Flash Photo Booth Application

Ugurcan Demir 150150208
Istanbul Technical University
Computer Engineering Department
Email: ugurcandemiir@gmail.com

Abstract--- Photo Booth is a software application for taking photos and videos with a camera. In this paper, an approach for another photo booth application is proposed. It is based on basic Cv2 filters, Keras, and Dlib library implementations. Evaluation results are presented and discussed.

I. INTRODUCTION

Photography takes an instant out of time, altering life by holding it still. Why should we miss it when we have the chance to capture that moment and keep our memories alive? So, I wanted to develop this application based on capturing memories in our lives. Flash Photo Booths has many different filters on it. It also creates filters related to current events and tries to make people happy.

This paper addresses the implementation of filters in the application combining OpenCV, NumPy, Dlib and Keras libraries. In Section II photo booths are explained in more detail and related work is presented. Section III presents the proposed approach for implementing and Section IV gives evaluation results of the proposed approach. The conclusion is given in Section V.

II. BACKGROUND

A. *Photo Booths*

Photo Booths display a preview showing the camera's view in real-time and save the taken images. Once the pictures have been taken, the users can save it easily on their computers. Features that can be found in photo booth applications are customizing the beauty of the users such as brightening the pictures, making the eyes sparkle more, changing the hair, fixing any blemishes by having them blurred.

Based on this, we implemented current event filters to make our application more usable. We also used some machine learning libraries to make our application cooler than other applications.

B. *Related Works*

Though not much, there is some work related to photo booth application. It is published by Apple Inc. as part of macOS and iOS. Photo booth was originally available only on Macintosh computers that had a built-in iSight camera running Mac OS as in figure 1.

Fig. 1. Photo Booth Application in MacOS

After the picture is taken effects can be applied by clicking on the "Effects" button. Photo Booth has two sets of image effects that can be applied when taking a picture. The first set contains photographic filters. Another set allows replacing the background with a custom backdrop [1].

III. PROPOSED APPROACH

Approach taken in this paper is based on several implementations. First, we used python as a programming language and Tkinter library to get user interface, shown in Figure 2.

Fig. 2. User Interface

A. *Filters Using Kernels with OpenCV*

In image processing, a kernel, convolution matrix, or mask is a small matrix. It is used for blurring, sharpening, embossing, edge detection, and more. This is accomplished by doing a convolution between a kernel and an image. By applying those special kernels, we created some filters for our application shown in figure 3. There are some special filters too like Canny edge detection.

The Canny edge detection algorithm is composed of 5 steps which are noise reduction, gradient calculation, non-max suppression, double threshold, and finally edge tracking by Hysteresis. For those filters, we basically use OpenCV functions to implement it.

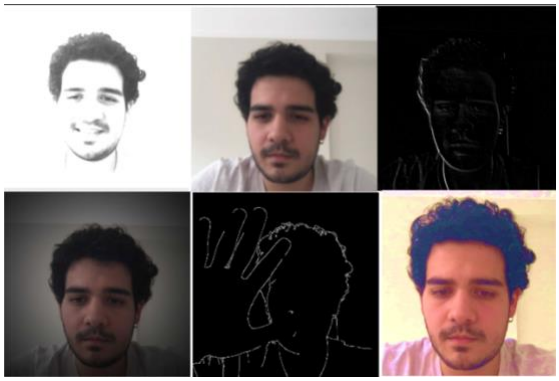

Fig. 3. Left to right - Sepia - Gaussian - Emboss - Vintage - Edge Detection – Saturation

B. *Implementation of Dlib Library into Filter*

We want to apply current event filters like Instagram and Snapchat filters. To do that, we did some research and found dlib library.

Using this library, we can detect faces in real time and predict the landmarks of the face easily. As shown in Figure 4, this is our template for face landmarks, and we found shape predictor file based on figure 3.

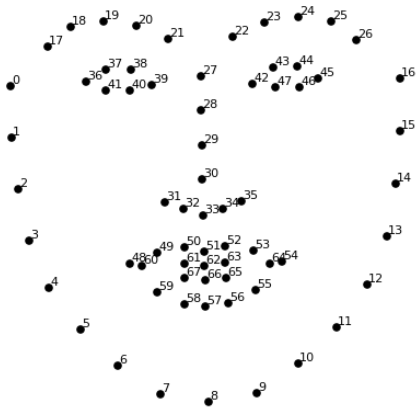

Fig. 4. Face marks Points

Depend on face marks points, we clarify mouth area using landmarks.part () functions. Then we used threshold and bitwise_and functions in cv2 library for our gif image. The final part, we added gif image to real time camera vision.

For this application, we created a current event filter for Corona Virus. Therefore, we used a mask gif. As shown the Figure 5, this our final result for our filter implementation.

Fig. 5. Designed Filter for Corona Virus

C. *Implementation of Keras Library into Filter*

As we mentioned earlier, our application purpose is to make people happier while using our application.

For this reason, we would like to detect facial expressions our users by using Keras library. We would apply convolutional neural networks to tackle this task. And we will construct CNN with Keras using TensorFlow backend. While using facial expression, we also implemented a face detection algorithm.

We have already found a suitable dataset before for our project. It has only five labels which are Angry, Happy, Neutral, Sad, Surprise. Our code basically detects the user's face and based on our dataset; it found the expression of the user.

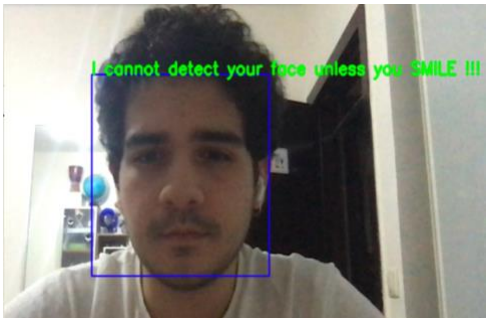

Fig. 6. Warning Message

Once the application detects the happy face, it shows the name of the user. If the user is not happy, application gives an advice message saying "I cannot detect your face unless

you smile" on the real time camera screen. It is shown in the Figure 6 and Figure 7.

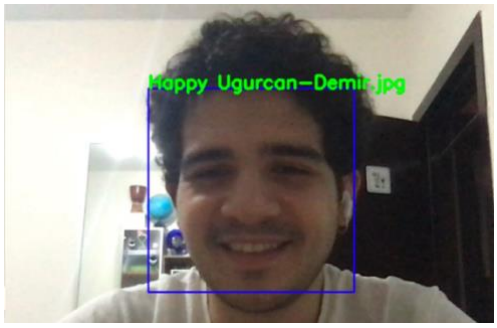

Fig. 7. Face Detection and Expression

IV. EVALUATION

Criteria for what is considered using different algorithms into the application and the idea of running those algorithms simultaneously is somewhat difficult to implement. This is because running them together into an application can cause some problems like not running fast or get some errors while using filters.

We improved our code and algorithm to run them together. Therefore, we wrote 4 different python script and imported them at the beginning of each script. By applying this method, we have successfully run.

V. CONCLUSION

This paper presented approach for photo booth application. We used different algorithm to accomplish this project. We learned the basic OpenCV functions and some machine learning libraries. It was a great opportunity to improve my image processing skills. We are going to improve our application and add new filters for our users.

REFERENCES

[1] https://en.wikipedia.org/wiki/Photo_Booth

[2] https://en.wikipedia.org/wiki/Kernel_(image_processing)

[3] https://sefiks.com/2018/01/01/facial-expression-recognition-with-keras