Automated Team Attendance

Design Document

19
Mohamed Youseff Selim
Mohamed Youseff Selim
Lance Demers – Chief Engineer Hardware
Brandon Johnson – Team Leader
Nathan Oran – Documentation Manager
Angela Schauer – Meeting Facilitator and Chief Engineer Software
Connor Sullivan – Testing Lead
sddec20-19@iastate.edu
https://sddec20-19.sd.ece.iastate.edu

Revised: 2/23/20 V1

Executive Summary

Development Standards & Practices Used

This project will be developed using test driven development and agile workflow. We will use paired programming and code reviews to make sure our software is high quality. We hope to develop a modular software system to simply future development and hopefully save ourselves time if we need to reimplement specific parts of the code.

Summary of Requirements

- Hardware
    - Project must run on a Raspberry Pi
    - Camera must be able to take clear pictures
    - Must be mounted in a way for the camera to see all of the classroom
- Software
    - Control the camera to take images of the classroom
    - Accurate detection of students from taken images
    - Determine which groups have absent members
    - Consolidate data and send a report to the professor
    - Professor must be able to create and adjust multiple seating charts for the system to work with

Applicable Courses from Iowa State University Curriculum

- CPR/S E 185/E E 285/COM S 227 - Introduction to programming
- COM S 228 - Data Structures
- COM S 309 - Application Development
- CPR E 288 - Embedded Systems

New Skills/Knowledge acquired that was not taught in courses

For several members of our team, the python programming language will be a new skill. This project also requires Machine Learning using Yolo and openCV which are new skills for all of us.

Table of Contents

List of figures/tables/symbols/definitions (This should be the similar to the project plan)

1 Introduction

1.1 Acknowledgement

We acknowledge Mohamed Youseff Selim for coming up with the idea of the project and also guiding our team through it.

1.2 Problem and Project Statement

No doubt, Team-Based Learning (TBL) positively influenced attendance in the classroom. However, teams are still suffering from absenteeism. Although a handful of attendance tools exist, none of these tools is adapted to the team-based classes, besides, none of them records the attendance without any interaction from the instructor/student. Moreover, these tools are consuming from 2 to 3 minutes from the class time, which will sum up over the semester to 135 minutes in the worst case. The proposed tool will record attendance at zero time.

1.3 Operational Environment

The Automated Team Attendance Tool will operate in a classroom environment. The final product will be mounted in classrooms used by Instructors. The product should be able to function in small and large lecture halls.

1.4 Requirements

Data Collection:

Image collection of the classroom, over multiple instances during class

Detect Students:

Optimize the accuracy of student detection using deep learning analysis

Determine Which Groups Have Absent Members:

Compare the locations of students to a provided team-based seating chart

Send Report to Professor:

Consolidate the collected data into an email sent to the professor of the class

Adjustable Seating Arrangements:

Professor has access to create and update team seating charts

1.5 Intended Users and Uses

The product is intended to be used by any instructors or teaching assistants wishing to record attendance automatically.

1.6 Assumptions and Limitations

Assumptions

- Instructor or TA will provide a seating chart
- Teams will be in the same spot for the whole semester
- Instructor or TA will decide when to capture the attendance
- Product will be used in a classroom environment

Limitations

- Algorithm struggles to detect objects that are densely packed
- Test to see how program performs on models of the raspberry pi that have lower performance specs

1.7 Expected End Product and Deliverables

May 2020: Test product having minimum functionality. Such as determining the number of students present for each team. Being able to take a seating chart as input to be used to determine any missing students.

December 2020: A final product that will capture the number of students present for each team based on a seating chart provided by the instructor. Once the program finishes it will email the instructor the number of students present for each team.

2. Specifications and Analysis

2.1 Proposed Approach

The proposed approach to solve the problem of lengthy attendance times in TBL classrooms is using object detection to take attendance. A camera will take a picture of the classroom and verify it against a seating chart provided by the professor. An email with information about the attendance of each group will then be sent to the professor. As of now, we have been researching and testing various free, open source object detection software on pictures of lecture halls to see if they are a viable option. For this project, no specific standards need to be followed.

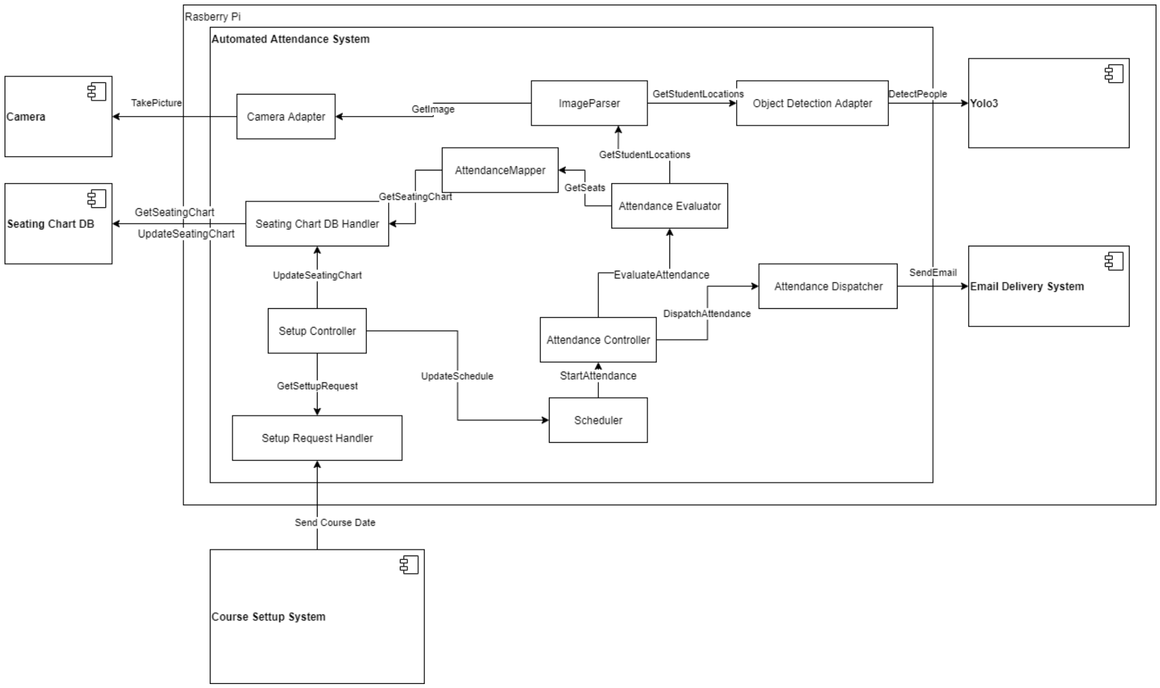
2.2 Design Analysis

After researching the available object detection software, we found that this proposed approach is feasible, specifically using the software YOLOv3. YOLO builds on top of OpenCV to create a system that can quickly and accurately detect and identify objects in a picture. To take the picture, we will have a camera connected to a RaspberryPi which will be connected to a server where it will run our code. We will be using Python for all the code relating to the object detection. For our project, we will take the data of the x,y coordinates of students that are detected and parse them so they are ready to be compared against the seating chart provided. To get the data of the seating chart, we will need a UI for the professor to use and analyze the data they give us.

2.3 Development Process

We will be using a mix of Test Driven Development and Agile processes. We want to be able to create code incrementally and includes a lot of user input, so we decided Agile was best. For each iteration, we want to use TDD to make sure all requirements we want are being met and our code does as we want.

2.4 Conceptual Sketch



3. Statement of Work

3.1 Previous Work And Literature

There are similar tools available to professors for taking attendance in large lecture halls. One product that is mainstream and widely used is top hat. Top Hat is designed for in-class quizzes and tests for large lecture halls. Additionally, many professors use Top Hat for attendance because Top Hat shows the students who participated in the in-class quiz and who did not. The disadvantage of Top Hat is that it takes time to record the attendance in class and wastes valuable time. Our tool does not require any class time to take attendance and the time saved used to teach students. Another similar product available is called Spotter. Spotter is an app that students install on their phone and it compares a students location to where the lecture hall is to take attendance. Spotter does have advantages when compared as it requires no camera or hardware to work, the students just need to download the app. It also will be more accurate as it tracks the individual student and our tools just identifies group size groups. One disadvantage of spotter is privacy. For the spotter app to work, it needs to know your location. This requires you to put a lot of trust into the app that it will not track you at all times. Our tool does not require any individual information and will not even know your name. The biggest privacy concern of our tool is that the raspberry pi will have pictures of the classroom stored on it temporarily.

3.2 Technology Considerations

Our project uses a wide variety of different technologies, The technology is split between the hardware and software. The hardware we use needs to take clear pictures and needs enough computing power to run our Automated team attendance software. The tradeoff with the hardware is the cost versus the computing power. Our goal is to minimize cost while being able to run the software effectively. The choice for our software technology is less straightforward. Our primary concern with our attendance tool is accuracy. We do not want to mark students as absent if they are present. Additional considerations are what programming language, libraries, and what machine learning framework / algorithm to use. For this project we chose python of other languages because it allows for a faster development process. The libraries that we are using are OpenCv and YoloV3. We chose these libraries because of the amount of documentation available and the reputation they have in the computer vision community. YoloV3 is also known to run much faster than other object detection algorithms and is important because a raspberry pi has limited resources.

3.3 Task Decomposition

In order to solve the problem at hand, it helps to decompose it into multiple tasks and to understand interdependence among tasks.

Our group split the responsibilities of the project into two main groups, software and hardware. The software team will focus on the algorithm of the automated attendance tool itself. The hardware team does not only set up the hardware, they are responsible for setting up the platform itself. This includes the backend which sets up the raspberry pi as a server that schedules the tasks to run at the appropriate time. One of the largest tasks for the hardware team is setting up the email server. The email server will email the instructors the attendance results for the day.

Both the hardware and software team will decompose their responsibilities into smaller tasks. The software team will need to complete the following tasks to insure that the object detection algorithm works. First they need to make the object detection as accurate as possible. This includes combining opencv, yolov3 and so given an input image, it can detect all persons and store their coordinates. Another task they will work on is determining what people belong to what group. Finally the last big task is comparing the group attendance we have to the seating chart given by the instructor. In addition to setting up the email server, the hardware team also needs to set up the hardware and scheduling. The hardware requires creating a program that sets up the camera on the raspberry pi and taking and saving pictures. The scheduling tasks include, running the attendance software multiple times and then sending the results to the instructor. The pictures taken will also need to be deleted for privacy reasons.

3.4 Possible Risks And Risk Management

We will need to make sure the program detects important data as accurately as possible, and is able to compare that data to a given seating chart. We will manage this by taking multiple photos if needed to make sure that the data we receive can produce the desired output.

We will need to make sure to consider the privacy of the students, and anyone who ends up being in the picture as well. We will manage this by deleting the image after the image has been successfully used to determine attendance, and after a predetermined time length. Any pictures that are either invalid or are determined to be bad will just be discarded immediately.

3.5 Project Proposed Milestones and Evaluation Criteria

Our first milestone will be being able to further develop the algorithm to be able to detect larger numbers of students grouped closely together. We can test this by taking some stock photos of large groups of people that we can determine the number of people in the photo, and running the algorithm on the photo to make sure it can count that many people.

Our second milestone will be being able to take a photo from the raspberry pi and comparing it to a given seating chart. We will test this by providing simple images to use that contain a small group, or 2 small groups, and then use a seating chart that we know what the result of the attendance should be.

Our third milestone will be being able to run the scheduled tasks. This includes being able to run the attendance software, possibly multiple times, emailing the professor the reported attendance, and deleting the picture after the program is finished with it. We can test this setting some output to see that our program can be automatically executed multiple times based on some given criteria. We can set up the raspberry pi to email one of us the results, or all of us in the case that a professor would like multiple people to be notified of the attendance. We can check to make sure that once the program is finished it no longer contains the image in memory.

3.6 Project Tracking Procedures

We will be using a Trell board to keep track of our progress on the project. We will be using the typical backlog, doing, and done trello tabs to keep track of our work.

3.7 Expected Results and Validation

What is the desired outcome?

How will you confirm that your solutions work at a **High level**?

We will determine the success of our solution by the solution's ability to consistently deliver the attendance information of each team in a given team-based learning class, by the number of persons each team has present (rather than individual attendance). In particular, this solution as a whole will be considered a success if it is less time consuming than manually taking attendance.

We will determine the success of our data collection requirement by our solution's ability to regularly collect images of the classroom at the correct scheduled times, proper locations, and done the proper number of times.

The success of our student detection requirement will be validated by determining whether the solution can use the collected images to correctly identify where all the students are in the given classroom, map each student to their correct team, supplied by an attendance sheet, and determine which teams are then missing students.

The success of our requirement to send the report to the professor will be measured by whether the system can reliably send the team attendance to the professor at the correct times.

Finally, our requirement for adjustable seating arrangements will be validated by our solution's ability to receive updated attendance lists and sheets from the professor, and apply these changes to the seating chart and/or schedule prior to the next class meeting time.

4. Project Timeline, Estimated Resources, and Challenges

4.1 Project Timeline

• A realistic, well-planned schedule is an essential component of every well-planned project

• Most scheduling errors occur as the result of either not properly identifying all of the necessary activities (tasks and/or subtasks) or not properly estimating the amount of effort required to correctly complete the activity

• A detailed schedule is needed as a part of the plan:

– Start with a Gantt chart showing the tasks (that you developed in 3.3) and associated subtasks versus the proposed project calendar. The Gantt chart shall be referenced and summarized in the text.

– Annotate the Gantt chart with when each project deliverable will be delivered

• Completely compatible with an Agile development cycle if that's your thing

How would you plan for the project to be completed in two semesters? Represent with appropriate charts and tables or other means.

Make sure to include at least a couple paragraphs discussing the timeline and why it is being proposed. Include details that distinguish between design details for present project version and later stages of project.

Tasks	S1 12	S1 13	S1 14	S1 15	S1 16	S1 FIN	Brk	S2 1	S2 2	S2 3	S2 4	S2 5	S2 6	S2 7	S2 8	S2 9	S2 10	S2 11	S2 12	S2 13	S2 14	S2 15	S2 16
	3/30/20	4/6/20	4/13/20	4/20/20	4/27/20	5/4/20	5/11/20	8/24/20	8/31/20	9/7/20	9/14/20	9/21/20	9/28/20	10/5/20	10/12/20	10/19/20	10/26/20	11/2/20	11/9/20	11/16/20	11/23/20	11/30/20	12/7/20
Software Team																							
Person Detection																							
Seating Chart & Schedule Submission																							
Attendance Mapping																							
Hardware Team																							
Pi Environment Configuration																							
Mount																							
Scheduler																							
Email Server																							
Full Team																							
Field Testing & Refining																							

4.2 Feasibility Assessment

We think the full completion of this project is possible within the two semesters given. However, this project does require a wide breadth of development that will require consistent progress throughout the time we have. Specifically, our greatest technical challenges will be learning to use and train the Yolov3 Neural Network with OpenCV to accurately find students every time and then mapping the locations of students to the seating chart to determine attendance. Developing accurate seat mapping algorithms and training the Network will take up a lot of our time.

Additionally to our major technical challenges, we need to develop the system to email results to professors, create an entire web app for professors to submit seating charts and schedule when the system takes attendance, and configure the raspberry pi to interact automatically with these elements. Finally, we need to create the mounting system that can work with any of the classrooms that use team based learning. At this point, we don't see any of these tasks as too complex for us to handle, but there are basically five or six large tasks that will take the efforts of all of the members of the team to complete.

4.3 Personnel Effort Requirements

The following table breaks down the expected amount of time for each member of the team for each task in order to complete the project. The estimations were made by assuming that each person works between 6 to 8 hours per week on their active task. Then, the estimation was calculated by multiplying each weekly average by the number of weeks for that task. Connor, Angela, and Nathan are the software team, and Brandon and Lance are the hardware team. So the total time spent on each task is the time each person spends on the task multiplied by the team size.

	Hardware Team	Software Team	Full Team	Total Time
Student Detection		28 hrs/person		84 hrs
Attendance Mapping		63 hrs/person		189 hrs
Seating Chart Submission		14 hrs/person		42 hrs
Email Server	28 hrs/person			56 hrs
Raspberry Pi Configuration	28 hrs/person			56 hrs
Pi Scheduler	35 hrs/person			70 hrs
Mount Design	14 hrs/person			28 hrs
Field Testing & Refining			21 hrs/person	105 hrs

4.4 Other Resource Requirements

To properly create our project, we will be needing test data readily available. This test data will be pictures of people sitting in a setting close to one of the Team Based Learning classrooms at Iowa State. With current conditions, we may have to utilize photoshop instead of using actual pictures. Once testing has been completed with the test data, we will move to testing in an actual classroom environment. We will need access to a Team Based Learning classroom and multiple students to act as a class if an actual class is not available.

4.5 Financial Requirements

For this project, we will need to pay for: RaspberryPis, cameras, and eventually a dedicated server to host our program on. As we do not know what the final setup will be, or how many classrooms will be outfitted with this product. Once we have a better idea of the exact hardware we will be using and how many we will need, and an itemized price list will be created.

5. Testing and Implementation

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, or a software library

Although the tooling is usually significantly different, the testing process is typically quite similar regardless of CprE, EE, or SE themed project:

1. Define the needed types of tests (unit testing for modules, integrity testing for interfaces, user-study for functional and non-functional requirements)
2. Define the individual items to be tested
3. Define, design, and develop the actual test cases
4. Determine the anticipated test results for each test case 5. Perform the actual tests
6. Evaluate the actual test results
7. Make the necessary changes to the product being tested 8. Perform any necessary retesting
9. Document the entire testing process and its results

Include Functional and Non-Functional Testing, Modeling and Simulations, challenges you've determined.

5.1 Interface Specifications

– Discuss any hardware/software interfacing that you are working on for testing your project

5.2 Hardware and software

– Indicate any hardware and/or software used in the testing phase

– Provide brief, simple introductions for each to explain the usefulness of each

5.3 Functional Testing

Examples include unit, integration, system, acceptance testing

5.4 Non-Functional Testing

Testing for performance, security, usability, compatibility

5.5 Process

– Explain how each method indicated in Section 2 was tested

– Flow diagram of the process if applicable (should be for most projects)

5.6 Results

– List and explain any and all results obtained so far during the testing phase

- – Include failures and successes
- – Explain what you learned and how you are planning to change it as you progress with your project
- – If you are including figures, please include captions and cite it in the text

• This part will likely need to be refined in your 492 semester where the majority of the implementation and testing work will take place

-**Modeling and Simulation**: This could be logic analyzation, waveform outputs, block testing. 3D model renders, modeling graphs.

-List the **implementation Issues and Challenges**.

6. Closing Material

6.1 Conclusion

Summarize the work you have done so far. Briefly re-iterate your goals. Then, re-iterate the best plan of action (or solution) to achieving your goals and indicate why this surpasses all other possible solutions tested.

6.2 References

This will likely be different than in project plan, since these will be technical references versus related work / market survey references. Do professional citation style(ex. IEEE).

https://pjreddie.com/media/files/papers/yolo_1.pdf

6.3 Appendices

Any additional information that would be helpful to the evaluation of your design document.

If you have any large graphs, tables, or similar that does not directly pertain to the problem but helps support it, include that here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc. PCB testing issues etc. Software bugs etc.