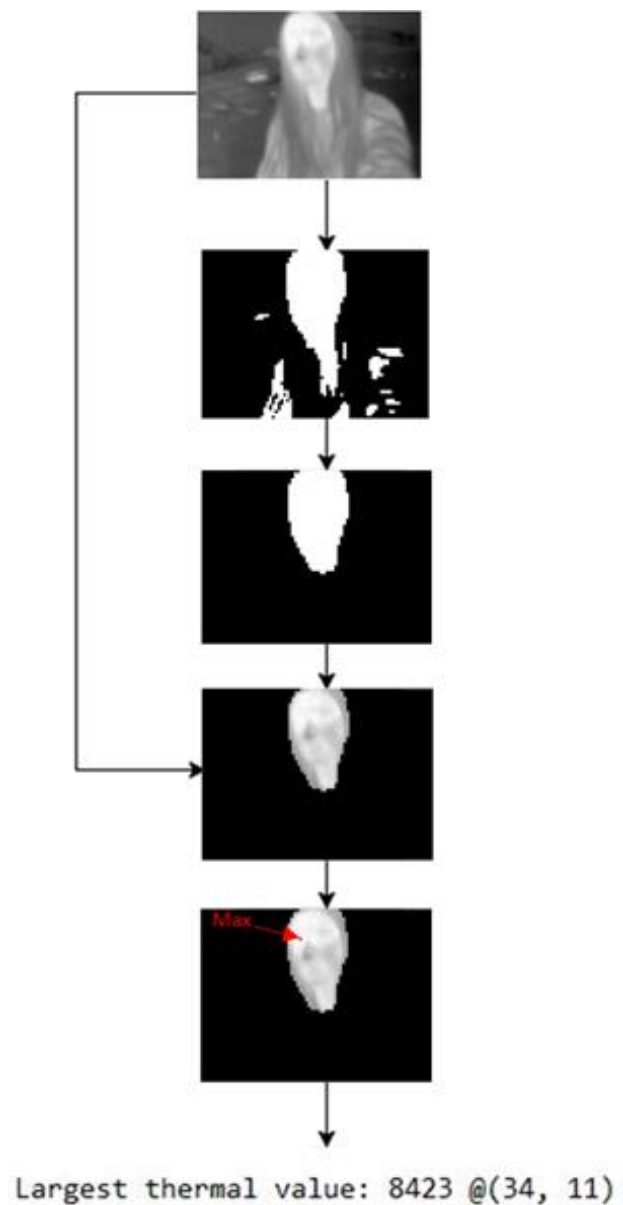


Figure 4: Example of my results of thermal image processing

Corner Cases of Thermal Image Processing

Glasses and facial hair obstruct the thermal camera and result in cold spots. Because our metric analysis will be comparative this may be fine but the user must consistently wear glasses or have the same facial hair. Although the user is not assumed to always be the warmest object, it is expected that something else that is as warm as the user is not taking up a large portion of the image behind the user. This leads to another possible corner case: large, warm objects such as windows in the background. In that case the user will not be excluded from the threshold or eroded with the opening. Another corner case is hot objects being around the outline or directly in front of the user. For example a candle could be sitting in front of the user. The flame would be included in the mask and would give an incorrect skin temperature that factors in the flame heat. Instructing the user to not have hot objects in the area is one option. Similarly, if two people are in the image the analysis will not be able to tell who is who.



Figure 5: Corner cases include glasses, facial hair, and large warm windows in the background. (3)

2.2. Color Image Processing

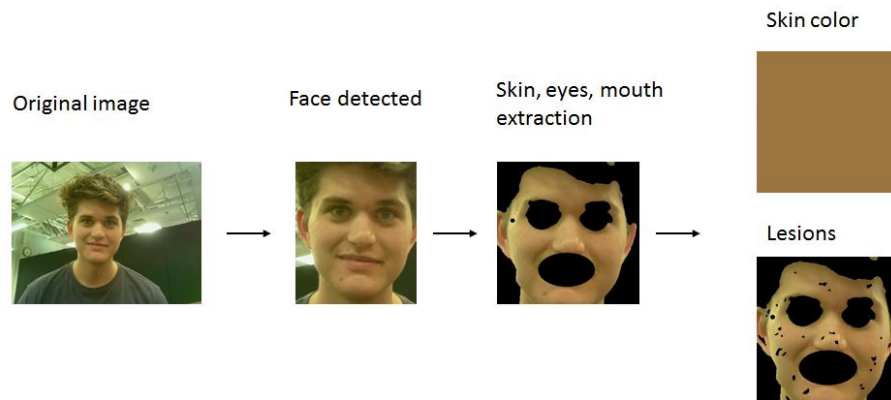


Figure 6: Broad overview of the color image process

2.2.1. Face and Face Feature Detection

Color image processing starts with wanting to find the face of the user, and then find the eyes and mouth as well. This is done using a pre-trained Harr Cascading classifiers provided by the OpenCV library. These classifiers are machine learning based and are robust enough for this application, however they do not represent the most advanced face and face feature detection techniques (e.g. such as neural networks). The output in the program is the ROI boxes for the face, two eyes, and mouth.

2.2.2. Skin Detection

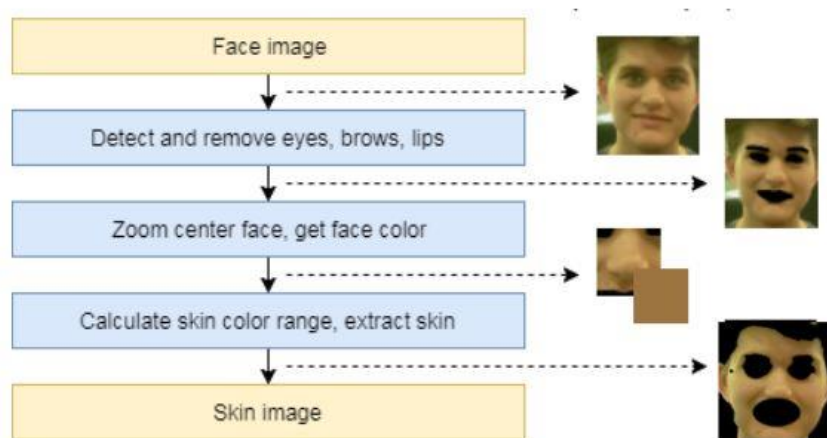


Figure 7: Skin detection process overview.

The idea behind skin detection is that if you know where the face is then you can have a pretty good approximate idea of what the skin is going to look like. This process was inspired by a process found in skin detection process in a paper on hand detection (7). The first step is to remove the parts of the face image you know are not skin, the eyes and mouth. The second step is to “zoom” into the face area and calculate the average pixel value (ignoring blacked out eye and mouth pixels) to approximate a skin color. That skin color is used to calculate an upper and lower bound for skin color ranges. That range determines which pixels are skin and which are not. Finally, some morphological operations can be performed to erode away outliers and close gaps in the mask.

Corner Cases

If the background is a similar color to the user’s face, then the process will include background as the face and perhaps get an incorrect skin color.

2.2.3. Skin Lesion Detection

A skin lesion is a section of the skin that is abnormal in appearance. We want to process the image of the skin and find out where the lesions are and what their characteristics (size and color) are.



Figure 8: Examples of skin with lesions.

This subsystem will have an extracted image of the user's skin as input. From that we have to detect these lesions. The steps for the design is:

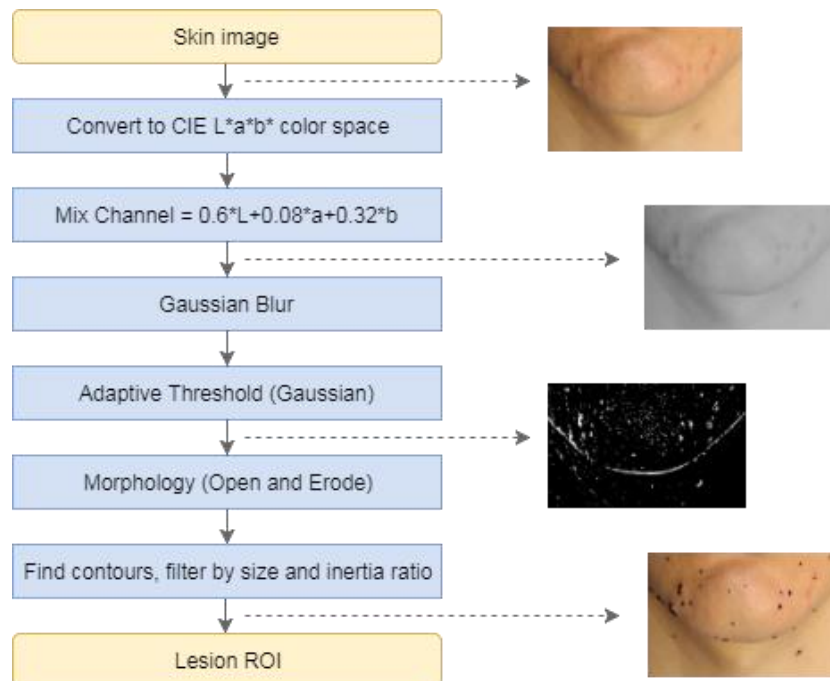


Figure 9: Overview of lesion detection process.

Step 1: Change Color Space, Remove Noise

Color image are normally understood as a vector of three color values: red, blue, green. This is good for viewing the image but not always good for processing images. You can change the color space of the image through a series of operations. The first one for lesion detection is to convert to the CIE L*a*b* color space. Then the three new channels are mixed according the formula in Figure 9. A lot of experimentation was used in order to determine this process for this change of color space. Originally it was as simple as removing the blue channel, which seemed to contain a lot noise. Then the L*a*b was found to generally to be better for lesion detection than other color space such as RGB, HSV, and YCrCb. Later it was found that the performance could be improved by changing the weights of each channel when mixed into one. In retrospect this experimentation was a pretty naïve approach and very time extensive. There are undoubtedly smarter techniques for doing this. The final step after changing the color space is applying a Gaussian blur to reduce noise.

Step 2: Adaptive Threshold

I discussed another threshold technique in the skin color section, Otsu's method, however this time it will not work. Otsu method considers the entire image when choosing a threshold value but because of uneven lighting conditions there is a need for the adaptive threshold technique. Adaptive thresholding applies different thresholds for different regions of the image based on the local conditions of that region. A factor in the size of those regions is the “block size” and is another factor in lesion detection performance.

Step 3: Morphology Close and Open

Just as with thermal imaging, we are left with a black and white “mask” that we want to adjust using morphology. The close operation will combine clusters of blobs and the open operation will erode the smaller blobs that are left over. The size of the open and close element used for these morphology operations are more factors in lesion detection performance.

Step 4: Find Contours, Filter

Now we are left with a mask where each white blob is associated with a lesion. Next we want to find the contours of each individual lesion. Contours are an array of pixels that form a border or outline of an object. In this case we want contours for each lesion so analyze each one individually. After finding the contours there will likely be some that do not belong, so the contours are filtered by size, shape, and color. If a contour is too big or too small, it will be removed as contained a lesion. Oddly shaped lesions can also be filtered, such as if they are too elongated. Similarly, if the color is too similar to the skin around the lesion then it is determined to not be a lesion.

Parameters

As mentioned, there are certain parameters that are adjustable to improve performance. These are: Gaussian filter k-size, adaptive threshold block size, and the size of the morphology open and close elements. Increasing the k-size of the Gaussian blur increases how “blurred” or smoothed out the image is. In other words, higher k-size means less details left over. The correct block size depends on size of the image and what size the lesions are. One round of lesions detection may work for the usual sized lesions but another round of lesion detection can be applied to catch very large lesions. If the block size goes up the Gaussian blur should go up with it as we are looking at larger areas (and thus remove more detail). Similarly, the sizing of the close element should also go up with block size. The open element needs to be increased as well, but less so. This understanding of the parameters is based on a lot of testing for various images.

Corner Cases

The way the process works right now is that it effectively detect lesions that are darker than the surrounding skin. However, a corner case is if the user has lesions that are lighter than the surrounding skin. This case should be addressed doing two rounds of lesion detection, one with the normal image and the other inverted. This can be fixed by changing the coefficients for mixing the image channels, however this would lead to having to perform the process at least twice. A smarter approach to color space selection

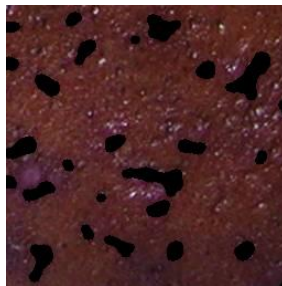


Figure 2: Notice that light, red spots are not detected on darker background. Those same spots on lighter skin would probably be detected

2.2.4. Skin Lesion Classification

With the contours of the lesion, we can find the color of each lesion and try to classify it compared to the average skin color. There are two classifications we are trying to make: red and dark spots. We make the distinction by looking at the colors in HSV color space. In general if the hue of the lesion is much less than the hue of the skin color surround the lesion we say it is red and if the value is much higher than the skin color we say it is dark. Once again, this process was stumbled into with time-extensive experimentation. Only with experience and reflection further down the road do I realize that this would have been an excellent case to apply machine learning classification algorithm.

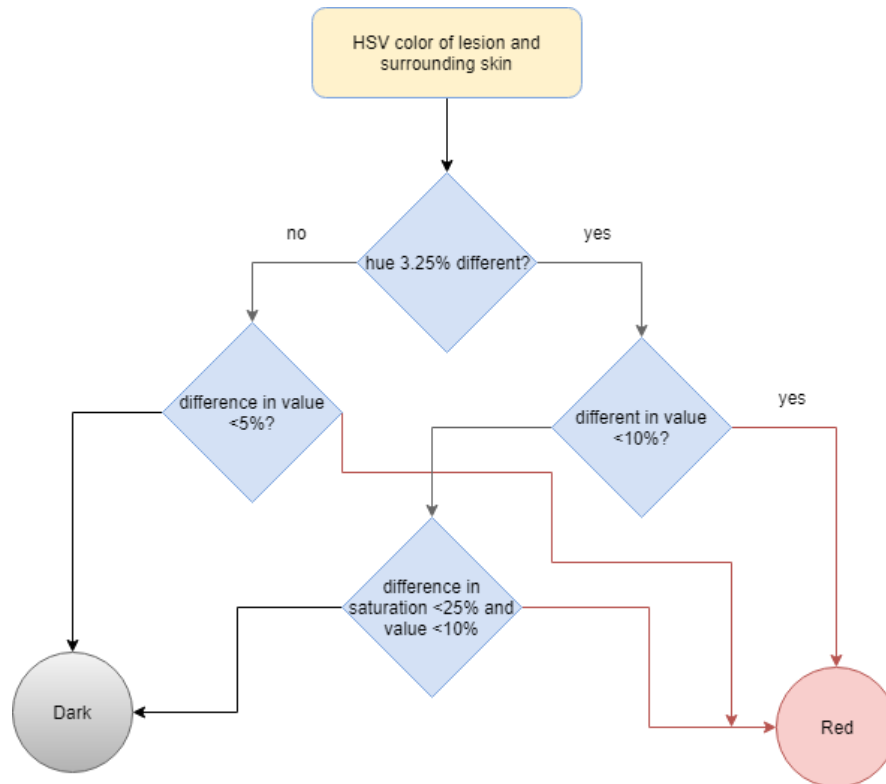


Figure 3: Overview for lesion classification.

3) Validation and Testing

3.1. Skin Temperature

Six images taken with our FLIR Lepton where used for testing. We have three normal images, two with glasses, two with beard, and one with glasses and beard, and one manipulated to have a “hot spot” in the left corner. In all cases we were still able to get a correct mask that excludes the background and contains the face. Glasses and beards don’t seem to have a significant effect, but nevertheless users should at least be told to be consistent with wearing glasses or not when using the Smart Mirror.



Figure 4: Summary visualization of results

Image	Glasses	Beard	Thermal Value	Correct mask?
1	0	0	8423	Yes
1 hot spot	0	0	8423	Yes
2	1	0	8418	Yes
3	0	0	8396	Yes
4	1	1	8425	Yes
5	0	1	8427	Yes

Table 1: Skin temperature validation data

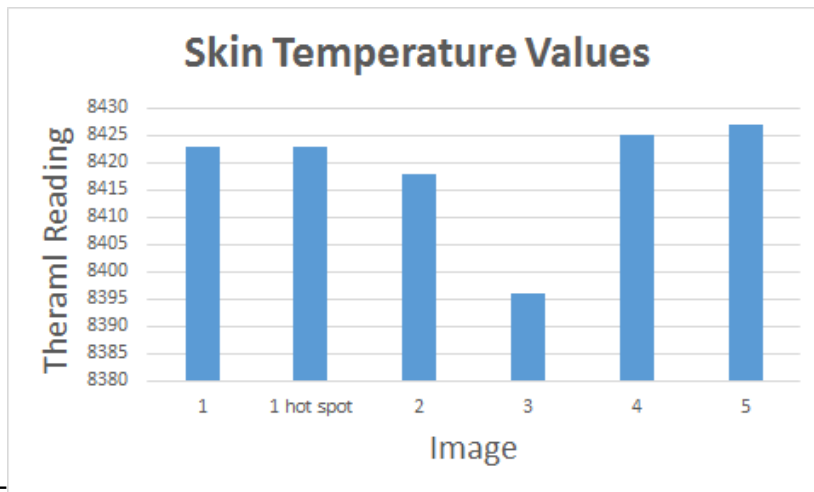


Figure 5: From our small sample it doesn’t seem glasses or beards effect temperature.

3.2. Skin Color

To validate skin color I used 7 images taken with our Raspberry Pi camera of everyone in our group and then also used 8 images of celebrities found on the web. Grading the performance is a bit subjective and theoretically a skin color that looks “off” here may be fine since our analysis will be relative to images taken in the same environment of the same person. However this round of validation raises some concerns about corner cases and further refinement of this techniques will need to be made. I determined the accuracy to be 13/15, or 86.6%.

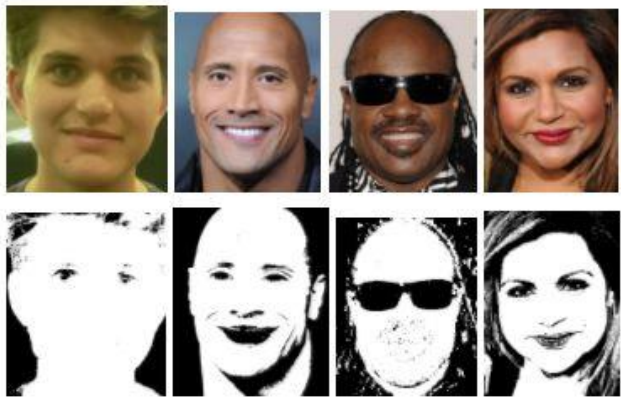


Figure 6: Sample output of skin detection results

3.3. Skin Lesion Detection

For lesion detection I have a picture of my own face and 20 others through online sources (5). These are from a range of different skin types, conditions in lighting, and severity of skin condition, and angles. Each face image was manually broken down into 143 smaller images for validation. For each image I manually counted the lesions and compared it to what the program counted. I kept track of how many lesions it had correctly identified (hit), how many it falsely detected, and how many it had missed. Because of the variety of images I did not have time to optimize the parameters each time and kept them constant unless performance suffered. The summary of the results are below. The accuracy was 92.7%, which is above the range we set to validate our subsystem.

		# lesions detected	733
		TP Hit	691
		TN Correct Rejection	718
		FP False detect	42
		FN Miss	69
		P Expected / Counted	760
		ACC Accuracy	92.70%
		PPV Precision	94.27%
		TPR Correct detection rate	90.92%
		FDR False detection rate	5.73%

Figure 7: Some examples and the summary of results of our lesion detection.

3.4. Skin Lesion Classification

For skin lesion classification we looked at 31 total lesions from 7 different skins. There are two classification, dark and red, as well as a default classification that says it is too similar to the average skin color. Just like for detection, a manual classification was made and then compared to the computed result. Below are the tables for our result. For dark lesions we have an accuracy of 87% and red has an accuracy of 91%, both meeting our goal for validation.



Figure 8: Sample output of lesion classification.

image #	#dark	# Hits	# Miss ID	# Miss	TP	Dark hits	13
0	1	1	0	0	TN	Correct rejection	15
1	1	1	0	0	FP	False ID	3
2	3	3	0	0	FN	Dark misses	1
3	5	5	0	0			
4	4	1	3	0	ACC	Accuracy	0.8750
5	2	2	0	1	PPV	Precision	0.9286
6	0	0	0	0			

image #	# red	# Hits	# Miss ID	# Miss	TP	Red hits	9
0	2	2	0	0	TN	Correct rejection	22
1	2	2	0	0	FP	False ID	0
2	0	0	0	0	FN	Red misses	3
3	1	1	0	0			
4	0	0	0	3	ACC	Accuracy	0.9118
5	1	1	0	0	PPV	Precision	0.75
6	3	3	0	0			

Figure 9: Result for red and dark lesion classification.

4) Conclusion

These processes to extract health metrics from images uses simple and middle level techniques. Overall these simple processes worked fairly well, such as the case of thermal image processing or skin detection. But while these simple techniques are powerful and seemed to be adequate at the time, it is now obvious more advanced techniques are out there for doing some of what we were trying to accomplish, such as with skin lesion classification. Thus the main improvement I would of made would have been having put more focus on using machine learning for some of the processes.

Another improvement that could have been made would have been with the data. There was a lack of open source data base for skin lesions or high quality faces with lesions. This restricted testing to a relatively small amount of images and people. Ideally there would perhaps been more effort to secure access to medical skin lesion archives out there, or there could have been a large effort on our end to recruit testers. A very large amount of additional data would have been preferred, especially if I had wanted to do some machine learning techniques.

An additional dimension to this analysis could have been one of health metrics themselves in the spirit of “big data” and trying to find trends that are hidden to simple human analysis. This certainly would have been impressive and made the mirror more helpful as a device, however the testing and validation may have been immense as you have to collect data for at least one person over a long time, more than the time of this project.

Finally, there should have been research or experiments involving the ways these data metrics could correlate with medical information. For example, we could approximate skin color over time and assume if the color gets “paler” that might be an indication of sickness. However in order to make that assumption correctly would require some research or even experiments to see if there is an actual correlation there.

Sources

- 1) [FLIR Lepton Datasheet](#)
- 2) [Temperature Of A Healthy Human \(Skin Temperature\)](#)
- 3) [FLIR Energy Loss Around Window \(Thermal Window Image\)](#)
- 4) [Automatic Detection of Melanin Spots in Atlantic Salmon Fillets](#)
- 5) Lesion Skin Sources
 - 5.1. [A Review of Acne in Ethnic Skin](#)
 - 5.2. [News-Medical](#)
 - 5.3. [Acne: American Osteopathic College of Dermatology](#)
 - 5.4. [Heathline](#)
 - 5.5. [Healthnbodytips](#)
 - 5.6. [Acne.org](#)
 - 5.7. [IDOJ](#)
- 6) [Acne image analysis: lesion localization and classification](#)
- 7) [Hand Detection using multiple proposals](#)