
Implementing an Eye Tracker with Modern Computer Vision Techniques

Anish Sahoo

CS 4973

Northeastern University

Boston, MA 02115

sahoo.an@northeastern.edu

Sharon Pan

CS 4973

Northeastern University

Boston, MA 02115

pan.sha@northeastern.edu

1 Overview

We propose to implement an eye/gaze tracker using modern Computer Vision techniques. This is an interesting problem, as it is hard to generalize eyes (many different eye shapes/colors), and requires real-time computation efficiency to make it viable for daily use. This problem combines several subproblems including eye-region detection, mapping eye to screen coordinates, and pupil localization under various lighting conditions, among others. A successful solution will enable applications in accessibility, driver safety technologies, and human-computer interaction. Our goal is to design and train a system that is accurate and lightweight, so it can be run on consumer-grade devices in real-world settings.

2 Literature Survey

1. Mesmoudi, S.; Hommet, S.; Peschanski, D. Eye-Tracking and Learning Experience: Gaze Trajectories to Better Understand the Behavior of Memorial Visitors. *J. Eye Mov. Res.* 2020, 13, 1-15. <https://doi.org/10.16910/jemr.13.2.3>
 - Mesmoudi and co's paper focused on using eye-tracking as a tool to quantify visitor experience and behavior in museums, as well as track how museums can have a role in learning among students.
 - We are interested in this paper because it discusses the way gaze trackers can be applied in the real-world with relatable implications. Furthermore, this paper has introduced variables that will be useful to analyze during our own eye tracking: gaze trajectory and viewing time.
2. Zhiwei Zhu and Qiang Ji, "Eye gaze tracking under natural head movements," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 918-923 vol. 1, doi: 10.1109/CVPR.2005.148.
 - A challenge with standard PCCR eye-tracking techniques is that the user needs to keep their head still for an uncomfortable amount of time in order to collect the most accurate results. Zhu and co's paper outlines a solution that allows the user to move their head freely and only requires one calibration procedure during the entire session rather than every time the head moves, making it more efficient.
 - We plan to implement our own gaze tracker with this solution in mind because it would make our model more user-friendly.
3. Rodrigues, J.A., Vieira de Castro, A., Llamas-Nistal, M. (2025). Integrating Eye-Tracking, Machine Learning, and Facial Recognition for Objective Consumer Behavior Analysis. In: Schmorow, D.D., Fidopiastis, C.M. (eds) *Augmented Cognition. HCII 2025. Lecture Notes in Computer Science()*, vol 15778. Springer, Cham. https://doi.org/10.1007/978-3-031-93724-8_5

- This paper introduces ML-based methods to use real time facial data and gaze data to learn about user fixations in certain areas of static images. We are interested in their approach in collecting real time data and computing gaze information.
 - Furthermore, we found this paper relevant due to its focus on a framework that improves the scalability, affordability, and accessibility of webcam-based eye tracking. By allowing gaze tracking from the user's own laptop, it appeals to our goal of targeting the general public.
4. Saxena, S., Fink, L.K. & Lange, E.B. Deep learning models for webcam eye tracking in online experiments. *Behav Res* 56, 3487–3503 (2024). <https://doi.org/10.3758/s13428-023-02190-6>
 - This paper explores gaze detection using webcam data, and outlines several methods and benchmarks to train and evaluate a gaze detection model.
 - We plan to take note of their roadblocks and shortcomings and attempt to make something that improves on these approaches.
 5. Krafka, K., Khosla, A., Kellnhofer, P., Kannan, H., Bhandarkar, S., Matusik, W., & Torralba, A. (2016). Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2176-2184). <https://arxiv.org/abs/1606.05814>
 - This paper introduces a massive dataset of over 2.5m frames of gaze data from over 1400 people, and introduces a simple baseline training approach for eye tracking.
 - We plan to use the dataset mentioned in this paper and see where we can improve on this paper. Our goal is to learn from the literature and create a model that takes the benefits of these approaches and can train and run efficiently on modern consumer devices.

3 Potential Datasets

We have found a bunch of datasets on Kaggle that relate to various aspects of our training process (pupil detection, gaze detection, etc.), but further research is required to determine the specific datasets we will be using. Below is a list of datasets we plan to base our initial work on:

1. UnityEyes Eye Gaze Detection (kaggle.com/datasets/hazemfahmy/eye-gaze-detection)
2. Eye Movement Data Set for Desktop Activities (kaggle.com/datasets/namratasri01/eye-movement-data-set-for-desktop-activities)
3. Eye Detection Dataset (doi.org/10.34740/kaggle/dsv/11920378)

4 Plan of Activities

In the final report, we plan to have created a working gaze tracker that can be run on consumer-grade devices in real-world settings. The idea is that we will have two synchronized videos—one capturing the user's face and the other recording the screen they are viewing—and our gaze tracker would be able to analyze the facial video to detect the approximate area on the screen the user is looking at. Ideally, we will have a demo showing this functionality to add to our final report. We plan to follow this plan of work:

1. Train model to detect eyes/gaze
2. Model the projection of gaze onto screens, i.e. detect what the user is looking at
3. create a system/interface to run this and display this using visual techniques

As for workload distributions, we will first do research together to get a better idea of the scope and knowledge that would be needed to finish this project. We plan to work on all aspects of the project together, from collecting datasets to processing the data to implementing the model, and to be in constant communication.