Meet Our Team

Kevin Li
Amanda Yao
Ran Yoo

Susie Alptekin
Ziyang Ji
Batuhan Akcay
Sponsors:

Little Caesars Enterprises Inc.
Sponsor Company

Tim Somero
Lead Software Development

Kevin Hinks
Lead Software Engineer

Faculty:

Brent Griffin
Assistant Research Scientist, Electrical Engineering and Computer Science
1. Introduction & Project Review
   - Project Background
   - Expected Final Deliverable
   - Demo
   - Current Status

2. Critical Requirement
   - Critical Requirement 1: Classifier Accuracy
   - Critical Requirement 5: User Testing

3. Risk & Contingency
   - Current Risk 2: Poor Model Performance

4. Q&A Section
1. Introduction & Project Review

- Project Background
- Expected Final Deliverable
- Demo
- Current Status
PIZZA ANALYSIS

The best pizza
Every time

Motivation

COMPUTER VISION

+ MACHINE LEARNING

= HIGHEST QUALITY PIZZA

The best pizza
Every time
Expected Final Deliverable

1) Pizza Quality Analysis:
   - Measure and evaluate pizza features that affect quality
   - Determine pizza quality passes the LC standard

2) Feedback Generation:
   - Create feedback depending on the pizza features evaluated

3) Web Application:
   - Display the feedback to the employee on User Interface system
   - Alert the employee if pizza does not pass LC standards
Approach

Model Team
- Neural Network Models:
  - PizzaClassifier
  - PizzaDetector
  - BurntClassifier
  - PizzaTypeClassifier
  - PepperoniDetector
- VM with GPU on Azure

Data Team
- Image collection for training, validation, and testing sets
  - Internet
  - Sponsor-Provided
  - LC pizza purchases funded by MDP

UI Team
- Web application to connect frontend and backend
  - Flask
  - Nodejs
  - React
Current Status
Current Status: End-to-End System Diagram

Pizza Image → Pizza Classifier → Pizza Detector → Burnt Pizza Classifier → Pizza Type Classifier

Feedback A: Not a pizza
Feedback B: Burnt pizza

Pass Pepperoni Pizza
Pass Cheese Pizza
Pass Other Type Pizza

Fail
Current Status: End-to-End System Diagram

- **Pizza Type Classifier**
  - Pepperoni Pizza
  - Cheese Pizza
  - Other Type Pizza

- **Pepperoni Detector**
- **Distribution Evaluator**

- **Feedback C:** Good pepperoni pizza
- **Feedback D:** Incorrect number of pepperonis or pepperonis not evenly distributed
- **Feedback E:** Good cheese pizza
- **Feedback F:** Good other type pizza

**Pass**

**Fail**
Current Status: End-to-End System Example

Original Image

Pizza Classifier

Pizza Detector

There's a pizza in this image.

Model successful loaded!
Welcome to pizza quality checker!

Detecting pizza...

Pizza detected and cropped!
Current Status: End-to-End System Example

Pizza detected and cropped!

Pizza detected and cropped!

Pizza Type Classifier

Pizza burnt!

This is a pepperoni pizza.

Burnt Classifier

Checking if it is burnt...

Ops... Pizza's burnt...

Perfect timing!
Current Status: End-to-End System Example

Pepperoni Detector → Distribution Evaluator → Good quality pepperoni pizza!

This pepperoni pizza has 25 pepperonis. Pepperoni count in 4 quadrants: [7, 7, 4, 7]. Passed pepperoni tests!
## Current Status: End-to-End System Timing

### Mean Time with Testing Set on GPU:

<table>
<thead>
<tr>
<th>Model Loading Time:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean over 100 times</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prediction Time:</th>
</tr>
</thead>
<tbody>
<tr>
<td>PizzaClassifier</td>
</tr>
<tr>
<td>PizzaDetector + BurntClassifier</td>
</tr>
<tr>
<td>PizzaTypeClassifier</td>
</tr>
<tr>
<td>PepperoniDetector</td>
</tr>
<tr>
<td>DistributionAnalysis</td>
</tr>
<tr>
<td>Overall (prediction)</td>
</tr>
</tbody>
</table>
Current Status: Data

- Training and validation set collected from internet - 4000+
- Purchased sets of pizzas from Little Caesars with MDP funds ($87)
- Testing 1 - 352 images
- Testing 2 - 64 images

Current Image Annotation Method

Label 0: Pizza
0 = Not Pizza
1 = Pizza

Label 1: Pizza Type
1 = Pepperoni
2 = Cheese
3 = Other

Label 2: Pizza Shape
0 = Round
1 = Square

Label 3: Burnt?
0 = Not Burnt
1 = Burnt

Label 4: Topping Distribution
0 = Good Topping Distribution
1 = Bad Topping Dist
Current Status: User Interface

End to End Model System

This pizza looks delicious! Great job!

This pizza looks delicious! Great job!
Current Status: User Interface

End to End Model System

This pizza needs some improvement
Ooops... Pizza is burnt...

This pizza needs some improvement
Pepperoni not evenly distributed!

Try Again

Current Status: User Interface

End to End Model System

No pizza detected
2. Critical Requirements

- Critical Requirement 1: Classifier Accuracy
- Critical Requirement 5: User Testing
Critical Requirement 1: Model Development

**Title:** Classifier Accuracy Testing

**Pass Criteria:** accuracy on testing dataset:

<table>
<thead>
<tr>
<th></th>
<th>PizzaClassifier</th>
<th>BurntClassifier</th>
<th>PizzaTypeClassifier</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accuracy</strong></td>
<td>95%</td>
<td>95%</td>
<td>90%</td>
</tr>
</tbody>
</table>

**Yellow Status:** but we are confident we can accomplish this goal
Model Development Validation Method

Running individual classifier model on validation set, which is 20% images we split from training set

Running end-to-end system on testing dataset, which comprises of images we took from Little Caesars pizza and largely represents real-world image distribution.

<table>
<thead>
<tr>
<th></th>
<th>Validation result</th>
<th>Initial testing result</th>
<th>Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>PizzaClassifier</td>
<td>99.6%</td>
<td>82.9%</td>
<td>95%</td>
</tr>
<tr>
<td>BurntClassifier</td>
<td>97.8%</td>
<td>90.7%</td>
<td>95%</td>
</tr>
<tr>
<td>PizzaTypeClassifier</td>
<td>90.6%</td>
<td>84.0%</td>
<td>90%</td>
</tr>
</tbody>
</table>
Initial testing result failure cases analysis:

- Wrong Pizza Type (30% of failing cases)
- Not recognize square pizza (30% of failing cases)
- Pepperoni is folded when cutting pizza (10% of failing cases)
- Weird angle (10% of failing cases)
Title: User Interface intuitive navigation testing

Pass Criteria: at least 12 of 15 users should navigate the application from image upload to receiving feedback with minimal guidance within 2 minutes

Red Status: COVID-19 pandemic and cybersecurity concerns with contacting 15 LC kitchen employees and providing them with demo
User Interface Validation Method

1. Set up interface running locally (cannot deploy due to security reasons)

```
## SETUP

### Virtual Environment
Move to web_app directory
Run: python3 -m venv env
Run: source env/bin/activate
cd src
export FLASK_APP=uploading.py
export FLASK_ENV=development
export FLASK_DEBUG=1

### Installations
web_app dependencies (npm install)
flask_cors (pip install flask-cors)
torch (pip install torch torchvision)
detectron2 (python -m pip install 'git+https://github.com/facebookresearch/detectron2.git')
pandas (pip install pandas)
cv2 (pip install opencv-python)
imageio (pip install imageio)

### Spin Up Web App
In terminal window in web_app directory with virtual environment activated, run (npm start)
In separate terminal window in src directory with virtual environment activated, run (flask run)
```
2. For each of the 15 users, follow the script and ensure to record all data specified.

Hello and thank you for taking the time to assist in our testing suite. Using the MacOS software, we will conduct screen sharing so that you can interact with our interface.

“Please click the ‘Accept’ button as it appears on your Mac device”

Wait for connection to be established

“Alright, please verify that you can see my screen and have the ability to interact with the browser up”

Wait for confirmation

“Great, now I will direct you to the browser interface we will be testing”

Once on the correct browser for testing, proceed.

“Please follow the steps on the browser. Use the image we have provided. It is the only file located on this computer’s desktop”

Start stopwatch

“Please read us back the feedback given once you have properly followed the progression of the interface”

Wait until they read the feedback back to you

“Now that you have identified the feedback, should the pizza be served to the customer or should the pizza be thrown away and the order remade”

Wait until they respond and record if they successfully identify proper next steps with the pizza

Stop stopwatch and record time for this test

If time is over 2 minutes then record status as unsuccessful otherwise record status as successful

“Thank you for your time, your responses have been recorded. I hope you have a good day. Goodbye.”
3. Risk & Contingency

- Current Risk 2: Poor Model Performance
Risk & Contingency

Risk: Poor performance of neural network models on testing dataset

Impact: 8 (out of 10)

Trigger: Neural network models’ performance scores on the testing set is less than the threshold values described in user requirements 1.1b, 1.2b, and 1.3 in Appendix A: Project Requirements
**Probability:** 40% (as of Oct 1., DR3 Report)

<table>
<thead>
<tr>
<th>Model</th>
<th>Performance Type</th>
<th>Target Data</th>
<th>Validation Data</th>
<th>Testing Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pizza Classifier</td>
<td>Accuracy</td>
<td>95%</td>
<td>99.6%</td>
<td></td>
</tr>
<tr>
<td>Burnt Classifier</td>
<td>Accuracy</td>
<td>95%</td>
<td>97.8%</td>
<td></td>
</tr>
<tr>
<td>Pizza Type Classifier</td>
<td>Accuracy</td>
<td>90%</td>
<td>90.6%</td>
<td></td>
</tr>
<tr>
<td>Pizza Detector</td>
<td>AP50</td>
<td>95%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pepperoni Detector</td>
<td>AP50</td>
<td>90%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table: Performance of our neural network models (Oct 1.)
**Probability:** 70% (as of Oct. 8, DR3 Presentation)

<table>
<thead>
<tr>
<th>Model</th>
<th>Performance Type</th>
<th>Target Data</th>
<th>Validation Data</th>
<th>Testing Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pizza Classifier</td>
<td>Accuracy</td>
<td>95%</td>
<td>99.6%</td>
<td>82.9%</td>
</tr>
<tr>
<td>Burnt Classifier</td>
<td>Accuracy</td>
<td>95%</td>
<td>97.8%</td>
<td>90.7%</td>
</tr>
<tr>
<td>Pizza Type Classifier</td>
<td>Accuracy</td>
<td>90%</td>
<td>90.6%</td>
<td>84.0%</td>
</tr>
<tr>
<td>Pizza Detector</td>
<td>AP50</td>
<td>95%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pepperoni Detector</td>
<td>AP50</td>
<td>90%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table: Current performance of our neural network models (Oct. 8)
**Contingency Plan I:** Retrain neural network models with the current testing set for neural network to train on more data that represents real world image distribution.

**Cost & Benefits:** For accurate real life performance measurements, testing data needs to be hold out from the data used for training and fine tuning the neural network models. Therefore we will need to gather a new testing set of pizza images by purchasing more LC pizzas with MDP funds.

**Status:** Activated

**Responsible:** Data Team and Model Team (Ran, Batuhan, Kevin, Amanda, Ziyang)
Contingency Plan II: By November 13th, if accuracy is still lower than threshold values, we will try to use a single neural network model, PizzaEvaluationClassifier, instead of using the current end-to-end model system to aim for a higher performance score.

Cost & Benefits: In the short time between November 13th and the project deadline it might be easier to improve the performance of a single neural network instead of an end-to-end model system with multiple models however this approach would not allow us to give pizza-specific feedback like the end-to-end model system.

Status: Not Activated

Responsible: Data Team and Model Team (Ran, Batuhan, Kevin, Amanda, Ziyang)
ADDITIONAL SLIDES:
## List of All Critical Requirements

<table>
<thead>
<tr>
<th>Requirement Group</th>
<th>Requirement Target &amp; Units</th>
<th>Origin of Validation Method</th>
<th>Responsible for Completion</th>
<th>Status</th>
</tr>
</thead>
</table>
| 1                 | Model Development
PizzaClassifier detects pizza with at least 95% accuracy on testing dataset (i.e. images we took from Little Caesars pizza, which represents the image distribution of the problem we are tackling); BurntClassifier detects burnt pizza with at least 95% accuracy on testing dataset; PizzaTypeClassifier detects pizza with at least 90% accuracy on testing set | Student Developed           | Model Team, Data Team               | Yellow |
| 2                 | Model Development
PizzaDetector detects pizza with at least 95% AP50 accuracy on testing set; PepperoniDetector detects the pepperonis with at least 90% AP50 accuracy on validation set                                                                 | Student Developed           | Model Team, Data Team               | Yellow |
| 3                 | Model Development
Integrate existing models into an End-to-End system and return 90% correct feedbacks on testing set of 50 images                                                                                                                                                   | Student Developed           | Model Team                          | Yellow |
| 4                 | System Integration
End-to-End system can process 5 datasets of 10 images and give feedback in a total runtime of 1 minute per dataset                                                                                                                                                  | Student Developed           | Model Team                          | Yellow |
| 5                 | UI Development
At least 80% of 15 users can navigate the application from image upload to receiving feedback with minimal guidance within 2 minutes                                                                                                                                  | Student Developed           | UI/UX Team                          | Red    |
<table>
<thead>
<tr>
<th>Requirement Number</th>
<th>Requirement Group</th>
<th>Sponsor Priority</th>
<th>Requirement Target &amp; Units</th>
<th>Origin of Validation Method</th>
<th>Responsible for Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1a</td>
<td>Model Development</td>
<td>1</td>
<td>PizzaClassifier classifies pizza with at least 98% accuracy on validation set (i.e. images from online, similar distribution as training set); BurntClassifier classifies burnt pizza with at least 98% accuracy on validation set; PizzaTypeClassifier classifies pizza with at least 90% accuracy on validation set</td>
<td>Student Developed</td>
<td>Model Team</td>
</tr>
<tr>
<td>1.1b</td>
<td>Model Development</td>
<td>1</td>
<td>PizzaClassifier detects pizza with at least 95% accuracy on testing dataset (i.e. images we took from Little Caesars pizza, which represents the image distribution of the problem we are tackling); BurntClassifier detects burnt pizza with at least 95% accuracy on testing dataset; PizzaTypeClassifier detects pizza with at least 90% accuracy on testing set</td>
<td>Student Developed</td>
<td>Model Team, Data Team</td>
</tr>
<tr>
<td>1.2a</td>
<td>Model Development</td>
<td>1</td>
<td>PizzaDetector detects pizza with at least 95% AP50 accuracy on validation set; PepperoniDetector detects the pepperonis with at least 90% AP50 accuracy on validation set</td>
<td>Student Developed</td>
<td>Model Team</td>
</tr>
<tr>
<td>1.2b</td>
<td>Model Development</td>
<td>1</td>
<td>PizzaDetector detects pizza with at least 95% AP50 accuracy on testing set; PepperoniDetector detects the pepperonis with at least 90% AP50 accuracy on testing set</td>
<td>Student Developed</td>
<td>Model Team, Data Team</td>
</tr>
<tr>
<td>Requirement Number</td>
<td>Requirement Group</td>
<td>Sponsor Priority</td>
<td>Requirement Target &amp; Units</td>
<td>Origin of Validation Method</td>
<td>Responsible for Completion</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------</td>
<td>------------------</td>
<td>-----------------------------</td>
<td>-----------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>1.3</td>
<td>Model Development</td>
<td>2</td>
<td>Integrate existing models into an End-to-End system and return 90% correct feedbacks on testing set of 50 images</td>
<td>Student Developed</td>
<td>Model Team</td>
</tr>
<tr>
<td>1.4</td>
<td>Model Development &amp; Data</td>
<td>3</td>
<td>The model should be 100% retrainable with 5 different sizes of dataset and larger dataset should have accuracy greater than or equal to the smaller dataset</td>
<td>Student Developed</td>
<td>Model Team</td>
</tr>
<tr>
<td>2.1</td>
<td>System Integration</td>
<td>2</td>
<td>For 5 different datasets composed of 10 images End-to-End system can process and give feedback for each dataset under a total runtime of 1 minute</td>
<td>Student Developed</td>
<td>Model Team</td>
</tr>
<tr>
<td>3.1</td>
<td>User Interface</td>
<td>1</td>
<td>At least 80% of 15 users can navigate the application from image upload to receiving feedback with minimal guidance within 2 minutes</td>
<td>Student Developed</td>
<td>UI/UX Team</td>
</tr>
<tr>
<td>3.2</td>
<td>User Interface</td>
<td>2</td>
<td>Conduct user testing on 15 users and at least 80% of users correctly identify the given feedback through survey form</td>
<td>Student Developed</td>
<td>UI/UX Team</td>
</tr>
<tr>
<td>Number</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Risk</td>
<td>Poor User Experience</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Impact (1:best, 10:worst)</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Probability</td>
<td>50%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trigger</td>
<td>User research testing described in user requirements 3.1 and 3.2 in Appendix A: Project Requirements fail by October 30th</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contingency Plans</td>
<td>Redesign the UI according to feedback from the user research survey, which we estimate will take 2 weeks. Since 2 weeks will be allocated for redesigning, drop the low priority task to deploy the model and web interface to a virtual machine. Do a second round of user testing by November 20th, if possible and this time conduct user testing with both sponsors and other LC employees.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Responsible</td>
<td>UI/UX Team (Susie, Kevin)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Number | 2
---|---
Risk | Poor performance of neural network models on testing dataset
Impact (1:best, 10:worst) | 8
Probability | 70%
Trigger | Neural network models’ accuracy and AP50 scores on the testing set is less than the threshold values described in user requirements 1.1b, 1.2b, and 1.3 in Appendix A: Project Requirements
Contingency Plans | Retrain neural network models with the current testing set that represents real world image distribution. Gather a new testing set of pizza images in different conditions taken in a controlled environment, by purchasing more LC pizzas with MDP funds. Measure the accuracy of models with the new testing set. If accuracy is still lower than threshold values by November 13th, use a simple PizzaEvaluationClassifier instead of using the current end-to-end model system to aim for a higher accuracy score.
Responsible | Data Team and Model Team (Ran, Batuhan, Kevin, Amanda, Ziyang)
# Risk & Contingency

<table>
<thead>
<tr>
<th>Number</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk</td>
<td>Team member unavailable</td>
</tr>
<tr>
<td>Impact (1:best, 10:worst)</td>
<td>6</td>
</tr>
<tr>
<td>Probability</td>
<td>65%</td>
</tr>
<tr>
<td>Trigger</td>
<td>A team member becomes unavailable due to Covid-19 or unforeseen reasons</td>
</tr>
<tr>
<td>Contingency Plan</td>
<td>Drop the low priority task of evaluating multiple varieties of toppings. Restructure the subteams according to the remaining high priority tasks and redistribute tasks to responsible subteam members.</td>
</tr>
<tr>
<td>Responsible</td>
<td>Current Team Lead</td>
</tr>
</tbody>
</table>
Risk 2: Poor performance of neural network models on testing dataset

**Impact:** 8 (out of 10)

Models might:
- return “no pizza in image” feedback even if there is a pizza in image (Pizza Classifier)
- treat as if there is a pizza in image, even though there is none (Pizza Classifier)
- crop and rescale wrong part of the image (Pizza Detector)
- return “burnt pizza” feedback even if the pizza is not burnt (Burnt Classifier)
- execute quality evaluation for wrong type of pizza (Pizza Type Classifier)
- Detect pepperonis incorrectly and return wrong results for the distribution (Pepperoni Detector)

Our end-to-end model system follows a workflow process and if any of the neural network models in the system does not exhibit high performance, errors in quality analysis will occur and employee will receive incorrect feedback.
Risk 2: Poor performance of neural network models on testing dataset

**Trigger:** Neural network models’ performance scores on the testing set is less than the threshold values described in user requirements 1.1b, 1.2b, and 1.3 in Appendix A: Project Requirements

<table>
<thead>
<tr>
<th>Req. Number</th>
<th>Req. Group</th>
<th>Sponsor Priority</th>
<th>Req. Target &amp; Units</th>
<th>Origin of Validation Method</th>
<th>Responsible for Completion</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1b</td>
<td>Model Development</td>
<td>1</td>
<td>PizzaClassifier detects pizza with at least 95% accuracy on testing dataset (i.e. images we took from Little Caesars pizza, which represents the image distribution of the problem we are tackling); BurntClassifier detects burnt pizza with at least 95% accuracy on testing dataset; PizzaTypeClassifier detects pizza with at least 90% accuracy on testing set</td>
<td>Student Developed</td>
<td>Model Team, Data Team</td>
<td>Yellow</td>
</tr>
<tr>
<td>1.2b</td>
<td>Model Development</td>
<td>1</td>
<td>PizzaDetector detects pizza with at least 95% AP50 accuracy on testing set; PepperoniDetector detects the pepperonis with at least 90% AP50 accuracy on testing set</td>
<td>Student Developed</td>
<td>Model Team, Data Team</td>
<td>Yellow</td>
</tr>
<tr>
<td>1.3</td>
<td>Model Development</td>
<td>2</td>
<td>Integrate existing models into an End-to-End system and return 90% correct feedbacks on testing set of 50 images</td>
<td>Student Developed</td>
<td>Model Team</td>
<td>Yellow</td>
</tr>
</tbody>
</table>

Table: User requirements mentioned in the trigger of risk 2
Future Work: User Interface

- Web application only running locally and not yet deployed for remote access
- Alert user that image has been successfully uploaded
- Conduct user testing
# LCE Project Plan

<table>
<thead>
<tr>
<th>PHASE</th>
<th>MILESTONE</th>
<th>PERCENT COMPLETE</th>
<th>OCT</th>
<th>NOV</th>
<th>DEC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>COMPLETE SOFTWARE DEVELOPMENT</strong></td>
<td>6. Gather Feedback MDP: DR3</td>
<td>50%</td>
<td>38</td>
<td>39</td>
<td>40</td>
</tr>
<tr>
<td>Determine Quality of Pizza with Evaluator</td>
<td></td>
<td>60%</td>
<td>41</td>
<td>42</td>
<td>43</td>
</tr>
<tr>
<td>Develop User Interface and Browser</td>
<td></td>
<td>50%</td>
<td>44</td>
<td>45</td>
<td>46</td>
</tr>
<tr>
<td>System Integration</td>
<td></td>
<td>15%</td>
<td>47</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td><strong>FINAL TESTING AND ADDRESSING ISSUES</strong></td>
<td>7. Incorporate User and Sponsor Feedback</td>
<td>0%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic End to End Functionality Test</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CONTIGENCY TIME AND STRETCH GOALS</strong></td>
<td>8. Finalize Product and Design</td>
<td>0%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>FINAL DESIGN PRESENTATION AND DELIVERY</strong></td>
<td>9. Final Product Delivery</td>
<td>0%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Detailed Project Plan

<table>
<thead>
<tr>
<th>PHASE</th>
<th>MILESTONE</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPLETE SOFTWARE DEVELOPMENT</td>
<td>1. Gather Feedback MDP: DR3</td>
</tr>
<tr>
<td>Determine Quality of Pizza with Evaluator - Model/Data Subteam</td>
<td>2. Distinguish Good vs Bad Pizza with Pizza Evaluator</td>
</tr>
<tr>
<td>Collect Necessary Images Needed for Testing</td>
<td>3. Feedback Generator for Pizza</td>
</tr>
<tr>
<td>Train and Extract Evaluator Standards from Pizza Data (Pepperoni</td>
<td></td>
</tr>
<tr>
<td>Distribution, Burnt)</td>
<td></td>
</tr>
<tr>
<td>Evaluate the Extracted Quality with Binary Standards</td>
<td></td>
</tr>
<tr>
<td>Provide Feedback Based on Evaluated Binary Standards</td>
<td></td>
</tr>
<tr>
<td>Develop User Interface and Browser - UI/UX Subteam</td>
<td>4. Finalize Working User Interface</td>
</tr>
<tr>
<td>Develop Fully Functioning Local Interface and Backend</td>
<td></td>
</tr>
<tr>
<td>System Integration - All Subteams</td>
<td>5. Connect Evaluator to Interface</td>
</tr>
<tr>
<td>Connect Backend to End to End Model System</td>
<td></td>
</tr>
<tr>
<td>FINAL TESTING AND ADDRESSING ISSUES</td>
<td>6. Incorporate User and Sponsor Feedback</td>
</tr>
<tr>
<td>End to End Integrated System Accuracy and Efficiency Testing</td>
<td></td>
</tr>
<tr>
<td>User Interface User Testing</td>
<td></td>
</tr>
<tr>
<td>CONTINGENCY TIME AND STRETCH GOALS</td>
<td>7. Finalize Product and Design</td>
</tr>
<tr>
<td>Deploy Interface on Web/Little Caesar Servers</td>
<td></td>
</tr>
<tr>
<td>Develop Evaluator for Multiple and Non-pepperoni Toppings</td>
<td></td>
</tr>
<tr>
<td>FINAL DESIGN PRESENTATION AND DELIVERY</td>
<td>8. Final Product Delivery</td>
</tr>
</tbody>
</table>
PIZZA! PIZZA!
The best pizza every time!

PIZZA ANALYSIS

HIGHEST QUALITY PIZZA

COMPUTER VISION + MACHINE LEARNING

CONFIDENTIAL
What are we doing?

Final Deliverable:
A system to identify pizza quality: improve kitchen efficiency and help LC employees make perfect pizza consistently

Our System:

Classifier Models:
- Classify pizza types (i.e. cheese, pepperoni, others).
- Classify burnt vs not burnt pizzas.

Detector Models:
- Detect pizzas’ relative locations in the image.
- Detect pizza toppings (i.e. pepperonis) and analyze topping distribution.

User Interface:
- Give feedback to employees about the pizza quality through UI.
- Simple and functional UI for employees.

With 90% accuracy on our system!
What does this accomplish?

Classifiers:
- Identify pizza types: Hula Hawaiian, Supreme, …
- Identify pizza shapes: round, square, …
- Analyze for separate standards

Detectors:
- Detect topping types: black olives, ham…
- Analyze topping distributions
- Analyze for separate standards

With the help of franchise infrastructure