A more robust client-server infrastructure

- In the previous semester, both of the client-side training and inference applications transmitted images with SCP
- Scp requires credentials of the server to be stored in the client-side program, which can become a security vulnerability
- Such client-server interaction also limited the number of active client to 1
- A robust full stack application has been developed to streamline both training and testing

AI-based Automation in the Classroom (AAC)

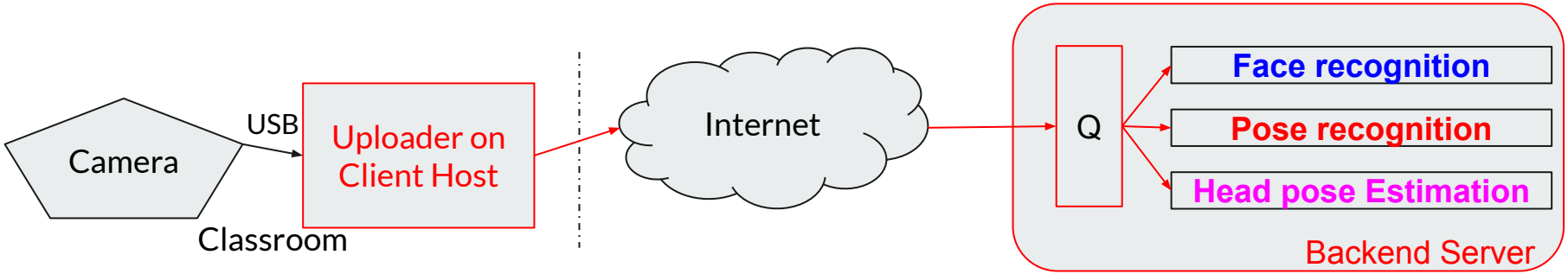
Final Presentation

Fall 2019 - Chieh-En Li, Chien-Hung Wang, Luis Materon, Chayaphon Pugkhem, Shreya Ilango, Elliot Edmunds, Wenbo Wei, Tanvi Bhardwaj, Shu Whai Teo, Po Yu Huang, Young Jin Jung

Goals, Context, Motivation

- Apply AI techniques in the classroom to improve:
  - Instructor experience
    - Expend less time on attendance tracking
    - Easier assessment of classroom interaction/participation
    - Gain feedback on Attention/Engagement
    - Offload work of responding to routine questions about course/syllabus, deadlines, resources
  - Student experience
    - Less classroom time spent on mundane tasks like attendance
    - More natural interaction (raise hands vs. press buttons on clicker)
    - Feedback on focus/engagement
    - Responsive, always available resource for getting questions answered interactively with natural language queries
- Two broad thrusts
  - **Thrust 1**: Automatically infer presence, interaction, engagement from images/video of classroom
  - **Thrust 2:** Virtual Teaching Assistant : Digital Assistant for classrooms; answers questions about course/syllabus

Thrust 1: Block Diagram + Team Responsibilities



Core Team: Client-server Software architecture; Parallel processing + Task queue design of backend server

Accuracy Optimization: Data science optimizations to enhance accuracy of recognition/classification

Engagement Estimation: New Backend functionality to measure student engagement in the classroom via head pose estimation

Outline

- Thrust 1:
  - Three Teams
    - Core Team
      - Wenbo Wei, Elliot Edmunds, Tanvi Bhardwaj
    - Accuracy Improvement Team
      - Shu Hwai Teoh, Po Yu Huang
    - Engagement Estimation Team
      - Young Jin Jung, Chieh-En “James” Li, Chien-Hung Wang, Luis Materon
- Thrust 2: VTA
  - One team
    - Chayaphon Pugkhem, Shreya Ilango

Core Team

Elliot Edmunds
Wenbo Wei
Tanvi Bhardwaj

Introduction

Team Purpose:

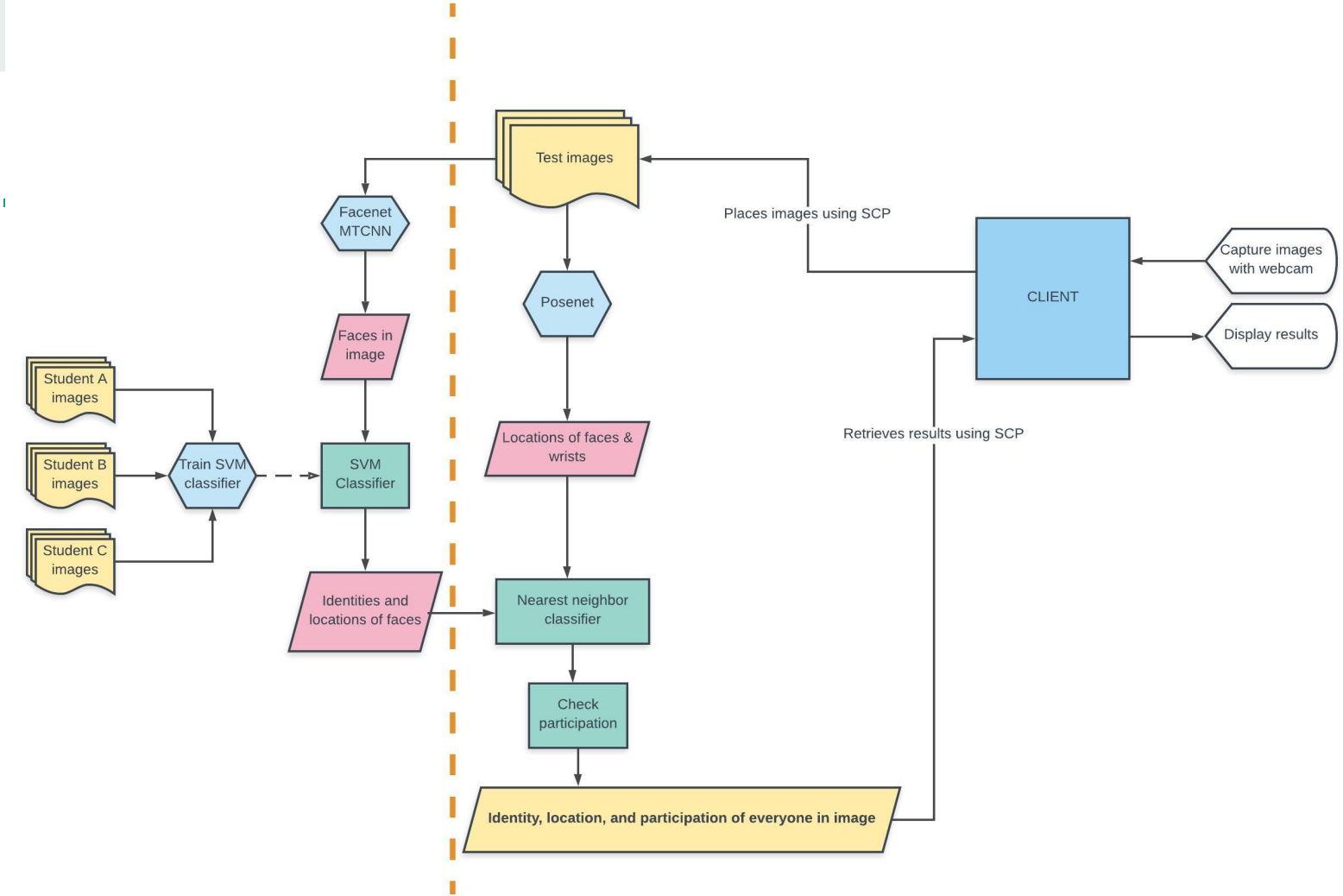
- Improve user experience
- Optimize efficiency
- Robust client-server infrastructure

Starting point

- Two semesters ago, the AAC team was able to successfully identify students in a classroom size of 25 people.
- The program was able to handle a 600x400 image in ~2.5s
- No server-client concept

Mid point

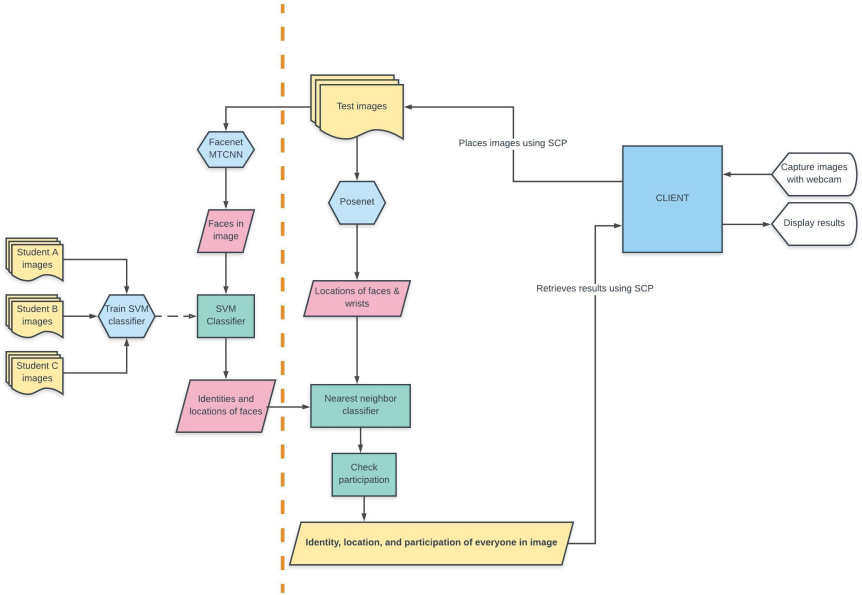
- Added posture recognition features to current model to detect participation
- Made posture and facial recognition features work together to tie identities to postures
- Allowed for processing to be done remotely (client & server)
- Made testing easier through the use of a webcam
- Made training process easier


Test images
Facenet MTCNN
Faces in image
Posenet
Locations of faces & wrists
Places images using SCP
CLIENT
Capture images with webcam
Display results
Retrieves results using SCP
Student A images
Student B images
Student C images
Train SVM classifier
SVM Classifier
Identities and locations of faces
Nearest neighbor classifier
Check participation
Identity, location, and participation of everyone in image

Current semester goals

- Improve image processing time
- Show performance for medium classroom size
- Develop robust client-server infrastructure for model training and testing
- Stretch goal: Improve training pipeline
- Stretch goal: Implement GPU

Optimizations to improve throughput



To improve throughput, we can:

1. Do these steps faster
2. Do multiple steps at the same time

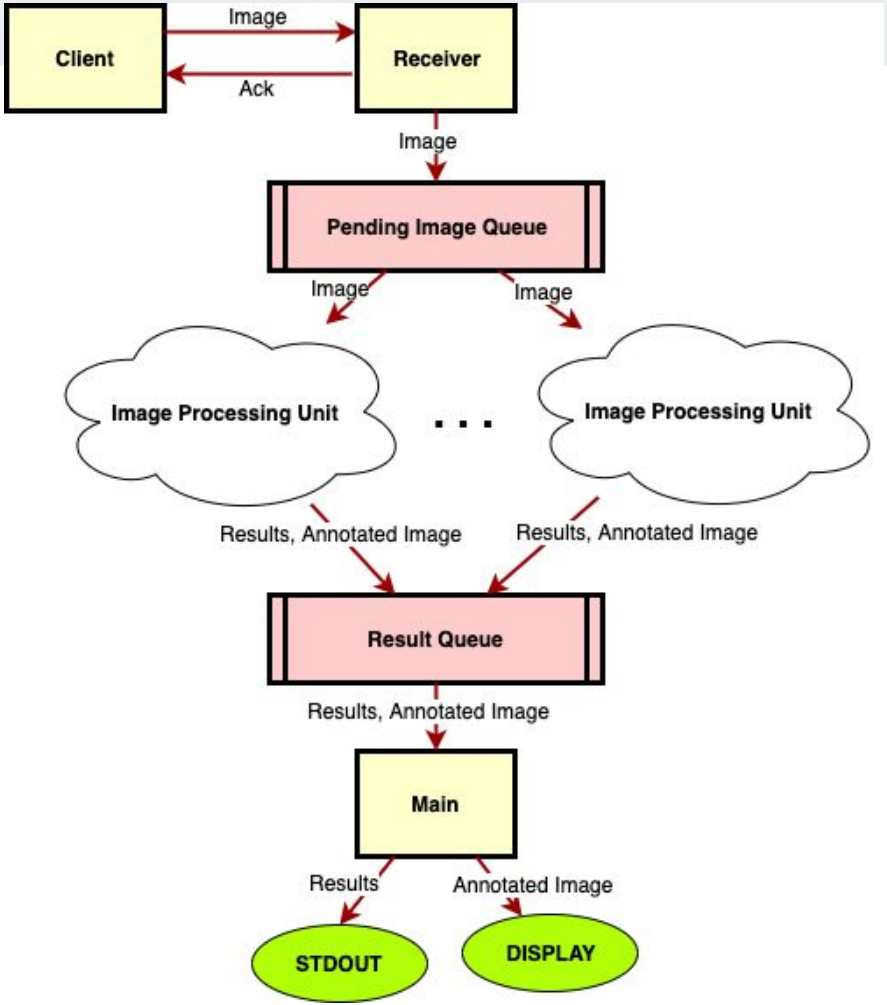
All this work was done sequentially on the server, leading to a low image throughput

We attempted to use a GPU to reduce the facial recognition time, but had difficulties (continued later).

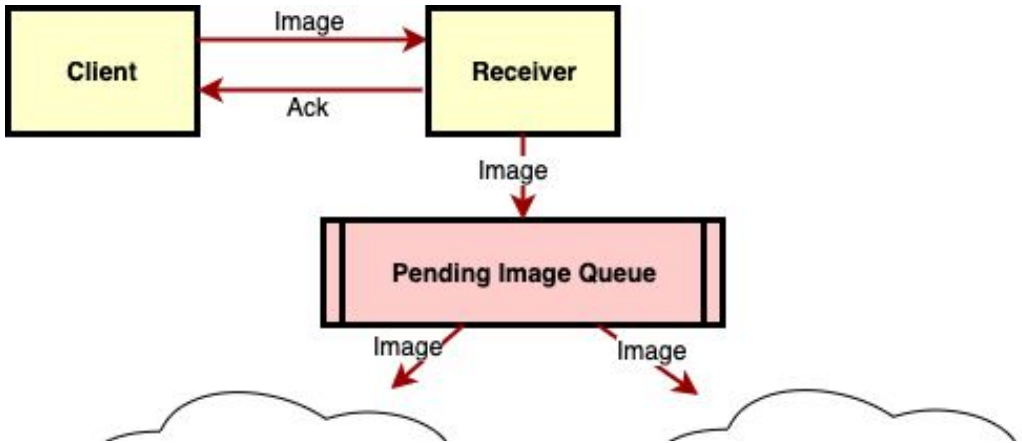
Old Process Layout

New Process Layout

Main contributions by Elliot Edmunds



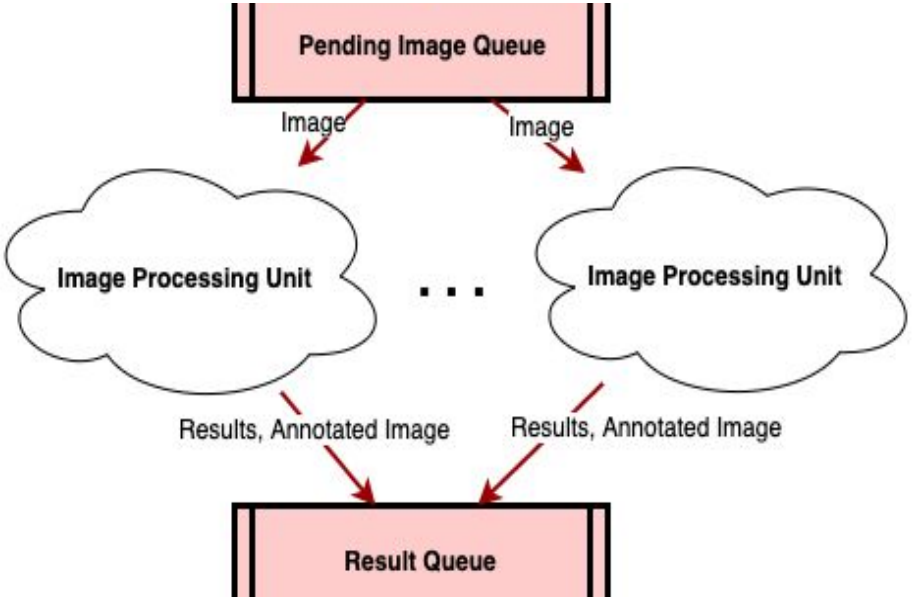
Asynchronous I/O



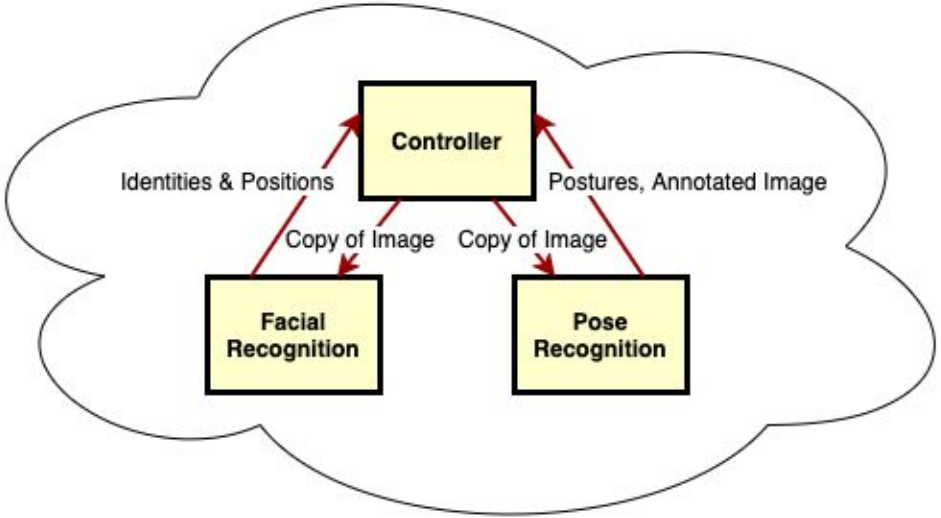
- By receiving images in an separate process, we free up the process that handles images to spend more time on images.
- Time savings are dependent on image transmission time, which is dependent on internet speed.
- Observed efficiency improvements: 40%.

Multiple Image Processing Workers

- As there are no cross-image dependencies, we are able to process images in parallel.
- The number of images that can be processed at a time is dependent on the number of cores available. We found that our CPU was maximized at 2 workers.
- Observed throughput improvements: 40%.
    - Latency is increased by 10%



Parallel Face & Posture Recognition



- As there are no dependencies between facial recognition and posture recognition, we are able to do them in parallel
- Observed efficiency improvements: 25%

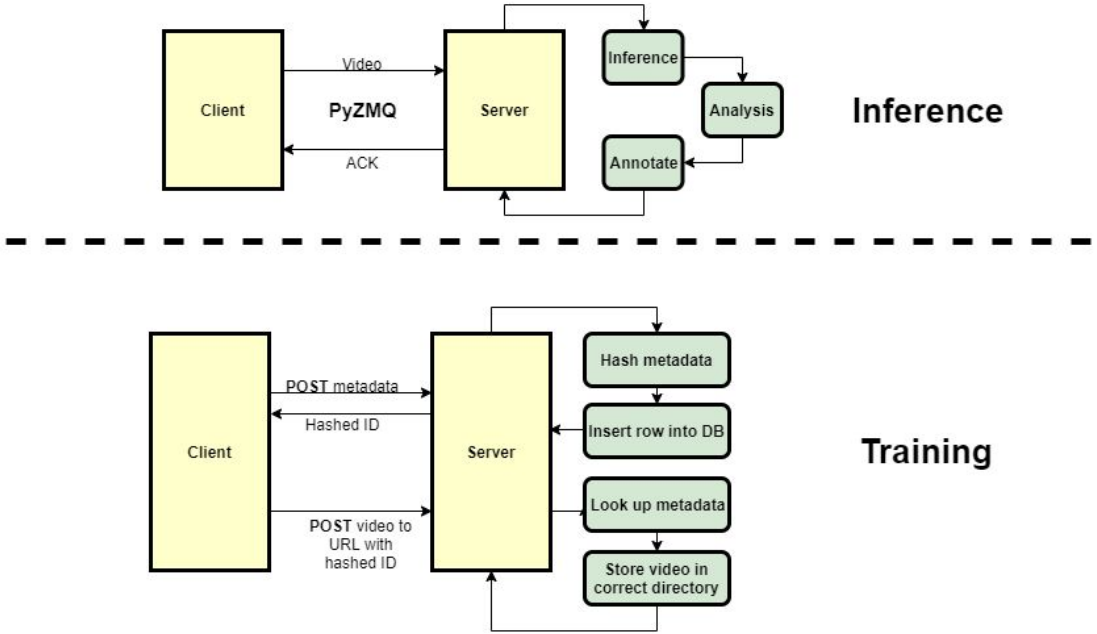
Inference: Transition from SCP to PyZMQ

- In this semester, PyZMQ, a distributed message queue that requires no message broker, is implemented to handle the transmission of livestream videos from client to server during inference
- Client-side app can be run on RPis, and multiple clients can upload live videos to the server simultaneously
- In practice, the server can spawn new processes containing different trained models on new ports
- Speed improvements dependent on network speed, observed 80% reduction in transmission time

Full stack application to streamline training

- In the previous semester, a GUI was developed with PyQt, OpenCV, and SCP that captured and uploaded selfie videos to the server
- In this semester, a full stack application has been developed that allows students to enter metadata (name, classname) and then capture selfie videos, which are uploaded to the server for training
- Backend developed under the Python Flask framework, and RESTful API endpoints are exposed for the frontend to POST metadata about the training object

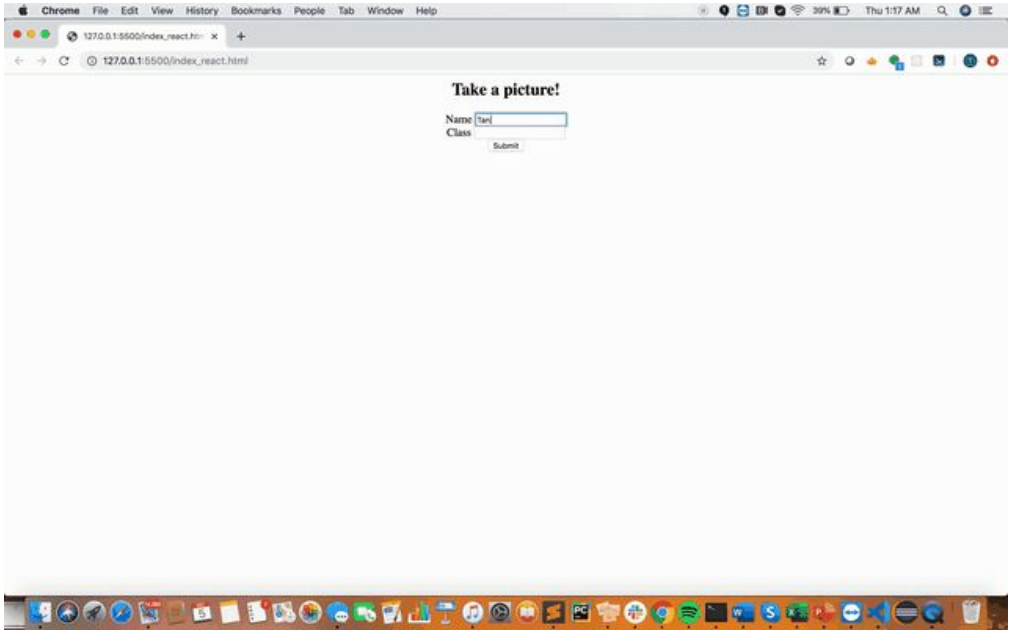
Client-Server Infrastructure



Technologies used for the Full Stack application

- BackEnd: Flask(Python Web Framework) and SQLAlchemy.
- FrontEnd: Initially attempted to use React JS, or React Native frameworks, but subsequently used the RecordRTC(JavaScript Library) for video recording, HTML and CSS for the user interface.

Front End Web Interface Demo



Stretch goal: GPU Implementation

- GPU recognized by Tensorflow when running in the tensorflow-gpu docker container and in the tensorflow conda image
- GPU crashes when the FaceNet model network is being loaded
- Debugging is expensive; have to physically reboot the server
- Future work: Reconfigure CUDA drivers

Results

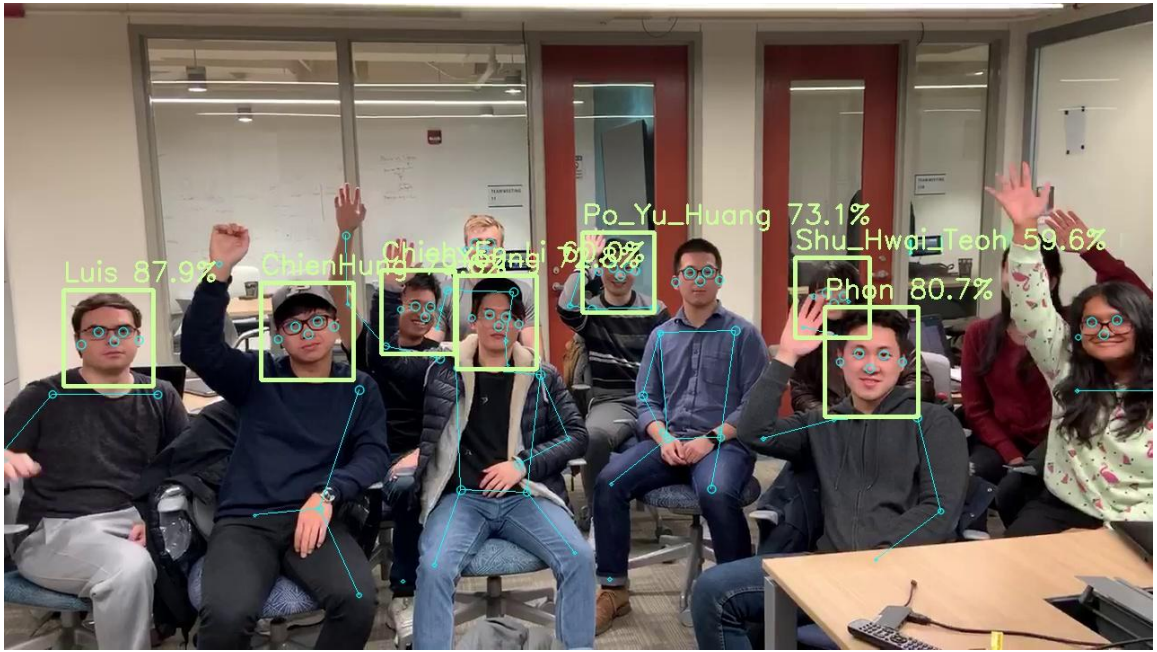


IMAGE RESULTS:

Po_Yu_Huang is participating

Chieh_En_Li is participating

Luis does not have good confidence scores, participation unknown

Phon is participating

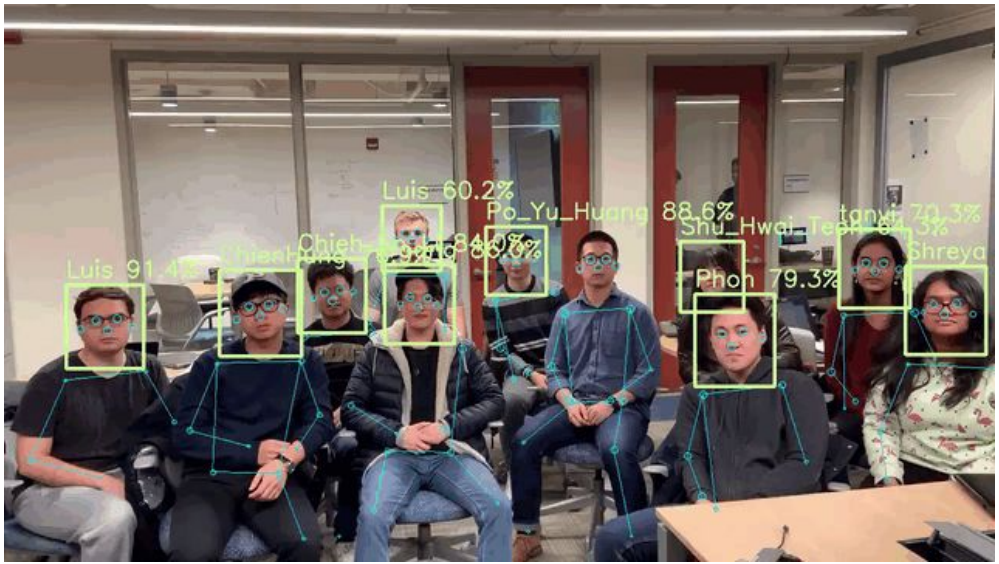
Wenbo is not participating

ChienHung is participating

Young is participating

Shu_Hwai_Teoh does not have good confidence scores, participation unknown

Cumulative Results



tanvi has:
- participated: 0
- not participated: 432
- unknown: 24

Po_Yu_Huang has:
- participated: 70
- not participated: 483
- unknown: 8

Phon has:
- participated: 2
- not participated: 395
- unknown: 96

Shreya has:
- participated: 1
- not participated: 451
- unknown: 80

Chieh_En_Li has:
- participated: 101
- not participated: 37
- unknown: 322

Shu_Hwai_Teoh has:
- participated: 0
- not participated: 271
- unknown: 45

Wenbo has:
- participated: 0
- not participated: 384
- unknown: 2

Luis has:
- participated: 0
- not participated: 521
- unknown: 44

ChienHung has:
- participated: 204
- not participated: 318
- unknown: 0

Young has:
- participated: 0
- not participated: 429
- unknown: 0

(continues...)

Future work

- Mobile application with react native
- Enable server to spawn multiple processes with different pretrained models
- Reconfigure CUDA drivers to use GPU
- Containerize with Docker for easy deployment

Accuracy Optimization Team

Team members: Shu Hwai Teoh
Po Yu Huang

Team purpose

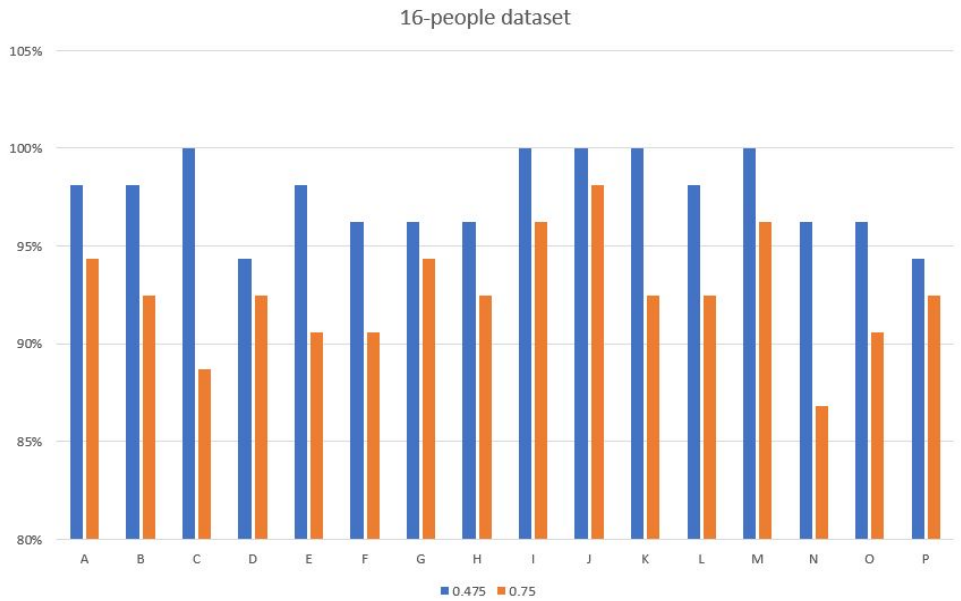
Problem: Perceived low accuracy of face recognition
Tasks:

- Build clean training datasets (with manual confirmation)
- Verify the (low) accuracy of the original model
- Improve the accuracy of the model

Accuracy of original model: 16-person dataset

- 16 unique labels: Person A thru Person P
- Each person
  - Training: 75 images
  - Testing: 54 images

- Accuracy
  - Probability threshold 0.75 : 93%
  - Probability threshold 0.475 : 98%



Findings

- Accuracy of the model decreases to 59% when tested with larger dataset (32-person dataset)
- Setting the threshold lower will cause the true recognition rate to increase, but it also tends toward a false recognition rate increase.
- Glitches between the true recognitions:
    - Changes of head orientation,
    - Overlapping of heads with other people
    - The person covers his or her face with their hands.

Methods

- Tune the parameters of SVM
  - Negative result: Did not work well
- Capture center points of eyes for each person; and compare to position in other frames
  - Positive result: Works well

Method: Capture Center Points of Eyes

- Capture the center point of eyes for each person from previous recognition
- During later recognition process:
  - Probability > 0.75: result accepted and the person is recognized
  - 0.75 > probability > 0.3: calculate the distance between the current center point and the previous center point
    - Distance between the two points < 150: result accepted and the person is recognized
  - Otherwise: result is dropped and the person is recognized as “unknown”

Method: Capture Center Points of Eyes

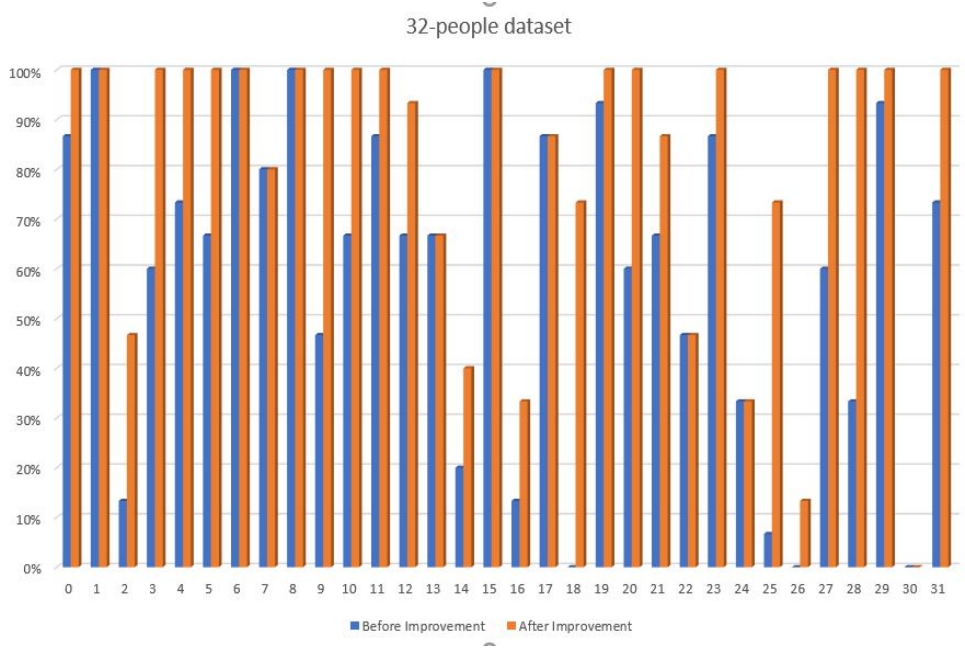
Key observation:

Many faces previously labeled “unknown” are now labeled accurately

Accuracy of model: 32-person dataset

- 0-31 person
- Each person
  - Training: 100~200 head pictures
  - Testing: 15 classroom images

- Accuracy
  - Before improvement:: 59%
  - After improvement:: 80%



Future Work

- Merge the improvement with structure of origin model
- Clothing as a contextual cue in facial recognition.

Contribution

- Po Yu Huang
  - Proposed the idea of capturing center points of eyes
- Shu Hwai Teoh
  - Build 16-people dataset
  - Proposed the idea of tuning the parameters of SVM
- Together
  - Train and test the origin model with 16-people dataset
  - Build 32-people dataset
  - Modified model for improvement
    - Tuning parameters of SVM (didn't work well)
    - Capture center points of eyes for each person to improve the glitches between recognition
  - Train and test the improved model with 32-people dataset

Engagement Estimation Team

Young Jin Jung
Cheh-En “James” Li
Chien-Hung Wang
Luis Materon

Team purpose - Starting Point

- Started as “Advanced Research & Development” group
    - Explore numerous possible functionalities to the core AAC project in the future to make the project a wholesome and more robust solution

Team purpose - Current Focus

- Seeking a method for measurement of students' engagement level
    - Head pose estimation is a great tool to map human attention, according to many studies
    - By analyzing a student's head pose, it is possible to determine whether individual student is engaged to the instructor or not
    - Can be extended further, as to measuring what percentage of student has the instructor lost/gained their interest, and that in when during the lecture

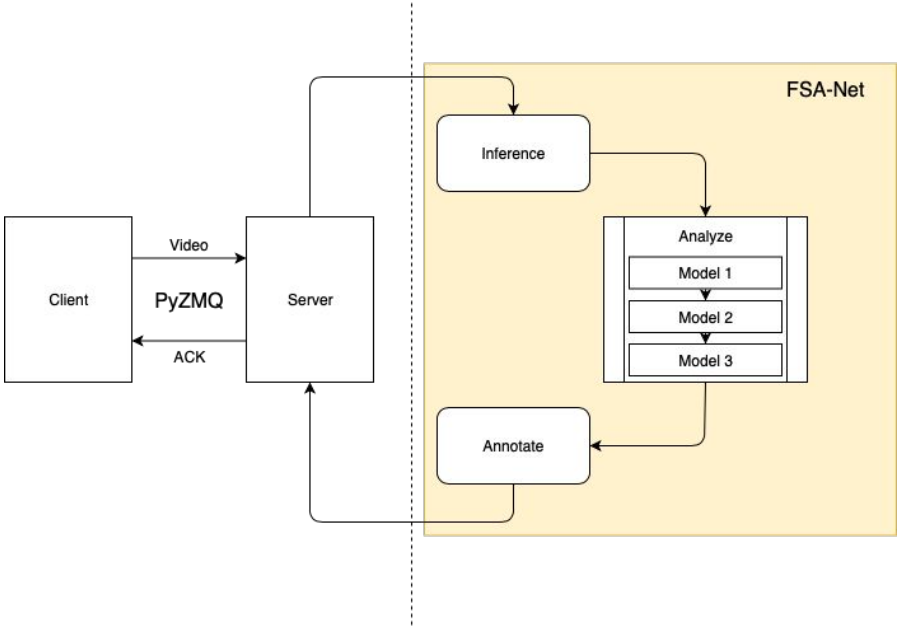
Design Challenges

- Initial approach: emotion recognition models
  - Rather illogical to map facial emotion to level of engagement
    - Many emotion classes (Happy, Neutral, Bored, Serious) can represent student is engaged
- Putting in consideration of privacy protection for students
  - Pictures of Purdue students must not leave outside of Purdue-owned server

Solution

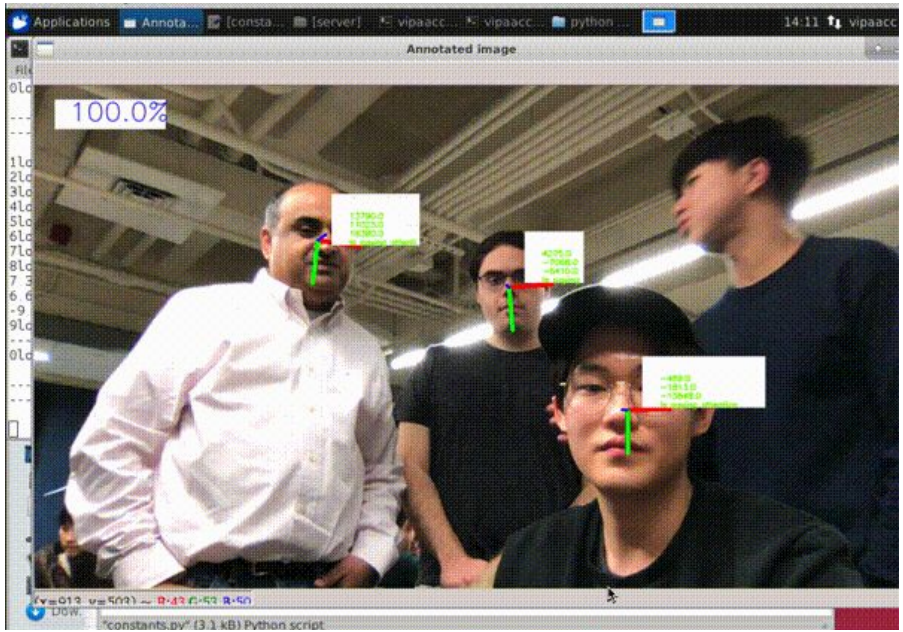
FSA-Net

- Most accurate and fast response
- Challenging modification due to:
  - More models to accommodate in server's environment
  - Depends on large number of modules
  - Heavy reliance on Cuda processing



Solution

FSA-Net

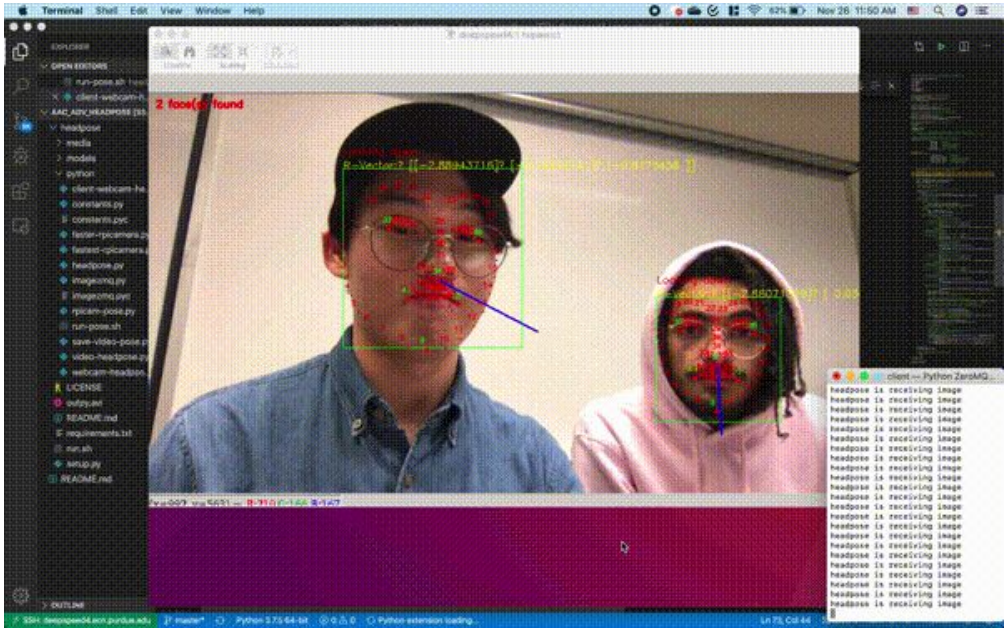


Individual Contribution

Young Jin Jung

- Conducted initial research on suitable head pose detection models for AAC project
- Set up initial development environment for implementation
- Simplified PyZMQ messaging setup, client and server-side, isolating from core AAC project
- Researched alternative solutions of FSA-Net, and actualized the implementation on the server

Individual Contribution



Individual Contribution

Chieh-En Li

- Created virtual environment for FSA-Net on the server to resolve its dependency issues
- Conducted debug on Cuda processing as well as GPU usage of FSA-Net
- Pilot application of landmark model on FSA-Net
- Implement landmark locating.

Cooperative Effort With Chien-Hung Wang

- Initial local implementation of FSA-Net
- Edge-TPU implement and debugging environment issue
- Visualize students' attention on the image

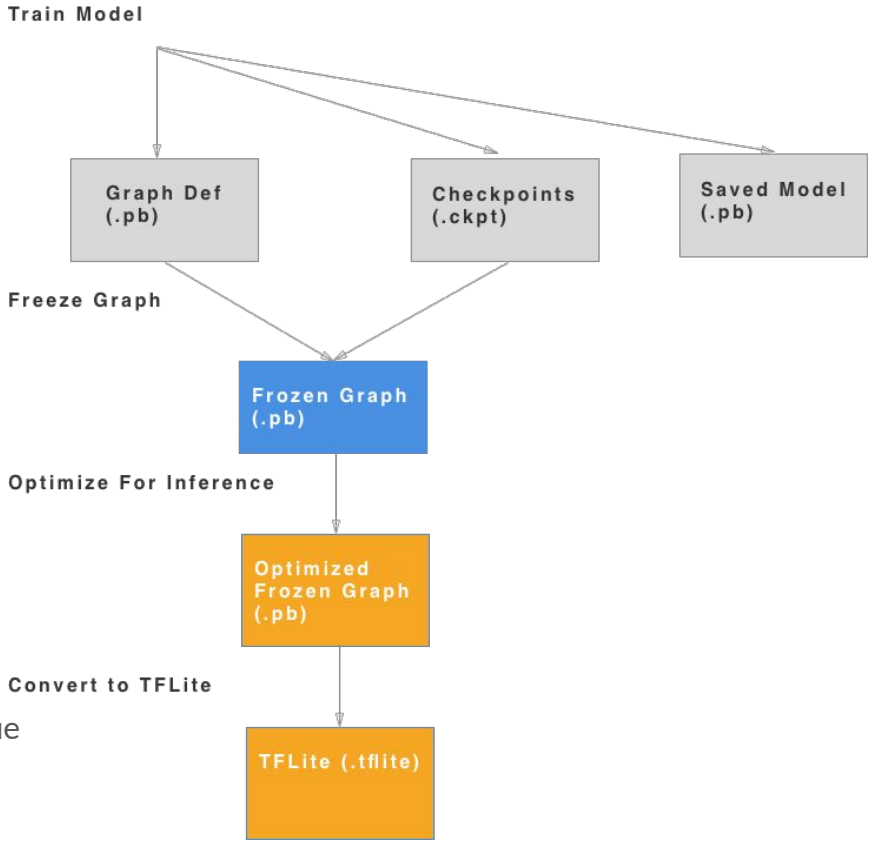
Individual Contribution

Chien-Hung Wang

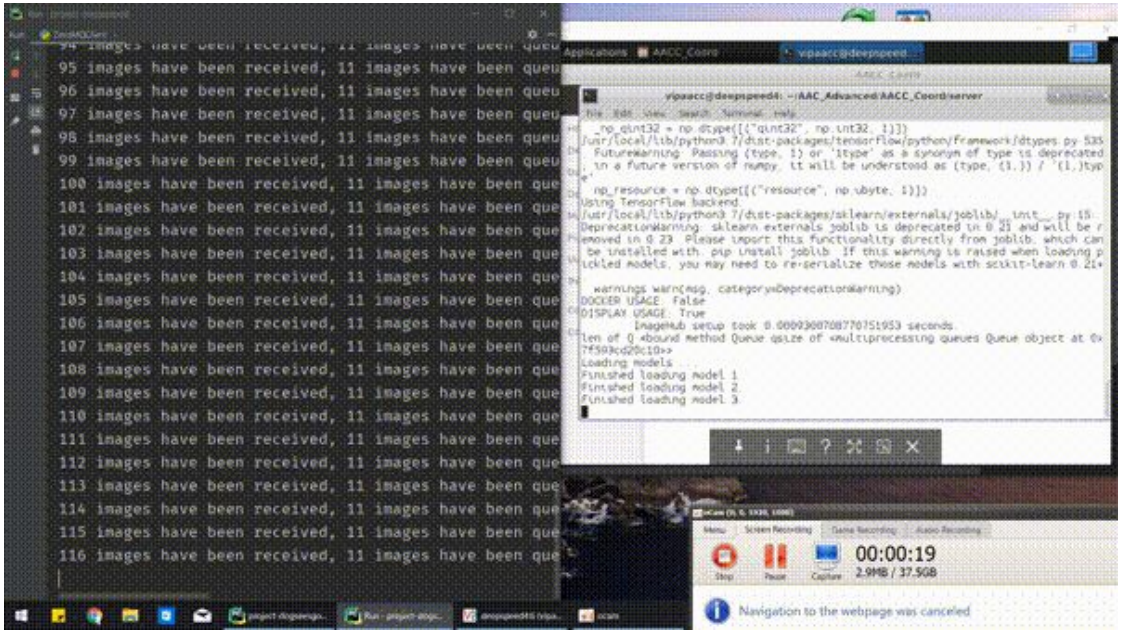
- Attempted to run Facenet Model on Google Coral Edge TPU (File conversion, Interpreting .tflit file)
- Added extra feature on FSA-Net annotation
    - Visualizing what percentage of individuals in frame is engaged

Cooperative Effort With Chieh-En Li

- Edge-TPU implement and debugging environment issue
- Initial local implementation of FSA-Net
- Visualize students' attention on the image



Cooperative effort - Chieh-En Li & Chien-Hung Wang



Individual Contribution

Luis F. Materon Botelho

- Independent research on emotion recognition models to measure student engagement
- Transition to research on feasible head pose estimation models after team's decision regarding emotion recognition
- Integrated head pose estimation model into core team's backend without interfering core features

Future Work

In upcoming semesters,

- Merging FSA-Net with the core features
    - Requires multiprocessing and, additional workers
    - Expected to provide measurement of individual students with labeled names
- Camera calibration optimization to tackle low recognition rate in sever camera angle
- Establishing master database to query attendance, number students' participation, or a period of time engagement level drops occured

Thrust 2: Virtual Teaching Assistant

Fall 2019: Chayaphon Pugkhem, Shreya Ilango

Project Overview

- Our team aims to create an AI-powered Virtual Teaching Assistant chatbot (VTA) to aid professors and teaching assistants in answering common questions about assignments, exams and scheduling appointments.

Phon Pugkhem: I want to schedule an appointment

What time do you want to meet?

Phon Pugkhem: Tomorrow 1.30pm

Which staff do you want to see?

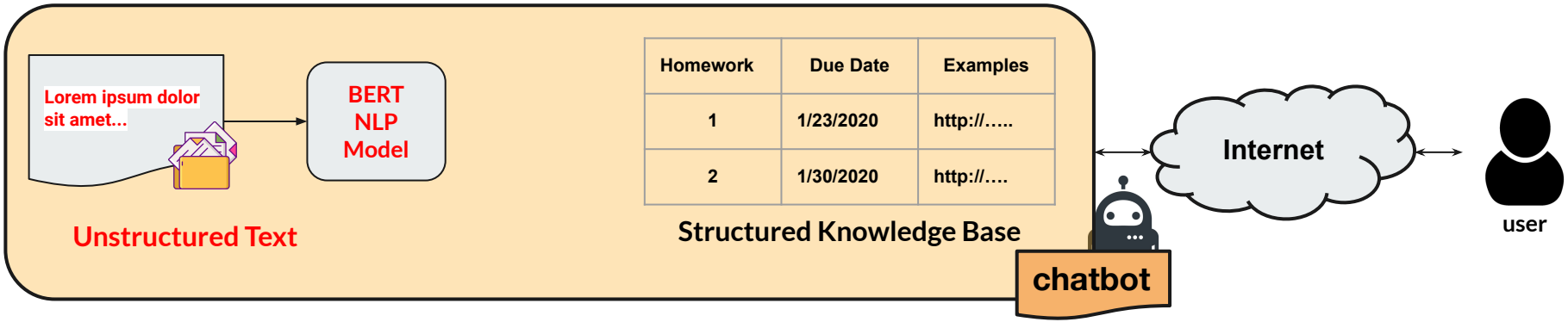
Phon Pugkhem: Professor Mithuna Thottethodi

All set: Appointment with Professor at 1.30 p.m. tomorrow.

Phon Pugkhem: Tell me everything about homework 11

Deadline: week 12, November 08
The assignment is more than 3 days late, the ECE 477 course staff may grade it at their discretion.
Temlpates:
https://engineering.purdue.edu/ece477/Course/Assignments/Template/11_EthicalAndEnvironmental.docx
Examples:
https://engineering.purdue.edu/ece477/Course/Assignments/Example/EthicalEnvironmentalEx2.pdf
https://engineering.purdue.edu/ece477/Course/Assignments/Example/EthicalEnvironmentalEx1.pdf
Other documents:
https://engineering.purdue.edu/ece477/Course/Assignments/Reference/enviro_refs.pdf

VTA: Block Diagram

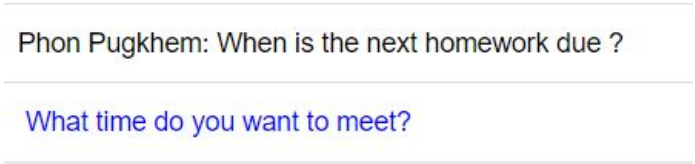


New: Natural language processing (NLP) for answering questions about unstructured text (e.g., Course Syllabus)

Old: Knowledge Base for answering questions about structured data (Calendar, due dates, files associated with homework)

Design Challenges

- The VTA can’t understand nuances and typos when a student asks a question.



- The website should be able to support multiple students chatting with the VTA at a given time.

Individual Contributions- Chayaphon

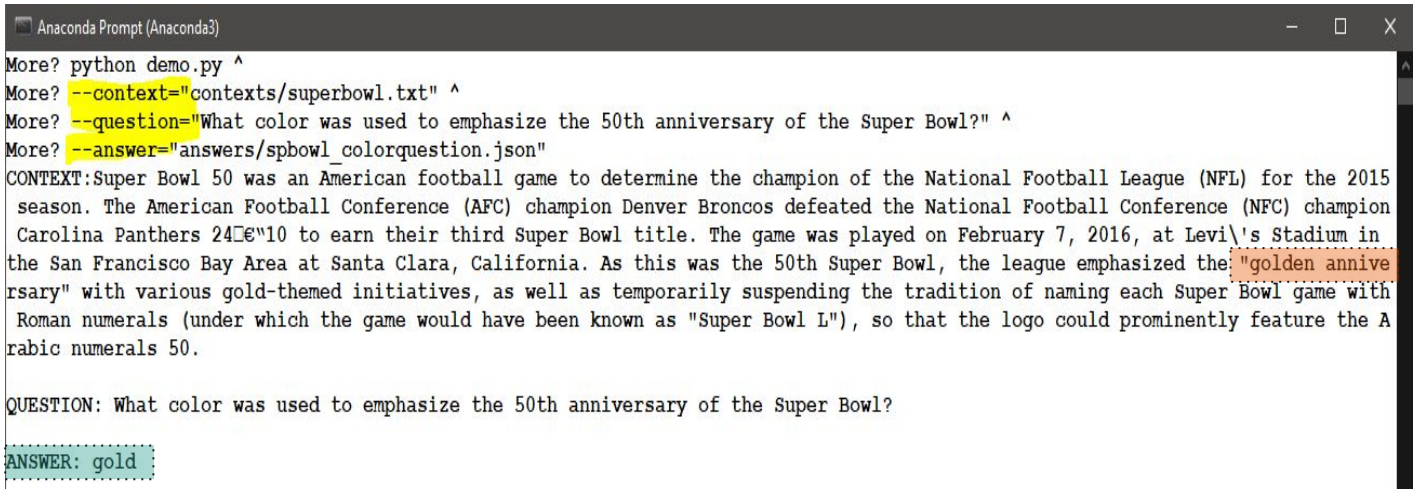
- Familiarized with the current VTA's code and updated the system's calendar.
- Researched about different methods to improve the VTA's question answering abilities and natural language understanding.
    - Results: **BERT** from Google gave best results especially in a question answering task with SQuAD datasets
- Fine-tuned BERT with SQuAD and wrote a simple script to demonstrate how our VTA can use BERT to improve it question answering ability

BERT and SQuAD Overview

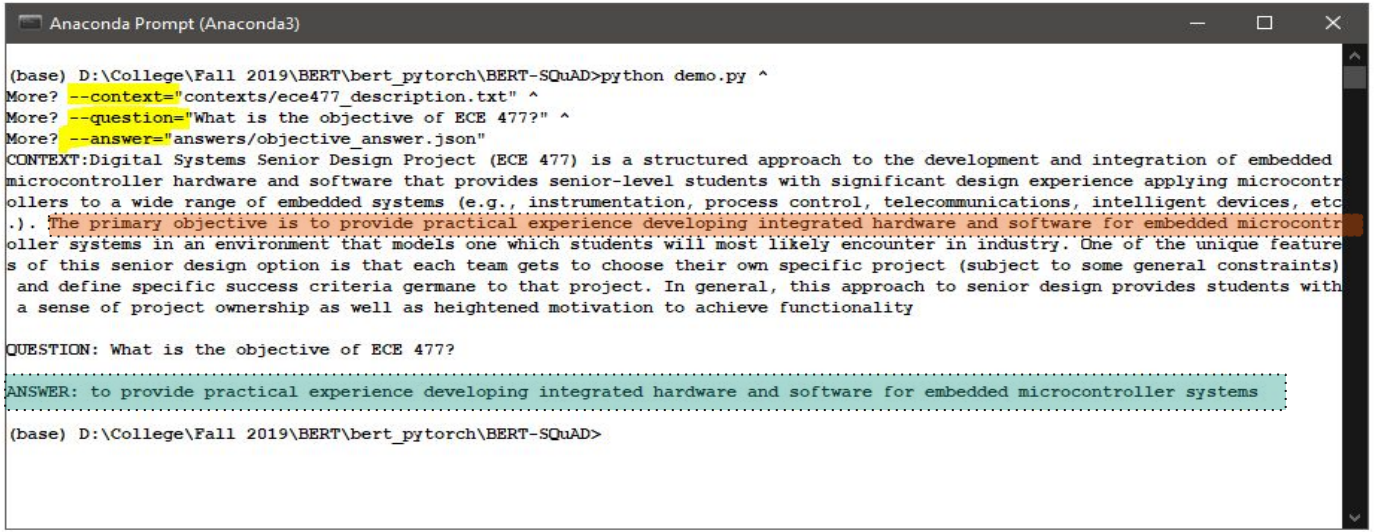
- BERT
  - Most recent pre-trained model from Google
  - Bidirectional and contextual representation.
  - Can be fine-tuned on new data to improve different NLP tasks: Question Answering, text classification , sentiment analysis, etc.
- SQuAD (The Stanford Question Answering Dataset)
  - Dataset for a question answering task
  - Contains a bunch of passages and questions regarding each passage.
  - Measures the model's ability to understand the context in each passage and answer the questions.

```
vipaacc@deepspeed4:~/VTA_BERTv1$ python3 squad1.1/evaluate-v1.1.py \
> squad1.1/dev-v1.1.json \
> tmp/squad_base/predictions.json
{"f1": 87.96098344624316, "exact_match": 80.64333017975402}
vipaacc@deepspeed4:~/VTA_BERTv1$
```

BERT Demo (1)


Anaconda Prompt (Anaconda3)
More? python demo.py ^
More? --context="contexts/superbowl.txt" ^
More? --question="What color was used to emphasize the 50th anniversary of the Super Bowl?" ^
More? --answer="answers/spbowl_colorquestion.json"
CONTEXT:Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24□€"10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi\'s Stadium in the San Francisco Bay Area at Santa Clara, California. As this was the 50th Super Bowl, the league emphasized the "golden anniversary" with various gold-themed initiatives, as well as temporarily suspending the tradition of naming each Super Bowl game with Roman numerals (under which the game would have been known as "Super Bowl L"), so that the logo could prominently feature the Arabic numerals 50.
QUESTION: What color was used to emphasize the 50th anniversary of the Super Bowl?
ANSWER: gold

BERT Demo (2)


Anaconda Prompt (Anaconda3)
(base) D:\College\Fall 2019\BERT\bert_pytorch\BERT-SQuAD>python demo.py ^
More? --context="contexts/ece477_description.txt" ^
More? --question="What is the objective of ECE 477?" ^
More? --answer="answers/objective_answer.json"
CONTEXT:Digital Systems Senior Design Project (ECE 477) is a structured approach to the development and integration of embedded microcontroller hardware and software that provides senior-level students with significant design experience applying microcontrollers to a wide range of embedded systems (e.g., instrumentation, process control, telecommunications, intelligent devices, etc.). The primary objective is to provide practical experience developing integrated hardware and software for embedded microcontroller systems in an environment that models one which students will most likely encounter in industry. One of the unique features of this senior design option is that each team gets to choose their own specific project (subject to some general constraints) and define specific success criteria germane to that project. In general, this approach to senior design provides students with a sense of project ownership as well as heightened motivation to achieve functionality
QUESTION: What is the objective of ECE 477?
ANSWER: to provide practical experience developing integrated hardware and software for embedded microcontroller systems
(base) D:\College\Fall 2019\BERT\bert_pytorch\BERT-SQuAD>

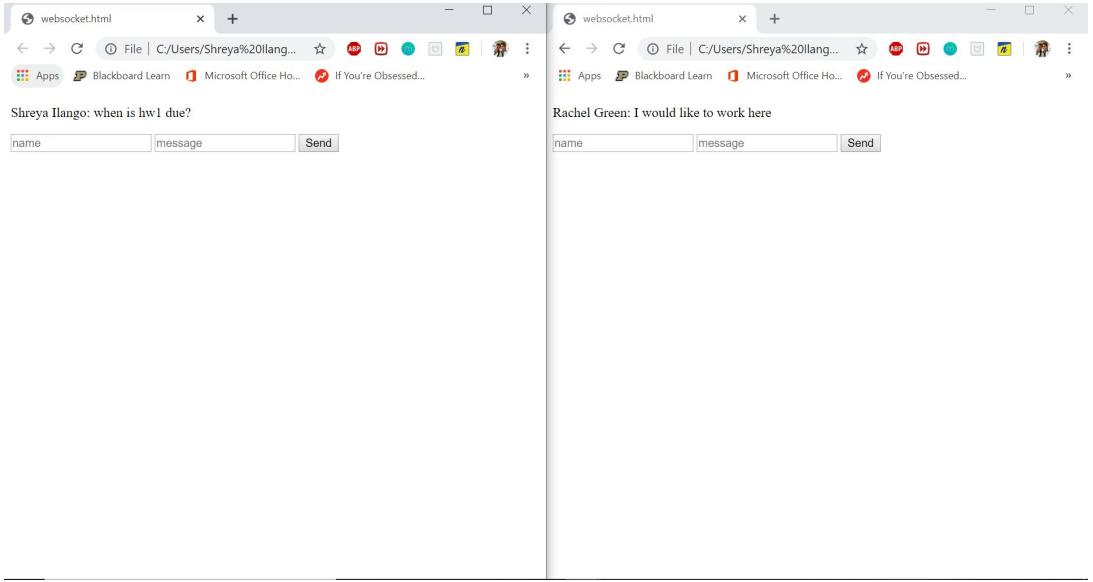
Individual Contributions - Shreya

- Learning the basics of ChatScript to understand the structure of the VTA.
- Researching methods on how to support multiple people chatting with the VTA at the same time:
    - Using WebSocket API to handle multiple connections at a time.
    - WebSocket allows a client and a server to exchange messages in a bidirectional way.
    - The other option is to create multiple copies of a bot in ChatScript, but that is very tedious and is possibly only useful for a small number of people.

Overview of WebSocket

- WebSocket is an API that enables web pages to use the protocol for two-way communication with a remote host.
- HTML5 WebSockets provide an enormous reduction in unnecessary network traffic and latency.
- HTML5 WebSockets-based applications place less burden on servers, allowing existing machines to support more concurrent connections.

WebSocket Demo



Goals for Next Semester

- Integrate BERT with ChatScript so that the VTA is able to recognize nuances and can provide the appropriate response to any question.
- Support multiple students communicating with the VTA at a point of time using WebSockets.

QUESTIONS?