Implementation of The Stanford FriendBlend Algorithm

Dadarkar, Ahmed
ahmed.dadarkar@gmail.com

Kumar, Rakesh
iamrakesh28@rocketmail.com

December 2020

Contents

1 Introduction

FriendBlend is an algorithm which was developed by students at Stanford University for blending two images with the same background featuring one person in each such that the final image consists of the same background with both people in it. The original algorithm is provided in the report [1].

1.1 Problem Description

Formally, the problem is as follows - given two images as input, each image having exactly one person and both images have the same background, we must

output a single image with both the people in that image and the image must have the same background. The outputted image must be as such that we visually perceive the image as a "true image", i.e. a naturally captured image.

1.2 Motivation and Use Case

The original authors of this algorithm had described a simple but important use case of this algorithm - Two people need to click an image of themselves together with a background, and assuming they do not have anyone else to click one for them, also the image produced by a selfie is not a satisfactory option. The solution to this problem is the FriendBlend algorithm - the two people individually click pictures with the background and then apply this algorithm to obtain an image in which both of them present (and one perceives it as being naturally captured).

1.3 Contributions of this Project

- **An implementation of the *FriendBlend* algorithm which closely matches the original algorithm.**
- **Experimental Results of the *FriendBlend* algorithm**

2 Algorithm

This section provides a brief description of the *FriendBlend* algorithm and discusses the results produced by the steps involved in the algorithm.

2.1 Brief Overview

The algorithm consists of the following steps,

1. **Color Correction:** Corrects the color in both the input images, so that the lighting in both the images is similar. This is done so that there are no artifacts in the blended image due to difference in lighting.
2. **Face and Body Detection:** This detects the face and body of the person in the first and second images. This is done since the steps ahead require the location of the face and body of the people in the images.
3. **Image Alignment:** It is possible that the second image which was taken may not have the exact same alignment as the first image due to reasons like - human error or imperfectness of the camera. Hence, before we proceed to blend the images, they must be aligned so that the blended image does not have artifacts of misalignment.
4. **Image Blending:** In this step, the two aligned images which have the same shape are blended together to form the final output image.

2.2 Color Correction

For this step the original algorithm performed *Contrast Limited Histogram Equalization in Lab Color Space* on both images. But, we had tried out the following two methods,

- Histogram Equalization on Y Channel in YCbCr space on each of the two images.
- Contrast Limited Adaptive Histogram Equalization (CLAHE) on Y Channel in YCbCr space on each of the two images.

For choosing which method to use, we had experimented on the two images one which had the same content except that one was lighter and the other was darker. The results are as follows,

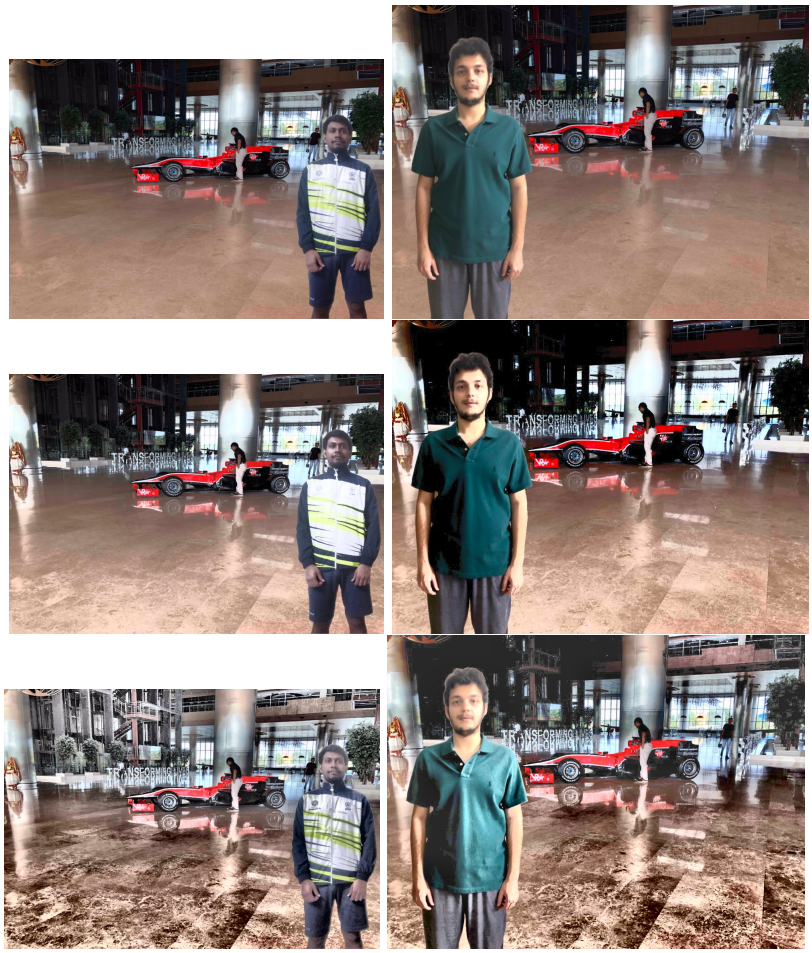

Figure 1: Original Images, Output of Histogram Equalization on Y Channel, Output of CLAHE on Y Channel

Clearly, we can see from figure 1, that Histogram Equalization provides better results, hence we decided to go with it.

2.3 Face and Body Detection

For Face and Body Detection, the original authors used a pre-trained *Haar Cascade* algorithm to first produce a bounding box about the face, and then used intuitive formulae to produce a bounding box about the body, these formulae are as follows,

assuming that the top left corner of the face bounding box is $(r^{face}_{topLeft}, c^{face}_{topLeft})$ and dimensions of the face bounding box is (h^{face}, w^{face}). Then top left and bottom right corner of the bounding box for the body is given as,

$$r^{body}_{topLeft} = r^{face}_{topLeft} - h^{face}$$

$$c^{body}_{topLeft} = c^{face}_{topLeft} - w^{face}$$

$$r^{body}_{bottomRight} = height(image) - 1$$

$$c^{body}_{bottomRight} = c^{face}_{topLeft} + 2w^{face}$$

We applied the exact same algorithm and obtained the following results,

Figure 2: Original Image, Output

Figure 3: Original Image, Output

2.4 Image Alignment

For this step, the original authors performed the following

1. Keypoints are detected in both the images using the ORB (Oriented FAST and Rotated BRIEF) algorithm, and then the keypoints points which lie within the body bounding box are removed since both images do not contain the same person.

2. The found keypoints are matched based on Hamming Distance. (For this step we used Brute Force method and Hamming Distance)

3. From the matched keypoints a transformation matrix (Homography) from the second image to the first image was computed.

4. Then this transformation matrix was applied on the second image to align it with respect to the first one.

We performed these steps and achieved the following results,

Figure 4: Original Image, Slightly Translated and (or) Rotated Image, Output

2.5 Image Blending

For this step, the authors had proposed to use the *Alpha Blending* algorithm if the closest horizontal distance between the bounding boxes is "large" (above or equal to a threshold - we chose 200 pixels), and when the distance is "small" (below the threshold), they had proposed to use *GrabCut* to segment out the person who is assumed to be in the foreground and then place her/him on the other image. For finding out which image had the person in the foreground they compared the area of the head bounding box - the one with the larger area of the head bounding box is the one in the foreground.

We applied this algorithm and found the following results, in the following image both friends are far enough for the algorithm to apply alpha blending

Figure 5: Image 1, Image 2, Image blended using Alpha blending

The following has both friends close enough for the algorithm to apply GrabCut,

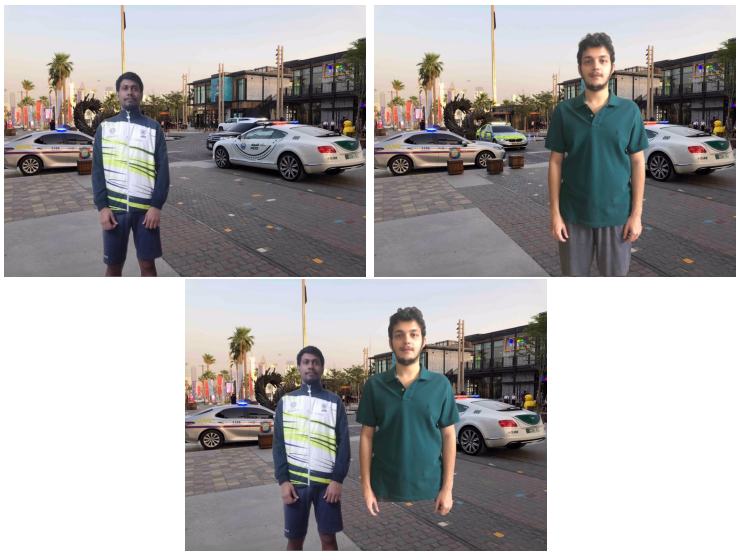

Figure 6: Image 1, Image 2, Image blended using GrabCut

Clearly, directly applying GrabCut lead to bad results.

3 Experiments

In this section we perform the algorithm on two pairs of images - one as part of the first experiment, and the other as part of the second experiment. We discuss each of the intermediate results which are produced by the algorithm.

3.1 Experiment 1

The input images are as follows,

Figure 7: Input Images

The Color Correction outputs are as follows,

Figure 8: Color Correction Outputs

The Face and Body Detection outputs are as follows,

Figure 9: Face and Body Detection Outputs

The detected keypoints are as follows,

Figure 10: Detected Keypoints

Matched keypoints are as follows,

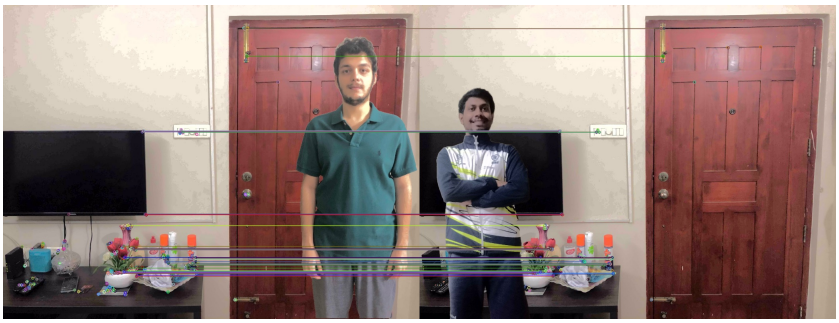

Figure 11: Matched Keypoints

Output of Alpha Blending is as follows,

Figure 12: Alpha Blending Output

Finally the result is as follows,

Figure 13: Image 1, Image 2, Blended Image

3.2 Experiment 2

The input images are as follows,

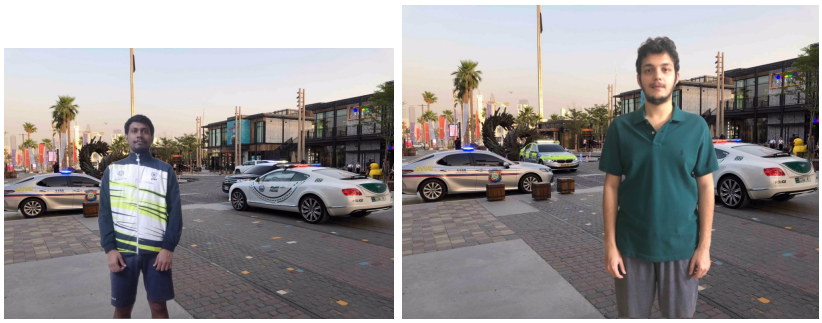

Figure 14: Input Images

The Color Correction outputs are as follows,

Figure 15: Color Correction Outputs

The Face and Body Detection outputs are as follows,

Figure 16: Face and Body Detection Outputs

The detected keypoints are as follows,

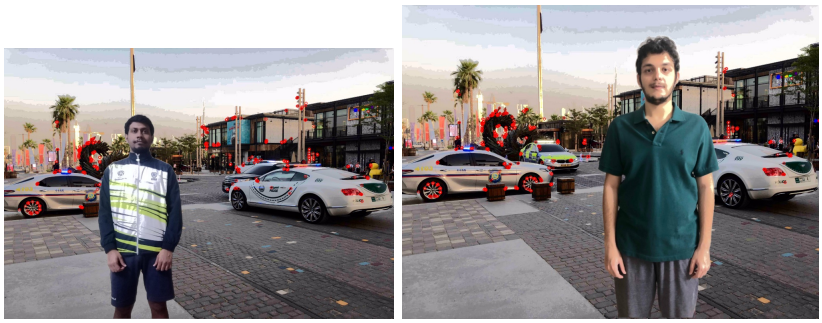

Figure 17: Detected Keypoints

Matched keypoints are as follows,

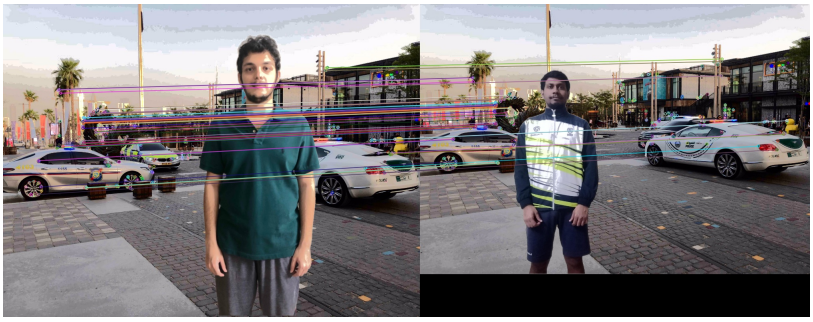

Figure 18: Matched Keypoints

Output of GrabCut is as follows,

Figure 19: GrabCut Output

Finally the result is as follows,

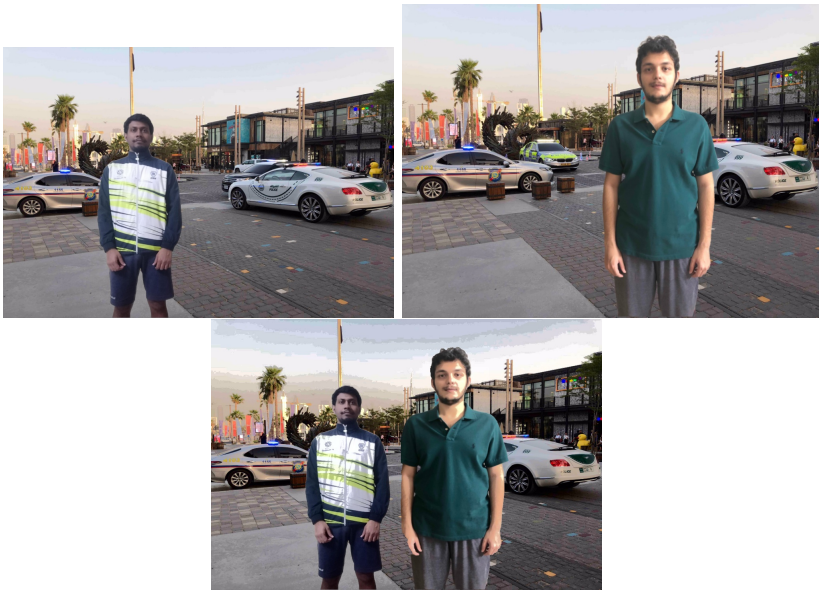

Figure 20: Image 1, Image 2, Blended Image

4 Conclusion

Hence, we were successfully able to implement the algorithm presented in [1] and reproduce the experimental results. Also, the implemented algorithm is quite fast and lightweight - hence can be used in mobile devices at a real time setting.

References

[1] Kevin Chen, David Zeng, Jeff Han. *Friend Blend*. Stanford University.