CS 498 Prototype Assignment

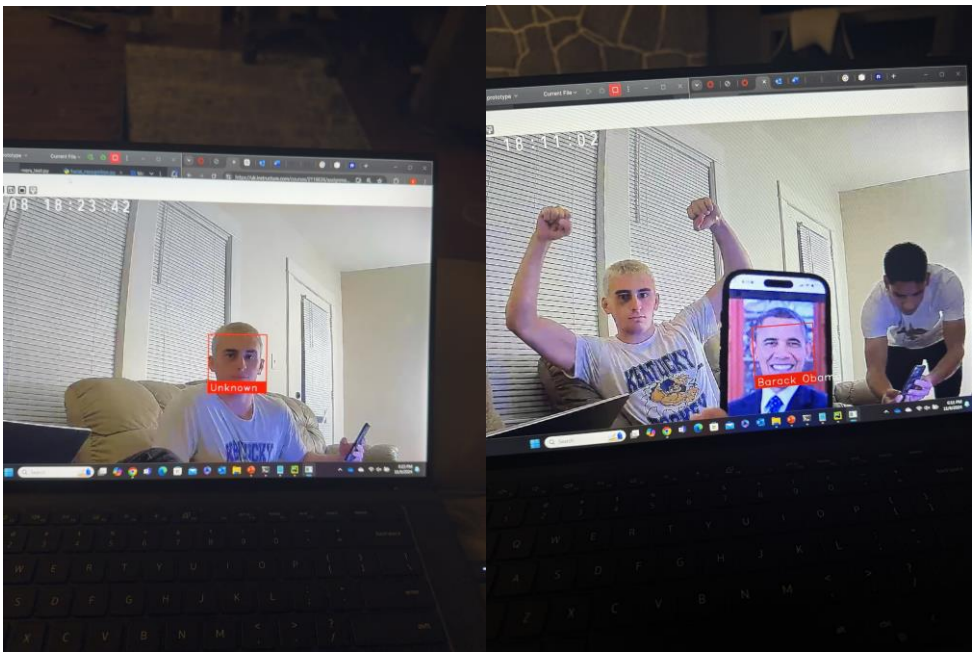
Evan Damron, Raymond Goodman, Chet Fasteen

1. **Prototype Branch: https://github.com/EvanDamron/Security-Camera/tree/prototype**
2. **User Stories Implemented:**
   - **Wi-Fi setup: As a homeowner, I want to connect the security camera to my home's Wi-Fi network so that it can stream footage to my computer.**
   - **Live View: As a homeowner, I want to view the live camera feed on my computer in real-time so that I can monitor what's going on in/around my house.**
   - **Unknown Face Alert: As a homeowner, I want to be notified when an unknown person comes into frame so I can quickly verify if they are supposed to be there.**
3. **We bought the TP-Link Tapo C100, which came with an app. Within the app we connected it to our Wi-Fi network easily, but accessing the live stream via Python was difficult. Eventually we figured out we had to disable the stream to the app and configure the camera with a custom IP address, and then we were able to access it's Real Time Streaming Protocol (RTSP) stream using OpenCV.**

   To implement facial detection, we used the face recognition software library found at https://github.com/ageitgey/face_recognition

   Getting basic face recognition working took very long, because we only have Windows devices, and Windows isn't officially supported by the library. So, we had to create an anaconda virtual environment to play around with different versions of cmake, dlib, and OpenCV until we finally found a working combination.
4. **For the third User story, we have not fully implemented the facial detection aspect yet, but we have basic functionality. So far, we are only able to detect the default faces, which are Barack Obama and Joe Biden. Adding more faces is as easy as adding images into a folder for the model to learn. We have also made the function to send an email notification to the user with a snapshot of the**

video stream, so now we just need to add new faces. Some demo pictures of the current state of facial recognition can be seen below (facial_recognition.py):



5. We collaborated a lot for this milestone, meeting in person for several hours a few different times. We all worked together to set up the camera. Chet implemented the code to send an email notification via Python. Evan and Raymond did pair programming on Evan's laptop to implement facial recognition. Raymond also worked on implementing the code to save video clips to the computer, which will be called whenever unknown people are detected.
6. This question doesn't apply to us because we aren't modifying a code base, and we dropped the ball on asking about a replacement question. However, we will take this space to discuss the face recognition library we are using. This library's documentation is very detailed, which will make the rest of our facial recognition move quickly. The lack of documentation on Windows was annoying, but there were third party sources that helped guide us. The documentation also provides several examples, so we will rely on these to create and fine tune our final version.