SFU
VISUAL
COMPUTING

CMPT 742 Practices on Visual Computing I – Final Project Report

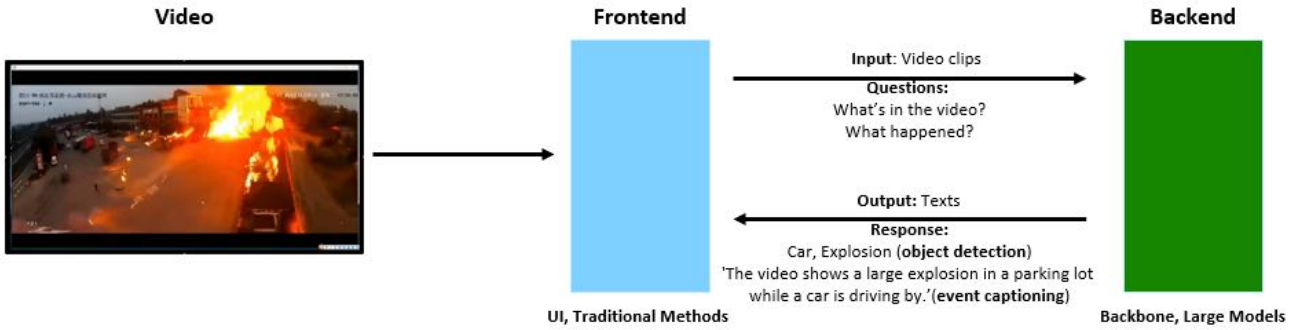
TotalRecall: A Software Framework for Capturing Events of Interests from Long Surveillance Videos

Description

Surveillance videos are typically long and it is tedious to spend time watching these to find key events. In the TotalRecall project, we aimed to tackle such issue and improve the efficiency with state-of-art models.

Overview

The Software Framework



Project Timeline

Milestone 1: Analysis & Baseline

Given the task of analyzing long surveillance video, we proposed three primary questions:

1) How to define and identify the temporal location of **key events**?
2) How to analyze **what** happens in these events?
3) How to ensure **efficiency and reliability**?

SFU
VISUAL
COMPUTING

We then developed a frame-difference-based methodology to extract frame energy metrics and established the core analytical tasks: key event detection, object labeling, and semantic event inference from video clips.

Milestone 2: Implementation

1. Developed a frontend user interface for result visualization
2. Implemented a backend server integrating two specialized video inference models
3. Established complete frontend-backend workflow
4. Optimized processing performance in both frontend and backend

Implementation

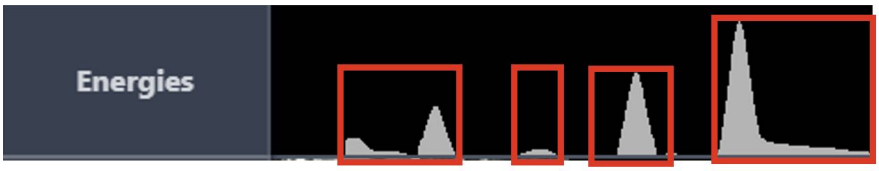
The Frontend

The frontend system serves two primary functions:

1. Implementation of traditional video analysis and visualization methods
2. Identification of potential events of interest for backend processing

The event identification process centers on a fundamental observation: significant events inherently create distinctive differences between consecutive frames.

By detecting regions of high frame difference, we maintain optimal recall rates for key events. This computationally efficient approach enables real-time processing in frontend JavaScript. Through the application of Gaussian smoothing and clustering algorithms, we identify time ranges with energy peaks, represented by four distinct video clips for further analysis.



Example of 4 events proposed based on energy peak analysis

SFU
VISUAL
COMPUTING

The user interface features an innovative Video Progress Bar incorporating three key elements:

1) Frame energy visualization
2) Frame thumbnail preview
3) 3D visualization highlighting moving objects

The trajectories of pedestrians and the raging flames can be clearly seen in the 3D view

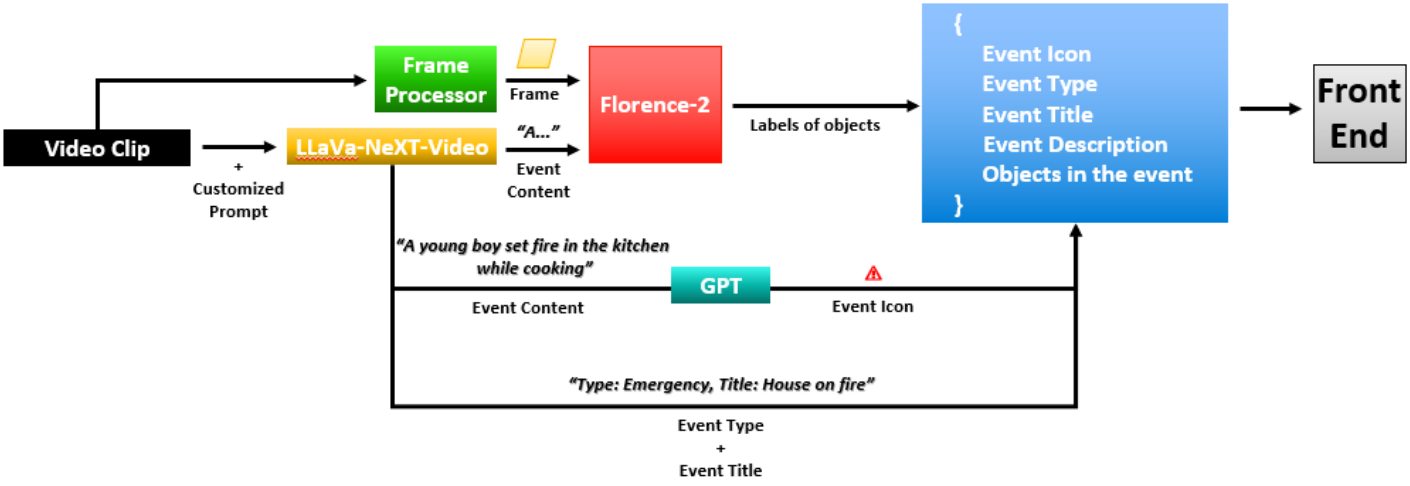
The Frontend provides initial event detection, while the Backend performs detailed analysis of identified segments.

Optimization:

We optimize video processing through ffmpeg.js, downsampling videos to 80x80 resolution at 2fps, enabling analysis of up to 5-hour videos while staying within browser memory limits (~2GB). Our parallel processing implementation using Web Workers achieves a 4x speedup over single-threaded execution, reducing processing time for 1-hour videos from 20 seconds to 5 seconds.

SFU
VISUAL COMPUTING

The Backend



The backend takes a video clip as its input, puts it into LLaVA with some customized prompts. The model gets the content and type of the event, accompanied with a title that summarizes the event.

The event content will be passed to GPT API to generate an event icon.

The model also has a frame processor that picks one key frame from the video clip and feeds it into Florence-2 with event content. So that it generates labels of objects in the current clip.

Lastly, the model combines everything into a json format response and passes it back to the frontend.

Optimization:

The entire pipeline is done in parallel to accelerate the processing time. The model also applies Fast-Attention and BitsAndBytes to overhaul the performance on consumer level GPUs for practical usage.

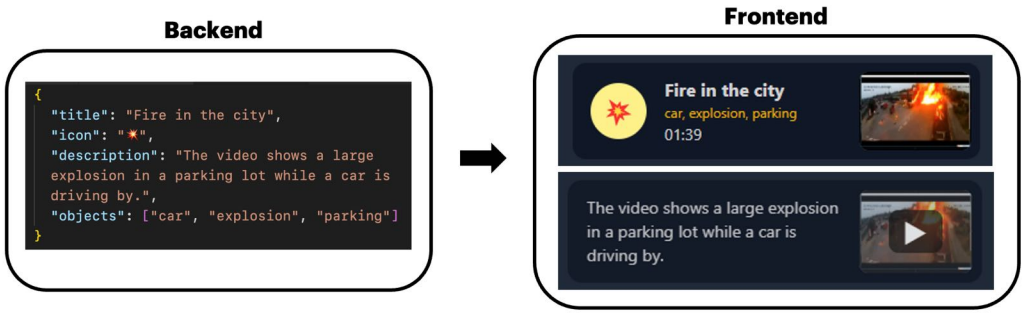
SFU
VISUAL COMPUTING

Combination

The frontend system processes backend responses for each event through specialized visualization methods:

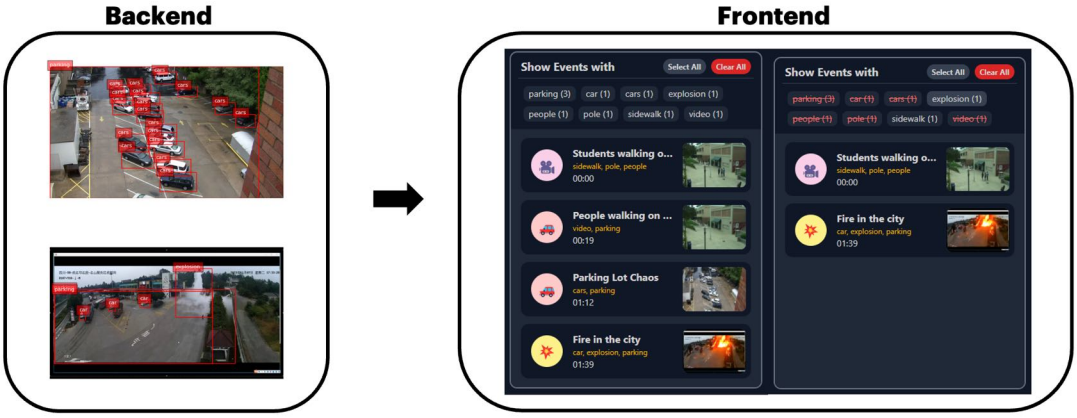
LLaVA model outputs are presented through interactive cards displaying:

1) **Front-side**: Title and emoji representation
2) **Back-side**: Detailed event description (revealed on mouse hovering)



Florence model outputs are processed to generate:

1) Interactive object **tags**
2) **Filtering** system for specific object types



Florence outputs are used for events filtering

These components integrate into a unified interface where all elements interact seamlessly, creating a cohesive user experience.

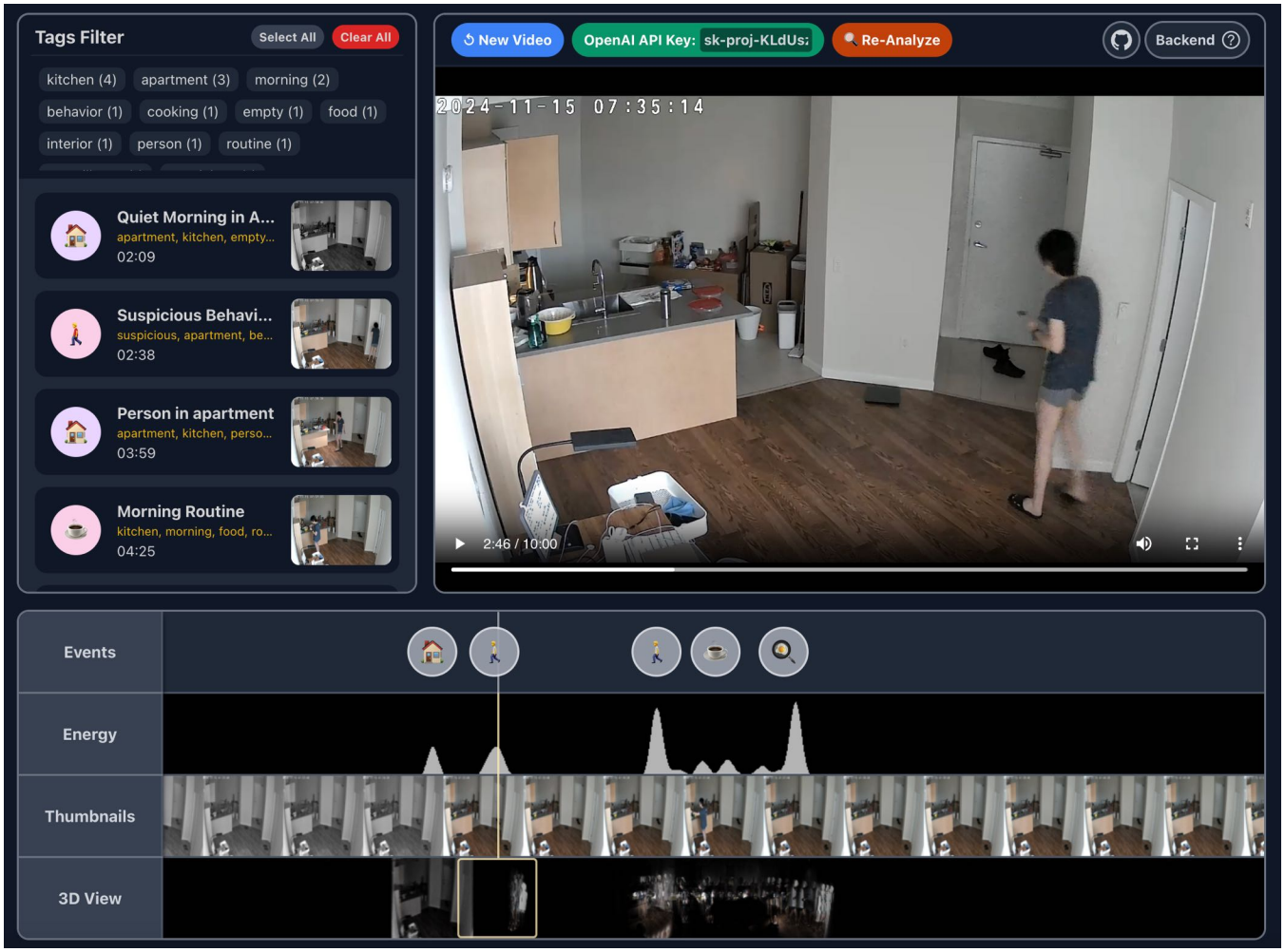
SFU
VISUAL COMPUTING

User Case

We demonstrate TotalRecall's capabilities through two distinct use cases that highlight both the frontend's standalone capabilities and the power of the complete frontend-backend integration.

Case 1: Rapid Event Location in Long-Duration Footage

To demonstrate the frontend's efficient event detection capabilities, we analyzed a home surveillance recording. The key challenge was to quickly locate specific moments of activity within this extended timeline.

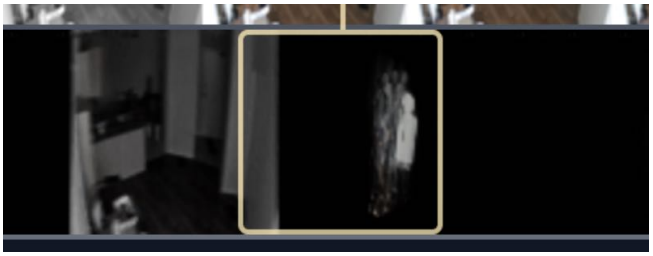


In the example above, our four-panel visualization interface (the bottom part) demonstrates the system's effectiveness in detecting and verifying events of interest.

SFU
VISUAL
COMPUTING

In the Energy plot, we observe several distinct peaks that indicate significant activity within the footage. These peaks are automatically processed by our frontend logic and marked in the Events plot as emojis, providing clear temporal indicators of important moments.

When clicking on an event emoji (🚶), the cursor will move to the corresponding position, and the interface reveals the presence of human activity in that frame. This is further reinforced by the 3D visualization panel, which provides a particularly intuitive representation of the event – displaying a clear human figure silhouette against the black background.



Human figure in 3D View

The 3D view serves as an especially powerful tool for rapid video analysis: periods of inactivity appear as pure black regions, while moments of interest are highlighted through distinct spatial patterns. This stark visual contrast enables users to quickly identify key events with the unaided eye, effectively reducing the time required for surveillance video analysis.

Through this integrated visualization approach, users can efficiently navigate through lengthy surveillance footage, focusing only on segments containing meaningful activity while confidently skipping uneventful periods.

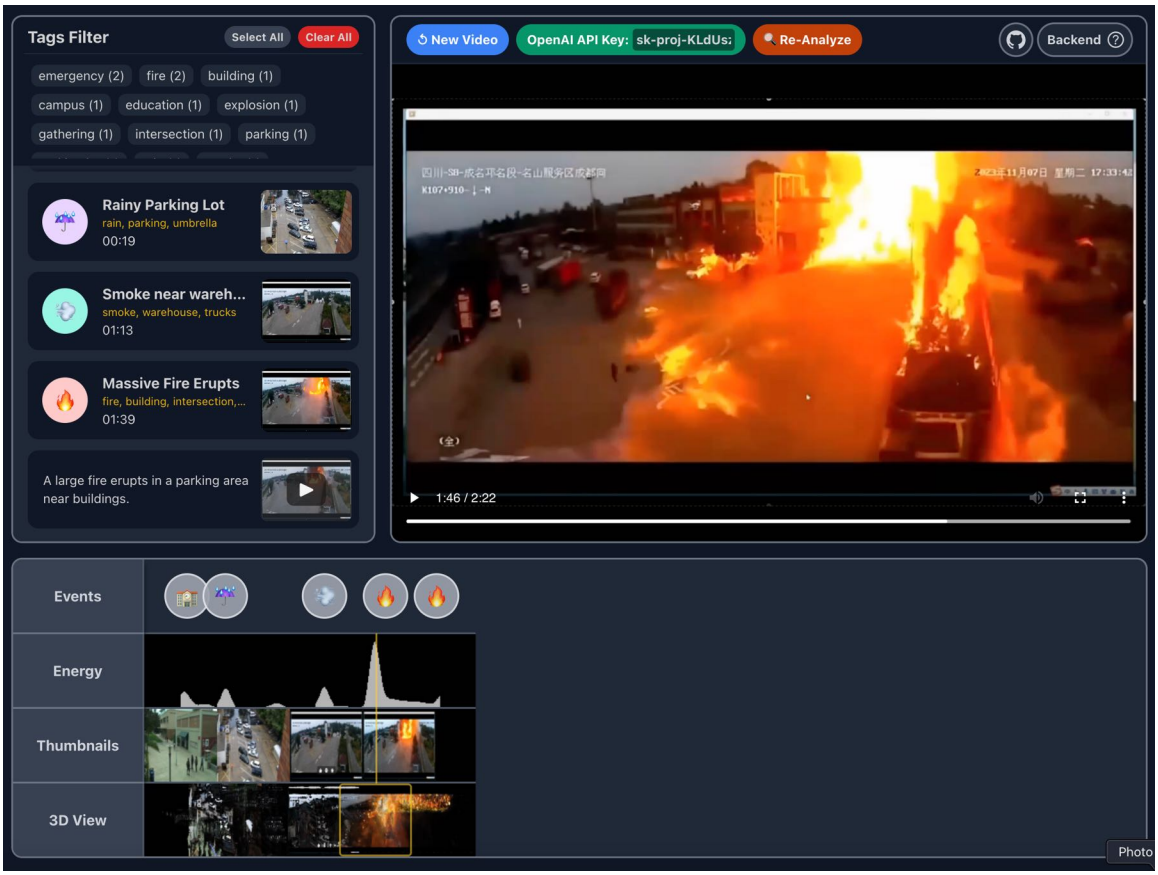
SFU
VISUAL
COMPUTING

Case 2: Multi-Scene Event Analysis

To showcase the system's comprehensive analysis capabilities, we tested it on a concatenated video sequence comprising three distinct surveillance scenarios:

1. Campus street footage
2. Parking lot surveillance
3. A large fire incident video

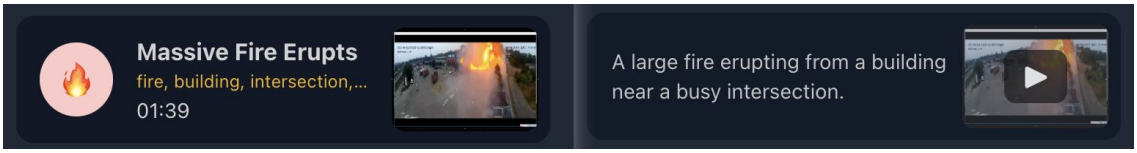
This diverse dataset challenged both the frontend event detection and backend analysis capabilities.



Events detected: campus walking students, rainy parking lot, and a big fire

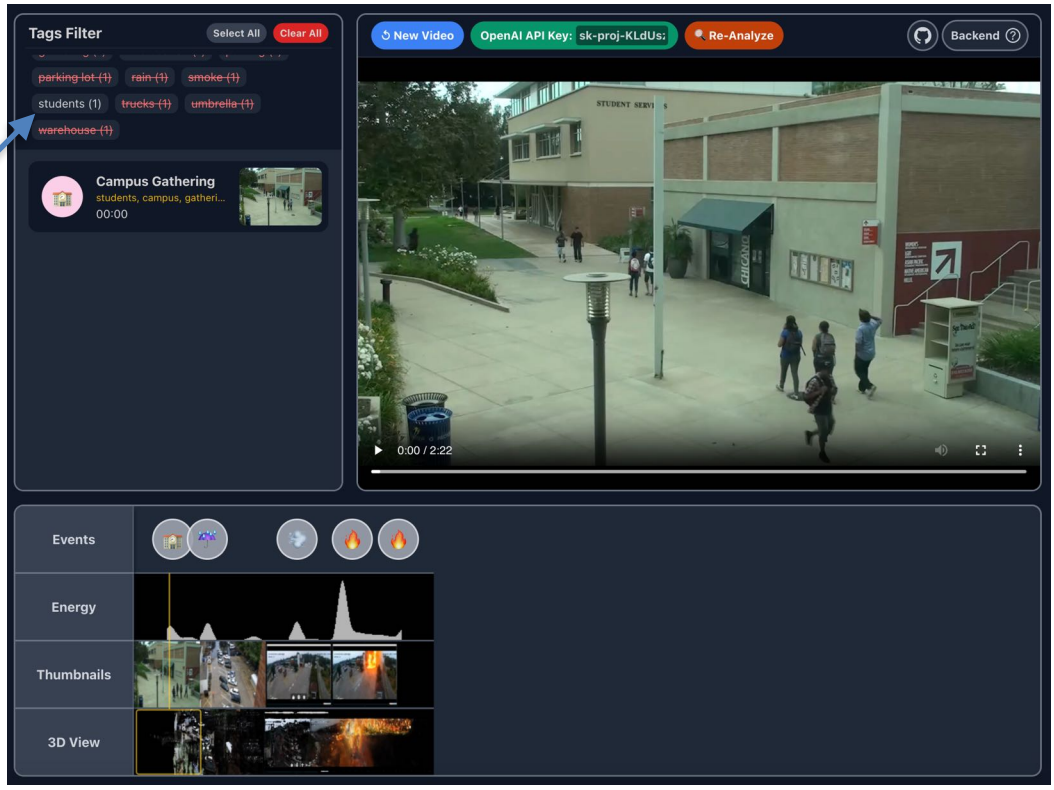
The frontend successfully identified all events across scenes, and generated distinct 3D views to help identify different types of events, especially the pedestrian movement and the raging flames.

For the backend analysis capabilities, the deep learning pipeline demonstrated sophisticated event understanding:



Backend generated title, emoji, tags and description for the event

- **Event Understanding**: The system accurately categorized both routine activities and emergency events, generating relevant tags and detailed descriptions
- **Object Detection**: Comprehensive tagging system identified key elements ("car", "pedestrian", "fire", "rain"), enabling efficient event filtering
- **User Interaction**: Tag-based filtering reduced analysis time, with event descriptions providing immediate context



We only filtered events that contains “students”

These use cases demonstrate TotalRecall’s versatility in handling both simple event detection tasks and complex multi-scene analysis scenarios. The system effectively bridges the gap between rapid visual scanning and detailed event understanding, making it a powerful tool for surveillance video analysis.

SFU
VISUAL
COMPUTING

Contact Person

Jiaqing Hu (Vincent) : jiaqingh@sfu.ca

Pengyu Cui (Raine) : pca119@sfu.ca

Contributor of the Project Idea

Jiaqing Hu (Vincent) Pengyu Cui (Raine)

Project Links

- **Online Demo**: https://pkucuipy.github.io/total-recall-webui/
  - An Example Video for Testing: https://raw.githubusercontent.com/PkuCuipy/total-recall-webui/refs/heads/example/example.mp4
- **Frontend Code**: https://github.com/PkuCuipy/total-recall-webui
- **Backend Code**: https://github.com/roast-my-resume/TotalRecall