Contents

List of Figures

List of Tables

1 Introduction

We often cannot judge based on the appearance of a product worn by a model how that product will actually look on ourselves. Compared to users, models may have different skin colour and other bodily features. Ideally, to help the user make a better informed purchasing decision, the model demonstrating the product should be customized to each user. To achieve this, there are already many applications developed for the virtual trying on of products in the beauty, cosmetics and garment industries which digitally modify the images of models to take on the appearance of the user [4] [5, 6].

For this project, we have a video in a mobile app demonstrating nail polish on a model hand to allow virtual try-on of different nail polish colours. We would like to edit this demo video so that the model appears to have the user's exact skin colour, to help the user better determine whether the nail polish colours look pleasant on the user's own hand. While it is possible to manually prepare a series of demo videos with models of different skin colours, preparing even a single video for virtual try-on is an extremely time intensive task. Moreover, each person has a particular skin colour and it's preferable to be able to tailor the video exactly to the skin colour of the user while the user is using the app.

To address these challenges, we propose developing an algorithm to incorporate into the app that quickly and automatically performs the image editing task. The user should be able to provide an image of their own hand as input, and the resulting edited model hand should be convincing and accurate to the skin colour of the provided user image. A wide range of user skin colours should be supported by a single model of mid-toned skin. Finally, the process should be quick to run on a mobile device, such that the user notices no significant time lag to see the resulting video frames upon inputting their own skin colour.

Currently, we aren't aware of an existing algorithm that satisfies all our specific requirements. While there has been a large body of work done addressing transfer of colour between images in general [7, 8, 9, 10, 11], only a smaller set of work specifically addresses transfer of skin colour [12, 13, 14]. All such studies address face skin colour rather than hand skin colour, which often means that more of the study is devoted to handling colour transfer of different, more complex aspects of the face [14]. Skin colour transfer of the whole body are sometimes performed in other, more general imaging processing applications, but in those cases, since the skin colour transfer is often only a small part of the whole process, the algorithm used is often relatively simple and not heavily designed for achieving accuracy to the user skin colour [5, 6]. In the related field of skin colour enhancement applications, the methods used often are not meant to make large changes to the user skin colour [15, 16]. Finally, algorithms developed by most of the prior studies do not appear to be meant for use with the limited resources on a mobile device.

Our goal is to develop a mobile compatible recolouring algorithm that would satisfy our requirements. As sub-objectives, we would like to first develop an effective algorithm and then optimize the algorithm's running time. We will focus solely on achieving convincing colour transfer in the algorithm, and assume that the location of the skin in the images

are already determined by an another process.

For developing and testing the algorithm, we will be using the OpenCV library in C++. OpenCV has a wide range of image processing tools and code in C++ should be easy to optimize and port to mobile platforms. We will develop the algorithm on a desktop computer, to allow for faster and easier testing, before we port the code to mobile. Our approach will be to test different algorithms on variety of hand images, starting with a naive approach and developing improved versions of the algorithm based on the results after each each iteration.

2 Background and Literature Review

2.1 Survey of methods for changing and matching skin colour in Photoshop

The are a wide range of online video tutorials available for adjusting human skin tone in individual images using Photoshop. The purposes of these videos include giving the subject of an image the appearance of a tan, matching the skin tone of the subject to a desired skin tone on another individual, or matching the skin tone of a subject's face to the rest of the subject's body, which is often a slightly different colour [17] [3] [2]. We surveyed a range of these videos and summarize the techniques of the most relevant videos with reasonably realistic results. See Appendix A for a more detailed description of three of these Photoshop processes.

Summary of Photoshop techniques

Levels and curves are frequently used for small brightness adjustments [1] [2] [3]. For large brightness adjustments, one technique was found (see Appendix A.1), where the skin area is brightened in a conversion to black and white, and then the luminosity blend mode is used to place the colour back into the image. Sometimes highlights and shadows are adjusted separately; curves or the "blend if" function (which blends in an effect only if the original pixel is above a certain threshold of brightness) can be used to achieve this effect [17].

There are many different methods to match colour, and the colour can be adjusted separately from the brightness or simultaneously - often one would affect the other [2] [3]. Methods for matching colour include matching the ratios of cyan, magenta and yellow by making adjustments with the selective colour tool, or using curves or levels on individual colour channels. Adjustments are made either by eye or to numerically match a target color [18] [2] [3]. Often to reduce the vividness of the colour adjustments the saturation must be slightly decreased [1] [2].

After all other effects are applied, the opacity of the overall effect is often reduced from 100% for a more natural appearance [1] [2].

Limitations of Photoshop techniques

These Photoshop techniques are generally meant to be tailored to each specific image that a human is adjusting. There are many junctures where the specific numerical amount of an adjustment often have to be judged by eye. While Photoshop has a method for automating processes using actions, the processes are meant for increasing ease of use by artists who can make additional adjustments and are familiar with the tool, rather than in commercial applications where the process is entirely automated [19].

Another limitation is that Photoshop operates at a higher level of abstraction than image

processing software making use of libraries such as OpenCV. Image processing code has much more control over processes that can be applied to images, and the regions on the image that processes are applied to.

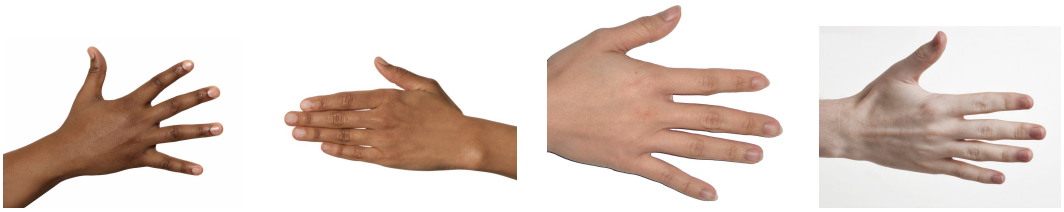
Finally, some Photoshop effects may be proprietary and are of course limited to the platforms that Photoshop supports, while a program developed with a platform such as OpenCV can be made open source and adapted to uses on a variety of different platforms.

3 Progress to Date

To accomplish the objective of recolouring the skin tone of a hand to a target colour, we wrote algorithms in C++ in Eclipse on OS X using OpenCV libraries. Eclipse is used to compile each iteration of the algorithm into a debug-mode executable program named Recolor. For ease of testing, as the algorithm is modified, we add more functionality to the Recolor program and retain the ability to use previous versions of the algorithm. We use a custom Python script to run new versions of Recolor from the terminal to test it. All of the relevant code and its versions are hosted on a git repository at https://github.com/tiantianhan/recolor

Recolor takes as input a hand image, a mask instructing it where to find the average skin colour of the hand, and a desired target skin colour. (Other flags and inputs are also used for testing purposes, see the Github repository readme file for a full description of the usage.) Recolor then outputs the processed image where the skin tone is adjusted to the target colour.

We iterated from simple to more complex algorithms, at each step testing the algorithm and evaluating the results. We tested progressive iterations on a set of hand images with varying skin tones. The images are are shown in Figure 1.



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

Figure 1: Different hand images used for testing

For each test, we called the Recolor program to transform the image of one hand to have the skin tone of the hand in another image, then visually compared the processed image to the image of the target hand. We performed the process on all possible combinations of our test images, paying particular attention to the extreme cases, such as transforming from Figure 1a to Figure 1d and vice versa, as well as cases that start with a hand with mid-tone skin such as in Figure 1b (as this is the most likely use case for applications that change the image of a model's hand to match a range of skin tones). We evaluated the resulting images subjectively, based on whether the processed hand looks believably like a hand naturally of that skin tone, and noted any flaws that we then attempted to correct with the next iteration of the algorithm.

In the following subsections we summarize the results of each algorithm and our evaluation of the results.

3.1 Simple brightness addition / subtraction

Algorithm

To begin, we performed a simple addition of a value to each of the rgb channels of the hand, such that the average colour of the hand in the processed image is equal to the average colour of the hand in the target image. The algorithm is shown in Equation 1.

$$r' = r + \delta_r \tag{1}$$

Where

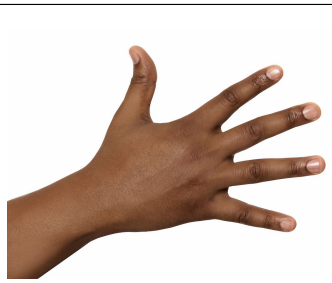
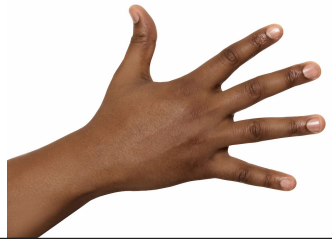
$$\delta_r = \bar{r_t} - \bar{r}$$

With the same equation applying for the g and b channels.

Results

The complete results are shown in Table 7 in Appendix B, and a portion is shown here for convenience.

Portion of test results of simple addition / subtraction brightening function from Table 7 in Appendix B

No.	Original	Target	Result
7.2			
7.4			

7.5			
7.6			

Evaluation

Images of darker skin tones and smaller changes from the original skin tone to the target colour to begin with (Row 7.4) tend to have better results than images with large changes (Rows 7.2, 7.6). In the case of changes towards brighter colours, this is because large changes force bright points in the original image to be truncated at white, and also causes dark regions on the image, such as shadows and grooves, to become significantly brighter and less close to true black, giving the image a "high-key" appearance (Row 7.2 and 7.5).

In addition, we noted that at this stage the transformation from a dark coloured hand to a very pale hand, or even from a mid-toned hand to a pale hand and vice versa is especially unconvincing. (Row 7.6, also see 7.3 and 7.10)

3.2 Proportional adjustment relative to average color

Algorithm

To correct for the effect of the bright spots in the image being over bright and the high-key appearance resulting from all the shadows being brightened, we used an algorithm that maps the black and white points of the image to the same value, and adjusts the colours in between to match the target average colour. The algorithm is shown in Equation 2.

$$r' = \begin{cases} \left(\frac{\bar{r}_t}{\bar{r}}\right) r, & \text{for } r \leq \bar{r} \\ 255 - \left(\frac{255 - \bar{r}_t}{255 - \bar{r}}\right)(255 - r), & \text{for } r > \bar{r} \end{cases} \tag{2}$$

With the same equation applying for the g and b channels.

Results

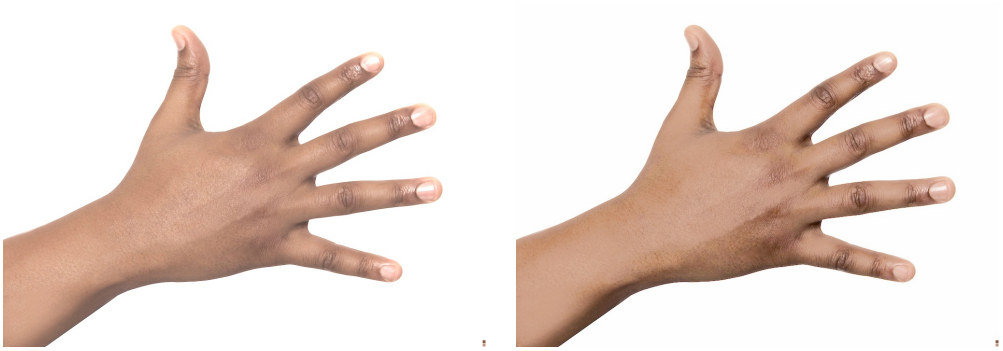
The complete results are shown in Table 8 in Appendix C, and a portion is shown here for convenience.

Portion of test results of adjusting proportionally based on distance of color to the average from Table 8 in the Appendix C

No.	Original	Target	Results
8.2			
8.4			
8.5			
8.6			

Evaluation

This method improved the appearance of cases with over-bright spots or "high-key" appearance issues, as Figure 2 shows:



(a) Simple addition algorithm (Equation 1)result

(b) Proportional adjustment algorithm (Equation 2) result

Figure 2: Comparison of algorithms 1 and 2 results for transforming a dark hand (Figure 1a) to a light hand (Figure 1c).

We noted however, that this method noticeably does not correct for, and even exacerbates slightly relative to the simple addition algorithm the dark spots at the joints and creases of a hand of darker skin tone when it is transformed to a lighter skin tone (Row 8.5). Other results are similar to the results of the simple addition algorithm.

3.3 Proportional adjustment with dark spot correction

Algorithm

We attempted to correct the dark spot issue by significantly reducing the absolute difference between dark pixels and the average colour, ensuring that the dark spots would instead have colours close to the average. We perform this correction on the output of the proportional adjustment algorithm.

$$r'' = \begin{cases} \bar{r'} - \dfrac{(\bar{r'} - r')}{\alpha}, & \text{for } r' < \bar{r'} \\ r', & \text{for } r' \geq \bar{r'} \end{cases} \tag{3}$$

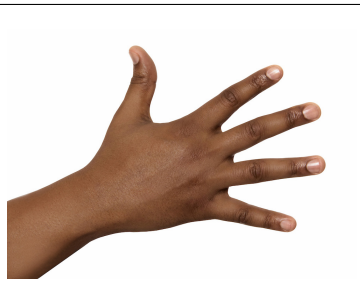
Where α is a constant, $\alpha > 1$. The same equation applies for the g and b channels.

Results

See Tables in Appendix D, for the complete results for $\alpha = 1.1$. A portion of the results is reproduced here for convenience.

Test results of proportional adjusting with correction for dark spots, $\alpha = 1.1$ from Table 9 in the Appendix D

No.	Original	Target	Results

9.1			
9.4			
9.5			
9.6			

Evaluation

As shown in Figure 3, the dark spots and creases noted in Section 3.2 are reduced.

(a) Proportional adjustment algorithm (Equation 2)result

(b) Proportional adjustment algorithm with correction (Equation 3) result

Figure 3: Comparison of algorithms 2 and 3 results for transforming a mid-toned hand (Figure 1b) to a light hand (Figure 1c).

We tried this effect for a range of α and found that $\alpha = 1.1$ gives an acceptably realistic result. A larger α would further reduce the dark spots on the skin but may begin to strongly brighten the shadows of the image, resulting in an unrealistic effect.

Up to the current iteration the more extreme changes of luminosity, such as from Figure 1a to Figure 1d and vice versa are especially unrealistic. Part of the reason is that the shadows, most prominent in Figure 1d may be causing the average colour of the entire hand to be of lower luminosity than it should be.

4 Future Work

References

[1] J. Shaver. (2014) How to change a person's skin color from dark to light in photoshop. [Online]. Available: http://www.designpanoply.com/blog/how-to-change-a-persons-skin-color-from-dark-to-light-in-photoshop

[2] Phlearn. (2014) How to match skin tones in photoshop. [Online]. Available: https://www.youtube.com/watch?v=5-DE49Muc0g

[3] PiXimperfect. (2017) How to match skin tone in photoshop cc 2017. [Online]. Available: https://www.youtube.com/watch?v=zEeknlHydfE

[4] T. Zhang, L. Cao, and W. Y. Wang, "The impact of virtual try-on image interaction technology on online shoppers' purchase decision," in *Proceedings of the 8th International Conference on E-Education, E-Business, E-Management and E-Learning*, ser. IC4E '17. New York, NY, USA: ACM, 2017, pp. 6–10.

[5] R. Shilkrot, D. Cohen-Or, A. Shamir, and L. Liu, "Garment personalization via identity transfer," *IEEE Computer Graphics and Applications*, vol. 33, no. 4, pp. 62–72, July 2013.

[6] L. Jun, Z. Mingmin, and P. Zhigeng, "Virtual try-on by replacing the person in image," *J. Comput. Aided Des. Comput. Graph. (China)*, vol. 27, no. 9, pp. 1694 – 700, Sept. 2015.

[7] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics Applications*, vol. 21, no. 5, pp. 34–41, Sep. 2001.

[8] F. Pitie, A. C. Kokaram, and R. Dahyot, "N-dimensional probability density function transfer and its application to colour transfer," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, 2005, pp. 1434 – 1439.

[9] X. Chen, D. Zou, J. Li, X. Cao, Q. Zhao, and H. Zhang, "Sparse dictionary learning for edit propagation of high-resolution images," in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA, 2014, pp. 2854 – 61.

[10] H. Chang, O. Fried, Y. Liu, S. DiVerdi, and A. Finkelstein, "Palette-based photo recoloring," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 139:1–139:11, Jul. 2015.

[11] Q. Zhang, C. Xiao, H. Sun, and F. Tang, "Palette-based image recoloring using color decomposition optimization," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1952 – 1964, 2017.

[12] L. Yin, J. Jia, and J. Morrissey, "Towards race-related face identification: Research on skin color transfer," in *Proceedings - Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, Seoul, Korea, Republic of, 2004, pp. 362 – 368.

[13] K. C. Seo, G. Shin, and S. W. Lee, "Interactive transfer of human facial color," in *Advances in Multimedia Information Processing - PCM 2005: 6th Pacific Rim Conference on Multimedia, Jeju Island, Korea, November 13-16, 2005, Proceedings, Part II*, Y.-S. Ho and H.-J. Kim, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 993–1003.

[14] Y. Yang, H. Zhao, L. You, R. Tu, X. Wu, and X. Jin, "Semantic portrait color transfer with internet images," *Multimed. Tools Appl. (Germany)*, vol. 76, no. 1, pp. 523 – 41, 2017.

[15] H. Aradhye, G. D. Toderici, and J. Yagnik, "Adaptive, selective, automatic tonal enhancement of faces," in *Proceedings of the 17th ACM International Conference on Multimedia*, ser. 2009. New York, NY, USA: ACM, 2009, pp. 677–680.

[16] J.-H. Lee, J. Heo, and Y.-S. Ho, "Skin tone enhancement and background change for mobile phones," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp. 2719 – 2726, 2010.

[17] Phlearn. (2016) How to get a tan in photoshop. [Online]. Available: https://www.youtube.com/watch?v=Har8CSjei2k

[18] M. Woloszynowicz. (2014) How to accurately match skin tones using selective color in photoshop. [Online]. Available: https://www.youtube.com/watch?v=IOYFhumpmv8

[19] ——. (2014) How to easily correct colors and match tones in photoshop. [Online]. Available: http://www.vibrantshot.com/how-to-easily-correct-colors-and-match-tones-in-photoshop/

A Photoshop techniques for changing skintone from select online video tutorials

A.1 Changing skin colour from dark to light [1]

Effect

Table 4: Screen captures from Photoshop tutorial for changing skin colour from dark to light.

Original	Result
	

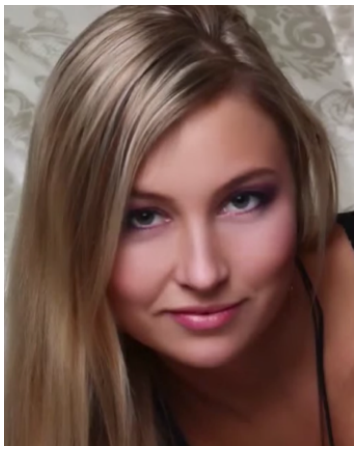
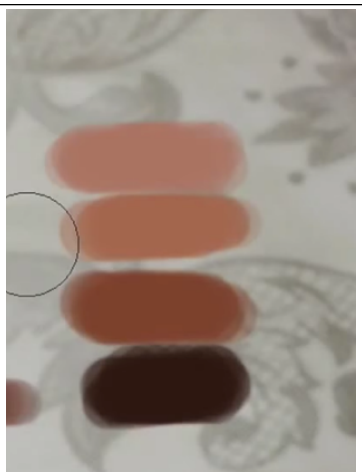
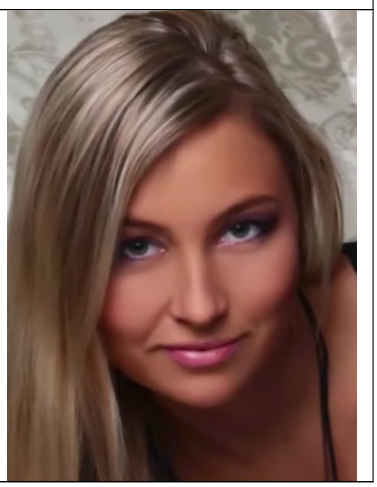
Summary of the process

- Levels adjustment layer - Make gamma value adjustment, adjusting midtones, and adjust the white point for overall brightening effect
- Curves adjustment layer - Reduce highlights resulting from brightening with a custom curve dipping at the highlights
- HSV adjustment layer - Reduce saturation, correcting the oversaturation caused by brightening the image
- Further brightening: obtain greyscale image; boost reds and yellows in the greyscale conversion, brightening the skin area, then use greyscale image to inform the original images luminosity; set this effect to a reduced opacity for a more natural appearance
- Adjust colours by eye with a colour balance layer

A.2 Matching the skintones of face and body [2]

Effect

Table 5: Screen captures from Photoshop tutorial for matching the skintones of face and body.

Original	Target	Result
		

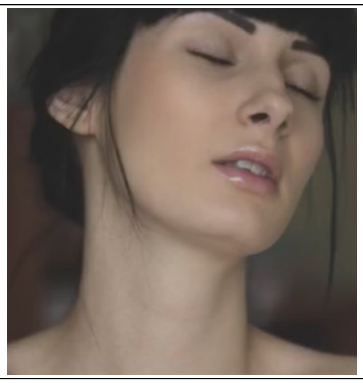
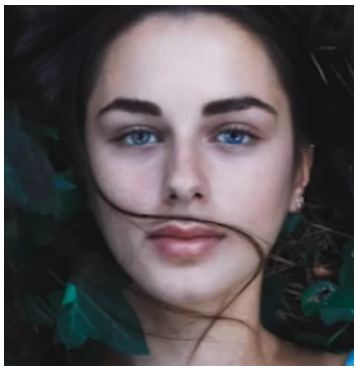
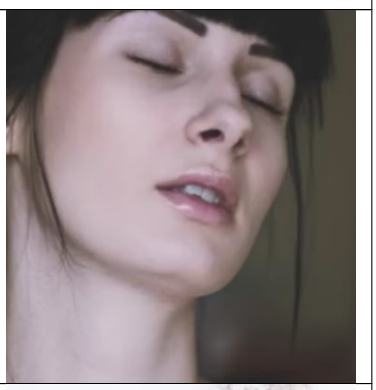
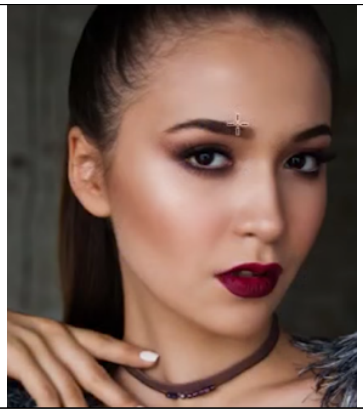
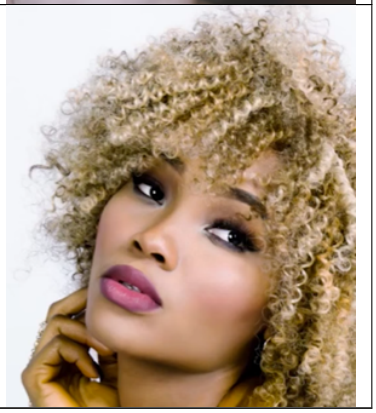
Summary of the process

- Sample a range of colours from face and body (the area with the desired colour) and determine by eye if face should become warmer or cooler, or lighter or darker
- Simultaneously adjust brightness and colour with levels adjustment for each colour channel, adjusting the output and input white and black points
  - For darker and warmer colours, add yellow and magenta to the image highlights while simultaneously dimming the image by lowering the output white points
- Reduce saturation to counteract oversaturation caused by colour adjustment
- Reduce opacity of effect for more natural appearance

A.3 Matching the skintones of portraits of different people [3]

Effect

Table 6: Screen captures from Photoshop tutorial for matching the skintones of portraits of different people.

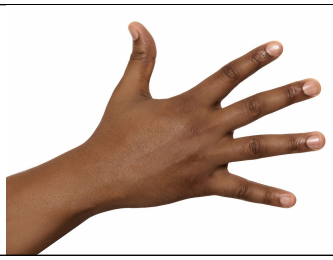
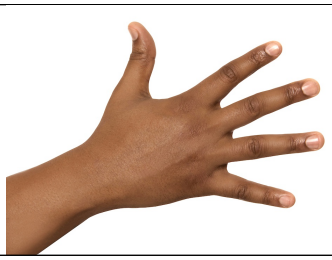
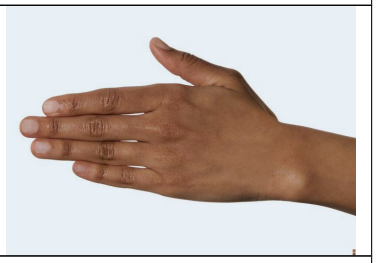
No.	Original	Target	Result
6.1			
6.2			

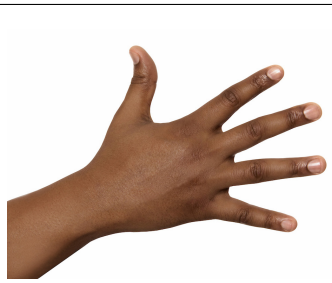
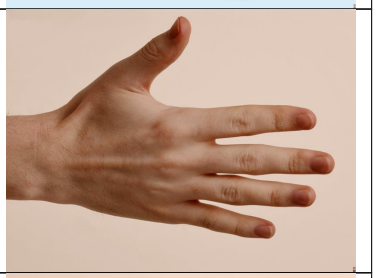
Summary of the process

- For both the target and original image, select an area on face with even skin tone and calculate the average colour. It is important to select the same areas on both images for the effect to work - if the cheek is selected for one person, the cheek should be selected for other person as well
- Using the curves adjustment layer, adjust the curves for each channel, manipulating the point of the original colour and the target colour such that the output of the original color is equal to the target colour
- Make some additional adjustments to the curve to change brightness and contrast
- Sometimes the colour curves will have to be further adjusted by eye; sometimes, different areas of the skin must be adjusted separately what works for the face may not work for the body areas

B Complete results for simple brightening

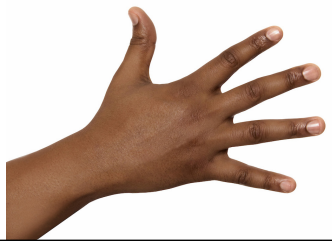
Table 7: Test results of simple addition / subtraction brightening function.

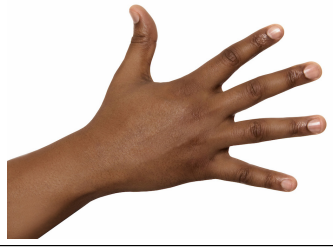
No.	Original	Target	Results
7.1			
7.2			
7.3			
7.4			
7.5			
7.6			

7.7			
7.8			
7.9			
7.10			
7.11			
7.12			

C Complete results for proportional brightness adjustment

Table 8: Test results of brightening proportionally based on distance of color to the average.

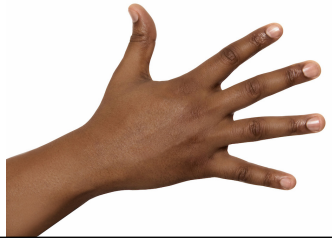
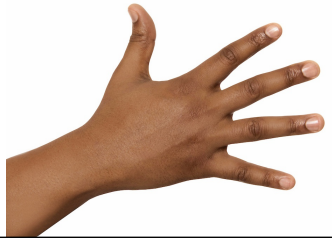
No.	Original	Target	Results
8.1			
8.2			
8.3			
8.4			
8.5			

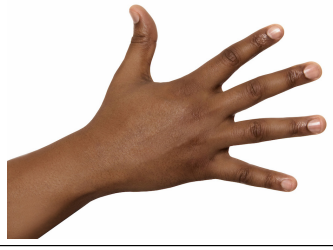
8.6			
8.7			
8.8			
8.9			
8.10			
8.11			

8.12			

D Complete results for proportional adjustment with darkspot correction, $\alpha = 1.1$

Table 9: Test results of proportional brightening with correction for dark spots, alpha = 1.1

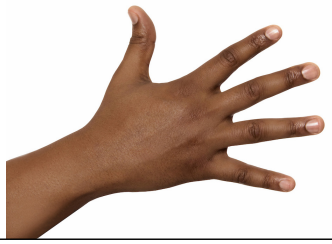
No.	Original	Target	Results
9.1			
9.2			
9.3			
9.4			
9.5			

9.6			
9.7			
9.8			
9.9			
9.10			
9.11			

9.12			

E Complete results for proportional adjustment with darkspot correction, $\alpha = 1.1$, calculating target average color with 5th percentile bright pixels

Table 10

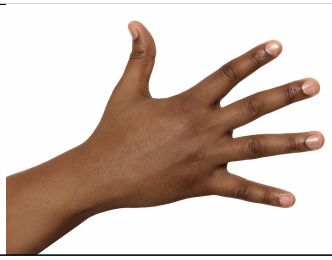
No.	Original	Target	Result
10.1			
10.2			
10.3			
10.4			
10.5			

10.6			
10.7			
10.8			
10.9			
10.10			
10.11			

10.12			

F Complete results for proportional adjustment with darkspot correction, $\alpha = 1.1$, calculating target average color with 10th percentile bright pixels

Table 11

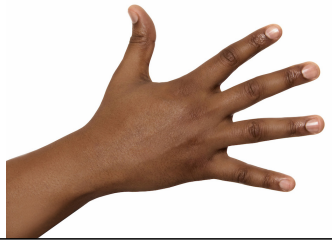
No.	Original	Target	Result
11.1			
11.2			
11.3			
11.4			

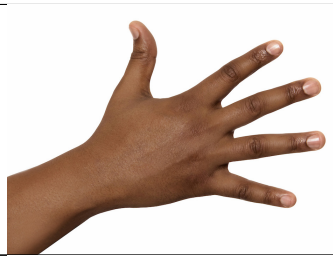
11.5			
11.6			
11.7			
11.8			
11.9			
11.10			

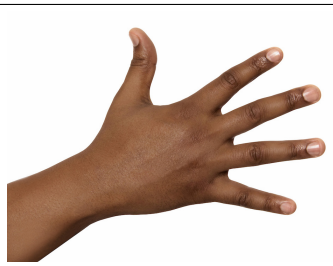
11.11			
11.12			

G Complete results for proportional adjustment with darkspot correction, $\alpha = 1.1$, calculating target average color with 25th percentile bright pixels

Table 12

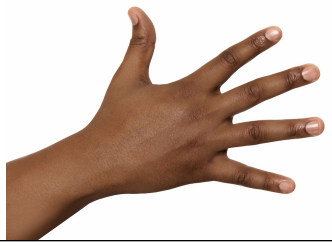
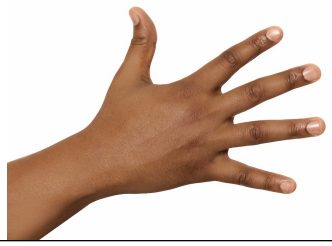
No.	Original	Target	Result
12.1			
12.2			
12.3			

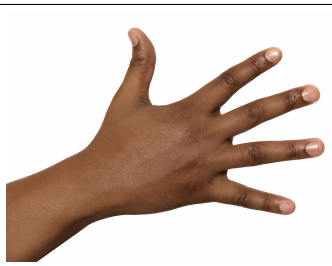
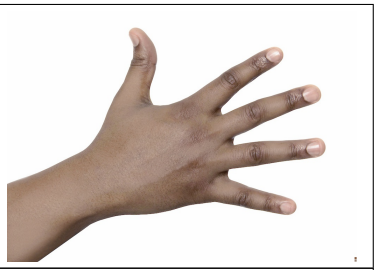
12.4			
12.5			
12.6			
12.7			
12.8			
12.9			

12.10			
12.11			
12.12			

H Complete results for proportional adjustment with darkspot correction, $\alpha = 1.1$, calculating target average color with 100th percentile bright pixels

Table 13

No.	Original	Target	Result
13.1			
13.2			

13.3			
13.4			
13.5			
13.6			
13.7			
13.8			

13.9			
13.10			
13.11			
13.12