BADGER VISION

A Solution for Face Blindness

Pinata Challenge, Pinata AI, Zoom Video SDK

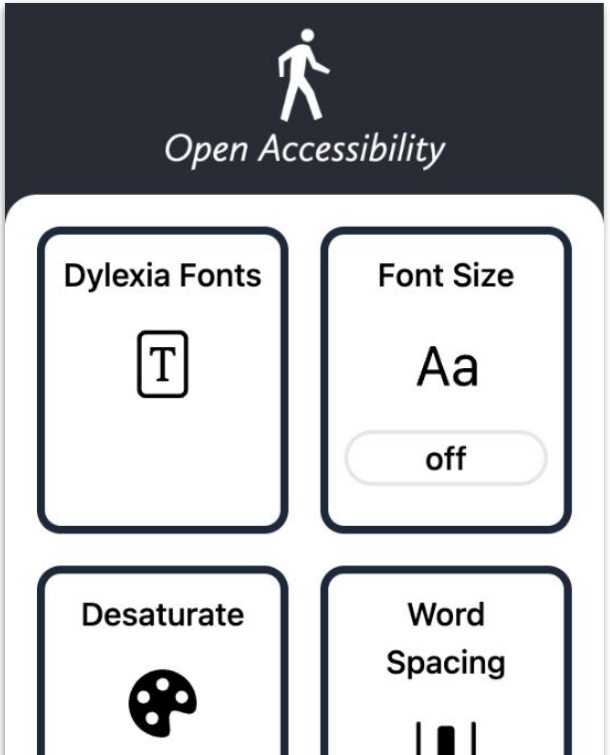
A project by UW-Madison Students

Max Maeder | Jeremy Kintana | Brennen Hill | Rahul Hathwar | Utkarsh Sharma

👓 Our Background

At a previous hackathon, we built a Chrome extension for **vision accessibility tools.**

We thought a good follow-up would be another accessibility tool that **uses AI classification!**

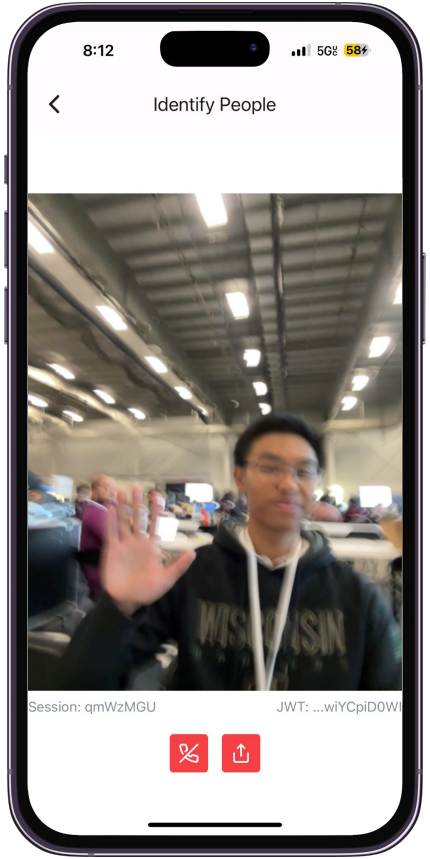


🤔 The Problem

Prosopagnosia is a condition where you struggle to recognize faces or struggle to interpret facial expressions.

- It affects about 2.5% of the population. In the USA alone, that's over 8 million people!
- "No therapies have demonstrated lasting improvements..." PsyPost

👀 Our Solution: Badger Vision



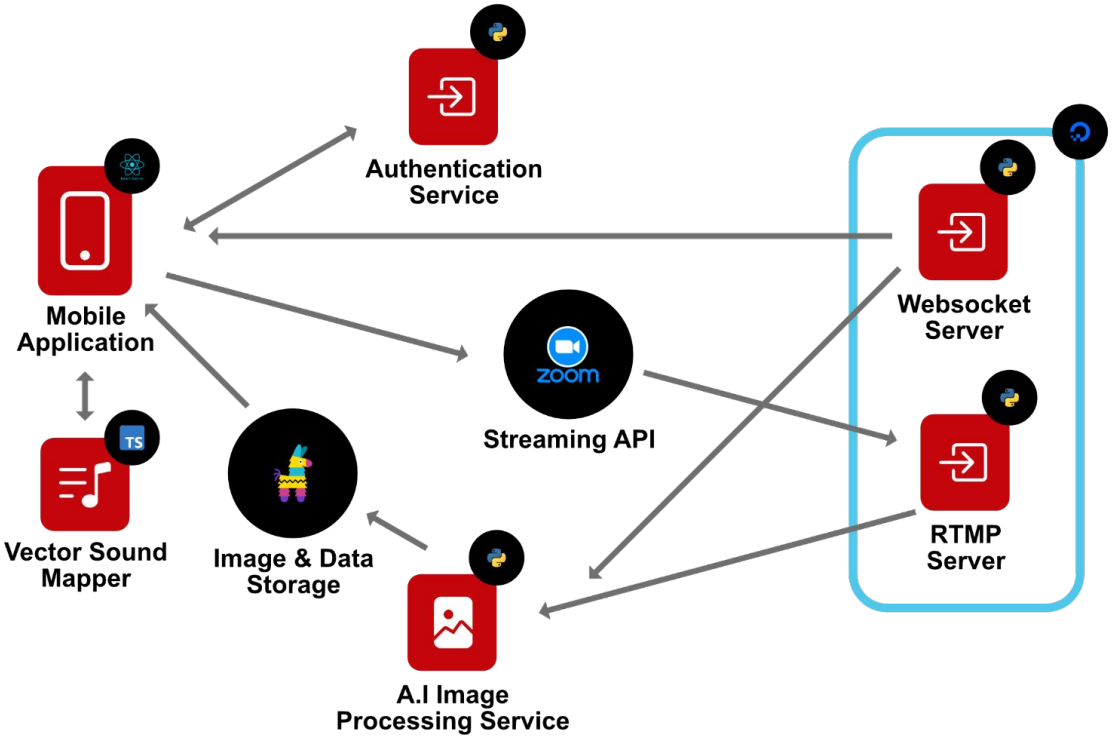
An open source, easy-to-use tool that **identifies emotions and faces.**

If the face is has a saved name, the tool says the **name of the person**.

If a face is unknown, the tool plays a **unique chime.** When that same face is seen again, the **same chime is played**.

Our model is also able to determine and announce the **emotion** of those present.

Demo Time!


Authentication Service
Mobile Application
Websocket Server
zoom
Streaming API
Vector Sound Mapper
Image & Data Storage
RTMP Server
A.I Image Processing Service

💻 How it Works

A user authenticates their Badger Vision app using a **QR code**, which downloads saved **AI artifacts** & configuration data from **Pinata**.

The user starts a session, and the app starts streaming video via our RTMP server to our AI recognition servers using the **Zoom Video SDK**.

The device can be placed discreetly in a shirt pocket.

Our AI recognition servers use **convolutional neural networks** and **deep learning** to identify unique faces and emotion seen by the device's camera, and sends detections back to the device over **WebSocket**.

Challenge: Pinata

We used Pinata for file sharing across our app.

We even wrote our *own* Pinata Python library to upload & download files!

For initial configuration: We use Pinata to store configuration info that authenticates a user of our app when they scan a QR code.

For management of facial recognition AI artifacts: We also began work on implementing Pinata to store and edit different labeled faces. This solution is not yet fully fleshed out and is still in development.

👨‍💻 Challenge: Zoom Video SDK

- The server generates a **Zoom Video session and JWT**, which the client retrieves by scanning a QR code.
- The client then joins the Zoom Video Session, and starts the video livestream
- Their camera streams from Zoom to our **RTMP server** where we run **computer vision analysis** to identify faces and emotions
- Most other copy pasted zoom examples into their apps
    - We know this because we read the code when there was no documentation and learned how to call the underlying methods
- When we talked with a zoom expert, Mr. Andrews he said our idea of live-streaming video through zoom to a server for real time data analysis was probably impossible
    - We proved him wrong.

🔨 Future Work

- Refine the codebase & **mobile client.**
- Run the application on AR and VR headsets, **including smart glasses.**
- Perform clinical trials on a **diverse population** to evaluate effectiveness.

🎉 Thanks!

Team Members:

- Rahul Hathwar
- Max Maeder
- Brennen Hill
- Utkarsh Sharma
- Jeremy Kintana

Built with 🦡

UW-Madison