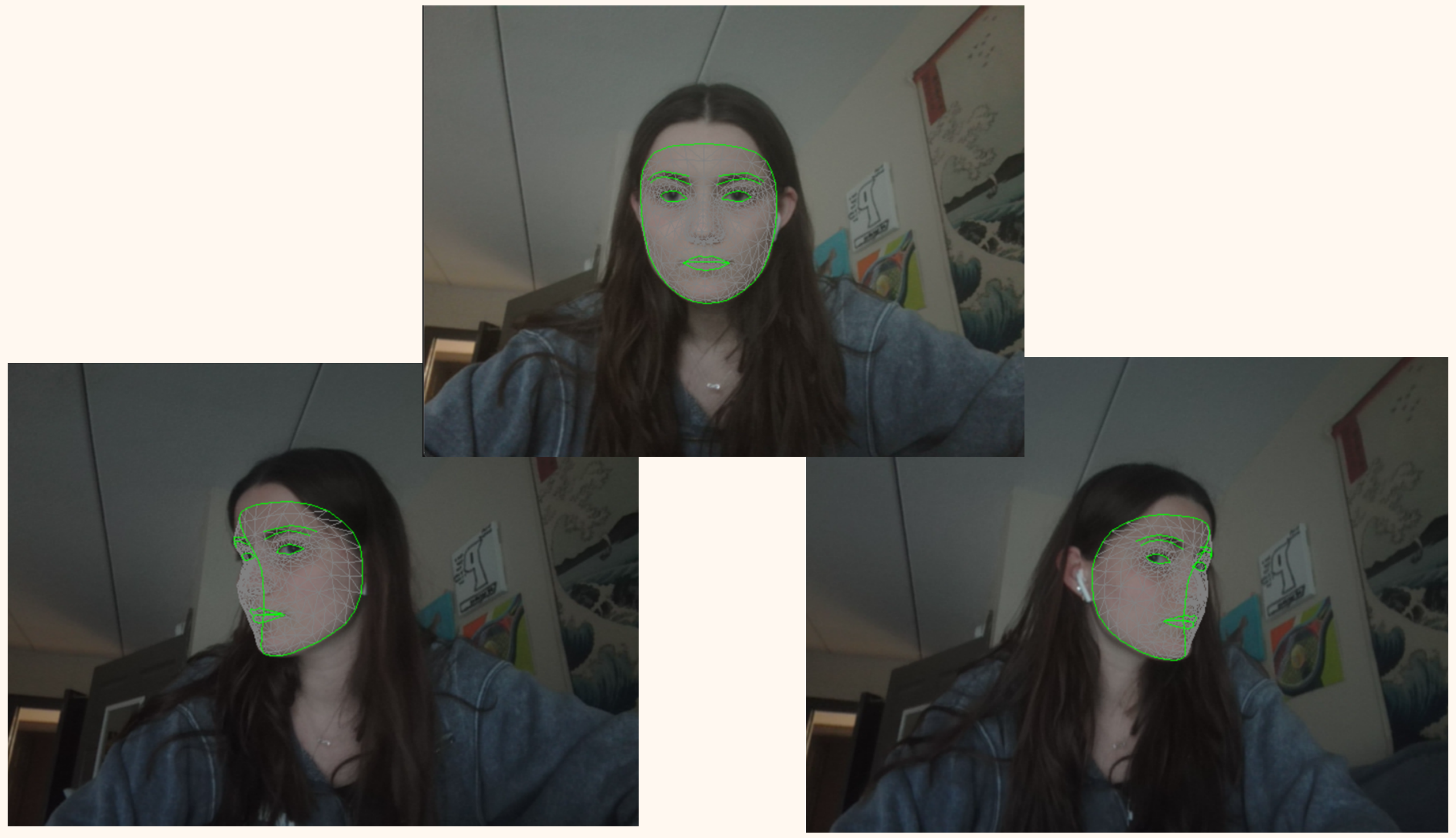
Avatars Runnin' on Dunkin'

Dr. Martin Hellwig Keelin Robbins, '26

Providence College Spring 2024

Department of Mathematics and Computer Science



Testing of computer vision facial recognition processes

INTRODUCTION

This research project was inspired by the recent removal of the security camera monitoring the activity at the on-campus location of Dunkin' Donuts. The video feed was publicly accessible to all the Providence College community for people to know whether the line was too long. However, issues arose with a publicly broadcast camera, mainly with privacy and security concerns. The final goal of "Avatars Runnin' on Dunkin'" is to remedy these risks, restoring the camera with its original purpose of displaying a live video feed of the activity. This program intakes the live video from the camera and imposes an avatar over the people in the frame, protecting their identities. Written in python, this program accesses public python libraries such as MediaPipe and OpenCV, which interact with the subjects' movements. Although it is currently in the prototype phase, this project aims to return the Dunkin' Donuts camera to all student and faculty usage.

Future Goals

1. Optimize the current code and fix the borders around the image segmentation to make it closer to the subject's body.
2. Create and model avatars to be imposed over the subjects of the video camera.
3. Use segmentation program to separate the people from the background, then impose the modeled avatars over the subjects and follow their movements.
4. After working with these programs on live feed from a laptop, move to a larger scale security camera to impose the avatars over multiple subjects in frame.
5. The final goal of this project is to be able to apply this program to the Dunkin' camera, which would restore it to the Providence College community.

Code Explained

1. Import Libraries: At the beginning of the code, the import libraries are stated. This allows the program to import standard computer vision libraries to access publicly accessible algorithms.

2. Storing of Data: Variables are then created to store the future drawings and visualizations from the algorithms run by the computer vision program.

3. Video Capture: The standard computer web camera must then be accessed and available to the program for it to be able to input and interact with the live video feed.

4. Segmentation: The program breaks down the image into pixels that it can accurately process. Then, it compares the individual components of the picture to other patterns of pixels from its library to perform the functions being asked of it from the user.

5. Color Grading: The image being processed by the MediaPipe library is written in BGR, as that presentation of viewing is easier for it to process. The image must then be converted back to RGB for its final presentation to the user, which is the presentation for normal video.

6. Segmentation Mask: This mask is then applied to the image, which covers the subject in the frame of the camera, so they are not visible. The dimensions of this mask can be changed within the source code to more accurately fit and mask the subject. This is the part that is seen by the user, which covers the camera subject's entire body.

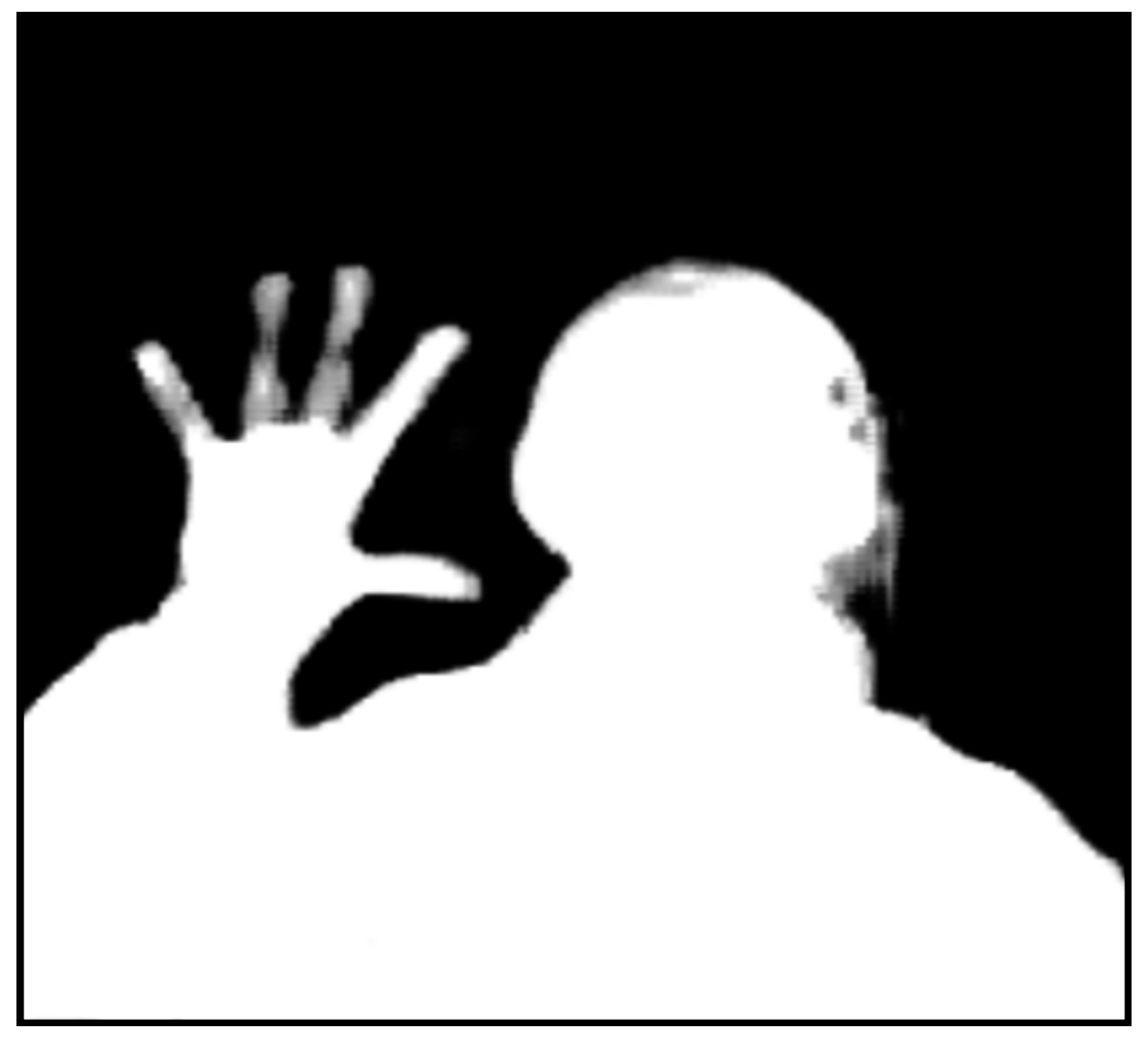
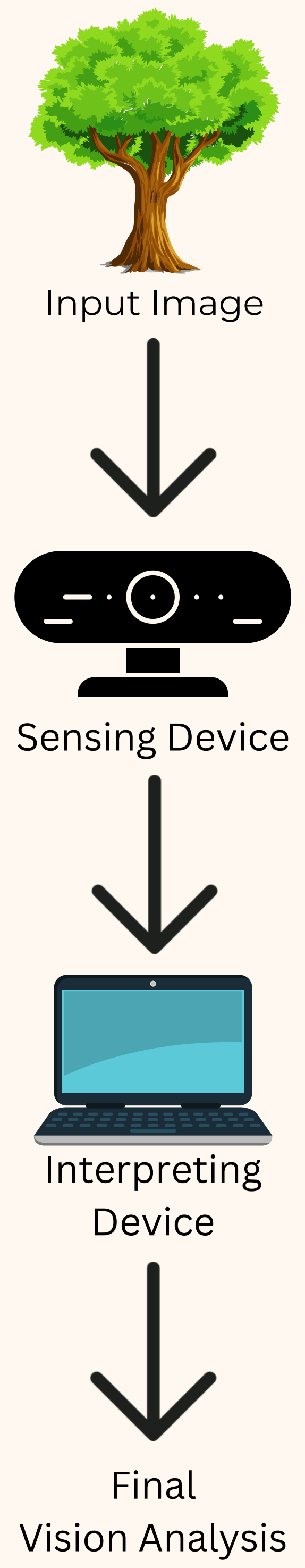


Image from program that represents image segmentation with a human subject in frame.

What is Computer Vision?

Computer vision is a type of artificial intelligence that deals with visual inputs, which then has the program interact with the feed given to it by the program. These computer programs require a lot of data to teach itself how to "see" the images it is being fed. Computer vision allows a program to mimic the ways in which people can process images and the world around them. The same ways people learn from experiencing their surroundings, vision modeling allows computers to run data analyses on prior images to make greater sense of current images being fed to it. These networks separate images into pixels that are accurately processed, with differences between frames or pixels analyzed to accurately understand or predict the images. The neural network compares patterns present in the current image to previous patterns detected from its extensive library. After processing the current feed, it is then recorded in the library for future reference, which is how it learns in the future. These libraries are publicly sourced, including examples such as MediaPipe and OpenCV, which were used in this research. Computer vision is becoming increasingly useful in countless different fields which require facial recognition, object detection, motion tracking, and have many more uses as the technology evolves in the future.


Input Image
Sensing Device
Interpreting Device
Final Vision Analysis