Intelligent Screw Organizer - ISO

Team 20
Jordon Balskus, Jordan Gyaltsen, Andrew McGrath, Rajesh Shahi

March 18, 2019
Advisor: Professor Eslami
ISO Team:

Jordan Gyaltsen  
CSE

Jordon Balskus  
CSE

Andrew McGrath  
CSE

Rajesh Shahi  
CSE
Outline

1. Problem Statement
2. System Review
3. CDR Accomplishments
4. FPR Deliverables
5. Demo
6. Questions
The Problem

- Many people and machines shops have loose screws laying around and do not want to invest the time and effort in sorting them. They also don’t want to spend the money in an industrial sorting machine.

- Throwing them out would be a waste of materials and only contribute to the growing trash problem our planet has.

We have a quick and low-cost solution for the problem: ISO
Backend Brains - Summary

Intel i7 running extensive Python code controls entire system

- Communicates with Raspberry Pi over SSH and SFTP
- Orients image then determines screw height and width
- Checks database to see if there is a match
- Sends command (over SSH channel) to move sorting slider into proper position
Via SFTP, receive screw image from Raspberry Pi

Rotate and fill image to align screw in vertical orientation

Complete Gaussian Blur, Grayscale, Canny Algorithm to image

Using pixel manipulations and custom algorithms, computes length and width of screw

TO DO: Add TensorFlow model that classifies head type to add additional reference criteria

Make decision based on data and move servo
Via SFTP, receive screw image from Raspberry Pi

- Rotate and fill image to align screw in vertical orientation
- Completes Gaussian Blur, Grayscale, Canny Algorithm to image
- Using pixel manipulations and custom algorithms, computes length and width of screw
- TO DO: Add TensorFlow model that classifies head type to add additional reference criteria
Enclosure - Conveyor and Raspberry Pi

- Conveyor System (controlled by motor driver PCB - To Do)
  - Feeds screws into enclosure system
  - Powered by driver which is controlled by Pi command

- Raspberry Pi
  - Turns on via SSH command from PC
  - On program inception, turns on conveyor belt via control signal and runs continuous ScrewDetection.py script
  - On detections, send image to PC via SFTP
  - Receive servo position detection decision via SSH and make the move
Enclosure - Conveyor and Raspberry Pi

- Conveyor System (controlled by motor driver PCB - To Do)
- Feeds screws into enclosure system
- Powered by driver which is controlled by Pi command
- Raspberry Pi
  - Turned on via SSH command from PC
  - On program inception, turns on conveyor belt via control signal and runs continuous ScrewDetection.py script
  - On detections, send image to PC via SFTP
  - Receive servo position detection decision via SSH and make the move
Sorting Mechanism

- Conveyor belt feeds into mouth of slide sorter
- Raspberry Pi sends commands via GPIO pins to servomotor
- 6 different bins will hold screws (5 being similar screws with 1 being miscellaneous)
Proposed CDR Deliverables

- Automatic Conveyor system
  - Take single screw and setup under camera
  - Able to pass screw to final sorting placement

- Integrate Backend Communication
  - Tell the sorting mechanism which bin to place current screw
  - Receive data from camera in real time

- Build individualizer mechanism
  - Separate out a single screw and setup under camera
  - Build individualizer system and enclosure for mount
Proposed CDR Deliverables

Automatic Conveyor system
- Take single screw and setup under camera
- Able to pass screw to final sorting placement

Integrate Backend Communication
- Tell the sorting mechanism which bin to place current screw
- Receive data from camera in real time

Build individualizer mechanism
- Separate out a single screw and setup under camera
- Build individualizer system and enclosure for mount
Individualizer Dilemma

If we had the budget of a medium-sized business, an automatic bowl feeder would be feasible. However:

- Custom bowl feeder would cost thousands ($$$$$$)
- Would be an engineering feat in and of itself
- **We have been focusing mainly on the CSE problems**

Nonetheless we will be researching and pursuing alternatives:

- Conical funnel at mouth of conveyor belt that singles screws
- Servo-controlled incline screw head holding system
- Hand fed
FPR Deliverables

- **Motor Driver PCB** - Jordon B
  - Conveyor Belt gear motor needs a variable speed driver and power supply.

- **TensorFlow Model Creation** - Jordan G
  - Implement machine learning model for head type detection
  - Incorporate head decision into current detection algorithm

- **Optimize Software Suite** - Andrew M.
  - Get detection code to work as flawlessly as possible

- **Aesthetic Improvements** - Rajesh S.
  - Upgrade enclosure structure and add professional touch

- **Final Prototype Completion** - All Members
  - Have final ISO system completed and working by FPR
Gantt Chart Status at MDR

Start SDP: Sep 10
PDR: Oct 15
MDR: Dec 10
CDR: Feb 15
FDR: Apr 28

- Backend System: 107 days, Sep 13 - Feb 10
- Sorting Mechanism: 21 days, Nov 3 - Dec 3
- Placement Mechanism: 68 days, Nov 20 - Feb 21
- Individualizer: 69 days, Jan 20 - Apr 25
Current Gantt Chart

<table>
<thead>
<tr>
<th>Task</th>
<th>Duration</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build TensorFlow Model</td>
<td>14 days</td>
<td>Mar 19</td>
<td>Apr 5</td>
</tr>
<tr>
<td>Implement Built Model</td>
<td>10 days</td>
<td>Apr 6</td>
<td>Apr 19</td>
</tr>
<tr>
<td>Design Motor Driver Circuit</td>
<td>8 days</td>
<td>Mar 19</td>
<td>Mar 28</td>
</tr>
<tr>
<td>Submit to Maker and Wait</td>
<td>7 days</td>
<td>Mar 29</td>
<td>Apr 8</td>
</tr>
<tr>
<td>Incorporate PCB and Put Everything</td>
<td>9 days</td>
<td>Apr 9</td>
<td>Apr 19</td>
</tr>
<tr>
<td>Optimize Code and Improve Aesthetics</td>
<td>24 days</td>
<td>Mar 19</td>
<td>Apr 19</td>
</tr>
</tbody>
</table>

CDR: Mar 18
FPR: Apr 19
Thank you

Questions?