Outlook

[CS 488T] Sprint 9 Report, Team 11 [stewartc]

From shelbyemailrelay@gmail.com <shelbyemailrelay@gmail.com>
Date Sun 4/6/2025 9:41 AM
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 9 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	59.0	3.0	9.0	7.4	7.4	1.8	8	0 (0%)
Stewart	12.0	69.0	2.0	16.0	8.6	8.6	4.0	8	0 (0%)
Taggart	10.0	55.0	3.0	10.0	6.9	6.9	2.5	8	0 (0%)
Team Total:	29.0								

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

Team Hours:

Sprint													
1	2	3	4	5	6	7	8	9	Total	Min	Max	Avg	Std
8.0	27.5	24.5	28.5	25.5	22.0	18.0	0.0	29.0	183.0	0.0	29.0	20.3	9.5

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

Taggart:

- Getting downloadable properly releasing on Github through Github Actions
- Getting downloadable properly releasing on Github through Github Actions
- Learning about FAISS
- Fixing minor problems on front / back end

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 100: Work on video analysis

Get our model working with video (two sprints expected)

Stewart

Activity 95: Intermediate presentation

Collecting documents and construct presentation for intermediate presentation (one sprint expected)

Activity 96: Start video analysis

Get the model to work on a video (two sprints expected)

Activity 97: Make video analysis more efficient

Make processing videos less time consuming (two sprints expected)

Taggart

Activity 98: Video Processing

Get YOLO model to work on videos (three sprints expected)

Activity 99: FAISS

Learning and using FAISS to detect when logos in different frames are the same. (two sprints 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 Current Sprint (expected to take one more sprint): The stats are on the webpage but could still look better

Completed Activities

These activities were completed during this sprint.

Activity 91.1: Border color

Opened in Sprint 7 by Stewart; expected to take one sprint.

Original description: Allow the user to select the border color of the bounding box

Progress in Current Sprint: Finished. User can select from a color wheel

---

Activity 92.1: Crop logos

Opened in Sprint 7 by Stewart; expected to take one sprint.

Original description: Crop the logos and display results/metrics about each logo

Progress in Current Sprint: Finished. Shows little icons of the logos detected in the result box. Also displays the area of the logo in relation to the entire image.

Activity 84.1: Changing web app to be a desktop app

Opened in Sprint 5 by Taggart; expected to take one sprint.

Original description: Using electron.js to wrap React app

Progress in Current Sprint: The electron app has now been configured for packaging.

Activity 94.1: Packaging Front and Back end

Opened in Sprint 7 by Taggart; expected to take one sprint.

Original description: Working on packaging each as a standalone then combining together.

Progress in Current Sprint: Application is now packaged and has an automated Github Actions workflow to automatically package, combine front end and back end, and then release.

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 aspect that is currently the easiest to understand is what we broadly need to do to be able to process videos instead of images. We understand that a video is just a bunch of frames/images pieced together, so we know we will just have to process lots of images to accomplish this.
Hardest:	The part of the current work that is hardest to understand is the FAISS algorithm which will help us determine whether or not the logo found in one frame is the same as the same logo in the next frame. It is hard to understand how to use it as we have found a very limited amount of documentation on it.

Approach

Easiest:	The easiest to approach part of the current work is updating our frontend design to work with videos now that we have finished image processing. The design of the video

processing is very similar to what we have done for images, so many of the components that we have built are reusable.

Hardest: Currently the hardest aspect to approach is designing the method we will be using to analyze and detect the logos within a video. We are trying to design a system that finds the correct balance of accuracy versus processing speed.

Solve

Easiest: The easiest to solve part has been packaging our backend into an executable since we were able to use the pyinstaller library which does the majority of the work for us.

Hardest: The aspect that is currently the hardest to solve for us has been getting GitHub Actions to package our application in a timely manner. We are dealing with an error when packaging the application for Mac, but we are having a difficult time solving/debugging the issue since much of the work/processes that Github Actions is doing for us is behind the scenes and out of our control.

Evaluate

Easiest: The easiest aspect of the current work to evaluate is whether or not our program is correctly identifying when there is a logo within a frame of a video as we can just watch the video and see if the boundary boxes were correctly drawn.

Hardest: The hardest to evaluate aspect is if our model performs the same time wise when the video gets longer and has multiple logos within a frame. Currently we are just applying it to very simple and short videos, so it is hard to tell what the effects will be once we add these other layers of complexity in.

Completion: 60%. We feel that we are on pace to successfully complete the project.

Contact: N/A

Comments: We do not currently have any issues, comments, or concerns regarding the project.

Report generated on Sun Apr 06 09:41:26 PDT 2025