SMART MIRROR

Nicole Lemieux
Austin Pursley
James Quinn
Amanda Wolken

SYSTEM OVERVIEW

Austin Pursley

04/28/2018

Contents

1) Image Processing: Overview

This subsystem processes images has two subroutines for each type of image: thermal and color. Each subroutine uses a series of image processing functions to extract health metrics. The thermal image subroutine finds the skin temperature of the user and the color image subroutine finds skin color and characteristics about skin lesions such as moles or acne.

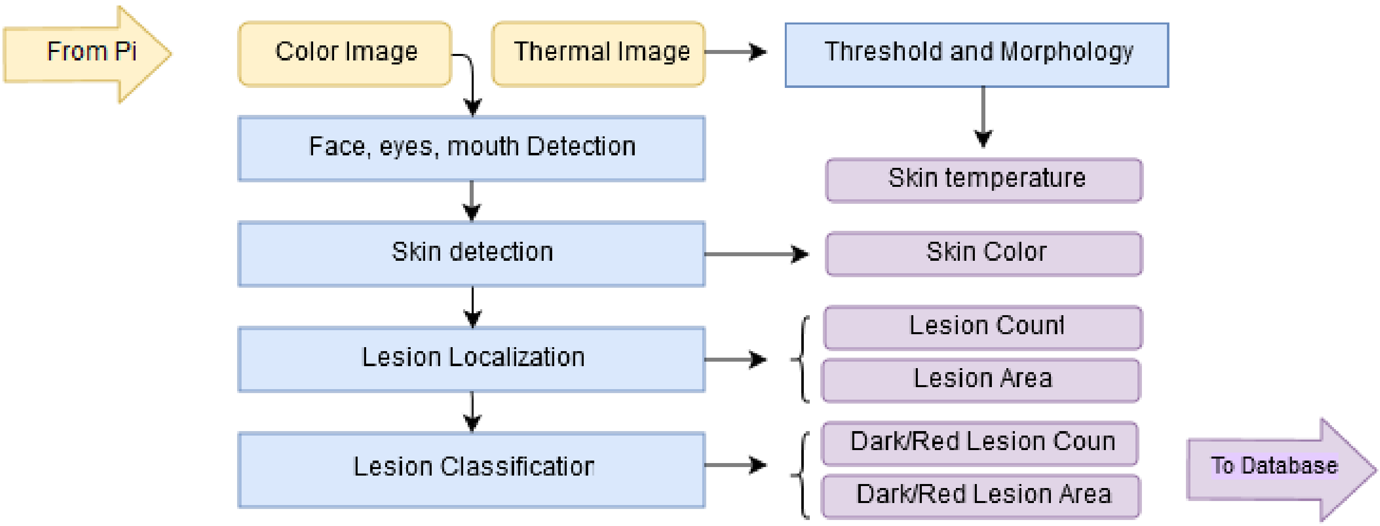


Figure 1: Overview diagram of image processing subsystem.

2) Design

2.1. Thermal Image Processing

The object was to accurately and consistently read the skin temperature intensity. The overall idea for this subroutine was to isolate the user's profile and take the highest intensity value from that reading. The FLIR Lepton thermal camera captures 80x69 images and the optimal operating temperature range is 14 °F to +149 °F (1). In general it is assumed that the coldest area will be the surface of an object in the background and the hottest area will be the user's skin. However there will be processing in place to deal with hot or cold objects in the background.

Step 1: Threshold

The first image processing function this subroutine performs is a threshold to exclude the colder section of the image. This creates a binary image where pixels with an intensity higher than the threshold value with be white and those lower will be black. This is the "mask", the area of the image that should approximate the profile of the user. This threshold value must be high enough to exclude the background but not so high that it could exclude the face. This function should also remove from the profile any cold objects, such as ice drinks in copper mugs or a window on a cold day.

The input for this threshold function starts as a text file that contains the matrix of thermal values that are integers around 8000-8500. This is converted into a OpenCV Mat object. During development I tested three types of threshold techniques: Otsu's method, a hard-coded threshold value, 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, this creates problems because OpenCV Otsu's function only works when the image values are normalized between 0-255. When these thermal images are normalized to 0-255, different images will have different values mapped to 0 and 255. This results in Otsu's method failing the case with a simulated hot spot in the background; the mask will only be the hot spot behind because it results in the highest variance. Then it will be eroded away by the morphological operations (Figure 2, the black square). Another method I tried was a hard-coded threshold value found with experimenting. This works but 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 and testing proved it had consistent results.

Figure 2: Simulated "hot spot" in the left corner. Due to constraints with OpenCV, Otsu's threshold does not work.

Step 2: Morphology: Open and Close

OpenCV's Morphological functions are used to adjust binary images. A morphology close operation will fill in gaps in the mask that may come from glasses or beards. A morphological open is the next function and will remove background objects that are hot, such as light bulbs or hot drinks.

Step 3: Bitwise AND, Finding Max Skin Temperature

What's left of the binary 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.

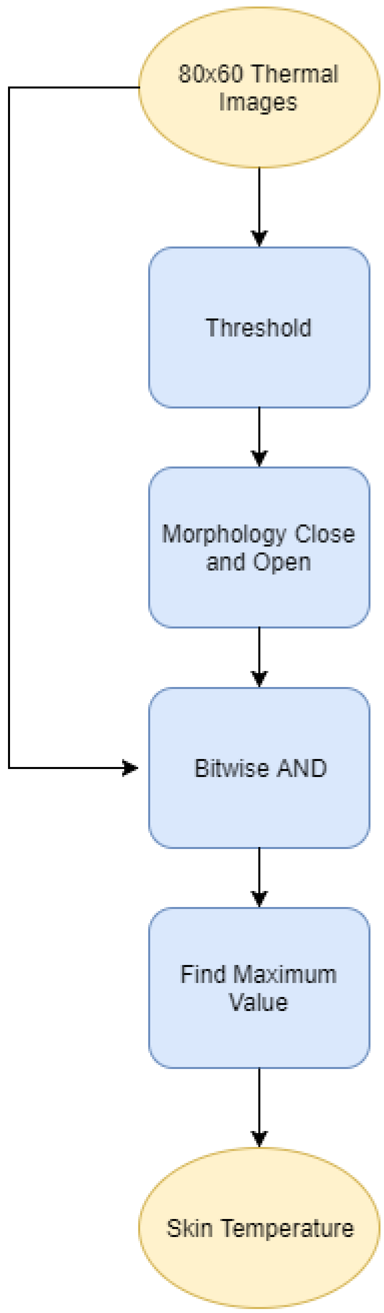


Figure 3: Step-by-step for the thermal image skin temperature subroutine

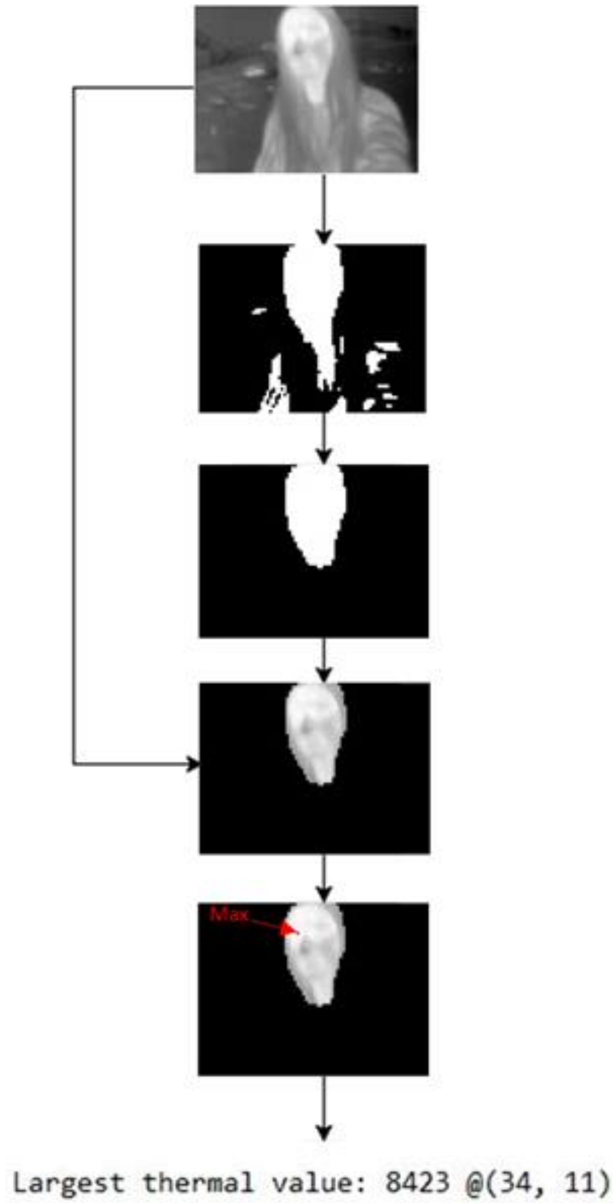


Figure 4: Example results

Corner Cases of Thermal Image Processing

Glasses and facial hair obstruct the thermal camera and result in cold spots. Because our metric analysis over time will be comparative, this will only be fine if the user consistently wears glasses or has 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. 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

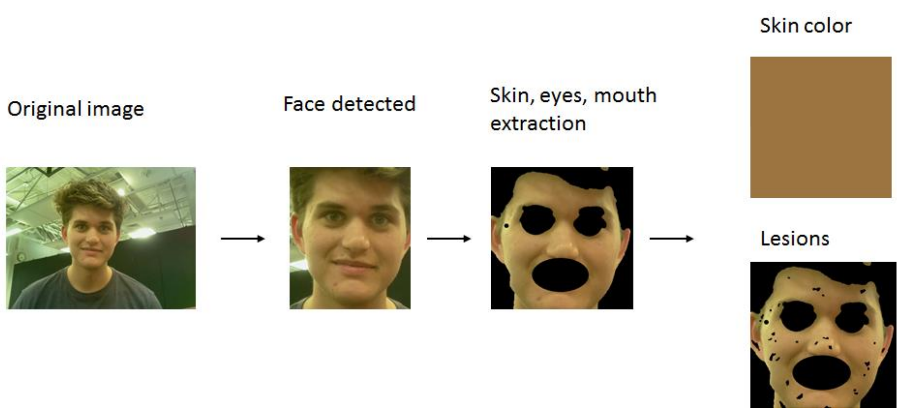


Figure 6: Broad overview of the color image subroutine

2.2.1. Face and Face Feature Detection

Color image processing starts with wanting to find the face of the user, 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. neural networks). The output in the program is the ROI boxes for the face, two eyes, and mouth.

2.2.2. Skin Detection

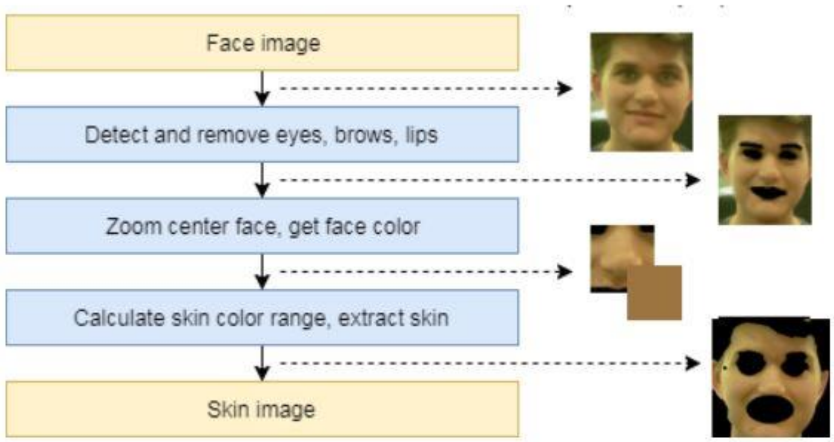


Figure 7: Skin detection process overview.

The idea behind skin detection is that if you know where the face is then you can have an approximation of what the rest of the skin looks like. This process was inspired by a process found in skin detection process in a paper on hand detection (7). After face detection, 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 range of skin colors. 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 artifacts such as hair or background objects are a similar color to the user’s face, the process will include those artifacts with 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. The steps for the design is:

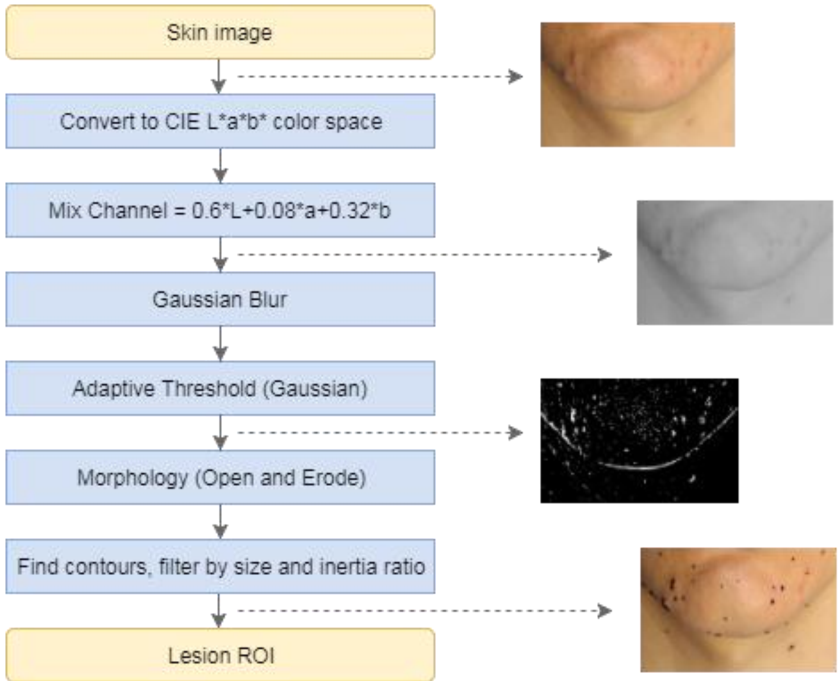


Figure 9: Overview of lesion detection process.

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. With OpenCV you can change the color space of the image through a series of operations. The first step 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. Finally, a Gaussian blur is applied to reduce noise.

I had to do a lot of experimentation to determine the best color space. Originally it was as simple as removing the blue channel, which seemed to contain a lot noise. Then I discovered the L*a*b was better for lesion detection than the other color spaces (RGB, HSV, and YcrCb). Later I found 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.

Step 2: Adaptive Threshold

I discussed Otsu's method in the skin color section, 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 adaptive thresholding i.e. applying 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 we can analyze each one individually. OpenCV has a function for finding the contours of binary images. After finding the contours there will likely be some that do not belong, so the contours are filtered. OpenCV can give certain characteristics of contours, such as size 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 lesion 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 is that it effectively detects 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 could be addressed doing two rounds of lesion detection, one with the normal image and the other inverted. Another fix could be changing the coefficients for mixing the image channels, having a different mix for different kinds of lesions. However this would lead to having to perform the process at least twice.

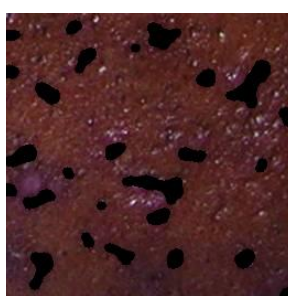


Figure 2: Notice that light, red spots are not detected on darker background. Those same spots on lighter skin would probably be detected

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 surrounding 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. Due to ignorance and time constraints, I unfortunately did not get to apply machine learning for this classification.

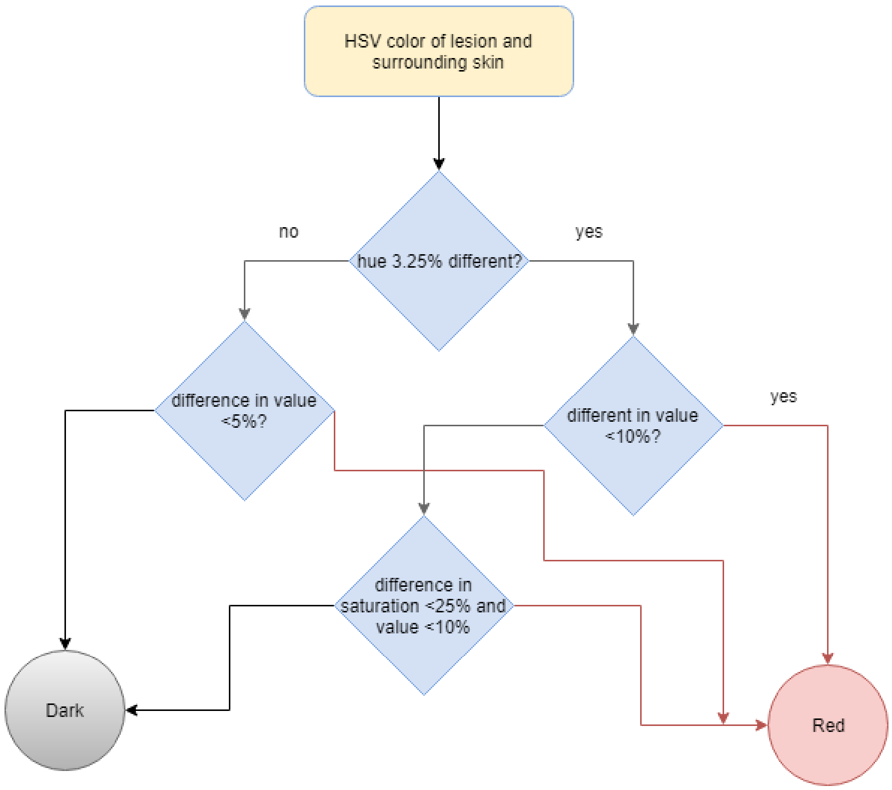


Figure 3: Overview for lesion classification.

3) Validation and Testing

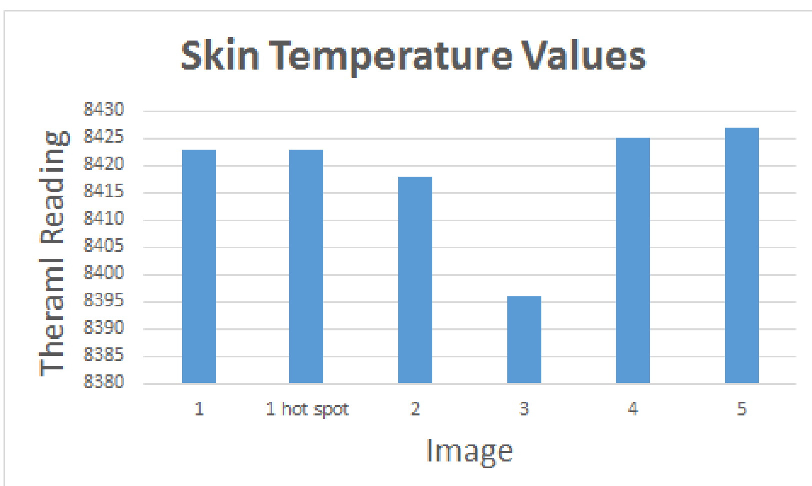
3.1. Skin Temperature

Six images taken with our FLIR Lepton were 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 (Figure 4, Table 1, Figure 5). 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 8 images of celebrities found on the web (Figure 6). 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. I determined the accuracy to be 13/15, or 86.6%.

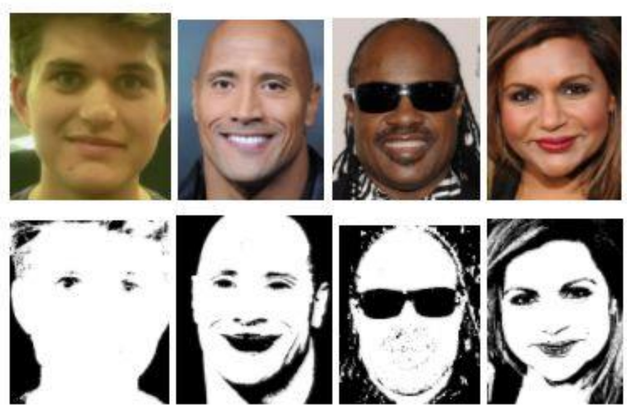


Figure 6: Sample output of skin detection results

3.3. Skin Lesion Detection

For lesion detection testing I used a picture of my own face and 20 others found through online sources (5). These are from a range of different skin types, conditions in lighting, severity of skin condition, and angles. Each face image was manually broken down into 143 smaller images for validation. I manually identified and counted lesions to compare with the program’s result. I kept track of how many lesions the program had correctly identified (hit), how many it falsely detected, and how many it had missed. Because of the variety of images. The summary of the results are in Figure 7. 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. Results are shown in Figure 8 and 9. 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			
0	1	1	0	0	TP	Dark hits	13
1	1	1	0	0	TN	Correct rejection	15
2	3	3	0	0	FP	False ID	3
3	5	5	0	0	FN	Dark misses	1
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			
0	2	2	0	0	TP	Red hits	9
1	2	2	0	0	TN	Correct rejection	22
2	0	0	0	0	FP	False ID	0
3	1	1	0	0	FN	Red misses	3
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 I was trying to accomplish, such as with skin lesion classification. Thus the main improvement I would of made would have been having more focus on using machine learning for some of the processes.

I also wanted better 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 collect our own data. 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 difficult as you have to collect data for at least one person over a long time.

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 / experiments.

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