Bin There Dump That

Jessica Lew, Lauren Tan, Tate Arevalo
Team C6

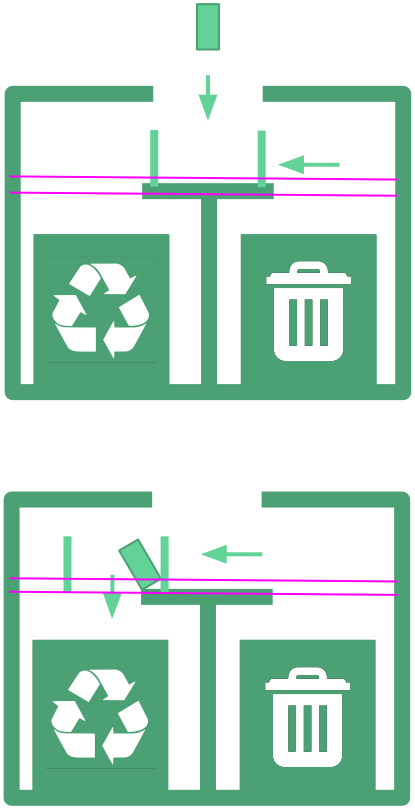
Use Case - AI Trash Can

Problem: People don't recycle

Scope: Automatic trash can for sorting recyclables vs. non-recyclables one item at a time

- Target audience: CMU campus
- Sensors: camera, inductive, IR, etc.
- ECE Areas: software, signals, hardware

"31% of people recycle a recyclable item"
-Covanta



Requirements - Classification

- 1) Recyclables:
    - Tin/Aluminum cans, metals
    - Newspapers, paper, cardboard
    - Plastic bottles
    - Glass

  Most commonly recycled items

  2) Non-recyclables
- Recyclable Accuracy Rate: >31%
- Non-Recyclable Accuracy Rate: >75%
    - Contamination Rate: <25%

Human error estimated by EPA and Covanta

Technical Challenges:

- Distinguishing visually similar objects
- Acquiring large amounts of data to train model
- Training model and tuning parameters

Requirements - Mechanism

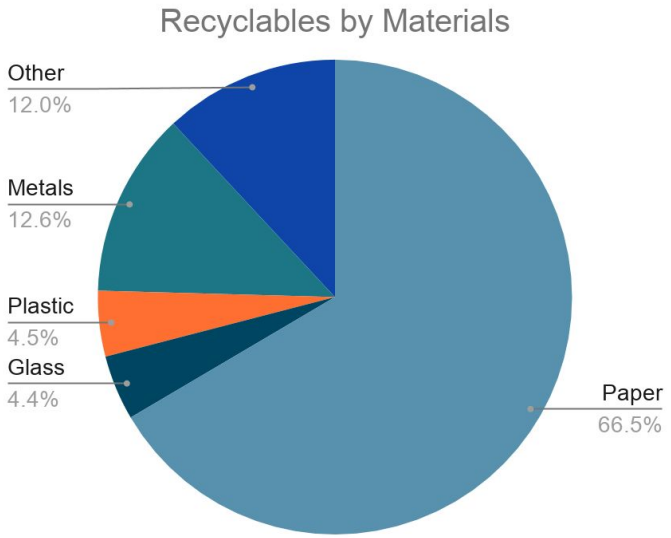
- Reliably move trash into correct bin
  - 99% Accuracy
- Support small to medium-sized objects
  - Size: 10x10x5 inches
  - Fits common items on campus (plastic water bottle, etc.)
- Latency: 2-3 seconds for sorting each item

Technical Challenges

- Handling variety of objects (i.e. small or flat objects)

Requirements - Sensor Input

- Sensors
  - Camera for general classification
  - Motion sensor for detecting item
  - Inductive sensor for metals
  - Capacitive sensor for plastic, glass, paper
  - IR/LDR sensor for plastic, glass
- Paper accuracy: +14%

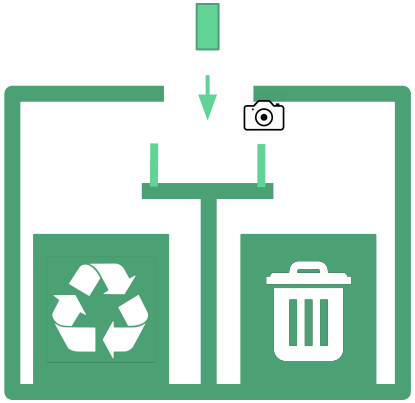


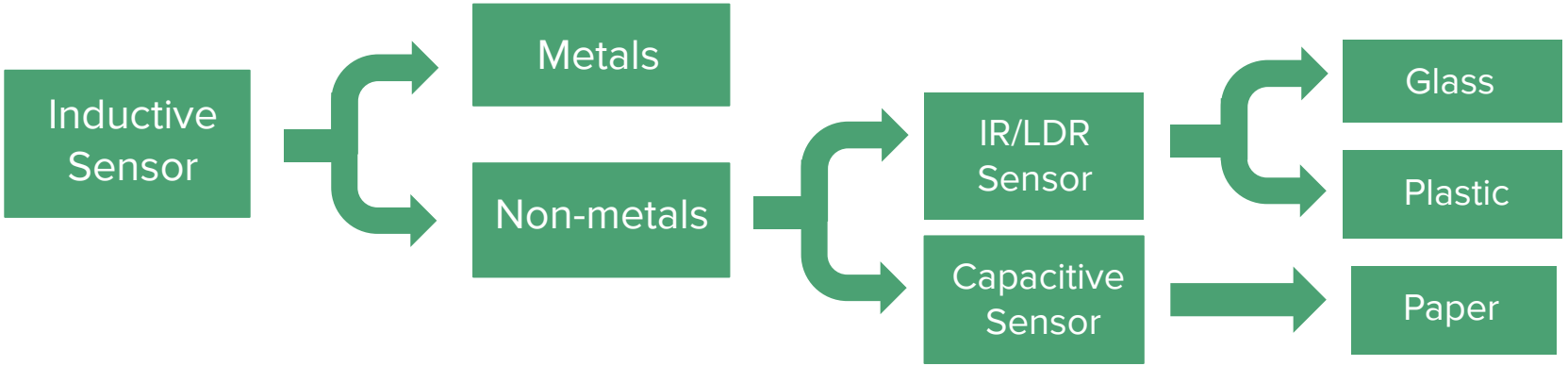
Technical Challenges

- Coordinating multiple sensor inputs, tuning parameters
- Filtering out background noise
- Sensor placement (range requirements)

Solution Approach - Sensor Input

- Embed sensors into sorting box
  - Camera next to the trash chute opening
  - Inductive sensor under plexiglass platform
  - IR and LDR sensors attached to the side of the box





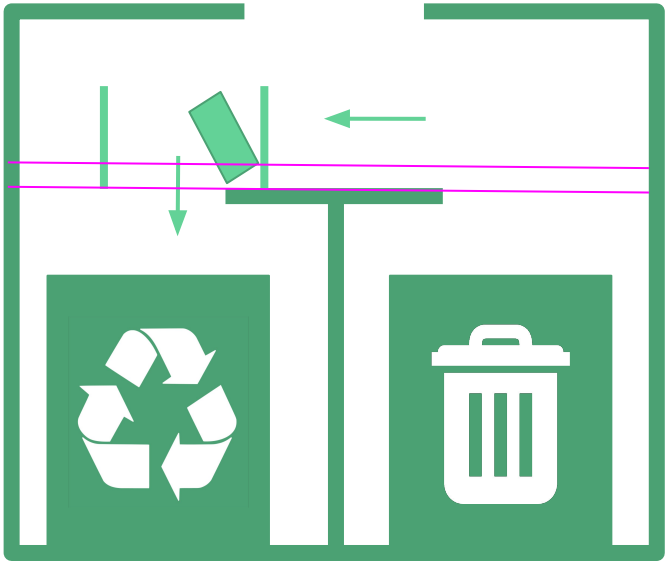
Solution Approach - Classification

Deep learning to classify garbage

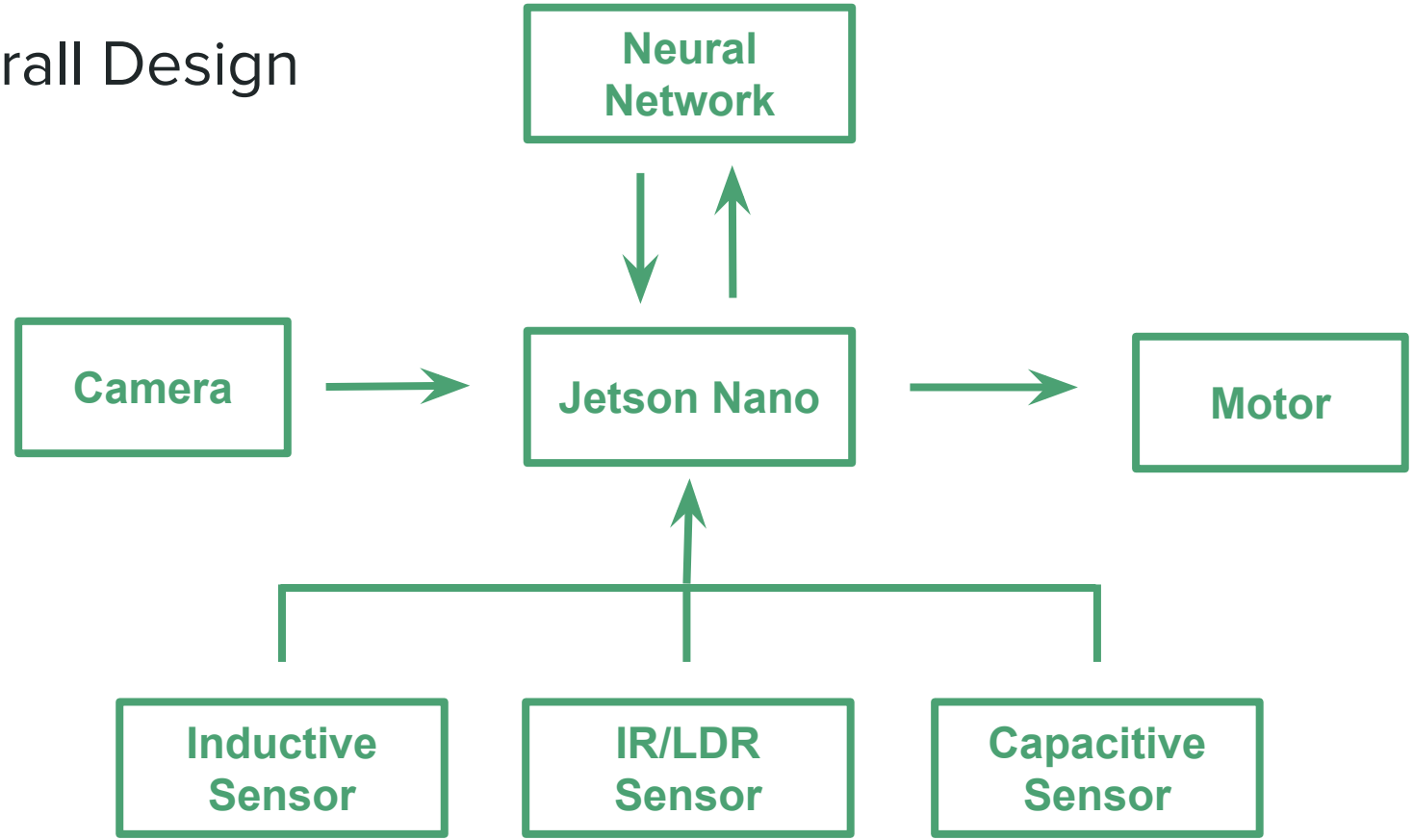
- Pytorch/TensorFlow to train classifier
  - Use existing dataset
  - Use pre-existing model
    - Ex: 95% accuracy with ResNet
- Nvidia Jetson Nano

Solution Approach - Mechanism

- Linear motor actuator to move sorting box and push trash into bin
- Lower platform is stationary
- Hardware
  - Nema 17 Stepper Motor
  - DRV8825 Motor Driver
  - Controlled by Jetson Nano



Overall Design



Testing, Metrics, Verification

- Testing strategy
  - Verify classifier/mechanism with different items (different materials, sizes, transparencies)
  - Verify classifier at different angles
  - User testing
- Latency
  - Time delay between placing item and physically sorting item
  - Overall 2-3 seconds
- Accuracy
  - Classifier
    - Recyclable accuracy >31%
    - Non-Recyclable accuracy >75%
  - Mechanism: 99%

Tasks and Division of Labor

- Classifier
  - Data collection: everyone
  - Selecting and building the classification model: Lauren, Jessica
  - Training the model: Lauren, Jessica
  - Testing the model/parameter tuning: Lauren
- Mechanism/Hardware
  - Hardware design (mechanism and sensors): Jessica, Tate
  - Building mechanism/infrastructure: Tate
  - Integrating sensors with Jetson Nano: Tate

	TASKS	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Finals
		2/1	2/8	2/15	2/22	3/1	3/8	3/15	3/22	3/29	4/5	4/12	4/19	4/26	5/3	5/10
	Milestones				Proposal Presentation		Design Presentation	Design Report				Interim Demo			Final Presentation	Final Report/Demo
1	**Mechanism**															
	Finalize Mechanism Design				JT	JT										
	Order Mechanism Parts					T										
	Build Mechanism									T						
	Connect motors to Jetson Nano								T							
	Build sorting box						J									
	Test Mechanism										T					
2	**Sensors**															
	Circuit Schematics					J										
	Order Parts				L	L										
	Connect sensors to Jetson Nano						J									
	Collect Data						All	All	All							
	Test Sensors							T	T	T						
3	**Classifier**															
	Find datasets					LJ										
	Research Models					LJ	LJ									
	Build Model							LJ								
	Train model for images								LJ							
	Manually train model									LJ						
	Test classifier (images)										LJ					
	Manually test classifer (images)										LJ	LJ				
4	**Integration & Testing**															
	Integrate camera with box								T							
	Integrate motion sensor								J							
	Integrate sensor with image classifier								LT	L						
	Test classifier (images, sensors)											L				
	Test overall (classifier, mechanism)											All				
	Slack time												All	All	All	All