Problem Statement
- Estimate 6D pose of query object in cluttered scene
  - Inputs: RGB-D image of scene, RGB-D image of object
  - Output: X, Y, Z, Roll, Pitch, Yaw of object
- Train model on simulated dataset and evaluate on real-world dataset

Motivation
- Real-world data for deep learning approaches to robotic object manipulation: difficult & expensive to obtain
- Deep neural networks trained on simulated robotics tasks may not transfer directly to real-world tasks
- Domain randomization in simulation improves transferability

Simulated Data
- 4000 RGB-D scenes of objects on a table simulated in Unity3D
- 58800 examples of an object’s pose in randomized scene
- 55 objects and their RGB-D images

Real Data
- Shelf Dataset from Amazon Picking Challenge
- 2130 examples of an object’s pose in a cluttered scene of a shelf
- 27 objects and their RGB-D images
- Shelf Dataset provides object-to-world transformation matrix, and camera extrinsic matrix, which we use to extract the 6D pose of object from the camera’s perspective (same as Sim Dataset)

ResNet Encodings
- Transfer learning with fixed CNN as fixed feature extractor
  - ResNet-18 pre-trained on ImageNet
  - Last hidden layer for encodings
  - Depth images triplicated as input

Model A
- Baseline: hierarchical combination
- Architecture selection MSE losses:
  - Train 4.83, train-dev: 5.56

Model B
- Immediate combination, pass through extra layer to output
- Architecture selection MSE losses:
  - Train 1.76, train-dev: 6.3

Model C
- Hierarchical combination, but pass scene through extra layer
- Architecture selection MSE losses:
  - Train 4.91, train-dev: 5.52

Model D
- Hierarchical combination, pass through extra layer to output
- Architecture selection MSE losses:
  - Train 4.15, train-dev: 5.74

Model E
- Immediate combination, pass through three extra layers to output
- Architecture selection MSE losses:
  - Train 2.15, train-dev: 6.10

Models

Experiments
- Architecture Selection:
  - Trained for 100 epochs on 3.5% of dataset, minibatch size 10
  - MSE loss as sum of position and orientation losses
- Model B achieved lowest bias
- Train Set Size Analysis:
  - Trained model B for 70 epochs
  - Changed minibatch size to 40
  - Losses with 7% of dataset:
    - Train 0.44, train-dev 6.66
  - Losses with full dataset:
    - Train 2.48, train-dev 6.91, dev 9.78
    - Moderate variance reduction
  - L2 Penalty Hyperparameter Selection:
    - Trained for 50 epochs on 10% of dataset
    - A = 10^-10: train 5.14, train-dev 6.58
    - A = 10^-1: train 4.32, train-dev 6.97
    - A = 10^-2: train 1.74, train-dev 6.97
- Full Dataset Training:
  - Trained model B for 70 epochs
  - Losses: train 4.94, train-dev 5.27, dev 6.33, test 8.13
  - Position loss:
    - Train-dev 0.85 (RMSE 0.92 m), test 2.2 (RMSE 1.47 m)
  - Orientation loss:
    - Train-dev 4.4 (RMSE 120 deg), test 6.0 (RMSE 140 deg)
  - L2 penalty reduces sim2real mismatch

Challenges and Next Steps
- L2 penalty search with full dataset training needed
- Sim2real transfer may require further domain randomization
- Final layer of ResNet may not provide appropriate encoding for pose estimation task
- Providing single viewpoint of query object may be too difficult, especially for orientation estimation
- End-to-end pose estimation may require more complex architecture or multi-stage approach, as in (3)

References