Automated Team Attendance Tool

Angela Schauer | Lance Demers | Nathan Oran
Brandon Johnson | Connor Sullivan

Faculty Adviser/Client:

Mohamed Selim

SDDEC 20-19

Problem Statement

Attendance is an important part of Team Based Learning, and instructors must take time during each class to go through roll call. This time adds up and by the end of the semester, can equal multiple lectures worth of time. Although a handful of attendance tools exist, none of these tools are adapted to team-based classes, and none of them record the attendance without any interaction from the instructor or student.

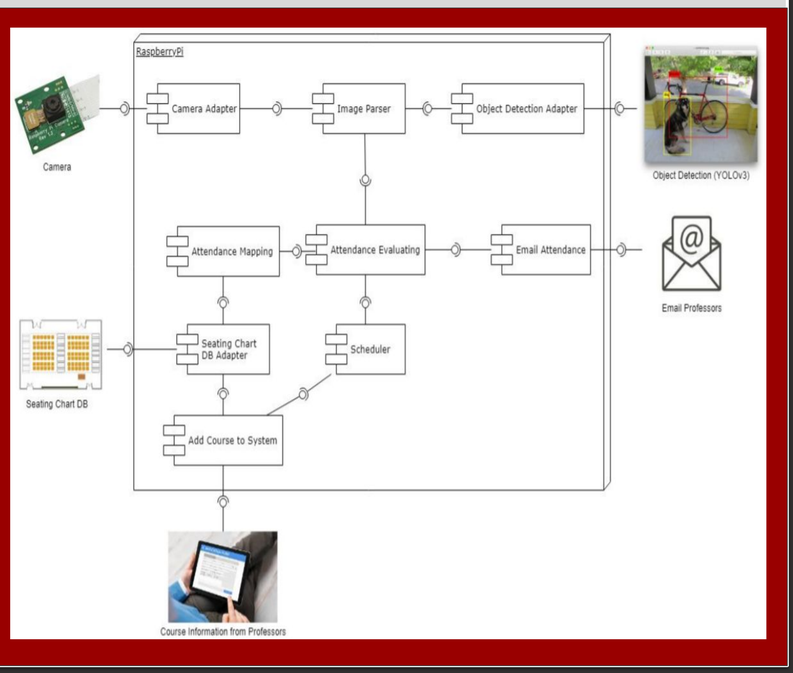
Solution

The proposed tool will record attendance with zero cost to the instructor and student. An image will be taken of the classroom which will then be processed with object detection software to verify if a student is in a seat. An email can then be sent to the instructor with attendance for that class period. This will all happen automatically after the instructor enters the class schedule and seating chart into the program.

Intended Users and Uses

Professors will use the application to automate attendance image capture. A secondary use case that has been proposed is to use this software for coronavirus contact tracing.

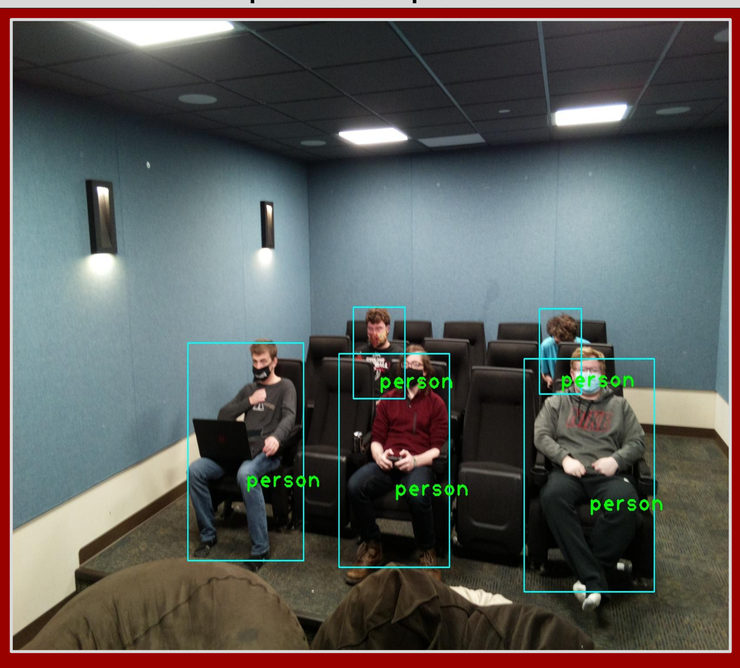
Concept Diagram



Testing

Due to the restrictions of COVID-19, and in-person basis of this project, it was difficult to test our system. During Spring semester, we used images of lecture halls from the internet to do some initial testing with the object detection software. We were able to meet once during Fall semester to set up the Pi and take an image of a classroom at Iowa State with our senior design team in the seats. An additional test was done with a few more subjects at an off campus site. The team working on the database used MySQLWorkbench and Postman to test the API.

Off Campus Site Sample Test Result



Technical Details

Hardware:
- Raspberry Pi 4
- Raspberry Pi Official camera
- Mount

Software:
- Python
- YOLOV3
- Linux
- C#
- HTML
- SQL

Design Approach

We knew there would be a lot of third party systems to interact with, so we wanted to utilize the facade pattern between any third party components. We designed our code this way so that it would be easier to interchange components, or fix things if necessary. We decided to go with the Raspberry Pi because it was cost effective, and provided us with all the functionality that we would need. Since we were using a Raspberry Pi we knew we needed software that was going to be small enough to fit on the limited storage. For that reason we went with YoloV3 since it could provide all of the object detection needs that we have, while also being lightweight enough to be put on the Raspberry Pi. One limitation we found with YoloV3 is that it had trouble detecting large numbers of objects packed tightly together. To solve this, we decided to split up the image into smaller sections, and then run the algorithm on the partitioned sections to improve the accuracy of detection.

Results and Findings

YOLO has difficulty identifying objects when they are closely grouped together or small. Students sitting at the back of a classroom are both, so a way to more accurately identify them had to be created. Our solution is to split the image into four quadrants, send each part through the object detection software, then combine the results from each part of the image.

During testing in the classroom, we found that the Raspberry Pi official camera did not have a wide enough view to capture all of the lecture seats. A wider lensed camera or addition to the Pi camera to create a wider lens are possible solutions.

The SQL database was easy enough to set up, but the API was more difficult to create than expected, as no one in our group had much experience in creating one from scratch.

Design Requirements

Functional Requirements:
- Image collection of the classroom, over multiple instances during class
- Optimize the accuracy of student detection using deep learning analysis
- Determine groups that have absent members by comparing the image with a provided seating chart
- Email the attendance to the instructor
- Provide instructors with the ability to create and update a team seating chart
- Storing the image for no longer than a day, and deleting the image for security purposes

Non-Functional Requirements:
- Should be faster than previous attendance methods.
- Should be as accurate as previous attendance methods.

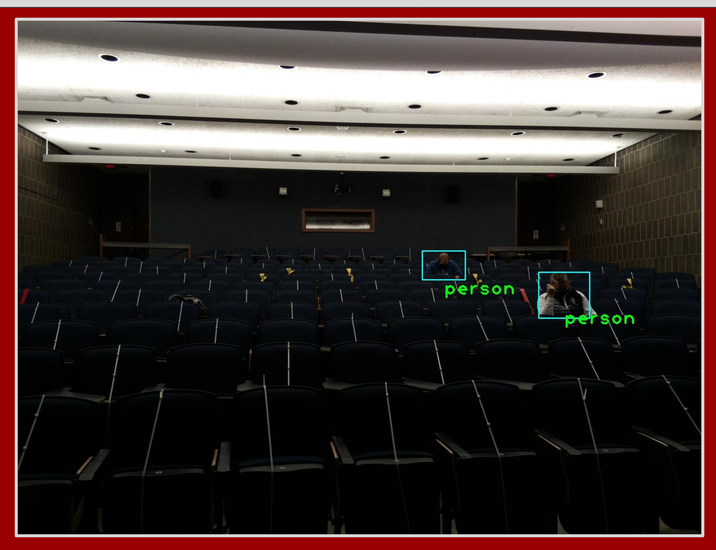
Operating Environment:
- Classroom
- Good line of sight
- Proper lighting
- Internet access

Standards:
- Hardware
  - Schematic
  - Bill of materials
- Software
  - Source code
  - In-line comments
  - External documentation

User Process

- Classes, Classrooms, Professors, Seating Arrangements, and Capture times are stored in a mySql Database and accessed through an ASP.NET Core API.
- The raspberry pi will retrieve all class data and capture times for its classroom at the start of each day.
- Once the Raspberry pi Camera captures the image, it will be processed via Yolo V3 to detect present students.
- Then, Yolo Data is compared to the seating chart information from the professor, and compile the attendance for the class.
- Finally, the raspberry pi drafts an email with the attendance report to send to the professor.

Carver Lecture Hall Sample Test Result



Future

There are still parts of this project that need to be implemented before it can be fully working. A large part is figuring out where the camera will be permanently mounted in each room, then using that image create a map of static locations for each seat in the classroom. Then, the algorithm to map an instructor given seating chart to this map would need to be created. Emailing attendance to an instructor, the UI for submitting a seating chart, and scheduling the Raspberry Pi to take images at the correct time also still need to be fleshed out.