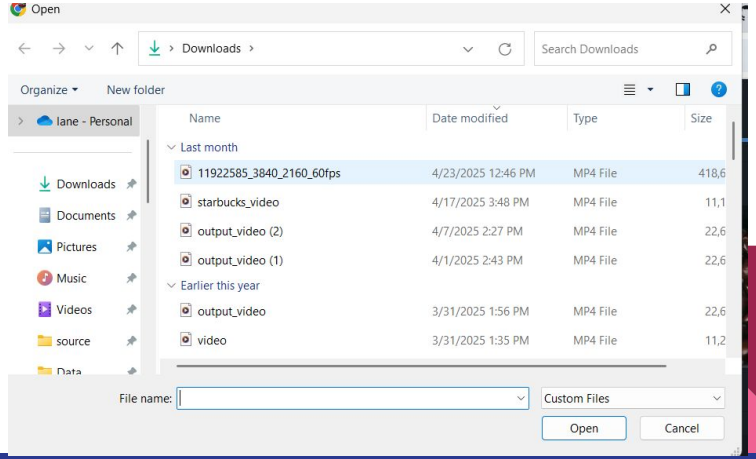
Trademark Analysis & Identification Tool (TRAIT)

Team 11 Sprint 14 Lane Keck, Logan Taggart, Caleb Stewart

Validate video/image extensions -Lane

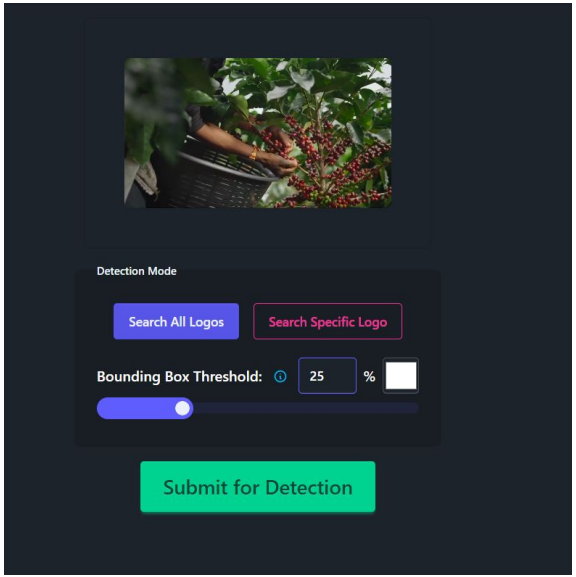
You can only upload files that are videos or images

- There is no way to for users to upload files that are not valid image or video formats
- .png, .jpg, .mp4, … (other file extensions we can upload)
- File explorer filters out different file types



Video Preview and Bounding Box Color -Lane

When a user uploads a video it will automatically play their video on loop, helps the user know what video they uploaded for processing.



The user can now also specify the color of the bounding box for video as is seen in the screenshot.

Loading bar when processing video

Processing a video may take a lot of time, so to tell the user how far along the video is processed, we created a progress bar representing how much of the video has been processed

This was done by streaming the current progress of the video every 1 second whenever a video is being processed.

Calculated by:

(current_frame / total_frames) * 100 → Returns a percentage

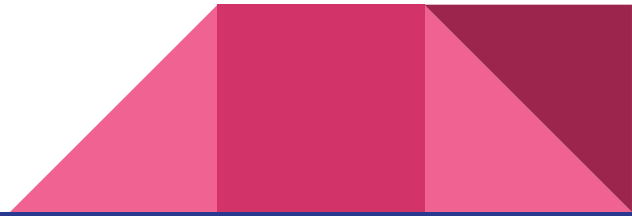
Smooth bounding box tracker

As we showed last week, this is done for the general video search.

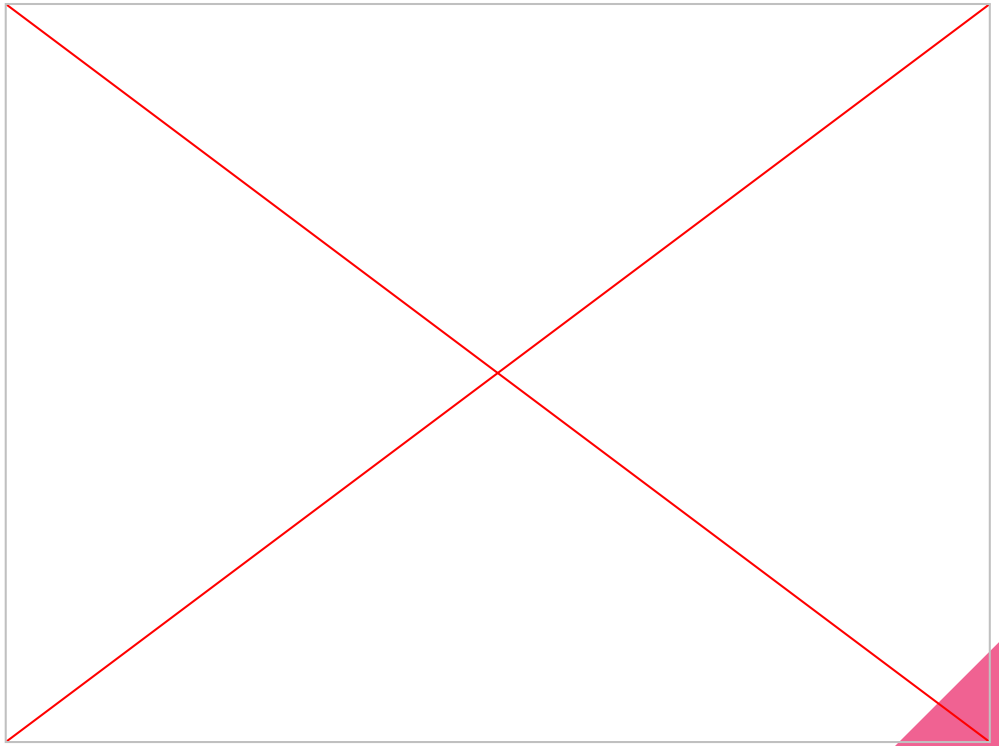
For specific video search we are following a very similar logic idea, except we only are switching to detect every frame after a logo is successfully determined to be a match.

This allows us to slightly reduce the time effect that detecting every frame has on our application since our model could find a logo that is not what we are looking for and thus we would have no logical reason to switch our detection frequency.

This has been both tested and implemented.



Smooth bounding box tracker demo

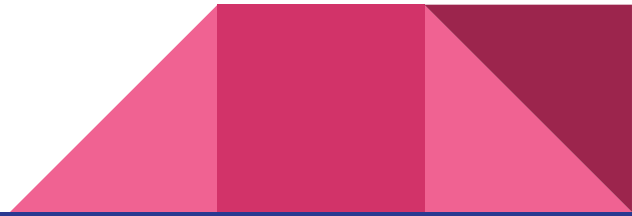


Lazy Loading

To speed up the initial loading process of our application we will be using lazy loading where we will relocate the importing of large libraries from the top of the files to inside the functions.

This will cause the libraries not to be loaded into our application's memory until they are needed as the individual functions are triggered.

Major libraries that we will be relocating that should reduce the initial loading time of our backend include ones like Numpy, Torch, Ultralytics, etc.



Still need to do

Make the submit button into a cancel button when a video is being processed

- This would allow for the cancellation of the logo detection process

Try to fix overlapping bounding boxes (One logo being detected twice)

Minor cosmetic improvements and changes

Package application into one executable file that users can download and use without installing additional dependencies