

# Project Progress Report

**Team name:** *Sigmoid Chi*

## Instructions

- What are the skills of the team members?

We are all confident Python programmers and our JavaScript skills range from adequate to expert. Two of our members are relatively comfortable with tackling the Tensorflow library. We have also previously worked in a team project before so understand our team dynamics and personal work tendencies, so don't anticipate running into any logistical/organizational problems during this process.

- What is your idea?

We would like to improve WebGazer by removing the calibration stage where users have to click to help the program learn their gaze. Currently we hope to train WebGazer with the provided data so that the program *does* begin with prior knowledge, and continue training the model as the user clicks/looks at the screen. Hopefully this will allow us to report when gaze predictions actually obtain some degree of accuracy, rather than assuming this is the case from the start and training as the user continues interacting. Additionally, we would also like to attempt at implementing a different regression model given the training we hope to execute beforehand.

- What data will you use?

The frames dataset provided in the class directory and potentially the full dataset if we need more information to train on.

- What software will you use?

- JavaScript
- Python
- TensorFlow

- Who will do what?

We haven't determined specific roles yes, but anticipate that the majority of our team members should become 'experts' on Tensorflow for building and training the model and the remaining member(s) should gain expertise with the current state of WebGazer in order to incorporate the trained model with the library such that upon initialization, WebGazer can obtain a better/more informed accuracy score instead of just assuming that it is accurate to begin with.

- What progress have you made so far?

We have familiarized ourselves with the WebGazer library.

- What problems do you foresee or have?
  - A new/different model will be too computationally expensive and may not perform well in real-time. We may also sacrifice a lot of time training with the risk of being unable to observe enough improvements.
  - Limited experience in JavaScript may make the initial learning curve to get the ball rolling on this project steep.
  - Building a Tensorflow model from scratch may prove difficult especially from the difficulties encountered in the previous project.
- Is there anything that we can do to help?
  - We're not really sure how we can implement a pre-trained model within the WebGazer library.
  - Support on setting up a Tensorflow model from scratch will be extremely helpful.