Skin Detection & Filters

Joecel Eman Carpio and Floyd Russel Hernandez

INTRODUCTION

Human skin detection, which is usually performed before image processing, is the method of discovering skin-colored pixels and regions that may be of human faces or limbs in videos or photos. Many computer vision approaches have been developed for skin detection. A skin detector usually transforms a given pixel into a suitable color space and then uses a skin classifier to mark the pixel as a skin or a non-skin pixel. A skin classifier explains the decision boundary of the class of a skin color in the color space based on skin-colored pixels. (Majeed, et. al., 2020). Skin detection can be divided into three methods: pixel-based (Nadian and Talebpour 2011), region-based (Zhu and Cai 2011) and hybrid methods (Tan et al. 2012). The pixel-based method classifies each pixel in the picture into skin or non-skin families. The region-based method identifies regions with similar features in the picture. From the perspective of whether to explicitly establish a skin model, skin detection can be divided into machine learning-based methods and traditional methods. Machine learning methods construct a skin detector, generally using supervised methods. Traditional methods generally establish a skin model explicitly. Most of the existing skin detection works are in the medical field, such as the diagnosis of melanoma, skin cancer and other diseases (Khan et al. (2021c) and Khan et al. (2021d)).

DATASET

The application detects faces using cv2.CascadeClassifier which takes in a dataset of Haar Cascades as input. This trained dataset can be accessed from the following github link:

https://github.com/opencv/opencv/tree/master/data/haarcascades

ALGORITHM

Step 1: Reading Webcam Feed

The application reads webcam video using cv2.VideoCapture(0).

Step 2: Face Detection with Haar Cascades / Viola-Jones Face Detection Technique:

The application detects faces using cv2.CascadeClassifier which takes in a dataset of Haar Cascades as input. Using the face detection algorithm, we can apply this on the webcam feed.

Step 3: Mask out Skin Pixels if Face Detected

Once a face has been detected, we can confirm that a human is present. Create a mask for the skin color using a set upper and lower boundary of BGR values. This application was tested with HSV colorspace and BGR color space, after the initial tests we decided to use the BGR color space.

Step 4: Color Shift

Shift the values of pixels that are greater than 0 according to the mask from the previous step. The values are shifted based on the state of the application (red/blue/green). For example, if the "RED" button is pressed skin color is changed to red pixels.

EXPERIMENT RESULTS

Facial Detection Output: Individual Faces and Multiple Faces

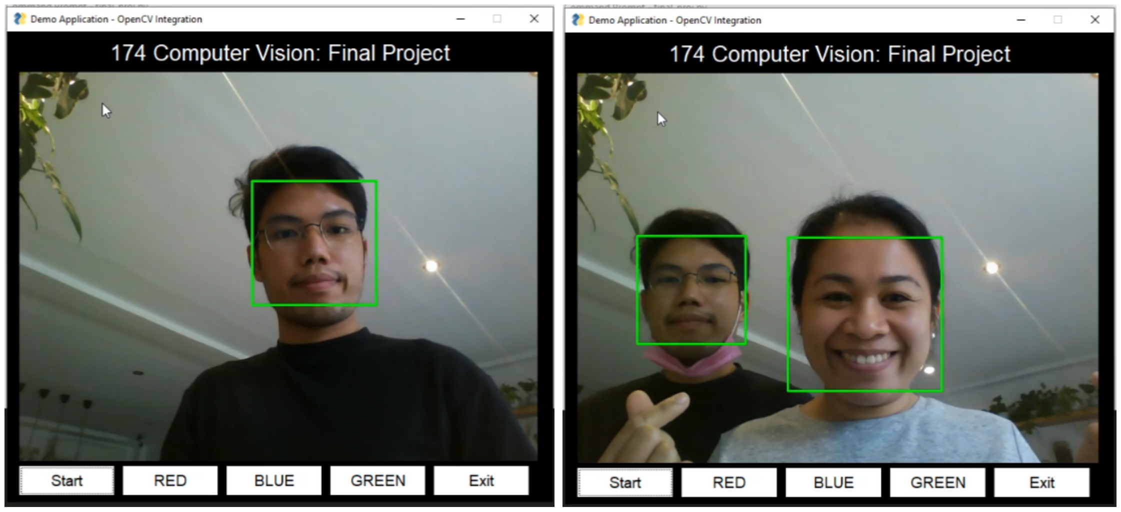
Demo Application - OpenCV Integration
174 Computer Vision: Final Project
Start
RED
BLUE
GREEN
Exit
Demo Application - OpenCV Integration
174 Computer Vision: Final Project
Start
RED
BLUE
GREEN
Exit

Figure 1. Face Detection in Individual and Multiple Faces

Facial Detection Output: Failure to detect face in condition described below.

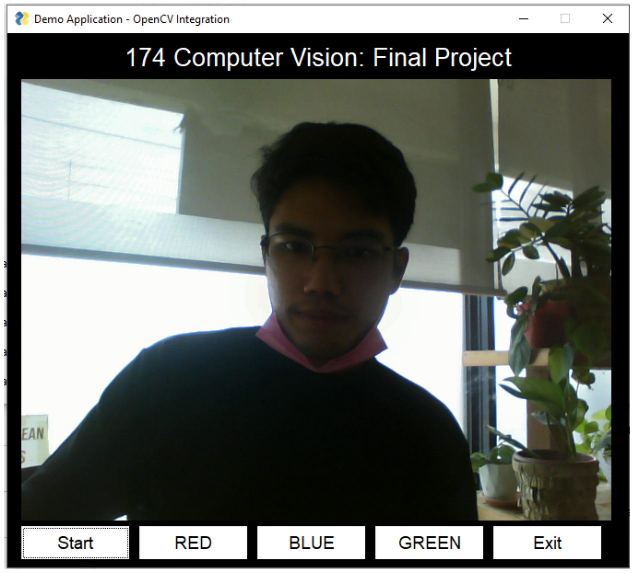
Demo Application - OpenCV Integration
174 Computer Vision: Final Project
Start
RED
BLUE
GREEN
Exit

Figure 2. Screenshot of application output when the human subject is behind a bright background

Skin Filter Application Output:

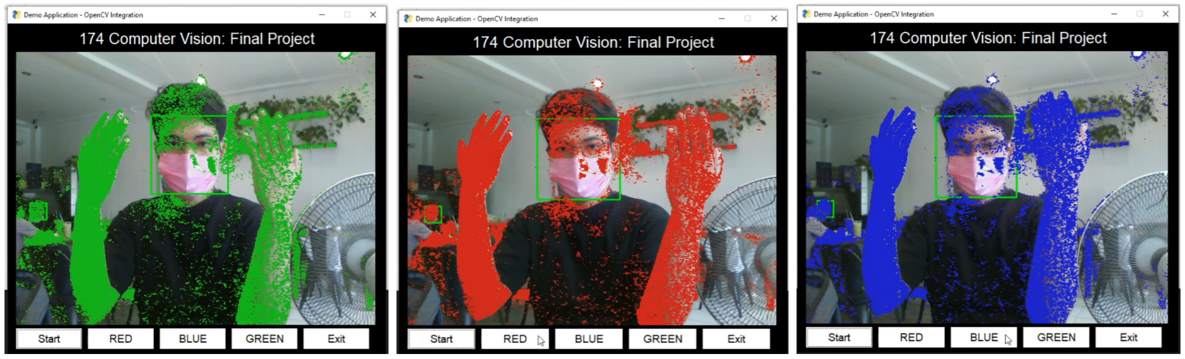
Demo Application - OpenCV Integration
174 Computer Vision: Final Project
Start
RED
BLUE
GREEN
Exit

Figure 3: Skin Filter Application Output in three different colors

The results as shown in the screenshots of the application output when a face has been detected and the button for Green, Red, and Blue has been pressed respectively. The output is able to detect most of the facial features and skin colors in good lighting conditions. One recommendation for similar experiments in the future would be to cross check color spaces for skin colors so that background elements with similar colors are not masked out.

REFERENCES

[1] Face Detection : https://towardsdatascience.com/face-detection-in-2-minutes-using-opencv-python-90f89d7c0f81

[2]Face Detection with Haar Cascade: https://towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd08#:~:text=So%20what%20is%20Haar%20Cascade,Simple%20Features%E2%80%9D%20published%20in%202001.

[3] Dataset of HaarCascades: https://github.com/opencv/opencv/tree/master/data/haarcascades

[4] Majeed, A. H., Mohammed, S. G., &amp; Aldujaili, A. (2020, February). Image segmentation for skin detection. Journal of Southwest Jiaotong University. Retrieved May 31, 2022, from http://jsju.org/index.php/journal/article/view/479

[5] Nadian, A., and A. Talebpour. 2011. Pixel-based skin detection using sinc function. IEEE Symposium on Computers & Informatics. doi: https://doi.org/10.1109/ISCI.2011.5958934. [Crossref], [Google Scholar]

[6] Zhu, J.-Q., and C.-H. Cai. 2011. Region growing based high brightness skin detection. 10th International Symposium on Signals, Circuits and Systems. doi: https://doi.org/10.1109/ISSCS.2011.5978652. [Crossref], [Google Scholar]

[7] Tan, W. R., C. S. Chan, P. Yogarajah, and J. Condell. 2012. A fusion approach for efficient human skin detection. IEEE Transactions on Industrial Informatics 8 (1):138–47. doi:https://doi.org/10.1109/TII.2011.2172451. [Crossref], [Web of Science ®], [Google Scholar]

[8] Khan, M. A., M. Sharif, T. Akram, R. Damaseviciu, and R. Maskeliunas. 2021d. Skin lesion segmentation and multiclass classification using deep learning features and improved moth flame optimization. Diagnostics 11 (5):1–26. doi:https://doi.org/10.3390/diagnostics11050811. [Crossref], [Web of Science ®], [Google Scholar]

[9] Khan, M. A., N. Hussain, A. Majid, M. Alhaisoni, S. A. C. Bukhari, S. Kadry, Y. Nam, and Y.-D. Zhang. 2021b. Classification of positive COVID-19 CT scans using deep learning. Computers, Materials & Continua 66 (3):2923–38. doi:https://doi.org/10.32604/cmc.2021.013191. [Crossref], [Web of Science ®], [Google Scholar]