FRIEND BLEND

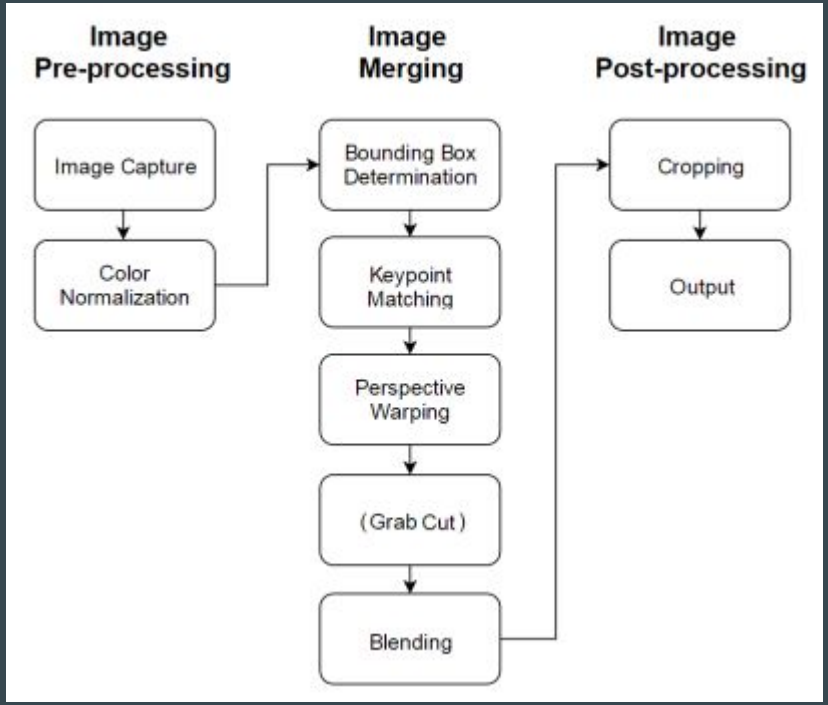
- Team Name: FB
- Team Mentor: Prathyakshun Rajashankar
- Team Members:
  - Ritvik Agrawal (Team Leader)
  - Kirandevraj
  - Vivek Chwandela
- Github Repo Link: https://github.com/Kirandevraj/FriendBlend

Problem Statement

Friend-Blend is an application that merges two portraits of different people to create a single, multi-person photo.

Pipeline of the project

1. Color Correction
2. Face and body detection
3. Keypoint Matching
4. Homography
5. Blending
    a. Alpha Blending
    b. Grabcut
6. Limitations & Analysis
7. Presentation



Color Correction

- **Convert RGB to LAB color space**

  This allows to modify the lightness channel of the image without interfering with the color.

- **Perform CLAHE and convert back to RGB space.**

  This makes sure both the images have similar background lighting conditions to avoid artefacts during blending.

Original Image

Color Corrected Image

Face and body detection

- Haar cascaded classifier for frontal faces is used for extracting a bounding box for the face of each person by taking fixed ratios of height and width of the face..
- The original paper uses a static scale for detecting faces in the input image which doesn't work with many images in the dataset that we collected.
- To generalize this, we iterate through different scales until a face is found, making it robust to many different scales.

Results

Formula for bounding box

$$x_{left}^{body} = x_{left}^{face} - 1.5 * w$$

$$x_{right}^{body} = x_{left}^{face} + 2.2 * w$$

$$x_{top}^{body} = x_{top}^{face} - h$$

$$x_{bottom}^{body} = height(image)$$

Keypoint detection & Matching

- ORB keypoint detector is used to determine mapping between the 2 input images with Hamming distance being the preferred distance measure.
- Total of 10000 key points are extracted from the image
- Only key points outside the bounding boxes of each subject in the images.

Keypoint Detection

Image 1

Image 2

Valid Key Points

Image 1

Image 2

Keypoint Matching

- BFMatcher is used for keypoint matching using Hamming distance as a metric.
- Algorithm:

Algorithm 1: How to take keypoint matches

1 Remove all keypoint matches present within the bounding boxes of the persons present in the image.
2 **if** *total no. of matches* <= *500* **then**
3 | Take top 20 keypoints matches
4 **else**
5 | Take top 40 keypoint matches

Keypoint Matching

BFMatcher results for keypoint matching

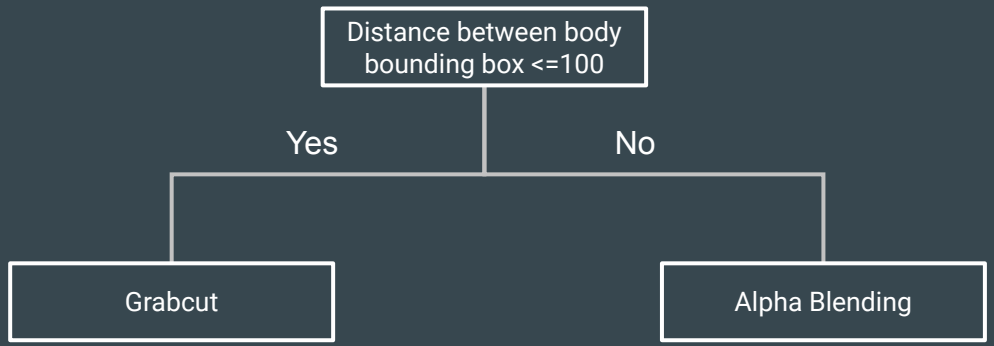
Homography

- Using the top k keypoint matches selected by the previous step, we create a homography transform.
- RANSAC algorithm is used to single out the best transform.
- Next, we warp one image using the homography transform to match with the other image which enables us to align two images without additional artefacts when merged together.
- The determination of the foreground and background images is important when image subjects overlap but does not need to be done if the body bounding boxes are distinct.

Homography Results

Blending

- Blending is the final step to merge two images.
- We follow two approaches for blending the images depending on the relative locations of the persons in images.
- If both the locations are at least 200 pixels apart from each other then we do alpha blending else grabcut is done.




distance

Alpha blending Algorithm

Algorithm 2: An algorithm to perform alpha blending in FriendBlend

1 $colStart$ = Rightmost edge of the bounding box for the left subject.

2 $colEnd$ = Leftmost edge of the bounding box for the right subject.

3 $stepSize = \frac{1}{colEnd - colStart}$

4 **for** *i in range(colStart, colEnd)* **do**

5 $stepCount = i - colStart$

6 $resImg[i^{th}column] = (1 - stepCount * stepSize) * leftImg[i^{th}column] + (stepCount * stepSize) * rightImg[i^{th}column]$

Alpha Blending Results

Input 1

+

Input 2

=

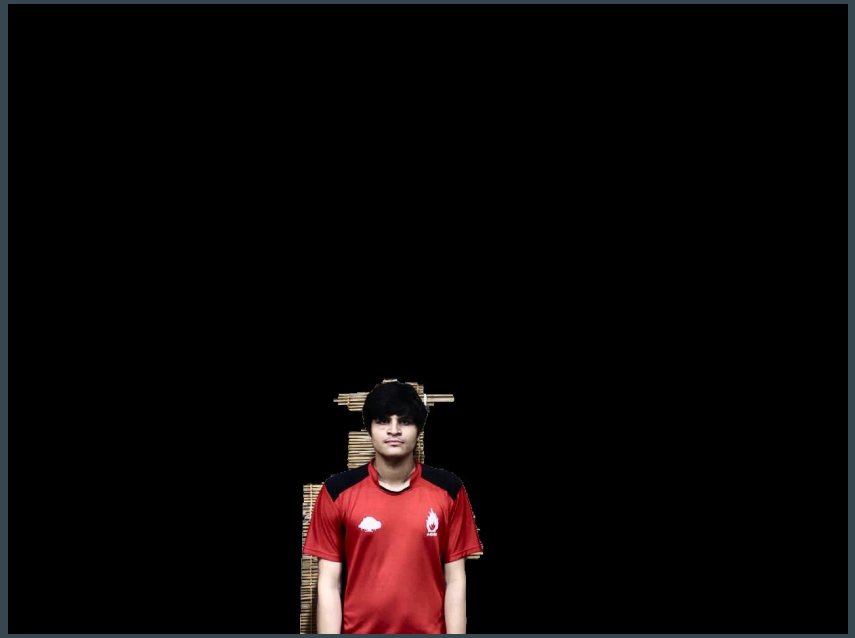
Output from the algorithm after alpha blending

Grabcut

- Grabcut is a technique used to segment out the foreground from background pixels.
- This techniques requires labelling a few for sure foreground pixels and probable foreground pixels.
- Internally it tries to model the pixels into two classes, foreground and background using Gaussian Mixture Models.
- This outputs an image containing only the predicted foreground pixels, which acts like a mask.

Grabcut - Binary Mask

Input to grabcut with sure foreground and probable foreground



Binary mask output from grabcut

Grabcut Result

Input 1

+

Input 2

=

Output from the algorithm after grab-cut blending

Edge-Artifacts

After doing Gaussian Blurring

Where everything fails, use Gaussian filter

Limitations

- Homography being a very crucial step of the process, this algorithm inherently includes all the limitations of homography.
- One of the major challenges was to make grabcut give better results. Still a challenge!

Homography

Improper Alignment after Alpha Blending due to poor homography.

Color Normalization Error

Original Image

Result

Grab-Cut

Contributions

- Color Correction - Vivek
- Face and body detection - Ritvik (50%), Kirandevraj (50%)
- Keypoint Matching - Ritvik (40%), Kirandevraj (30%), Vivek (30%)
- Homography - Kirandevraj (60%), Vivek (40%)
- Blending - Vivek (40%), Kirandevraj (30%), Ritvik (30%)
  - Alpha Blending
  - Grabcut
- Limitations & Analysis - Kirandevraj (70%), Ritvik (30%)
- Project Presentation - Ritvik (50%), Vivek (50%)

THANK YOU !

- Suggestions ?
- Questions ?