Group 4: Autonomous Sorting of Trash Objects Based on Structure using Gazebo Robot Simulation

Dhyakesh Soundararajan, Saffat Ahmed, Kevan Mehta
Introduction

Goal: Create autonomous robot that can detect and sort various trash objects

Reason: Assisting with proper disposal and sorting of waste

Objectives:

- Detecting and Differentiating between various objects
  - DOPE - Deep Object Pose Estimation
  - Image Classification with Tensorflow
- Consistent and Accurate Grasping of objects
  - MoveIt Motion Planning Framework
Related Work - DOPE

Pre-trained Models for Pose Estimation

- Utilized pre-trained models for initial pose estimation.
- Requires all objects to be known ahead of time
  - Useful for small dataset testing in simulation
  - Potentially not as useful in real world circumstance, too many shapes and objects to train all of them

ROS Implementation Challenges

- Struggled to get working on our systems. Missing certain files and libraries
- If implemented successfully, would require additional training for additional models if needed
- May attempt implementation again in future
## Related Work - Image Classification

### Model summary

<table>
<thead>
<tr>
<th>Model: &quot;sequential_8&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Layer (type)</strong></td>
</tr>
<tr>
<td>---------------------</td>
</tr>
<tr>
<td>conv2d_25 (Conv2D)</td>
</tr>
<tr>
<td>max_pooling2d_25 (MaxPooling2D)</td>
</tr>
<tr>
<td>conv2d_26 (Conv2D)</td>
</tr>
<tr>
<td>max_pooling2d_26 (MaxPooling2D)</td>
</tr>
<tr>
<td>conv2d_27 (Conv2D)</td>
</tr>
<tr>
<td>max_pooling2d_27 (MaxPooling2D)</td>
</tr>
<tr>
<td>flatten_8 (Flatten)</td>
</tr>
<tr>
<td>dense_16 (Dense)</td>
</tr>
<tr>
<td>dense_17 (Dense)</td>
</tr>
</tbody>
</table>

---

Total params: 11169218 (42.61 MB)
Trainable params: 11169218 (42.61 MB)
Non-trainable params: 0 (0.00 Byte)
Implementation

- Gazebo for Simulations and Collecting Classification Images
- MoveIt for Grasping
- Classification using Tensorflow
- Current Model trained on boxes and cans
Collecting Sample Input Images Demo

https://youtu.be/Qx4bh3ISySg
Demonstration of Current Implementation

https://youtu.be/W6Qgg94Xr7E
Experiments and Data

Dataset stats:

Train:
- Box: 50
- Can: 51

Val:
- Box: 15
- Can: 15

Test:
- Box: 10
- Can: 10
Experiments and Data

Training and Validation accuracy

Graph (5 epochs)
Analysis of Results/Limitations

Testing Data sample

To evaluate the model
Analysis of Results/Limitations

Confusion matrix
Box [[7, 3], [0, 10]]
Accuracy: 0.85
Testing model on the fetch robot camera

```python
model.save('custom_image_classifier_box_can.h5')
# Provide the path to the image you want to classify
image_path = '/content/image_158.png'

# Make a prediction
class_index, class_confidence = classify_image(image_path)

# Print the result
print(f"Predicted Class Index: {class_index}"),
print(f"Confidence: {class_confidence}"))

1/1 [==============================] - 0s 131ms/step
Predicted Class Index: 0
Confidence: 1.0
```
Future Work

Improve Object Recognition

- Enhance the robot's ability to recognize diverse trash items.
- Handle deformable or complex objects effectively.
- Distinguish between different materials for precise sorting.

Improve Grasping

- Implement models with defined potential grasps
- Using CNN’s for grasp learning
- Focus on top-down grasping instead of 6D grasping
References
