Trademark Analysis & Identification Tool (TRAIT)

Team 11 Final Presentation

Lane Keck, Logan Taggart, Caleb Stewart

Problem Overview & Problem Statement

The Problem:

- Marketing teams lack efficient tools to measure brand visibility in visual media
- Manual analysis is often time consuming, error-prone, and costly
- Difficult to measure ROI on advertising campaigns and sponsorships

Our Solution:

- A computer vision based desktop application
- Automated logo detection and tracking in images and videos
- Calculated analytics on brand visibility and exposure

Target Users & Use Cases

Primary Users:

- Marketing teams trying to evaluate their campaigns ROI
- Brand owners measuring social media presence
- Event organizers demonstrating sponsor value
- Potential sponsors trying to assess visibility opportunities

Key Use Cases:

- Track advertisement performance within in videos and images
- Measure sponsor logo exposure at events
- Compare brand visibility against competitors

Sprint Progress

Winter Quarter - **Image Processing**

- Model training
- Multi-embedded voting system
- Achieved goal of functional image processing with logo matching

Spring Quarter - **Video Processing**

- Video analysis with FAISS integration
- Advanced features, such as progress tracking, smooth bounding box, and specific logo search
- Achieved goal of functional video processing with logo matching

Architecture Overview

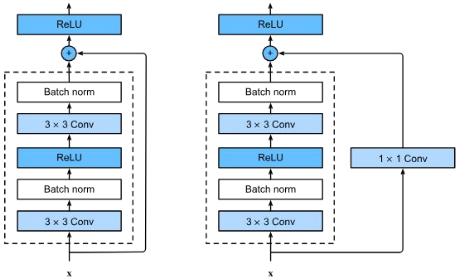
Frontend: React js with Electron js (to make it a desktop application)

Backend: Python Flask

AI/ML Stack: YOLOv8, CLIP, BEiT, ResNet embeddings, and FAISS

Deployment Stack: Docker, Bash Scripting

ultralytics
YOLOv8



docker

YOLO Machine Learning

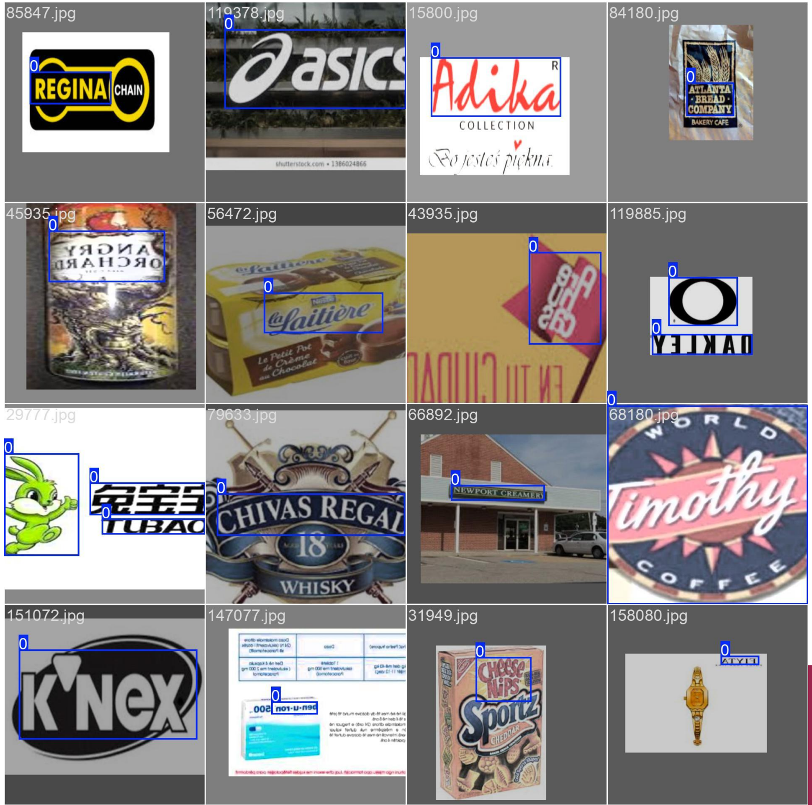
YOLO - You Only Look Once

YoloV8 was trained on the LogoDet-3k dataset

- Comprised of 200,000+ images, with 3,000 different categories

Performance Metrics:

- Recall: 76.73%
- Precision: 81.02%



Custom Logo Matching System

Multiple Embedding Models: We use 3 different models to generate embeddings for each logo (BEiT, CLIP, ResNet)

Two Similarity Metrics: Cosine Similarity and Euclidean (L2) Distance

Voting System:

- Each model-metric pair ‘votes’ on whether a match exists, given a preset threshold
- This reduces false positives significantly

FAISS Integration:

- Stores embeddings of previously seen logos
- Allows for fast similarity search
- This prevents redundant voting by recognizing known logos instantly

Project Evolution: Initial Plan → Final Implementation

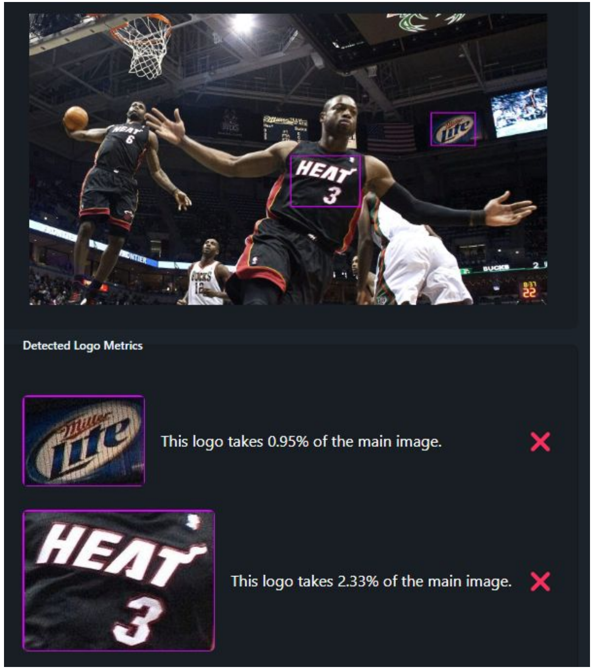
Initial Plan:

- Deploy the application on the web using services like Render
- The goal was to make it accessible anywhere via a browser
- Challenge: <u>Free tier limitations</u> couldn’t support the heavy computational load (ML models)

Final Implementation:

- Switched to a desktop application using Electron
- Runs locally on the users machine
- This allows us to:
  - <u>Run the application on the users system resources</u>
  - <u>Not rely on cloud service limitations</u>
  - <u>Produce a faster performance for model and similarity search</u>

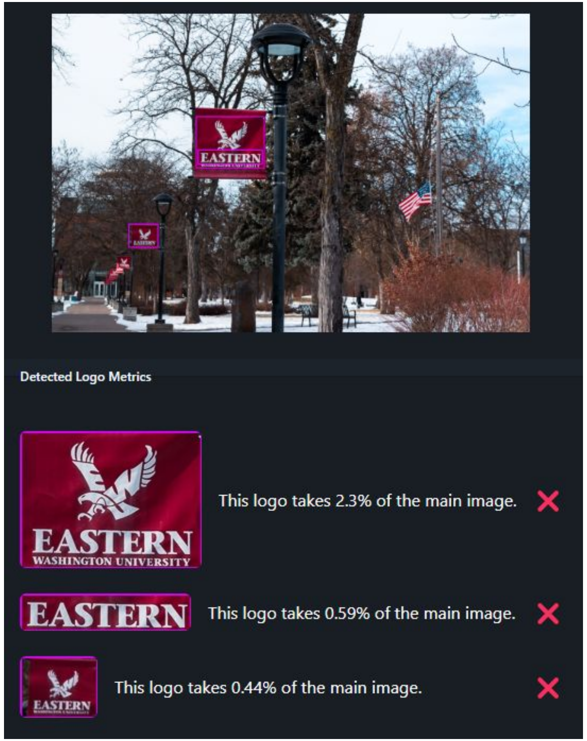
Image General Search Demo



Detected Logo Metrics

This logo takes 0.95% of the main image. ✕

This logo takes 2.33% of the main image. ✕

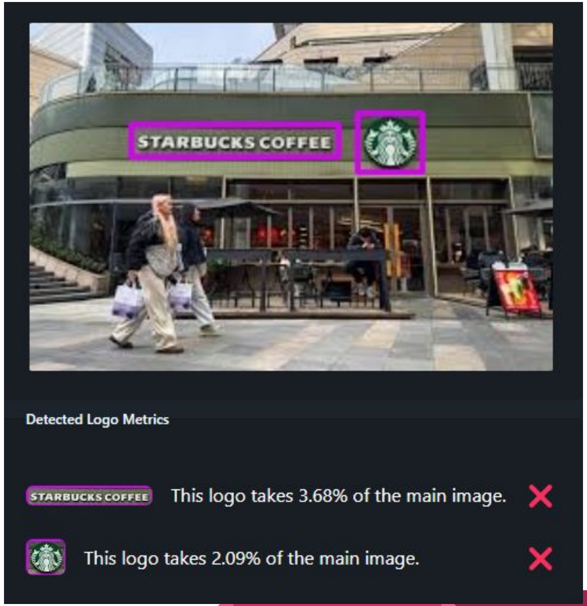


Detected Logo Metrics

This logo takes 2.3% of the main image. ✕

This logo takes 0.59% of the main image. ✕

This logo takes 0.44% of the main image. ✕

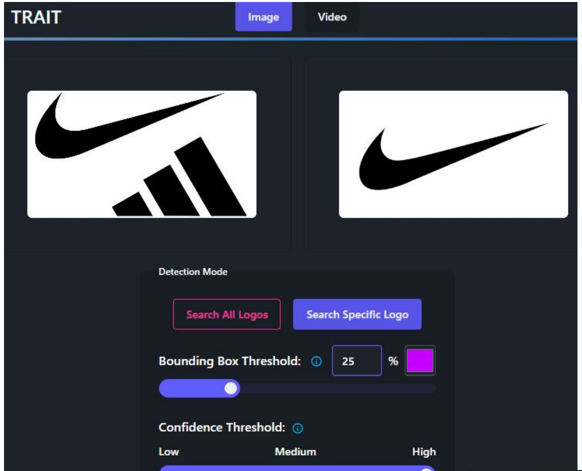


Detected Logo Metrics

This logo takes 3.68% of the main image. ✕

This logo takes 2.09% of the main image. ✕

Image Specific Search



Detection Mode

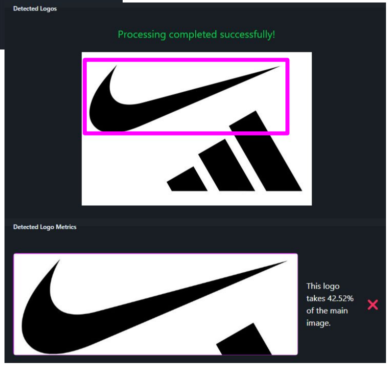
Search All Logos | Search Specific Logo

Bounding Box Threshold: 25 %

Confidence Threshold:

Low Medium High

High

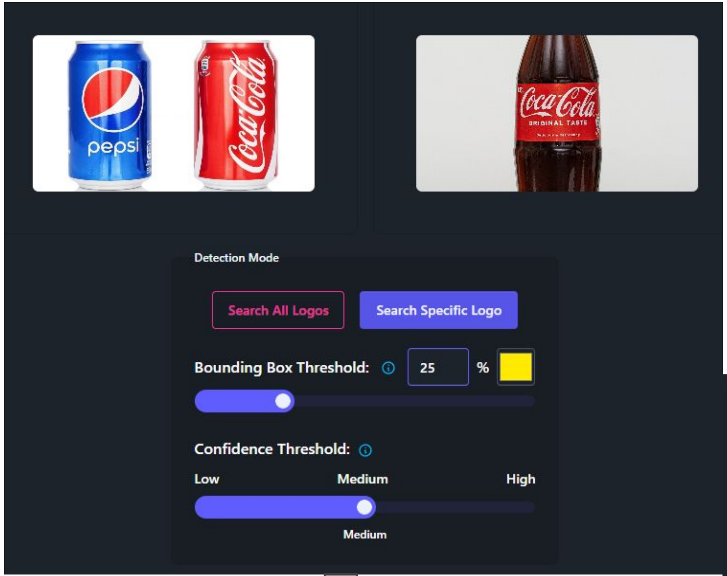


Detected Logos

Processing completed successfully!

Detected Logo Metrics

This logo takes 42.52% of the main image.



Detection Mode

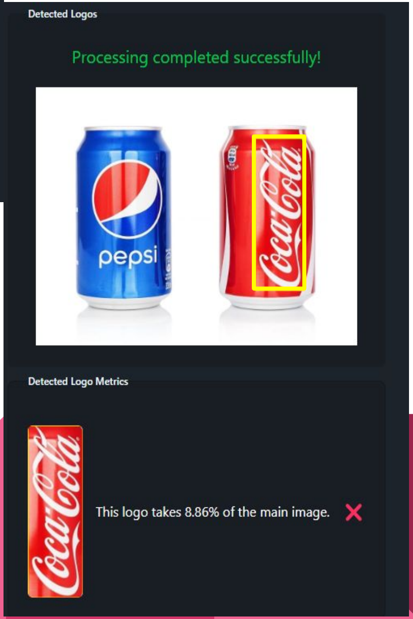
Search All Logos | Search Specific Logo

Bounding Box Threshold: 25 %

Confidence Threshold:

Low Medium High

Medium



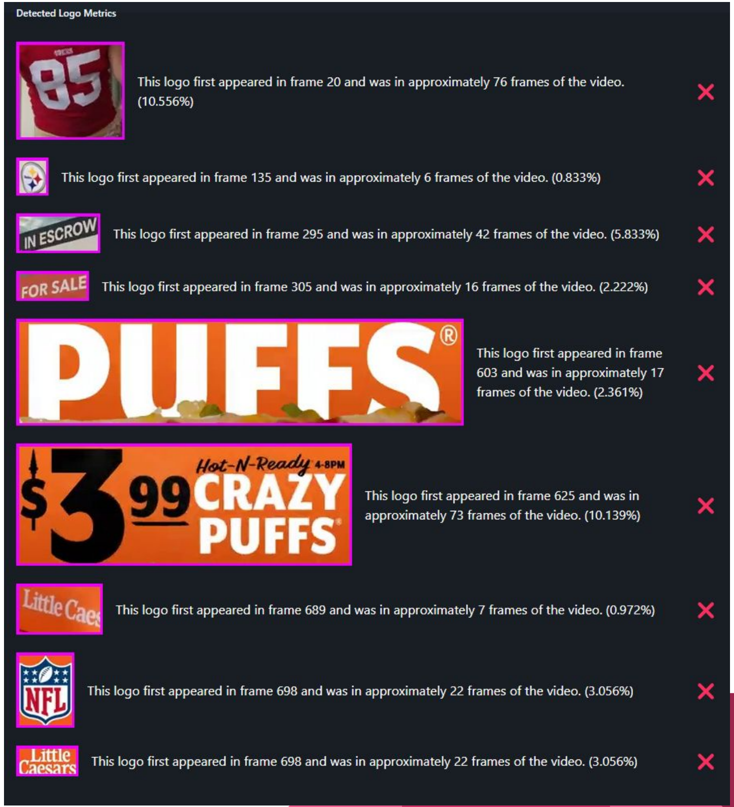
Detected Logos

Processing completed successfully!

Detected Logo Metrics

This logo takes 8.86% of the main image.

Video General Search



Detected Logo Metrics

This logo first appeared in frame 20 and was in approximately 76 frames of the video. (10.556%)

This logo first appeared in frame 135 and was in approximately 6 frames of the video. (0.833%)

This logo first appeared in frame 295 and was in approximately 42 frames of the video. (5.833%)

This logo first appeared in frame 305 and was in approximately 16 frames of the video. (2.222%)

This logo first appeared in frame 603 and was in approximately 17 frames of the video. (2.361%)

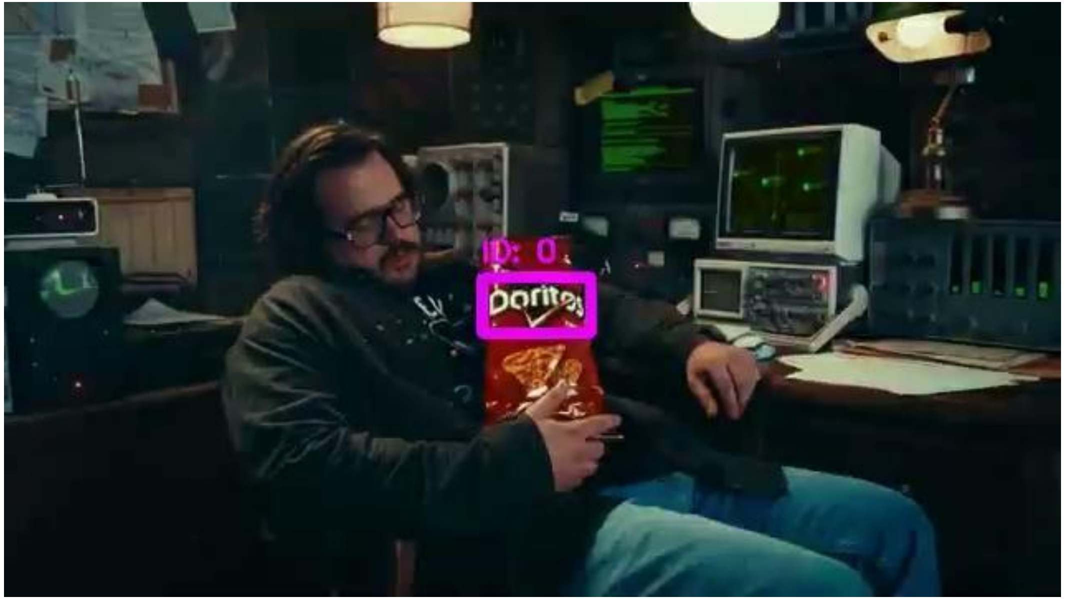
This logo first appeared in frame 625 and was in approximately 73 frames of the video. (10.139%)

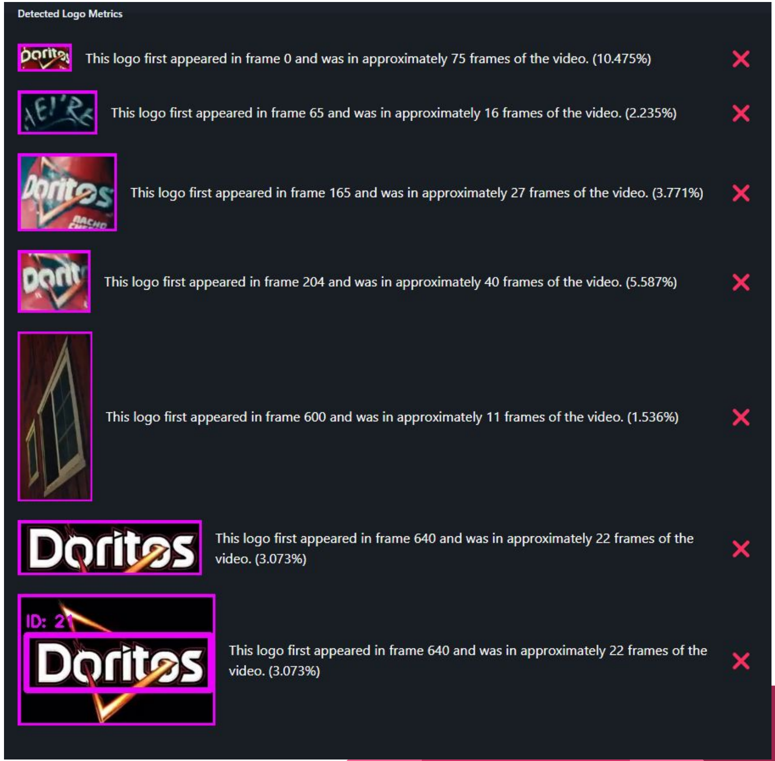
This logo first appeared in frame 689 and was in approximately 7 frames of the video. (0.972%)

This logo first appeared in frame 698 and was in approximately 22 frames of the video. (3.056%)

This logo first appeared in frame 698 and was in approximately 22 frames of the video. (3.056%)

Video General Search Part 2




Detected Logo Metrics

This logo first appeared in frame 0 and was in approximately 75 frames of the video. (10.475%) ✕

This logo first appeared in frame 65 and was in approximately 16 frames of the video. (2.235%) ✕

This logo first appeared in frame 165 and was in approximately 27 frames of the video. (3.771%) ✕

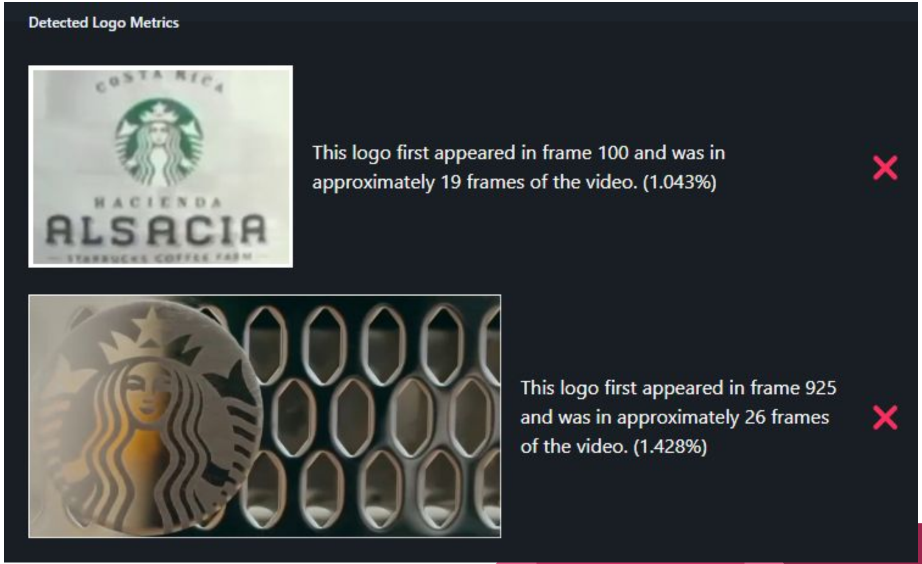
This logo first appeared in frame 204 and was in approximately 40 frames of the video. (5.587%) ✕

This logo first appeared in frame 600 and was in approximately 11 frames of the video. (1.536%) ✕

This logo first appeared in frame 640 and was in approximately 22 frames of the video. (3.073%) ✕

This logo first appeared in frame 640 and was in approximately 22 frames of the video. (3.073%) ✕

Video Specific Search


Detected Logo Metrics

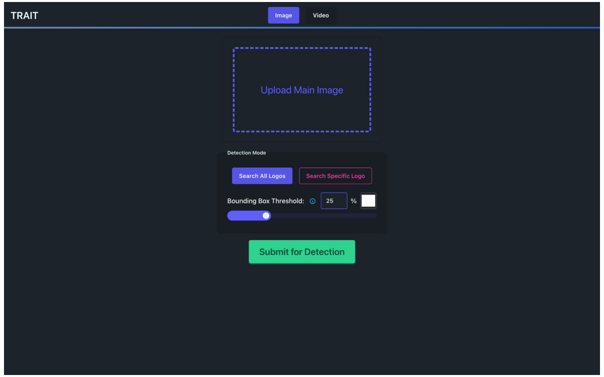
This logo first appeared in frame 100 and was in approximately 19 frames of the video. (1.043%) ✕

This logo first appeared in frame 925 and was in approximately 26 frames of the video. (1.428%) ✕

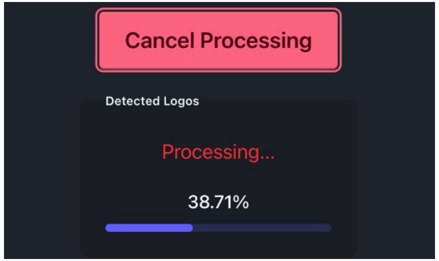
User Experience & Interface

Desktop Application Features:

- File Validation
- Cancel Button
- Progress Bar
- Metric display with logo removal options



Bounding Box Threshold: 25 %



38.71%

Attempted User Friendly Design:

- Minimal technical experience is required to run this application
- Simplified confidence thresholds

Application Performance

Technical Performance:

- Initial Load Time: Application launches pretty quick, but the first process for a piece of media takes about 30 seconds longer
- Video Processing: ~1 minute for a 10-second video
- Accuracy: Significant reduction in false positives with voting system implementation

<u>Performance depends on user’s machine as well as image and video size</u>

Challenges Overcome

Logo tracking in video: Implemented FAISS and adaptive frame processing

Image similarity search: Implemented with our custom voting system

Computational complexity: Solved by pivoting to a desktop application

Development Challenges:

- Model training (33+ hours)
- Cross platform compatibility
- Codec copyright issues (H.264 Codec)
- Making it an actual desktop application

Future Plans and Scalability

- Improve our YOLOv8 model training so that it has higher logo detection accuracy metrics
- Make improvements to the application’s overall processing time
    - This could be done by using ONNX instead of PyTorch to run the YOLO model?
- Develop advanced mode where users are able to tune more of the parameters
    - This would give more control to those with more ML knowledge while keeping it simple for those who don’t
- Optimize for long-term scalability and maintainability
    - Signing the download with certificates
    - Simplifying download process for an average user

What We Learned

- Training, using, and deploying a YOLOv8 model for real-world object detection tasks
  - Using and tuning different parameters associated with the model
- Process of identifying similarity between images
  - Extracting features with different embedding algorithms
  - Using different methods of determining level of similarity (Cosine, L2, FAISS)
- Complexity of building cross-platform applications
- How to containerize applications using Docker to solve this
  - Understanding how to build images and create/run containers from them