GauchoEats
Dining halls are routinely congested during peak meal hours
Slow UI when interfacing with the UCSB Meal webapp
Current Solution

- Live camera footage of the dining hall entrance

Drawbacks

- Dining halls can still be congested inside if there is no line outside
Our Solution

- Provide continuous metrics real time
  - “Best time to dine”
  - Students in line
  - Capacity
  - Menu
- Package the metrics into voice assistant application
Implementation

- Used an AWS stack
  - AWS Elastic Compute Cloud
  - AWS Elastic Beanstalk
  - AWS DynamoDB
  - AWS Lambda
  - AWS Skills Development Kit
  - AWS S3
- OpenCV
- UCSB Dining API
- GnuPlot (C++)
High Level Architecture

User

Invokes Gaucho Eats

amazon alexa

Exchange requests and responses

AWS Lambda

Reads plot

Queries and updates

Amazon S3

Update plot

Updates metrics

Amazon DynamoDB

Retrieve camera feed

Process images

UCSB Dining API

OpenCV
Use Case: Dining Hall Capacity

Precondition: GauchoEats Skill has been invoked
- **Alexa** sends **Lambda** a formatted request
- **Lambda** prompts **DynamoDB** for data
- **OpenCV** object detection model updates **DynamoDB**
- **Lambda** formulates response and sends to **Alexa**
- **Alexa** dictates request and displays skill card

Postcondition: Number of students in dining hall is returned
Use Case: Dining Hall Line Plot

Precondition: GauchoEats Skill has been invoked

- **Alexa** sends **Lambda** a formatted request
- **Lambda** prompts **S3** for plot image
- **Gnuplot** plotter interface updates **S3**
- **Lambda** formulates response and sends to **Alexa**
- **Alexa** dictates request and displays skill card

Postcondition: Image of Line over Time plot displayed to Alexa App
## Non-Functional Requirements

<table>
<thead>
<tr>
<th>Continuous Updates</th>
<th>Natural Conversations</th>
<th>Concise Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real time information</td>
<td>Robust conversations</td>
<td>Concise questions and responses</td>
</tr>
<tr>
<td>Present value proposition compared to manual lookup</td>
<td>Various ways of phrasing intents and questions</td>
<td>Ask a single question w/out being further prompted</td>
</tr>
<tr>
<td></td>
<td>Feel like talking to a real person</td>
<td>Return comprehensible information</td>
</tr>
</tbody>
</table>
OpenCV Centroid Tracker

Before

After
Functionality (In Progress)

- Students in line
  - Current solution: # of persons currently being tracked in camera FOV
- Self-running object detection scripts on servers
  - Blocker: Video streaming not working
- Robust Alexa responses
  - Need: More time to cover all edge cases
Challenges

- Camera placement not ideal
- Dining Hall entrance orientation
- Stale object detection model
Future Features

- AWS Sagemaker (data analysis)
- Predictive insights (dining duration, wait times)
- Dining camera video stream (“show <diningHall>”)
Thanks!

Any questions?