Outlook

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

**From** shelbyemailrelay@gmail.com <shelbyemailrelay@gmail.com>
**Date** Sun 4/20/2025 5:23 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 11 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 | 8.0 | 75.0 | 3.0 | 9.0 | 7.5 | 7.5 | 1.6 | 10 | 0 (0%) |
| Stewart | 8.0 | 86.0 | 2.0 | 16.0 | 8.6 | 8.6 | 3.6 | 10 | 0 (0%) |
| Taggart | 7.0 | 70.0 | 3.0 | 10.0 | 7.0 | 7.0 | 2.3 | 10 | 0 (0%) |
| Team Total: | 23.0 | | | | | | | | |

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

Team Hours:

| Sprint | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 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 | 231.0 | 0.0 | 29.0 | 21.0 | 8.7 |

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

Taggart:

- Working on MOSSE tracker
- Debugging general video search
- Implementing general video search
- Implementing 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 108:** Fine tune our website

Update designs, make sure everything works (two sprints expected)

Stewart

**Activity 107:** Make FAISS implementation more efficient

Make the FAISS implementation from backend to frontend more computationally efficient (two sprints expected)

Taggart

**Activity 109:** Getting video to send to frontend

Encoding video to send to the front end (one sprint expected)

### Continuing Activities

These activities were continued from the previous sprint.

---

**Activity 93.1:** Style new stats that are returned

Opened in Sprint 7 by Keck; expected to take two sprints.

**Original description**: Make the stats look better

**Progress in Sprint 9** (expected to take one more sprint): The stats are on the webpage but could still look better

**Progress in Sprint 10** (expected to take one more sprint): Still working on it

**Progress in Current Sprint** (expected to take one more sprint): Stats are being returned but not styled yet

---

**Activity 102.1:** Implement FAISS General Search

Opened in Sprint 10 by Stewart; expected to take one sprint.

**Original description**: After FAISS works in interactive notebook, implement it into actual backedn

**Progress in Current Sprint** (expected to take two more sprints): General FAISS partially implemented. Trying to make more efficient.

---

**Activity 104.1:** Implement FAISS specific search

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

**Original description**: After FAISS works in interactive notebook, implement it into actual backedn

**Progress in Current Sprint** (expected to take one more sprint): Still need to implement. We will get the general search working first, and then implement the specific search.

---

**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 Sprint 10** (expected to take two more sprints): The model is now working on videos and detects logos within frames, but still needs to be actually implemented into the backend code.

**Progress in Current Sprint** (expected to take three more sprints): General logo search is implemented but needs more work to send video from back end to the front end in a reasonable time amount.

---

**Activity 99.1:** FAISS

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

**Original description**: Learning and using FAISS to detect when logos in different frames are the same.

**Progress in Sprint 10** (expected to take one more sprint): FAISS now detects first instance of logo then stores it for similarity lookup later. This works in our testing but still needs to be actually implemented into backend code.

**Progress in Current Sprint** (expected to take one more sprint): FAISS is implemented into back end, but needs to be tested to make sure it consistently works as expected.

---

**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** (expected to take two more sprints): MOSSE tracker now works in testing code and tracks the object based off of the initial boundary box, but it still needs to be implemented into our backend.

## Completed Activities

These activities were completed during this sprint.

---

**Activity 100.1:** Work on video analysis

Opened in Sprint 9 by Keck; expected to take two sprints.

**Original description**: Get our model working with video

**Progress in Current Sprint**: Our model does work with video now

---

**Activity 106.1:** Connecting new front end to backend

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

**Original description**: Using the new video interface, make it functional

**Progress in Current Sprint**: The front end sends video to the backend

---

**Activity 97.1:** Make video analysis more efficient

Opened in Sprint 9 by Stewart; expected to take two sprints.

**Original description**: Make processing videos less time consuming

**Progress in Current Sprint**: FAISS indeed did reduce the runtime of processing a video

---

**Activity 101.1:** FAISS General Search Working

Opened in Sprint 10 by Stewart; expected to take one sprint.

**Original description**: Prove that FAISS can work. Write this code in an interactive notebook for testing purposes

**Progress in Current Sprint**: FAISS works for our use case.

---

**Activity 103.1:** FAISS Specific Search Working

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

**Original description**: Get a working prototype of a specific logo search while using FAISS. for video search

**Progress in Current Sprint**: Specific search works well with FAISS

## 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:** | The easiest thing to understand is the importance of FAISS in our pipeline. We need a way to classify multiple frames of images/logos together, and FAISS gives us a way to do that. We also know that we need a way to process videos in the backend efficiently. |
| **Hardest:** | The most difficult conceptually has been figuring out why our video analysis takes so long to process, despite individual image detection being very fast. There are many possible bottlenecks, such as frame handling, embedding similarity checks, or the model itself, and identifying the exact cause has proven challenging. |

| Approach | |
|---|---|
| **Easiest:** | Surprisingly, implementing the FAISS vector database has been relatively straightforward. We started trying to implement our pseudocode idea within interactive notebooks to test the approach. After getting a working prototype, we knew we could have a successful solution in our production code. |
| **Hardest:** | The hardest to approach has been optimizing performance in video processing. There's no clear single fix, and it involves investigating everything from model performance to how embeddings are stored and compared across frames. |

| Solve | |
|---|---|
| **Easiest:** | It has been easiest to solve the FAISS general search in a notebook environment. Once FAISS was installed and configured, building a working demo for general similarity search using logo embeddings was achievable in one sprint. |
| **Hardest:** | The hardest aspect to solve has been building out the specific logo search using FAISS. Determining the right similarity threshold and integrating consistent embeddings across frames in a live backend environment is more complex than the general case. |

| Evaluate | |
|---|---|
| **Easiest:** | It's easy to evaluate whether our backend is functioning properly using HTTP response codes and console output. If an error occurs or a feature isn't working, we typically get clear, immediate feedback. |
| **Hardest:** | Evaluating the effectiveness of our video processing is difficult. While we can see the outputs, it's hard to tell whether all logos are correctly detected or missed, and it's not clear why detection slows down at certain points in a video, or why the video is so slow to send back to the frontend. |

**Completion:** 70%-75%. At this pace, we believe we are likely to succeed by the deadline, though backend efficiency will require ongoing effort.

**Contact:** N/A

**Comments:**

---

Report generated on Sun Apr 20 17:23:03 PDT 2025