University of Arkansas – CSCE Department
Capstone I – Preliminary Proposal – Fall 2020

Emotion & Gesture Tracking Browser Extension for use in Classrooms

Caleb Duke, Allison Frye, Gage Robinson, Cassidy McManus, Ryan Rau

Abstract

COVID-19 has altered the way we interact with large groups, forcing us to turn to a digital environment. The academic world has been hit the hardest, forcing children to literally grow up, academically and personally, via Teams, or other video sharing platforms. They must learn how to read or do math all while their teacher resides in a tiny square on their screen. The teacher must learn how to have a relationship with their students while keeping them engaged in course work. Both participants lack severe human interaction, beyond just the verbal cues.

By creating an emotional and gesture tracking system, we can restore a small piece of human interaction by emulating how a real classroom would function. Creating a chrome extension that can read in a student's emotion or capture certain gestures in real time and relay that back to the teacher can provide a small sense of relief on their side. Teachers can easily see who is interested, confused, or just left the class as naturally as they could if they were face to face. This will remove the awkward silences when a question is asked or helps teachers learn how engaging they truly are in lecture.

1.0 Problem

With the rise in popularity of online classes, students are finding it increasingly more difficult to stay engaged while attending lectures. The lack of any kind of face-to-face interaction between professors and students easily leads to students not having as much of a drive to pay attention and attempt to understand the material being presented. Moreover, professors often find it nearly impossible to gauge how students are feeling based off lack of physical and verbal cues given during the current lecture.

Ultimately, it is the responsibility of the student to attend online lectures, take notes, and ask questions. However, this style of instruction simply does not easily lend itself to normal classroom interaction amongst students and between students and professors. The implications of this range in severity. Some students may simply do worse in a particular class if they cannot engage like they normally would, while others may adjust to this new style of teaching and have a difficult time transitioning back to in person classes at a later date.

Without a proper solution to this issue, we will see a steady decline of a student's well-being in both academic success and social skills. Teachers can no longer engage with their students in the same way they once could, and students often feel like they are missing out on the true college experience. By not having a concrete solution, motivation for holding a productive learning environment will be lost, therefore emulating human interaction to some degree will help maintain classroom dynamic.

2.0 Objective

The objective of this project is to develop an application that allows interpersonal connections through online learning. With this application, our group hopes to bridge the current emotional and physical gap between teachers and their students due to virtual learning for more engagement between the two. We plan to do this using facial and gesture recognition to track the student's emotions, thoughts and behaviors during the class time which provides live feedback for the instructor. Through all of this, students and teachers can have a close to normal classroom experience once more.

3.0 Background

3.1 Key Concepts

These problems facing us today are nothing new, even if COVID-19 has definitely exacerbated them. In a 1998 peer-review article by Jill Galusha, Md, published by the US Institute for Education Sciences, she notes that distanced-learning students at that time felt had a feeling of isolation and where insecure about self-evaluation and their own progress [1].

The first key concept is the use of non-verbal feedback from students to their instructors. We propose that restoring ubiquitous bidirectional learning may help alleviate some difficulties in online classrooms. By this, we mean to restore the ability of students to non-verbally respond to the instructor at any time without interrupting the learning experience of others. In in-person classrooms, students do this largely through posture, eye-contact, and facial expression. This means that instructors can learn from non-verbal student feedback and adjust their own teaching style and presentation methodologies to increase student engagement. However, this form of feedback has yet to be properly implemented in an online teaching environment.

3.2 Related Work

With the advent of Amazon's Rekognition software in 2016 [2], a powerful computer vision tool, we see a new avenue to restore effective in-class student feedback in an online environment. Particularly relevant to our project is the ability of Amazon Rekognition to analyze and report facial expressions from source images in a prompt and detailed manner [3].

4.0 Design

4.1 Requirements and/or Use Cases and/or Design Goals

Our application falls under two use cases: one as a student and one as a teacher. According to one's role, different views will be presented along with functionality. Below are the respective requirements:

<u>Student Requirements</u>

1. Allow students to login and direct them to a modal asking for a classroom password. LDAP may be incorporated.
2. Ask students if they may have access to the camera.
3. Allow students to use Zoom as usual.

<u>Teacher Requirements:</u>

1. Allow teachers to login, while generating a classroom code to send to students.
2. Allow teachers to conduct a Zoom based class experience while being able to view live reactions.
3. At the end of a session, a report will be generated to allow the teacher to analyze their students emotional state over the class period

4.2 High Level Architecture

We are taking a microservices approach by compartmentalizing the functionality of our project. As these stand-alone services, the core parts will be the backend, ML Models, and the Chrome Extension.

<u>Backend</u>

Our backend will be comprised of two parts: the database and the RESTful APIs. As of now, we plan to test for confusion, contentment, or long periods of inactivity/ absence, by taking a succession of pictures and averaging out which mood the student is in. This data will be sent directly to the database to allow us to pull the data when crafting the final report as well as being sent to the teachers live reports table. No personal student information will be stored, however if we plan to have a focus on a grade school setting, the data could be stored for each student and evaluated at the end of a certain time period.

The APIs will allow communication between AWS Rekognition, database, and the extension. The language will be selected when the time comes to connect all the components, however JavaScript and Python are top contenders. Ideally, the Zoom feed will be sent directly to AWS Rekognition, deciphered appropriately, then the emotions will be sent to the database as well as the teacher's live feed.

ML Model

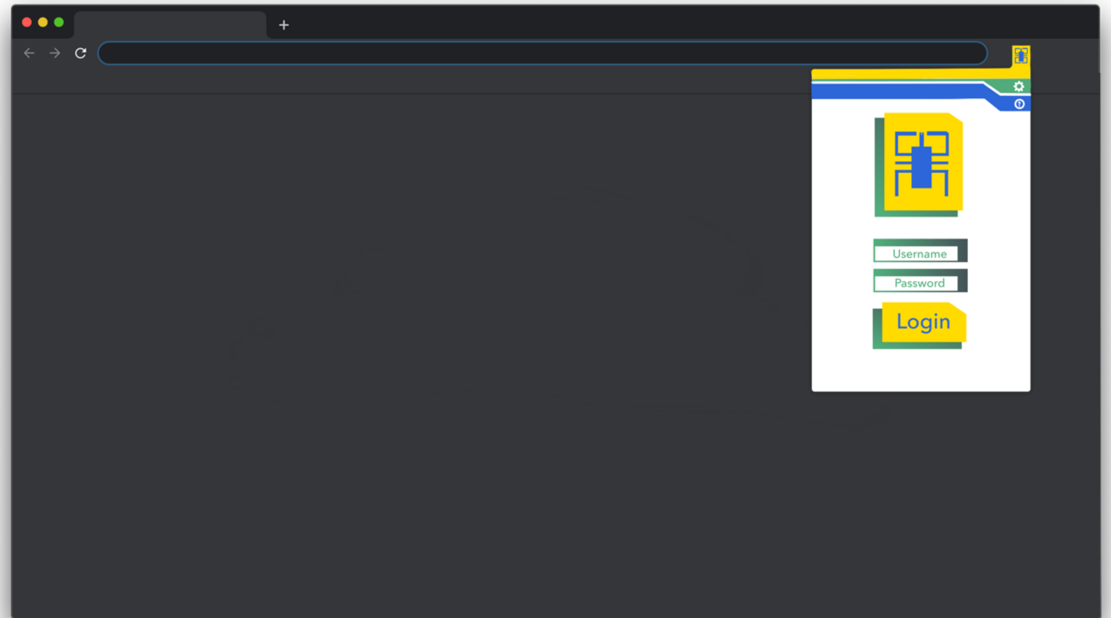
At its most basic state, our project will consist of AWS Rekognition machine learning models. Rekognition, provides us with easy to access pretrained ML models which can return the perceived emotions of a person from a still image, video clip, or video stream. Our chrome extension will provide the data (either images or video clips) needed by Rekognition and return its results of perceived emotions to our backend for further processing and formatting so that it can be displayed to the teacher in an insightful way.

Once base functionality is set up, we'll look into expanding the capabilities of what we can detect from the data from the extension. This will consist of either building off of the AWS environment or building our own models using tools such as Google Teachable Machine or PyTorch. With additional ML models we could detect gestures or inactivity on top of perceive emotion bettering the data we can return to the teacher.

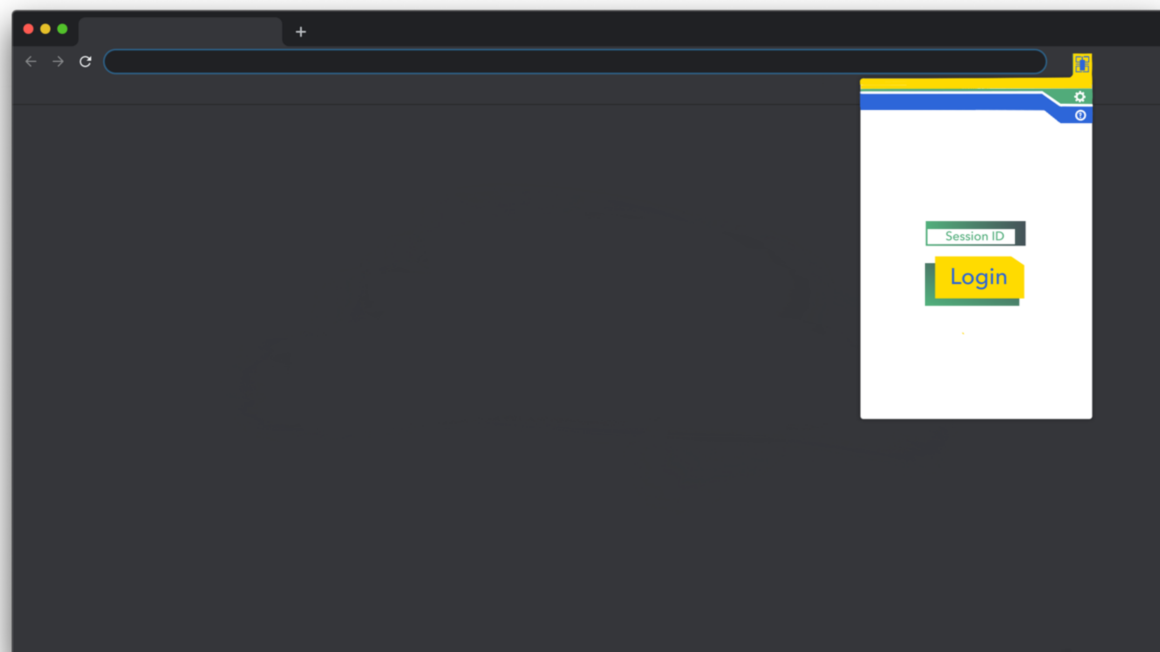
Chrome Extension

The Chrome extension will allow both the student and teacher to login and display the correct functionality. As the students will not need much of a visual aid, we will focus primarily on the teachers side, especially with the final report. Our two options are embedding a React App inside the extension or creating a URL that redirects the teacher to a site that is auto populated via the teacher's ID which is connected to the emotions stored in the database. The former option allows quick access to the data, however the latter would allow for more control development wise.

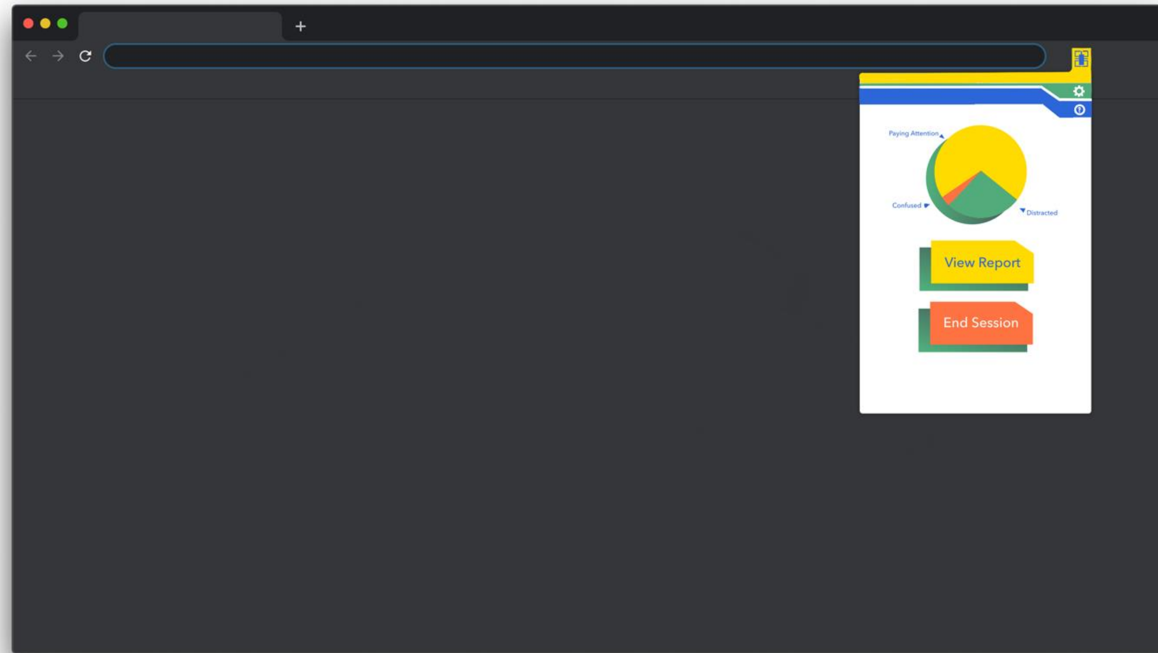
We have fabricated a series of mock-up designs for the user interphase:



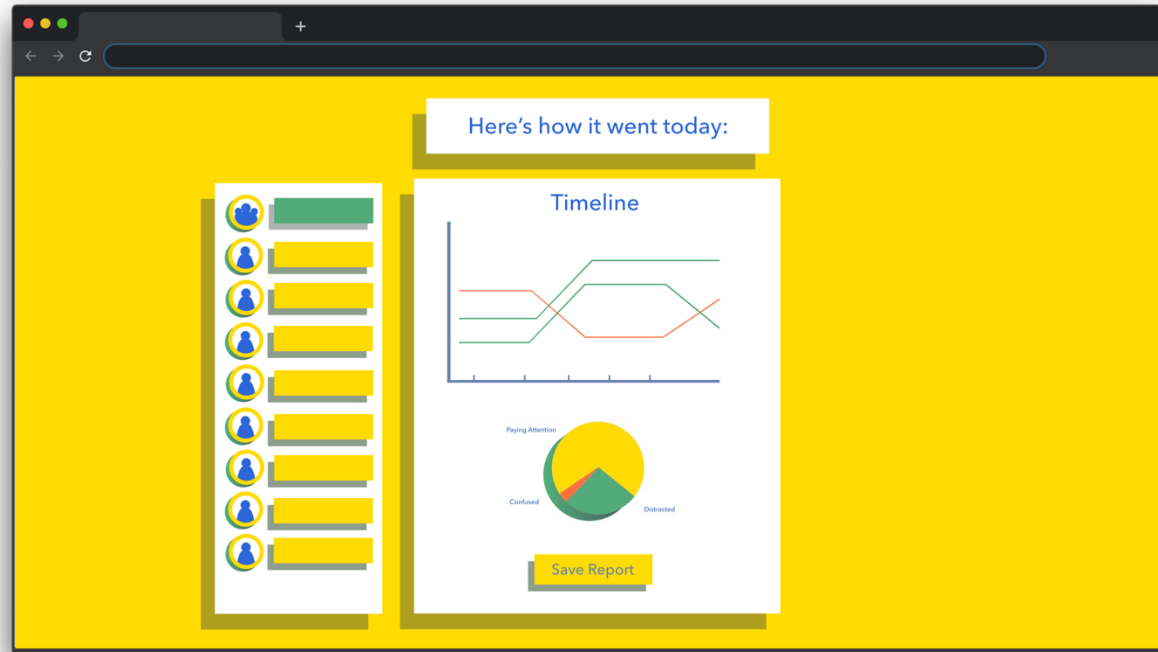
1: User Login for Students and Faculty



2: Session Login page for Students



3: Instructor’s view of session



4: Reports page generated for Instructors

Change title appropriately:

- For proposals: This is a high level design. Include some graphics and 1 or more pages of description of the proposed design (and steps you have taken to refine the design), including all major hardware and software components.
- For final reports: This is a (preliminary or final) detailed design that is fairly detailed, describing the architecture, technologies used, interface design (with screen shots), implementation, lessons learned, potential impact, future work, etc. 4-8 pages

4.3 Risks

Risk	Risk Reduction
Performance impact of browser extension's connection with backend and ML models	We'll test the performance impact of our extension sending video clips, still images, and a live video feed to determine which is the most optimal in terms of impact on the user's machine and feedback form the ML model
Real time data getting out of sync between users and DB	Implement web socket connection between our browser extension and backend to ensure that data is updated in real time. Additionally, include a level of error checking that will warn the user when a connection has been lost.
Users accessing data which they should not have access to	Implement a role system to distinguish between students and teachers. Also ensure that a user is authenticated when calls to our API are made.
Privacy concerns	Address privacy concerns in a terms agreement informing the user of what data is being accessed.

4.4 Tasks

1. Prelim Research: We'll begin by taking a deeper dive into the technologies we plan to use to ensure that by the spring we have a level of familiarity and comfort of what we'll be using. During this time, we'll also get a better idea of what our limitations are when it comes to our ideas that go past the scope of our originally presented project. With that information, we'll be better able to adjust our future plans to ensure that everything is reasonable and within our capabilities to ensure success.
2. Initial Setup: During this time, we'll setup up the basic frameworks for each component within our project ensuring that all team members have a running local copy along with access to the parts of our projects that will be hosted elsewhere.
3. Backend, Extension, ML Models: Splitting up the work and working in parallel on the three separate components will allow us to quickly get the ball rolling on this project. During this time our focus won't be to get things to work with one another but rather to work on their own and be ready to connect to the other parts. At this time, we'll also have a better understand of how the data in each step will be formatted which will aid us later.
4. Combining Components: With the three components being setup independently we will then get together and start bridging the gaps making all the components talk to one another.

Mainly solid communication to and from the backend API from the extension and AWS ML models.

5. Expand AI: With AWS Rekognition mainly only supporting emotion states from one's face we think it would be beneficial to expand upon this in our project. That said, we'll spend this time either implement other premade ML model or by created models of our own to include gesture recognition along with anything else we can think of along the way. This task will be influenced mainly by our finding in our prelim research.
6. Reporting: On the teacher side of the user experience, we'll want to provide the teacher with a nice interface to see all the data we collect. During this time, we'll design and build a reporting site that will present that data in a way that can be insightful to the teacher. This will include graphs that will show timestamped emotions along with the other data we end up collecting with additional ML models.
7. Optimization and Testing: Though we plan on testing everything as we build it, its good to take some time to really push it to its limits. At this time, we'll try to break our program exposing any bugs that may have slipped through the cracks and address them as needed. Additionally, we'll use this time to clean up any spaghetti code that might be hindering performance.
8. Final Touches and Report: At this point everything should be working, and all the kinks should be worked out. With that all that's left is to make everything look nice and to wrap up the project with the final report.

4.5 Schedule

Tasks	Dates
Sprint 1 (Prelim Research) • Iron out DB structure • Determine best languages for backend • Explore AWS Rekognition capabilities & limitations • Explore other methods of training ML models for use in our app • Look into limitations of browser extension	11/7 - 1/11
Sprint 2 (Initial Setup) • Get both our DB and base Backend set up on servers • Set up AWS environment, ensuring team members have access	1/11 - 1/25
Sprint 3 (Backend, Extension, ML Development) • Backend: Basic CRUD operations on DB, set up for connection to & from extension and AWS • AWS: Set up Rekognition models, get ready for it to accept data from backend • Browser Extension: Prepare for communication with backend, create	1/26 - 2/9

background task for image handling, get access to user's camera.	
Sprint 4 (Merge Components Together) • Link up extension to send and receive info from backend • Link up backend to send and receive data from AWS	2/10 - 2/24
Sprint 5 (Expand AI & Start Reporting Page) • Begin expanding upon ML models, adding more emotions/gestures • Begin teacher reporting page	2/25 - 3/11
Sprint 6 • Further expand upon ML models & reporting page • Begin to look into additional features	3/12 - 3/26
Sprint 7 • Continue with additional features • General optimization & testing project wide	3/27 - 4/12
Sprint 8 (Final Touches, Report, Optimization) • Clean the project up, final touches, final report, etc…	4/13 - 4/27

4.6 Deliverables

- Browser Extension: Browser extension which has access to the user's camera and sends that data to our backend API. Extension will also provide UI for the user to
- Backend: API + Database Schema Documentation: Document with all the APIs' endpoints, inputs, and outputs. Included will also be the languages used. In addition to API information, the database scheme and language will be listed.
- Machine Learning (Emotion & gesture recognition) models:
- Design Document: Document explaining how the three components interact with one another. All technologies will be listed.
- Final Report
- GitHub Repository List: As we are taking the microservices approach, each component will have their own repository, thus a location in which all links are consolidated will be given.

5.0 Key Personnel

Caleb Duke- Duke is a senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has formerly served as a graphic designer and project lead at the UITS Professional Development department. He is currently working for Charger Marketing in Front-End web development and Graphic Design, as well as a research support assistant at the UARK College of Education and Health Professions. He is planning to continue to work with Charger Marketing after graduating in May 2021.

Allison Frye- Frye is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. She has completed the majority of the core CSCE courses ranging from Programming Foundations to Algorithms. She has completed internships with Michelin and Cerner as a Software Engineering Intern with a focus on backend work. Currently, she interns with Affirma as a Software Engineering Intern, serves as a First-Year Engineering Peer mentor, and works as an Extra-Help assistant in the CSCE Tech Office. She's set to work at Cerner after graduation.

Cassidy McManus- McManus is a senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. She has completed relevant courses such Programming Paradigms, Software Engineering, and other higher level CSCE courses, all of which have provided foundational skills for this course. McManus currently is a lab member in Dr. Di's TruLogic Lab, has web development experience using Python Django through UATeach, and works as a lab instructor for Digital Design. She is planning on attending graduate school at the University of Arkansas after graduating in May.

Ryan Rau – Rau is a 4^{th} year Computer Science and German major at the University of Arkansas. He has completed a majority of the core CSCE courses including courses such as Algorithms and Software Engineering. Additionally, he is currently taking Artificial Intelligence which could prove to be useful when it comes to expanded out ML models. Outside of classes, Rau works at Affirma as a Software Engineer Intern working on a variety of projects including frontend design and business data processing.

Gage Robinson- Robinson is a senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed a majority of his Computer Engineering classes. Currently he is taking Embedded Systems and will take Computer Architecture his last semester. He had a web development internship which was cancelled due to COVID-19 restrictions. He is set to begin employment as an Embedded Systems Engineer at Honeywell in Kansas City, Missouri at the end of May 2021.

8.0 Facilities and Equipment

Description of all facilities and/or equipment required and/or utilized for the complete project.

7.0 References

[1] Galusha, Jill M., "Barriers to Learning in Distance Education", Institute of Education Sciences, 1998. https://eric.ed.gov/?id=EJ584170
[2] "Amazon Rekognition – Wikipedia", https://en.wikipedia.org/wiki/Amazon_Rekognition
[3] "Rekognition – Video and Image – AWS", https://aws.amazon.com/rekognition/?blog-cards.sort-by=item.additionalFields.createdDate&blog-cards.sort-order=desc