Gesture Recognition for Initiating Human-to-Robot Handovers

Jun Kwan, Chinkye Tan and Akansel Cosgun
Problem Statement

- Object handovers can happen in two directions
  - Robot-to-Human
    - The robot delivers a requested object to a human
  - Human-to-Robot
    - The robot acquires an object from a human
- Develop a system to recognize the act of the human handing an object over to the robot
Overview

- Extract features relevant to the task
  - Object detection
  - Keypoints detection
  - Head pose estimation

- Train a classifier
  - Detect the existence of a handover gesture based on the extracted features
Object Detection Module

- Faster R-CNN with Detectron2 engine
  - Proposed by Ren et al. [1]

- Presence and location of object extracted
  - Bounding box around the object returned if object is detected
  - x, y, width and height

Body Keypoints Detection Module

- Keypoints R-CNN with Detectron2 engine
  - Proposed by He et al. [2]

- Coordinates of various joints in the body extracted
  - 11 keypoints are selected (upper body)

- Centralized around the detected object
  - Object centric frame

Head Pose Estimation Module

- Multi-loss Resnet50 architecture
  - Proposed by Ruiz et al. [3]

- Multiple losses are designated for different Euler angles

- Multi-task cascaded convolution networks (MTCNN) [4]
  - Face detection

Multi-layer Perceptron

- An input layer, four hidden layers, and an output layer

- Feature vector
  - The 1st parameter
    - The presence of an object in the scene
  - The 2nd to 23rd parameters
    - Pixel coordinates of upper body keypoints
  - The 24th to 26th parameters
    - The yaw, pitch and roll of the head orientation
Custom Dataset

- We designed a custom dataset to train the multi-layer perceptron

- A total of 25 videos were recorded in various environments
  - Containing a total of 2506 images

- Each image was labelled ‘1’ denoting a handover scenario or ‘0’ otherwise
Custom Dataset
Skeleton Images

- ResNet50 used instead of Multi-layer Perceptron
- Features are placed on a black image and fed into a CNN
End-to-End Method

- Standard CNN used as a baseline
- Raw RGB images used as input to the CNN

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>End-to-end (Alexnet)</td>
<td>50.0</td>
</tr>
<tr>
<td>End-to-end (Resnet50)</td>
<td>89.4</td>
</tr>
<tr>
<td>CNN on skeleton images</td>
<td>83.3</td>
</tr>
<tr>
<td>MLP (absolute pixels)</td>
<td>90.1</td>
</tr>
<tr>
<td>MLP (relative to object)</td>
<td>90.6</td>
</tr>
</tbody>
</table>
Video Demonstration
Discussion

- The system with object centric frame is more robust
  - Absolute position of human no longer taken into account

- MLP system outperforms skeleton image CNN system
  - MLP receives features directly
  - CNN has to decipher features from the skeleton images
Future Work

- Temporal information to be included
- Use of other communication cues, e.g. verbal and gaze
- Dataset to be more robust
- Ablation study of each module
Thank You