Allsky Camera Network for Detecting Bolides
Milestone 3

Members
Tyler Turner, tturner2021@my.fit.edu
Vincent Quintero, vquintero2021@my.fit.edu
Jean-Pierre Derbes, jderbes2021@my.fit.edu
Charles Derbes, cderbes2021@my.fit.edu
Faculty Advisor/Client
Csaba Palotai, APSS, cpalotai@fit.edu

Progress Matrix for Milestone 3:
If completion is not 100%, relative workload was preserved. For example, if the task was 50% complete by the end of the milestone each person's contribution to that 50% is 50% of what their initial contribution would have been.

Task	Completion	To Do	Tyler	Vincent	Jean-Pierre	Charles
Replace current C++ camera code	100%	0%	10%	35%	45%	10%
Implement Server API	99%	1%	50%	0%	50%	0%
Implement Client API	99%	1%	20%	0%	20%	60%
Begin writing CLI	30%	70%	30%	10%	50%	10%
IoT style setup	99%	1%	20%	0%	10%	70%
Classification	99%	1%	0%	33%	33%	34%

Start writing UI	0%	100%	20%	70%	0%	10%
Create setup process for node	75%	25%	75%	0%	25%	0%

Discussion of each accomplished task for the current Milestone:

- Task 1: Goal was to replace the C++ camera code (primarily the sentinel.cpp file) that contained functionality for detecting events, recording events, and storing events. Successfully replaced the C++ code with python code using ffmpeg. The code records videos with a duration of 10 minutes and an overlap which enables events occurring in the overlap to be fully contained in the current or next video being sent to the server.
- Task 2: Goal was to add some polish to the server API by fixing some bugs and making it reliable for use by the frontend. Server API is essentially complete and simply needs more testing. Videos successfully are sent from the client to the server.
- Task 3: Goal was to implement the video sending and rework the node state reporting. Video sending is now working (see task 1). Correct state being reported back to the server is still blocked by non camera/main detection related C++ code needing to be rewritten.
- Task 4: Goal was to write CLI for the researchers to be able to have access to our internal tools. Classification CLI has been successfully implemented and allows a user to manually run classification on a bolide event video or a set of videos. Useful when used on a set of videos and for tuning the model.
- Task 5: Goal was to work through edge cases, try to solve permissions issues for connecting to wifi programmatically, and to add a captive portal. Edge cases have been worked through and wifi can be connected to programmatically (although the permissions workaround is sketchy and needs to be looked at). Captive portal still needs to be done.
- Task 6: Goal was to build a classifier using CNN to act as a powerful filter that eliminates a large majority of junk events. Goal was met and the classifier's accuracy was satisfactory (95%) on the test data set. We have yet to test it in the wild and will be looking to make modifications based on how it performs.
- Task 7: Goal was to begin building the UI if everything else was finished. Everything else was not finished and is not stable therefore we cannot begin implementing the UI, although we have come up with ideas on how it should function and look.
- Task 8: Goal was to create a process for setting up nodes which included installing OS, packages, and testing the node. An Ansible playbook that installs all of the necessary requirements on a new node has been created. We will look into testing the hardware of the node before sending it out to ensure they are all sent out in prime condition.

Discussion of contribution of each team member to the current Milestone:

- Tyler Turner:
  - Looked into captive portal for IoT
  - Worked heavily on both APIs and implemented video sending
  - Node setup (Ansible playbook)
- Vincent Quintero:
  - Implemented video composites
  - Implemented data augmentation for training model
- Jean-Pierre Derbes:
  - Trained and tuned classification model
  - Implemented classification pipeline
- Charles Derbes:
  - Designed classification pipeline
  - Implemented object proposer and molder

Task Matrix for Milestone 4:

Task	Tyler	Vincent	Jean-Pierre	Charles
Implement UI	10%	50%	0%	40%
Polish Server	50%	20%	30%	0%
Polish Client	30%	20%	20%	30%
UI Tests	0%	0%	50%	50%
Server Tests	50%	0%	50%	0%
Client Tests	0%	0%	50%	50%
Create setup process for Node	75%	0%	25%	0%

Discussion (at least a few sentences, ie a paragraph) of each planned task for the next Milestone:

- Task 1: Goal is to have a working frontend that we can present to the researchers for testing. We will use their feedback to enhance UX. The centralized UI should make accessing information much easier for them.
- Task 2: The server will most likely have bugs and/or performance issues and we will most likely want to change a few things around. We will need to stress test the server with all of the nodes sending videos at once to the server. The server also needs a set of pipelines that are triggered after receiving the videos.
- Task 3: Non camera related C++ code needs some looking at. Project structure might need to be revised and more testing around state needs to be done. Video sending will need some more testing as it is the core purpose of the node.
- Task 4: In order to make sure our UI is working as expected for any user, we will write end-to-end UI tests using Playwright. These tests will ensure that the functionality of the UI is working properly whenever we update or add new features to it. These tests will be run automatically in CI/CD.
- Task 5: To make sure that the server is working as intended, we will write unit tests for each of its moving parts. The database, endpoints, and video processing algorithms will need unit tests to ensure the correct functionality. The classification pipeline will also have its own set of tests to make sure nothing inside it breaks.
- Task 6: To make sure that the client is working as intended, we will write unit tests for the client's API: checking sensors (hardware), getting and updating node state, and getting node status.
- Task 7: Continuation of last milestone, node setup just needs a hardware testing suite. This would test all of the hardware making sure nothing is broken. The tests would be checking if the Pi is properly connected to all hardware components, and then testing the transmission and receiving of data between the Pi and the hardware component. For example it would test to make sure that the camera is connected and is capturing video correctly.

Dates of meetings with Client during the current milestone:

see Faculty Advisor Meeting Dates below

Client feedback on the current milestone:

see Faculty Advisor Feedback below

Dates of meetings with Faculty Advisor during the current milestone:

- Oct 30, 2024
- Nov 6, 2024
- Nov 13, 2024

Faculty Advisor feedback on each task for the current Milestone:

- Task 1: Dr. Palotai appreciates that the nodes are easier to maintain since camera code is written in python.

- Task 2: Dr. Palotai is pleased with the progress being made on the server api.
- Task 3: Dr. Palotai is pleased with the progress being made on the client api.
- Task 4: Dr. Palotai appreciates that we have added classification to the pipeline but thinks the rest should wait until the event gathering is finished.
- Task 5: Dr. Palotai is eager to get the IoT finished since it would prevent users from messing up the node.
- Task 6: Dr. Palotai is pleased with classification but emphasizes having the main functionality working before adding fancier features.
- Task 7: Dr. Palotai agrees the UI should come after the backend is implemented, and is eager to finally be able to see and use the new system.
- Task 8: Dr. Palotai is very pleased with the node setup process since it is important to standardize the software aspect of the box and ensure the boxes are all of the same quality.

Faculty Advisor Signature: _____ Date: _____