CS 445

FALL ‘23
FINAL PROJECT

SnapBlend.

Pranav Nagarajan
Advaith Yeluru
Divyam Arora
Nazar Kalyniouk

Prof: Derek Hoiem

Table of Contents

1. Motivation and Impact

The decision to create pre-existing and custom Snapchat filters was a result of us wanting to pursue a project that explores computer vision techniques as well as staying relevant to the cultural impact of these technologies. We were able to gain hands-on learning experience with creative expression and technical innovation. By recreating popular Snapchat filters as well as creating our own, we were able to show that accessibility for this type of creative expression is feasible and easy to use. Looking at this project high-level, we are able to mold our perspective and choices by immersing ourselves in this digital environment. This project has provided us with an avenue for visual expression and reimagination, showcasing the future of cultural expression in the digital age.

2. Approach

As a team, our goal for each of the filters was to firstly understand the end result, then work backwards using modules and libraries. We took advantage of pretrained models and built-in library functions to serve as useful building blocks. For the facial accessory filters, we relied heavily on Dlib for face detection and pretrained facial landmark models. For backgrounds, segmentation models like DenseNet proved to be very useful for isolating silhouettes. In addition, enhancement filters were also implemented, using PIL to get the desired effect. OpenCV face detection and text overlays were also very useful, specifically for filters like the CCTV Face Recognition Feed Filter. The key goal was to identify which mix of models and libraries would come together to reach the desired effect.

3. Results

In the end, our team's approach to these challenges resulted in the successful implementation of Snapchat's most popular filters. We leveraged many libraries and modules in Python to achieve our desired results. For example, using Dlib for face detection and pretrained facial landmark models, we were able to successfully overlay accessories like sunglasses and mustaches onto faces. These results display our strong understanding in image manipulation and processing techniques and pave the way for more sophisticated and specific applications. Since Snapchat filters are widely popular and used every day, replicating these filters and making our own filters showcases our skills in image processing in a topic that is related to current trends.

Face Detection Sunglass Filter :

Face Detection Crown Filter :

Face Detection Mustache Filter :

Face Detection Face Blur Filter :

Meme Face Swap :

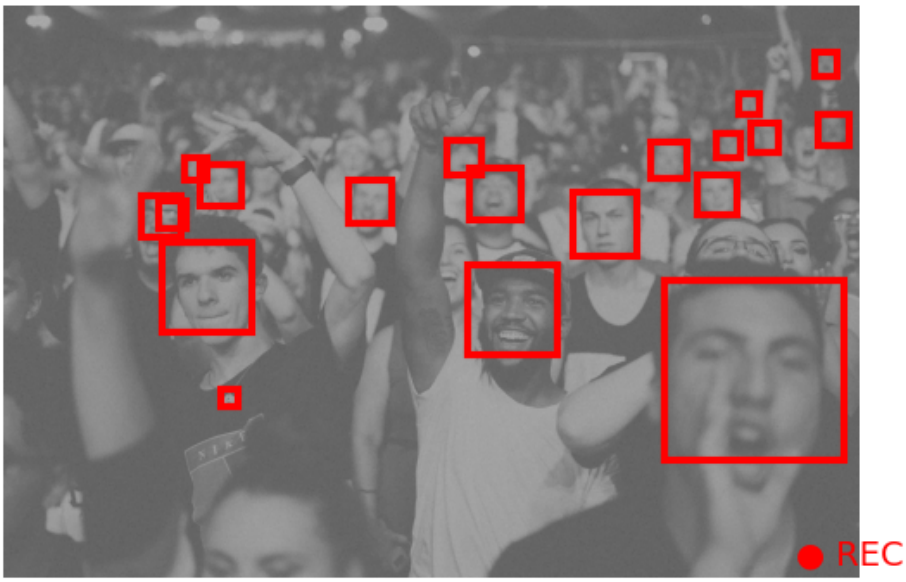
Image Artist Filter :


Alien Filter :


X-Ray Filter :

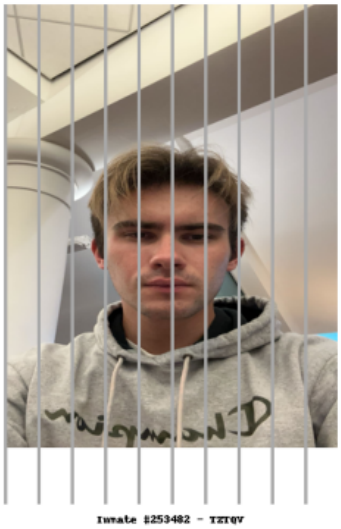
Thermal Face Filter :

CCTV Face Tracker Filter :

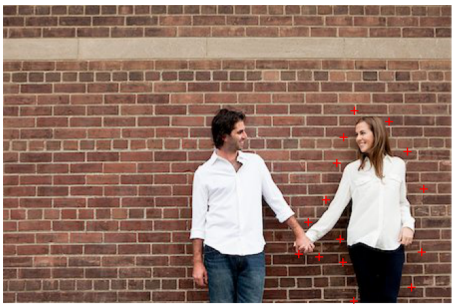


Neon Effect Filter :

Mugshot Filter :



Object/Person Remover :



4. Implementation Details

Programming Language: Python

List of Modules and Libraries Used:

- OpenCV - (Open Source Computer Vision Library), used for computer vision and image processing tasks.
- Dlib - Library used for face recognition purposes.
- Numpy - Library for numerical computations.
- Tensorflow - Deep learning library used for building and training neural networks.
- Matplotlib - Plotting library used for creating visualizations.
- PIL - (Python Imaging Library), used for opening, manipulating, and saving different image file formats.
- Random - Module used for generating random numbers.
- String - Module that provides a collection of constants, classes, and functions specifically related to string manipulation.
- Google.colab.patches - Module that includes functions and utilities that help patch or modify certain functionalities.
- OpenAI - Used for parts of technical understanding and a rough start to the project.

List of External Data Used:

- shape_predictor_68_face_landmarks.dat
    - This file contains a pre-trained model designed to detect 68 facial landmarks.
- shape_predictor_81_face_landmarks.dat
    - This file contains a pre-trained model designed to detect 81 facial landmarks.
- shape_predictor_194_face_landmarks.dat
    - This file contains a pre-trained model designed to detect 194 facial landmarks.
- haarcascade_frontalface_default.xml
    - This file contains a pre-trained Haar Cascade classifier provided by OpenCV for recognizing frontal faces in images.

5. Challenge and Innovation

Our group planned out a systematic approach with each image processing task, aiming to replicate existing filters as well as make our own from scratch. Our strategy of approaching each filter was broken down into three steps: understanding the objective of the filter, writing and analyzing the code, and then implementing any necessary modifications. A challenge we faced was selecting the appropriate modules and libraries and researching them. Since these filters are achieving a very specific objective, it was important that we used the proper libraries to achieve the proper end result. Using the external data such as the face landmarks and haarcascade proved to be difficult. In the end, we were able to effectively implement our objectives by researching how to properly use them. Given the depth of our technical exploration as well as implementing many more types of filters than we previously expected in the original proposal, we anticipate receiving a high score. As a result, a 20/20 would be fair.

6. Final Notes

"Group members contributed similarly."

Pranav Nagarajan : Equal parts of the report and filters added
Advaith Yeluru : Equal parts of the report and filters added
Divyam Arora : Equal parts of the report and filters added
Nazar Kalyniouk : Equal parts of the report and filters added

All of the above team members worked together in both research and the technical implementation. There was no time where we split up the work and had each person achieve a separate working individual filter. We have equal contributions in finding and implementing the libraries we used for certain parts and the fillers we came up with.

Overall it was a fun experience putting to use the skills we learnt in this class and also thinking outside the box on how to apply some of the open-source information available to us to make something innovative our age-group loves and enjoys.

Working as a team surely made things more pleasant as it was easy to split things up and work on parts we were confident with.