1

Applying object image filter with a background subtraction

Jaewoo Cho

Abstract

Most of the popular social network applications have their own set of filters to attract more users. At first, the filter started as a photo effect. These days, famous apps like snapchat and instagram have integrated machine learning and augmented reality into filters. As technology got more advanced, it became easier to integrate machine learning and augmented reality into the app. This resulted in more companies to come up with an app using advanced technologies. For my final project, I made various image filters using openCV

I. Introduction

In order to make an image filter, First, I had to get an image file of the object that I wanted to make a filter with and also a background image of each filter. Then I divided my project into 3 big parts.

- Face recognition
- Apply a filter
- Apply a background filter

For detecting a face, I used haarcascade classifier from openCV. Although the stability of haarcascade classifier was fairly unstable, the accuracy of the classifier was good enough to implement a filter.

Applying a filter and background filter were very complicated and confusing part of the project. The challenging part of this task was to get a mask image of the object filter and also a mask image of the background and combine two images. Then for each object filter, the desired location of the object was all different and I had to come up with an offset value to put the object at the correct location of the video frame.

II. Related Work

I've found a multiples of work that could improve this project in the future.

1

Object segmentation.

From [Fundana, K. 2008], the paper talks about convexifying the energy function of the Chan-Vese method in order to find a global minimizer, so called continuous graph cuts. Using this method, I could improve the quality of segmenting an object from the background. Being able to separate the background with a foreground would improve the quality of the image filter.

Background subtraction.

From [Szeliski, Richard 2022] I followed the basics of threshold and clean up of image. I used the thresholding method to separate the background image and foreground image. However, solely using thresholding has limitations of accuracy of being able to subtract the background of the image and From [Saleh, F. S., Aliakbarian 2017], the paper talks about the use of two-stream network and it's shown that the two-stream network outperforms the state-of-the-art weakly-supervised video semantic segmentation methods. It gave me an idea of making more than one background class and having more than one class could be helpful in some scenarios.

Computer Vision: Algorithms and Applications, 2nd ed. Rick Szeliski

III. Methods

As it's stated in the introduction of the paper, to build an image filter, I separated this project into 3 big tasks.

Face recognition

- The face recognition part of the project was fairly simple and fast with using the inhibited haarcascade face classifier that detects person's face

Apply a filter

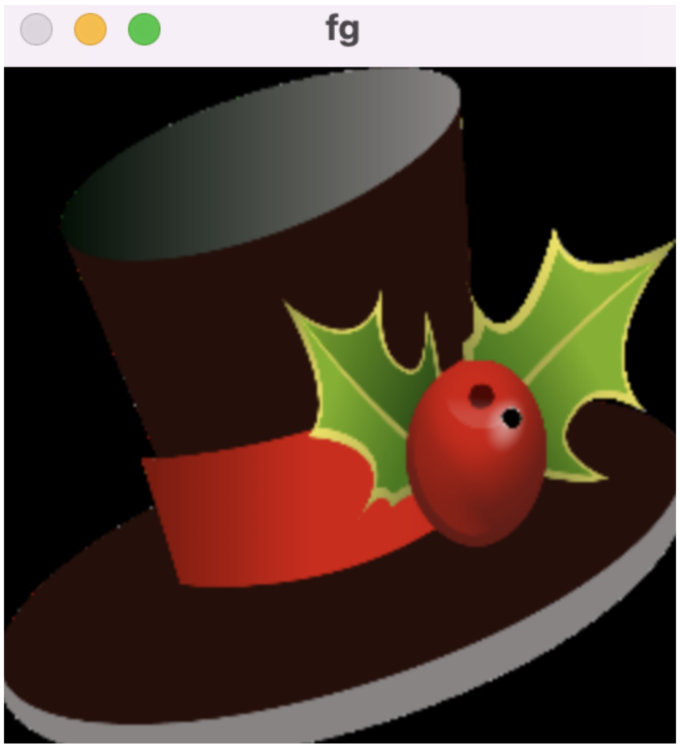
- The first step that I had done was to change the object image into grayscale to get a binary image of the object image with only the object part highlighted.

- Then I'm using the faces that've been detected by the first part of the project to get the x, y coordinates, width, and height of the bounding rectangle box of the face.
- Then for each object filter, I've used a guess and check method to come up with an algorithm that makes the filter fit into the correct location of the image.
- Using a new location coordinate points, width, and height of the filter image, I resized the object image to fit into the size of the mask image.
- Then I created a new base frame of plain dark background and bitwise() function to get a mask image of the object filter at a desired location.
- Lastly, apply bitwise() function one more time to combine the object mask image and a live video frame image.

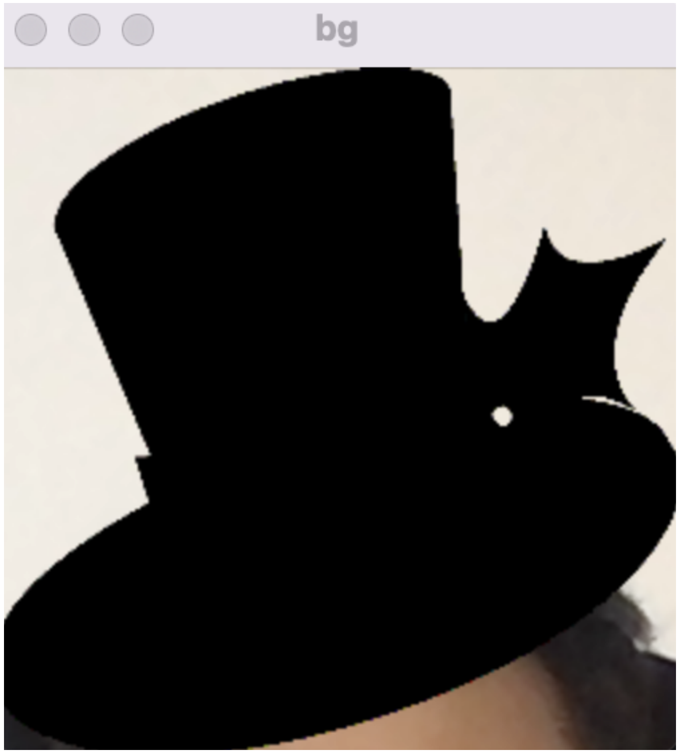
Apply a background filter

- Last task was to apply a background filter. The reason why I chose to apply a background filter after applying an object filter first was that the algorithm would be much more efficient if we combine a layer of images from the most foreground image to the background image.
- To apply a background filter, the first step was to threshold a live video frame to separate the person from the background. The thresholding was done by using inRange() function to only get the pixel values that are in range of white HSV values. For the input of the inRange function, I used the HSV range of (0, 0, 180) to (255, 255, 255) to exclude the white background. Then I used a dilation to remove the noise and increase the area of the segmented part.
- Then with the background mask image, I used the bitwise_not() function to get a negative image of the background mask image to get a foreground mask image.
- With both background and foreground mask images, I used bitwise_and() to combine a foreground with a live video frame and background with an input background image.
- Lastly, I combined the new foreground image with the new background image to get a background filtered image.

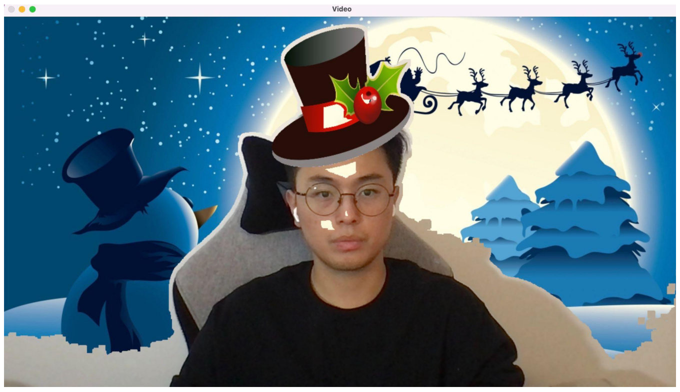
IV. *Experiments and Results*



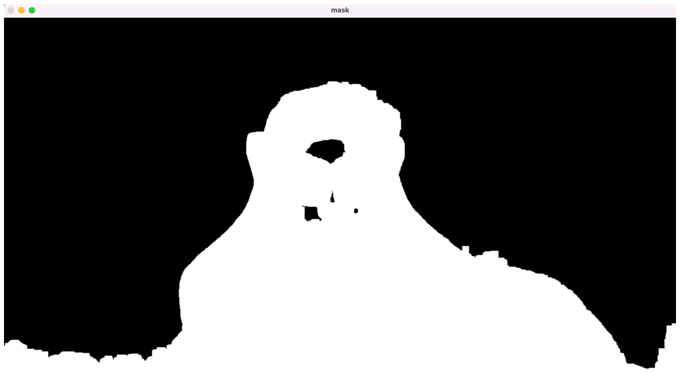
Foreground image of the object



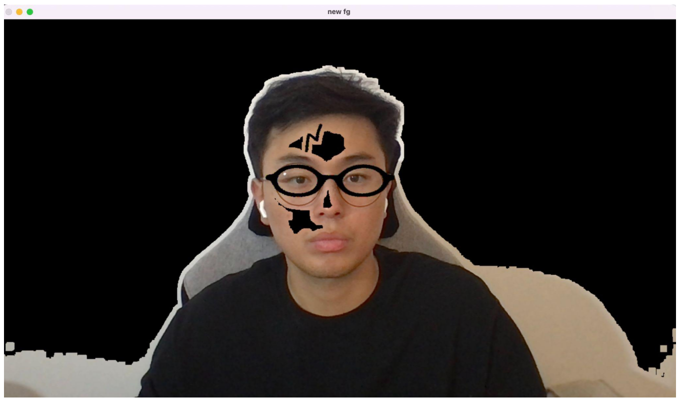
Background image of the object image



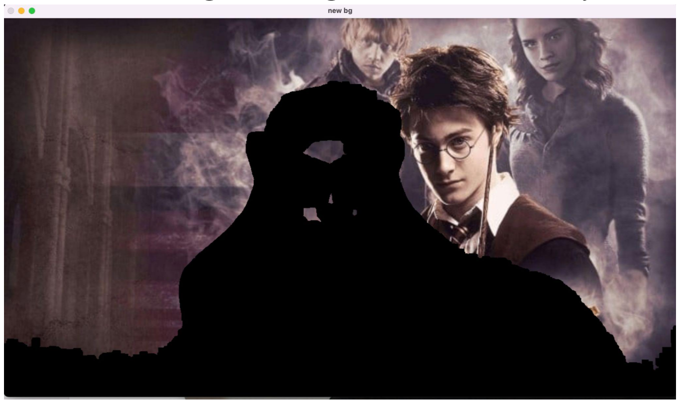
Combined image of an object image and a foreground



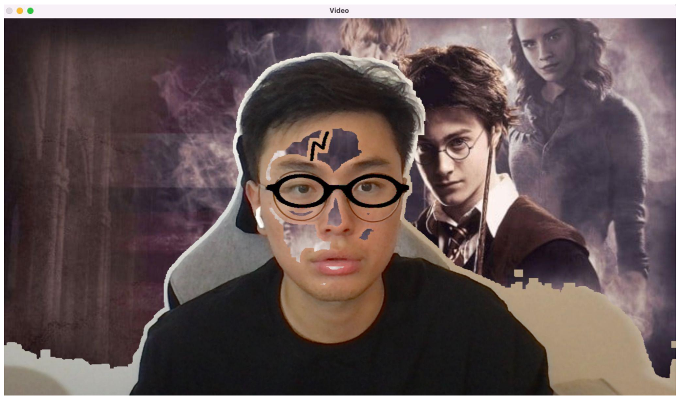
Background masking done by thresholding the image with HSV range of white color.



Foreground of an image found by bitwise function with masking image and a frame.



Background of an image found by applying bitwise function with an invert image of masking image and a background image



Combined image of foreground and background

V. *Discussion and Summary*

There are many aspects of this project that could be improved to get a more realistic filter

Background subtraction. I've separated the background and the foreground by thresholding an image with the range of white HSV values. My implementation wouldn't be able to separate a foreground image if the foreground image contains a pixel value that is in a thresholding range. If I make a further contribution to this project, I will try to find a better way to extract the background.

3D image filter. The current image filter that I've built only works from a perspective of the front view. The cascade classifier that I've used to detect a face only has an ability to detect a face of the front view. This is another area where I could improve for future work. I would try to make it so that it can detect any face in any perspective of view and also display an object filter in 3d plane as well.

VI. References

[1] Shaikh, M. U. (2021, August 25). *Making face filters with opencv*. Medium. Retrieved December 12, 2022, from https://medium.com/@mohd-uzair/making-face-filters-with-opencv-e3c928865239

[2] Shilu, S. (2018, December 12). *How to make Snapchat lenses?* Medium. Retrieved December 12, 2022, from https://medium.com/hackernoon/how-to-make-snapchat-lenses-f9eae861b5db

[3] *Work-IS-playing*. WorkisPlaying. (n.d.). Retrieved December 12, 2022, from https://grauonline.de/wordpress/?page_id=3065

[4] Fundana, K., Heyden, A., Gosch, C., & Schnorr, C. (2008). Continuous graph cuts for prior-based object segmentation. *2008 19th International Conference on Pattern Recognition*. https://doi.org/10.1109/icpr.2008.4760956

[5] Szeliski, R. (2022). *Computer vision: Algorithms and applications*. Springer.

[6] Saleh, F. S., Aliakbarian, M. S., Salzmann, M., Petersson, L., & Alvarez, J. M. (2017). Bringing background into the foreground: Making all classes equal in weakly-supervised video semantic segmentation. *2017 IEEE International Conference on Computer Vision (ICCV)*. https://doi.org/10.1109/iccv.2017.232