Outlook

[CS 488T] Sprint 13 Report, Team 11 [stewartc]

From shelbyemailrelay@gmail.com <shelbyemailrelay@gmail.com>
Date Sun 5/4/2025 4:35 PM
To Stewart, Caleb <cstewart15@ewu.edu>

Caleb,

This report describes the activities of your EWU Senior Project team over the previous self-evaluation period (usually Saturday through Friday). It contains only public information. Private information and comments, etc. are available only to the instructor. If you notice any discrepancies or have questions, please contact Dan Tappan at dtappan@ewu.edu.

Sprint 13 Team Report

Team 11: Trademark ID & Analysis Engine

· Lane Keck
· Caleb Stewart
· Logan Taggart

Logged Hours

The team is generally free to work whenever they want during the sprint. The expectation for a team of three members is 45 hours total (15 per member) on average. However, this number will vary throughout the course.

Individual Hours:

		All Sprints							
Member	Hours	Total	Min	Max	Avg[1]	Avg[2]	Std[2]	Count[1]	Missed
Keck	7.0	90.0	3.0	9.0	7.5	7.5	1.5	12	0 (0%)
Stewart	9.0	103.5	2.0	16.0	8.6	8.6	3.3	12	0 (0%)
Taggart	8.5	87.5	3.0	10.0	7.3	7.3	2.2	12	0 (0%)
Team Total:	24.5								

[1]including and [2]excluding missed submissions for required sprints

Team Hours:

Sprint																
1	2	3	4	5	6	7	8	9	10	11	12	13	Total	Min	Max	Avg	Std
8.0	27.5	24.5	28.5	25.5	22.0	18.0	0.0	29.0	25.0	23.0	25.5	24.5	281.0	0.0	29.0	21.6	8.1

The following is optional descriptions of daily work that is not captured as activities below:

Taggart:

- Working on packaging backend executable with Nuitka
- Minor code clean up/cosmetic changes
- Implementing final tracker for specific video search
- Debugging CSRT Tracker
- Implementing final tracker for general video search

Activities

Activities are member-defined units of work that are formally tracked from sprint to sprint (unlike the optional descriptions above). Every activity must be accounted for from its creation until it is completed or abandoned.

New Activities

These activities were created by during this sprint.

Keck

Activity 128: Create Preview for Video Upload

Autoplay video for user when they upload (one sprint expected)

Activity 129: Validate Extension Types

Users will not be able to upload just any file (one sprint expected)

Stewart

Activity 115: Add loading bar when processing video

Create a loading bar on the frontend showing how much of the video has been processed (one sprint expected)

Activity 116: Change bounding box color on videos

Allow the user to decide what color they want the bounding box color to be (one sprint expected)

Activity 117: Implement specific video search

Use prototype to implement specific video search into application (one sprint expected)

Taggart

Activity 118: Packaging backend into executable

Compiling backend into single executable with Nuitka. (two sprints expected)

Activity 119: Minor cosmetic changes

Making certain aspects look better. (one sprint expected)

Completed Activities

These activities were completed during this sprint.

Activity 114.1: Clean up code

Opened in Sprint 12 by Keck; expected to take one sprint.

Original description: Clean up all parts of the code

Progress in Current Sprint: The code has been cleaned up and made modular

---

Activity 110.1: Code clean up

Opened in Sprint 12 by Stewart; expected to take two sprints.

Original description: Make code more readable and reduce reused code.

Progress in Current Sprint: I successfully migrated the code so it can be used for images and videos.

Activity 111.1: Implement FAISS Specific Search

Opened in Sprint 12 by Stewart; expected to take one sprint.

Original description: Implement the prototype code of specific search

Progress in Current Sprint: The prototype was successfully implemented, and works well.

Activity 98.1: Video Processing

Opened in Sprint 9 by Taggart; expected to take three sprints.

Original description: Get YOLO model to work on videos

Progress in Current Sprint: Specific video search is now working and implemented.

Activity 105.1: MOSSE Tracker

Opened in Sprint 10 by Taggart; expected to take two sprints.

Original description: Implementing MOSSE tracker to interpret logo boundary boxes in between analyzed frames.

Progress in Current Sprint: I have finalized tracker design and abandoned the use of OpenCV trackers. This is now implemented for both video general and video specific.

Activity 112.1: Video Compression

Opened in Sprint 12 by Taggart; expected to take one sprint.

Original description: Manual compression of processed videos

Progress in Current Sprint: Videos now compress properly on both Windows and Mac before sending to front end.

Activity 113.1: Specific Logo Search

Opened in Sprint 12 by Taggart; expected to take two sprints.

Original description: Implementing specific logo search into application

Progress in Current Sprint: Specific logo search is working and implemented into our application.

Team Reflection

This section refers to the team's collective perception of and reflection on the project over this sprint.

The instructions are: Consider the following four pairs of questions hierarchically. They are not the same question. If you think they are, then you are likely not using an appropriate breadth and depth of software-engineering thought. This course is a practical application of the aspects of product, process, and people. We are trying to account for everything: not just to create a good product, but also to learn from the process to improve the people. Reflect on the experience of the entire team collectively over this sprint. You do not need to account for all work, just two examples that are most representative of easiest and hardest.

For reference, *understand* relates to the comprehension of what needs to be done; *approach* to how you think it should be solved; *solve* to implementing the actual solution; and *evaluate* to demonstrating to yourself and your team (if applicable) that the performance of your solution is consistent with everything else in the project. Remember The Cartoon from CS 350.

Understand	
Easiest:	For the rest of our project time we only have the small details to work out within our application. Most of the hard work has been done. The easiest aspect to understand is that we need to clean up our code and make it modular. In many places, there was duplicate code that was unnecessary and it is easy to understand that we need to condense our work
Hardest:	The hardest part to understand about what we are currently working on is the logo tracker and unexpected errors within our code. The logo tracker is used to help smooth out tracking but our old approach using the OpenCV tracker was not working. Also the logo tracker does slow down the process leading to longer process times. In the backend, the errors the code produces can be very hard to read and interpret leading to long debugging time.

Approach	
Easiest:	The frontend website is robust and easy to add components too. We want to update the website in order to make it more presentable and user friendly. We feel comfortable enough to get this done easily.
Hardest:	Our executable using Electron.js runs very inefficiently. This may be due to the large libraries and APIs that we use within our code to create the model, do embeddings, analyze videos, etc. This is going to be hard to approach because we may not have a solution for this

Solve	
Easiest:	It is easy to solve the small details and frontend components we want to create such as the loading bar and video preview whenever a user uploads a video to our website.
Hardest:	The hardest problem to solve right now is the false positive problem that we have. The model detects objects that are not logos and although our application is a prototype it would be nice to fine-tune or get rid of these false positives but that will not be feasible

Evaluate	
Easiest:	It is easy to evaluate if our logo tracker or frontend components are working because we can see it working when we boot up our project on localhost.
Hardest:	It is hard to evaluate how well our specific logo search works. We implemented a voting system between embeddings to make sure the application finds logos that

match but there could be other approaches to this that work better. It seems to be consistent through our testing.

Completion:	85
Contact:	N/A
Comments:	

Report generated on Sun May 04 16:35:35 PDT 2025