Magic Eyewear Try-On Software document

Ragaa Moustafa, Farida Hesham, Samiha Hussein, Yara Amr

June 24, 2021

Abstract

It is often hard to know what a pair of glasses would look like on a person's face, especially if they cannot try them. Magic Eyewear aims to solve this issue by providing online shoppers with the ability to try on different frames virtually. This paper explains the development process of the virtual try-on software, the various tools used to create it, and the Github repository for the source code.

1 Introduction

Magic Eyewear is a web application of two parts; e-Commerce and python-based smart-mirror software. Magic Eyewear is being developed for EyeCare Optics, an Egyptian eyewear company. As of the date of this paper, Magic Eyewear is still in the development process. That said, this document thoroughly explains the software part of it.

2 Aim

The aim of the Magic Eyewear try-on software is to make it easier on people to find the perfect pair of glasses without having to go to the physical store. They can choose whether to see a live-preview of how the frame looks on them, or just upload a picture or video and the software will apply their chosen frame.

3 Program Requirements

3.1 Anaconda Navigator

Anaconda Navigator [1] is a distribution containing different packages which are used in scientific computing such as, data science, image processing, and machine learning. These packages help to simplify the management and functioning of different applications.

3.2 Spyder IDE

Spyder [2] is an integrated development environment written in python, mainly used in scientific computing such as in Magic Eyewear case is image processing.

3.3 PyCharm IDE

PyCharm [3] is a Python-dedicated IDE that provides essential tools for Python developers.

3.4 Libraries

- OpenCV [4]
- Numpy [5]
- Boost [6]

- CMake [7]
- Xlib [8]
- Dlib [9]
- Scipy [10]

4 Program Details

4.1 Magic Eyewear using live preview

Landmark is a module in dlib library used in face detection, which contains different points(detector) that automatically detect faces. We applied the landmarks in 3 points which are (0,27,16) which are left side, middle side, and right side's face respectively. Then, we adjusted the width and height of the face so that the glasses change size when the user is near or far from the camera. The last step is to mask the eyes area to add the glasses into the face, we used thresholding which mask the glasses and adjusts it to be applied into the face.
figure 1 : Landmarks point reference

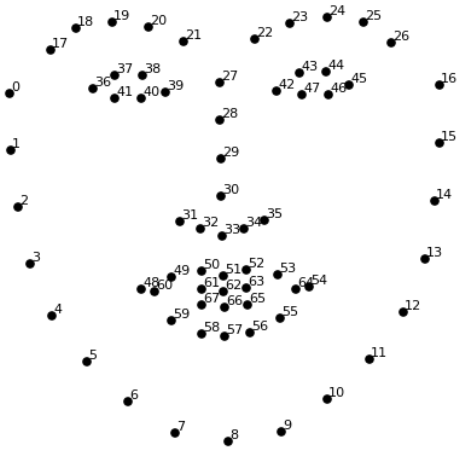

Figure 1: Landmarks points

4.2 Magic Eyewear using saved images

Haar cascade is an Object Detection Algorithm that uses edge or line detection features. a haar cascade was used in this algorithm to detect faces and eyes in an image. after detecting the eyes' position, the background of the glasses image is removed, and the image is scaled based on the eyes. the main library used is OpenCV. all the enhancements of the glasses picture have been done with the OpenCV library. At the end of the process, the glasses are perfectly overlayed on the face.

4.3 Magic Eyewear using recorded video

In order to apply the glasses on a recorded video, a module in the dlib library called Landmark was used. It is a module that detects the face and its shape and can find 68 different facial landmark points including chin, jawline, eyebrows, nose, eyes and lips. The algorithm first detects the face shape and its landmarks, resizes the glasses in order to fit any face shape, then overlays the face with the glasses. The image processing is done using OpenCV which is an open-source library mainly used in machine learning, computer vision, and image processing.

5 Program Preview

5.1 Screenshots

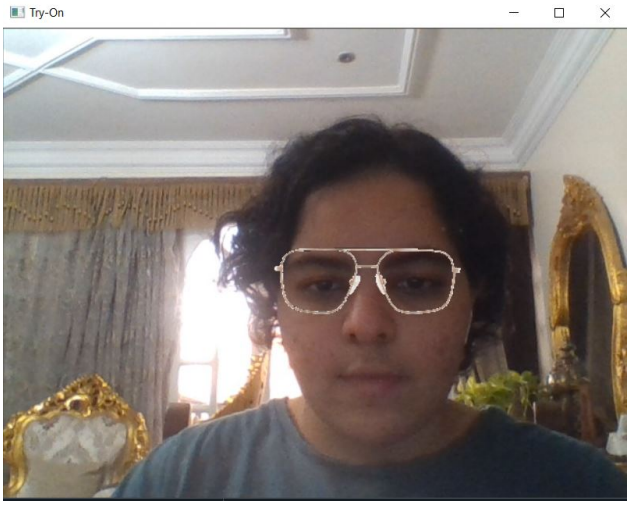

Figure 2: Live Preview

Figure 3: Using saved image

Figure 4: Using saved video

6 Conclusion

Three different algorithms were discussed in this paper. The first being the virtual try-on with a live preview. The second being the virtual try on using previously stored images. And the last being the virtual try on using recorded video. All of them aim to save time -and possibly money- by providing virtual try-on software and making glasses more accessible without the obstacle of being unable to visit the store.

7 Contributions

- Frame on previously stored image: Ragaa Moustafa
- Frame on a pre-recorded video: Farida Hesham
- Frame on live video: Samiha Hussien and Yara Amr

For the source code, refer to the Github [11] link of this project.

References

[1] URL: https://docs.anaconda.com/anaconda/navigator/.

[2] URL: https://www.spyder-ide.org/.

[3] URL: https://www.jetbrains.com/pycharm/.

[4] URL: https://opencv.org/.

[5] URL: https://numpy.org/.

[6] URL: https://www.boost.org/doc/libs/1_70_0/libs/python/doc/html/index.html.

[7] URL: https://cmake.org/.

[8] URL: https://pypi.org/project/xlib/.

[9] URL: http://dlib.net/.

[10] URL: https://www.scipy.org/.

[11] URL: https://github.com/yaraamrsalah/Virtual-Try-On-Image-Processing.