UMassAmherst
The Commonwealth's Flagship Campus

Mid-year Design Review:

Intelligent Screw Organizer - ISO

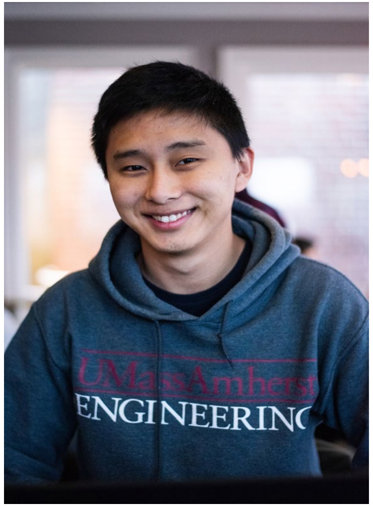
Team 20

Jordon Balskus, Jordan Gyaltsen, Andrew McGrath, Rajesh Shahi

December 10, 2018

Advisor: Professor Eslami

ISO Team:



Jordan Gyaltsen
CSE

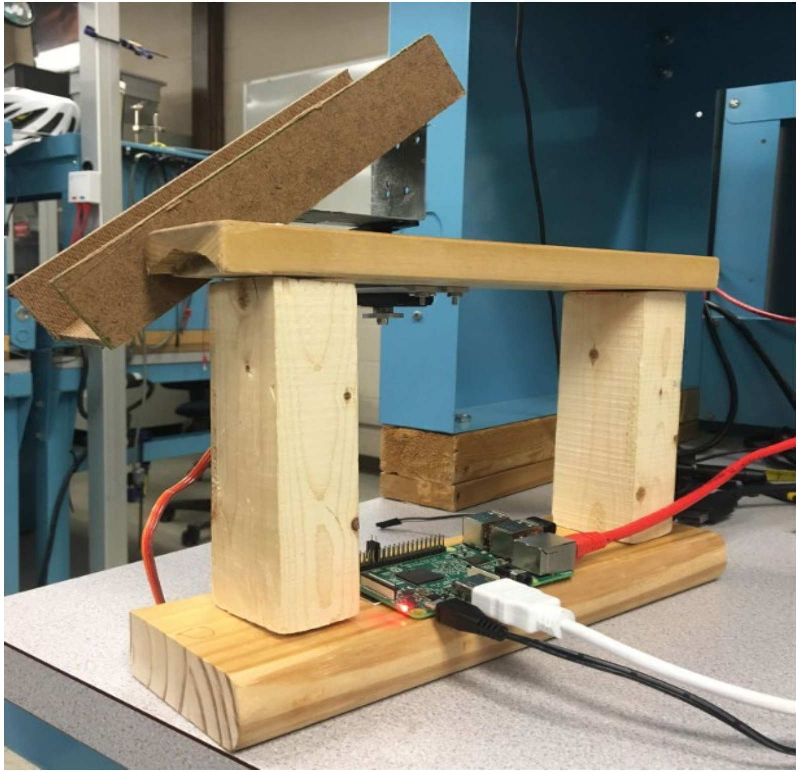
Jordon Balskus
CSE

Andrew McGrath
CSE

Rajesh Shahi
CSE

Outline

1. Problem Statement
2. System Specifications
3. Design System
4. CDR Deliverables
5. Demo



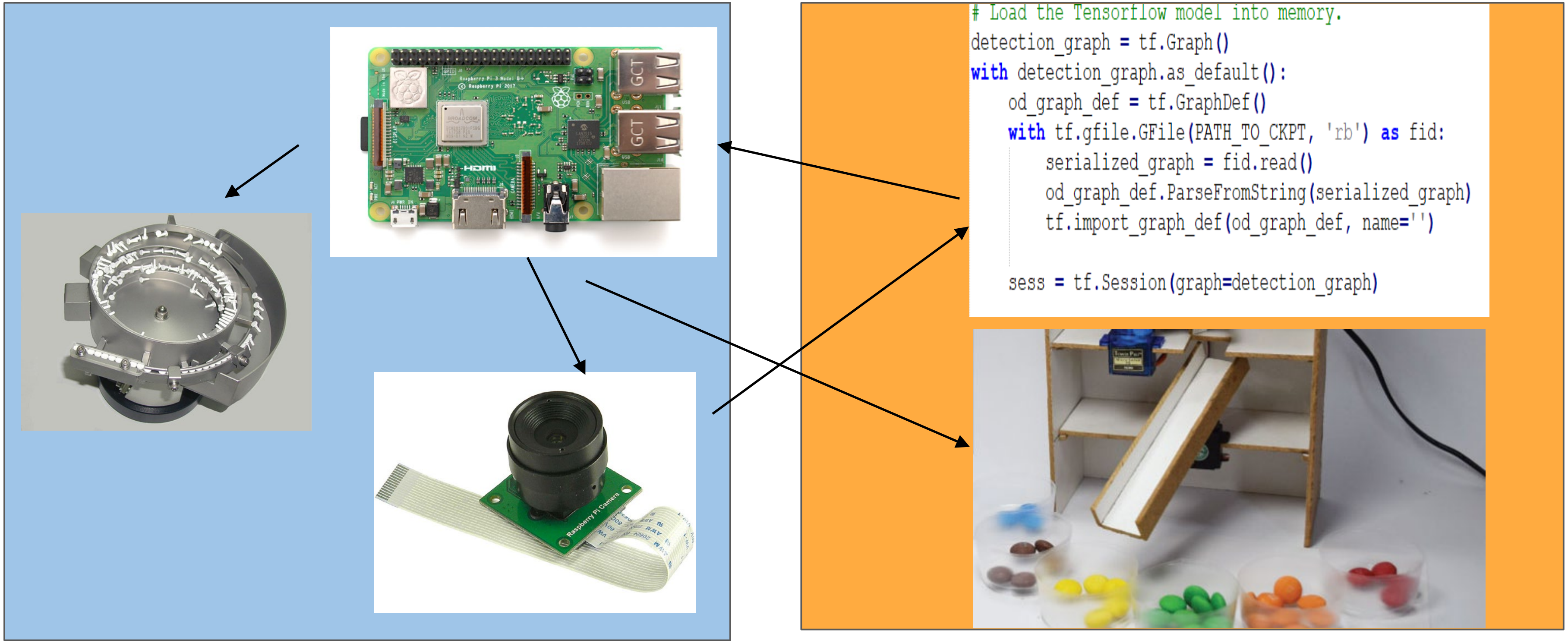
The Problem

- 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

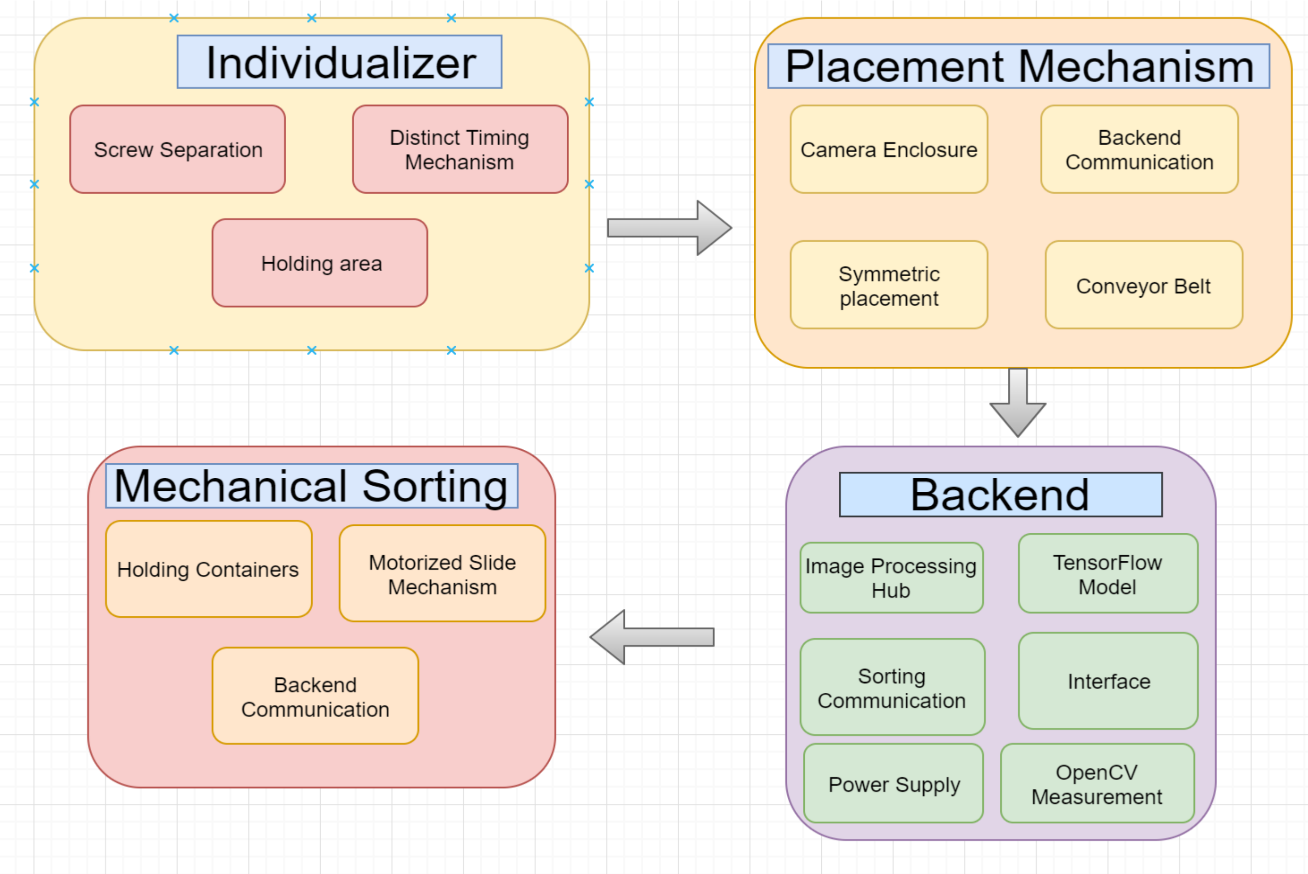
System Topology



```
# Load the Tensorflow model into memory.
detection_graph = tf.Graph()
with detection_graph.as_default():
    od_graph_def = tf.GraphDef()
    with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
        serialized_graph = fid.read()
        od_graph_def.ParseFromString(serialized_graph)
        tf.import_graph_def(od_graph_def, name='')

    sess = tf.Session(graph=detection_graph)
```

Block Diagram



Individualizer

Screw Separation

Distinct Timing Mechanism

Holding area

Placement Mechanism

Camera Enclosure

Backend Communication

Symmetric placement

Conveyor Belt

Backend

Image Processing Hub

TensorFlow Model

Sorting Communication

Interface

Power Supply

OpenCV Measurement

Mechanical Sorting

Holding Containers

Motorized Slide Mechanism

Backend Communication

Mechanical Requirements & Specifications

- Screw length between 0.5 inch - 3 inch
  - limit scope of mechanical complications
  - works for most screws in machine shop
- Sort between 5 different types of screws in a session
  - Excess screws placed together then resorted
- Sorter will be able to turn and deposit screw into correct bin within 6 seconds of identification

Image Processing Requirements & Specifications

- Real Time Interface Communication
  - Screws identified once individualized and scanned by Raspberry Pi camera (scanning at least 1 FPS).
  - Able to measure length and width of based on the intrinsic and extrinsic properties of the camera.
  - Able to detect headtype using TensorFlow neural network.

MDR Deliverables

- Final screw placement
  - Demonstrate the working of the mechanical system to complete sorting process.
  - After being scanned and identified, a few screws will be placed into a designated bins.
- Preliminary Image Processing
  - Have basic system that takes image of screw and imports it to our software suite
  - Demonstrate that we are able to detect screw length and diameter

MDR Deliverables

✓ Final screw placement

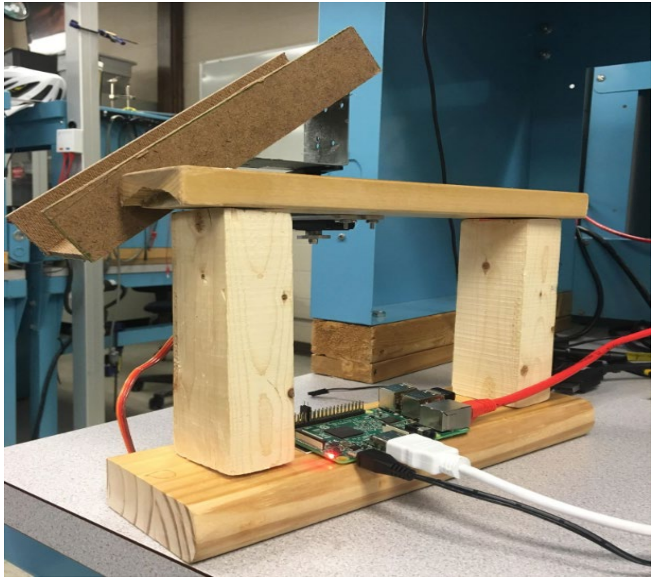
- Demonstrate the working of the mechanical system to complete sorting process.
- After being scanned and identified, a few screws will be placed into a designated bins.

✓ Preliminary Image Processing

- Have basic system that takes image of screw and imports it to our software suite
- Demonstrate that we are able to detect screw length and diameter

Mounted Mechanical Slide : Lead Rajesh

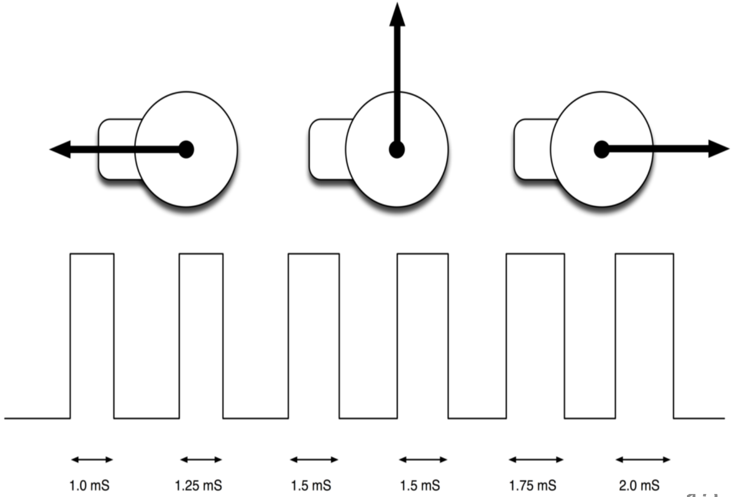
- Create physical stand to sorting slide to connect to.
- Securely adhere servo to the slide to prevent problems with weight shifting.
- Connect wires from raspberry pi to servo to allow for communication through the GPIO Pins.



Final Screw Placement : Lead Jordan G.

- Position of the servo motor is set by the length of a pulse.
  - The servo expects to receive a pulse roughly every 20 milliseconds. If pulse high for 1 millisecond, then servo angle will be zero. If high for 1.5 milliseconds, then 90°, if 2 milliseconds then 180°.
  - Servo range of rotation is 180°
  - To accomodate six positions for six bins, each bin designated a location of X(36°), where X is the bin number assigned to screw type.

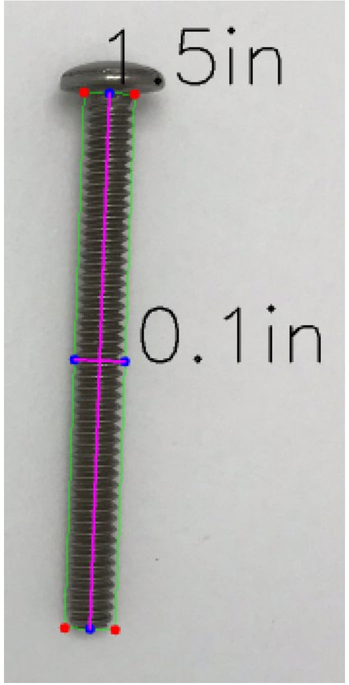
  Based on bin number, X, pulse signaled for 1 + X(.2 seconds)



Screw Characteristic Detector (Length/Width): Lead Andrew

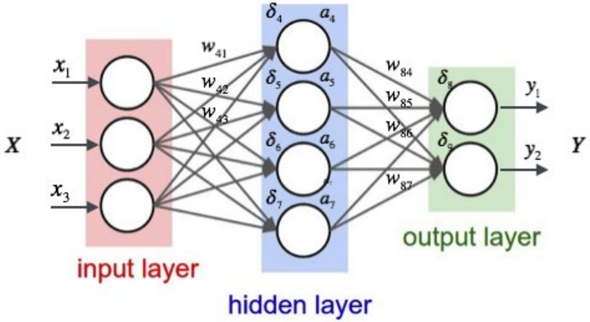
- Uses canny image detection in OpenCV to dilate and erode the image removing unwanted objects.

- Calculate length & width of each individual pixel.
- Use this value to calculate size of screw object
- Values can be stored in an array to compare with screws seen in the future.



Screw Characteristic Detector (TensorFlow): Lead Jordon B.

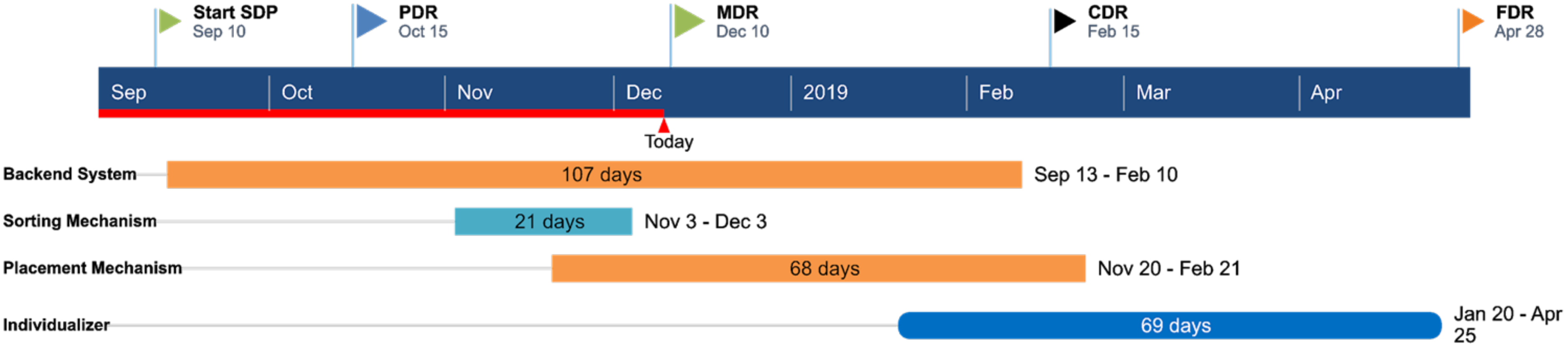
- Build Tensorflow model to demonstrate screw detection with a neural network
  - Utilizes machine learning
  - 5 Screw model with over 30 pictures of each screw taken (200+ photos)
  - Model trained until error is below 0.05
  - Can be used to detect a static number of screw characteristics



Proposed CDR Deliverables

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

Gantt Chart


Start SDP
Sep 10
PDR
Oct 15
MDR
Dec 10
CDR
Feb 15
FDR
Apr 28
Sep
Oct
Nov
Dec
2019
Feb
Mar
Apr
Today
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

DEMO

Thank you

Questions?