Exploring CNN-based Feature Transfer for Robot Affordances

Abhineet Jain, Oliver Kroemer,
Gaurav S. Sukhatme

USC Viterbi
School of Engineering

USC Robotic Embedded Systems Laboratory

Motivation

- Robots need to learn affordances of objects for manipulation
- Suitable features are needed for predicting affordances
- Deep neural networks can learn features, but they require a lot of input data
- Data is limited for new affordances

What do we explore?

- We use Transfer Learning to explore:
  - Learning features from previous affordances with many samples
  - Using learned features to predict new affordances with fewer samples

Workflow

- Create affordance patch dataset
- Preprocessing images
- Train CNN
- Create dataset from robot
- Transfer learned CNN features

Affordance patch dataset

- Initial dataset:
  - Affordance Detection of Tool Parts from Geometric Features
    Austin Myers, Ching L. Teo, Cornelia Fermuller, Yiannis Aloimonos. ICRA 2015.
  - Comprised of objects affording 7 affordances (grasp, cut, contain, support, pound, scoop, wrap-grasp)
  - Each object has multiple RGB images, corresponding depth images, and affordance labels for every pixel in those images.

Affordance patch dataset

- Details of the original dataset

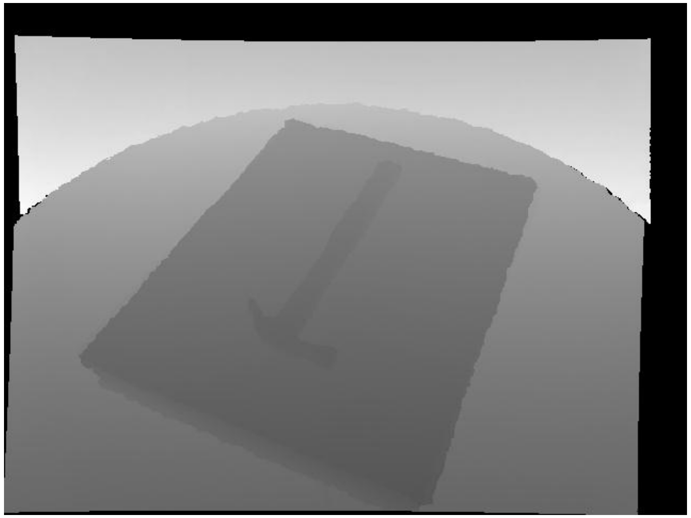
Object	Samples
Cut	**25**
Knife	12
Saw	3
Scissors	8
Shears	2
Scoop	**17**
Scoop	2
Spoon	10
Trowel	5

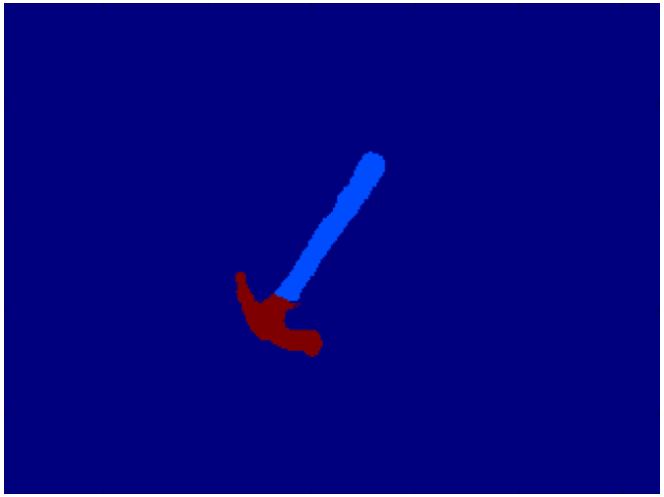
Object	Samples
Contain	**43**
Bowl	10
Cup	6
Ladle	5
Mug	20
Pot	2
Support	**10**
Shovel	2
Turner	8

Object	Samples
Pound	**10**
Hammer	4
Mallet	4
Tenderizer	2

Total	**105**

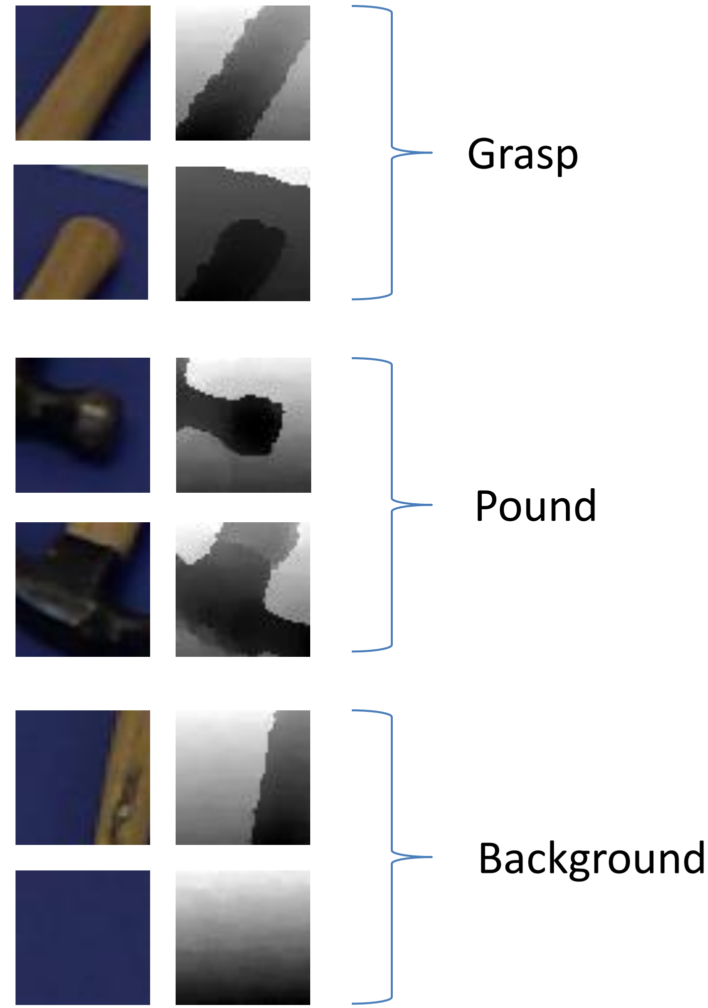
Affordance patch dataset





Affordance patch dataset

Image in the original dataset



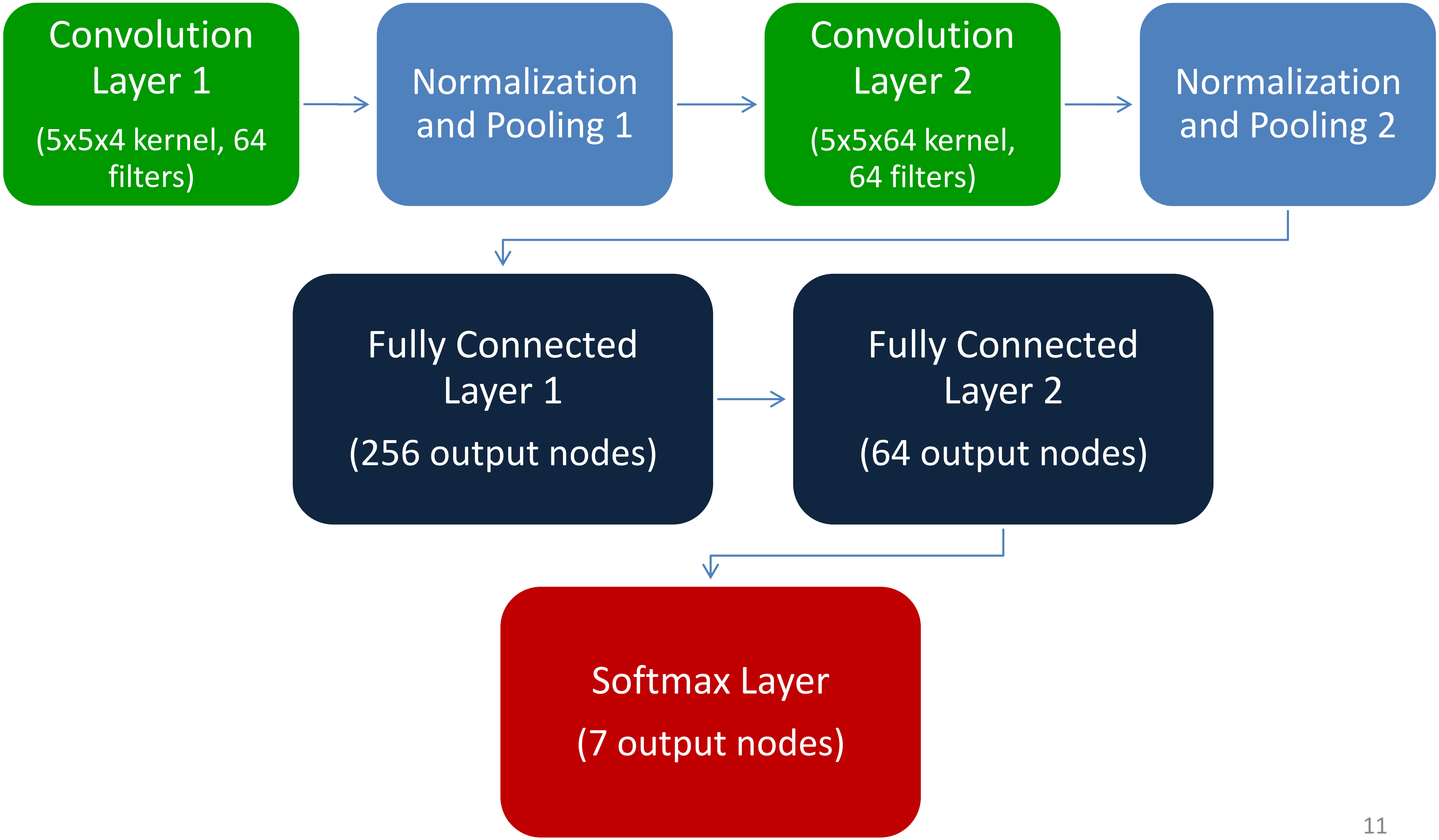
Affordance patch dataset

- For every affordance (7 categories + no affordance), 6250 patches. Total: 50,000
- 40,000 patches for training, and 10,000 for testing.
- We combined grasp and wrap-grasp categories
- Final labels: background, grasp, cut, scoop, contain, pound, support

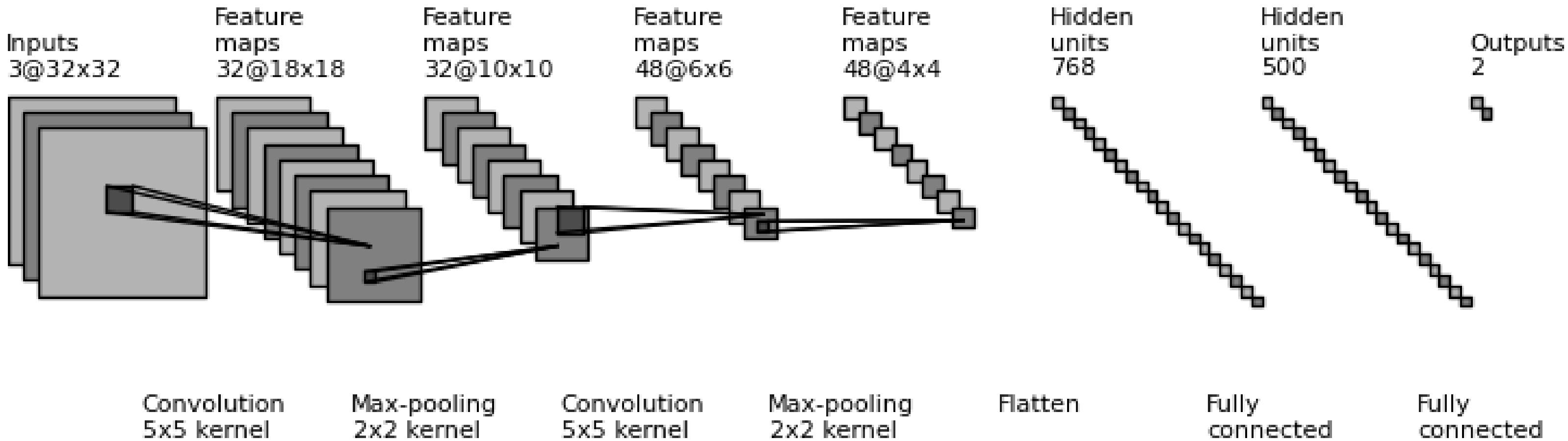
Preprocessing

- Whiten RGB image
- Whiten depth image
- Removes variations in intensity and lighting.
- The dataset becomes more generalized.
- Formally, the features are less correlated with each other
- All features have the same variance

Train CNN

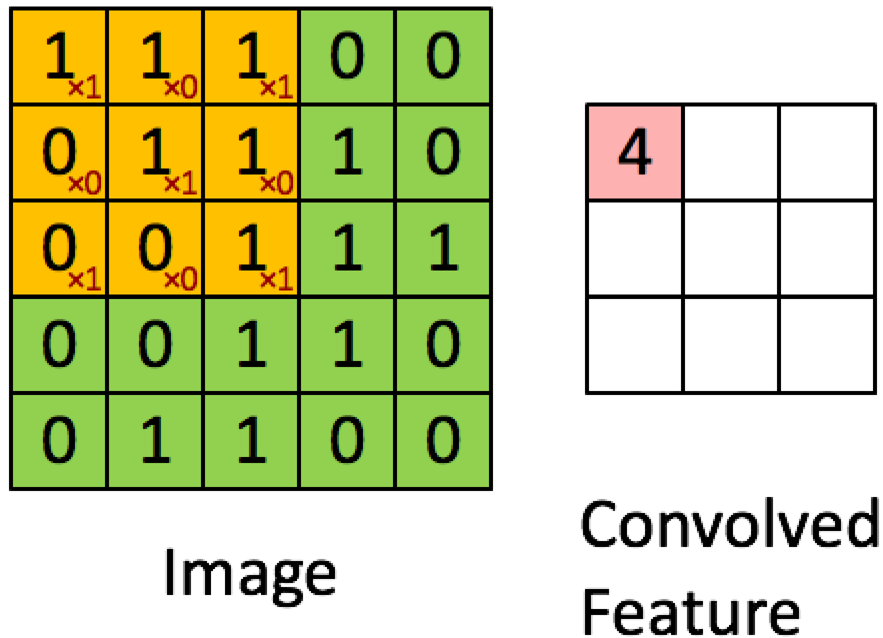

Convolution Layer 1
(5x5x4 kernel, 64 filters)
Normalization and Pooling 1
Convolution Layer 2
(5x5x64 kernel, 64 filters)
Normalization and Pooling 2
Fully Connected Layer 1
(256 output nodes)
Fully Connected Layer 2
(64 output nodes)
Softmax Layer
(7 output nodes)

Train CNN



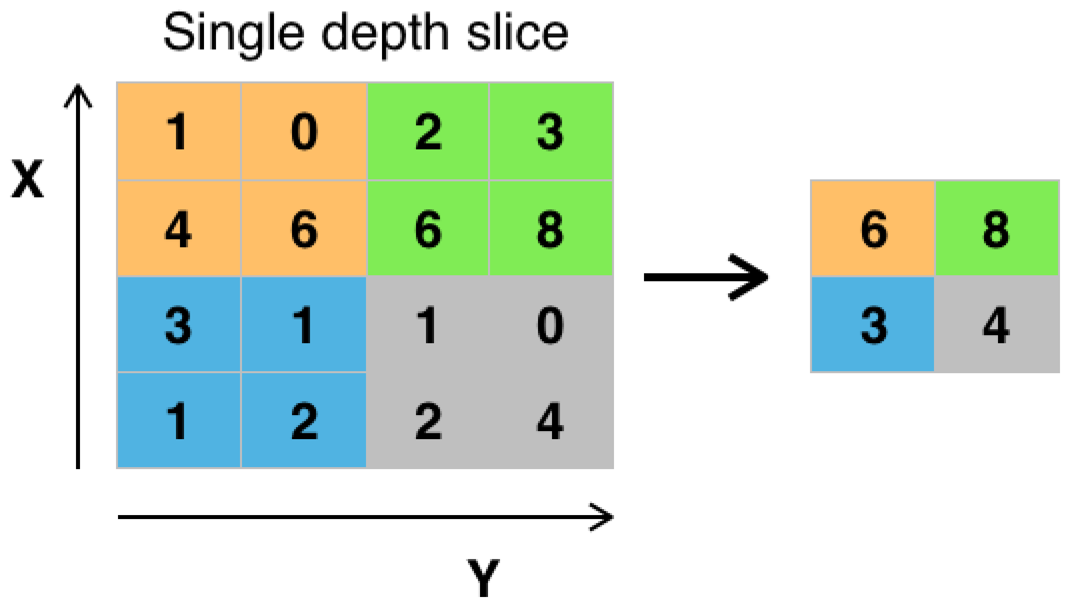
Sample CNN

Train CNN



2D Convolution

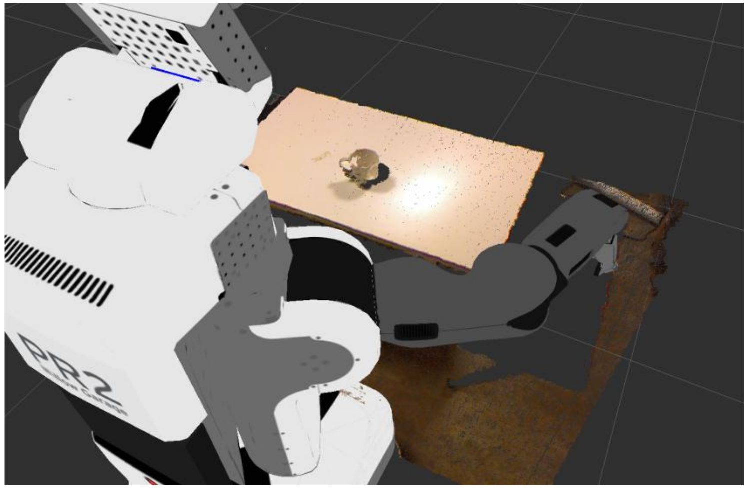
Train CNN

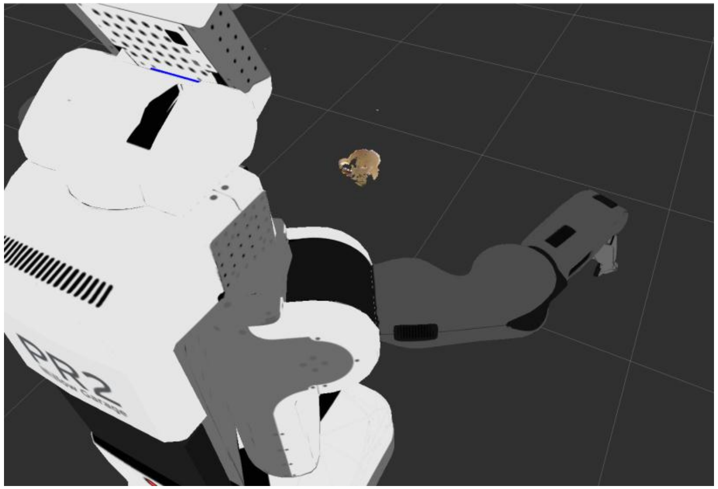


Max Pooling

Target dataset from robot

- Read Point Cloud data from Kinect
- Segment the table surface so that only object is visible



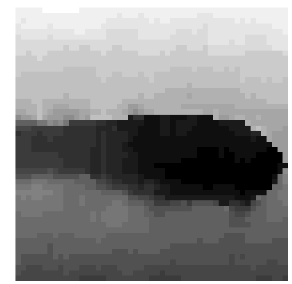


Target dataset from robot

- Read RGB and depth images from Kinect
- Select a random pixel, until a valid point on the Point Cloud is obtained

Target dataset from robot

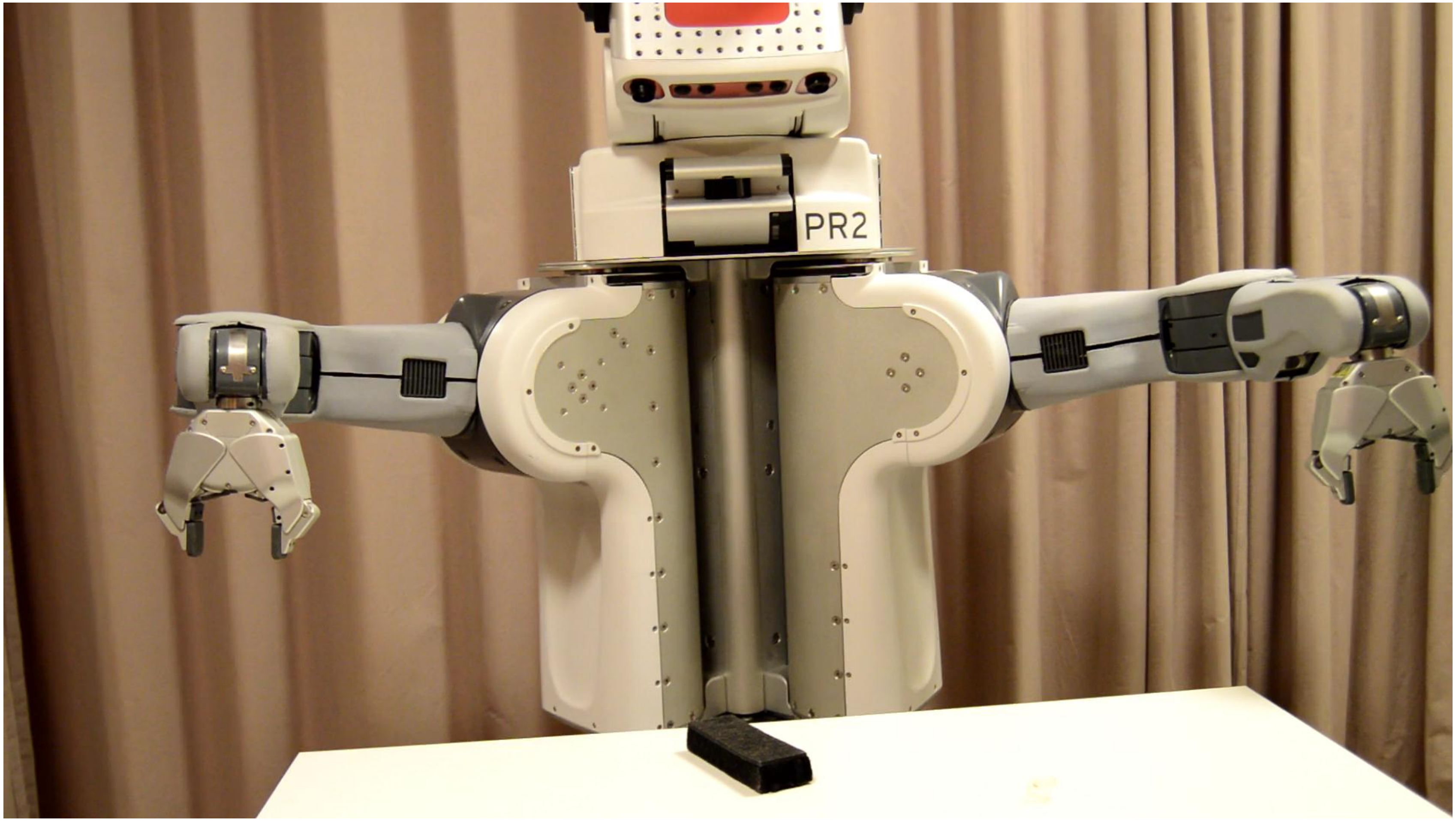
Image from Kinect



Target dataset from robot

- Set the orientation of the grasp at that point, in the direction of the normal at the point.
- Use the point and orientation to execute the grasp in 6 steps:
  orient, pre-grasp, grasp, lift, place, retract

Target dataset from robot



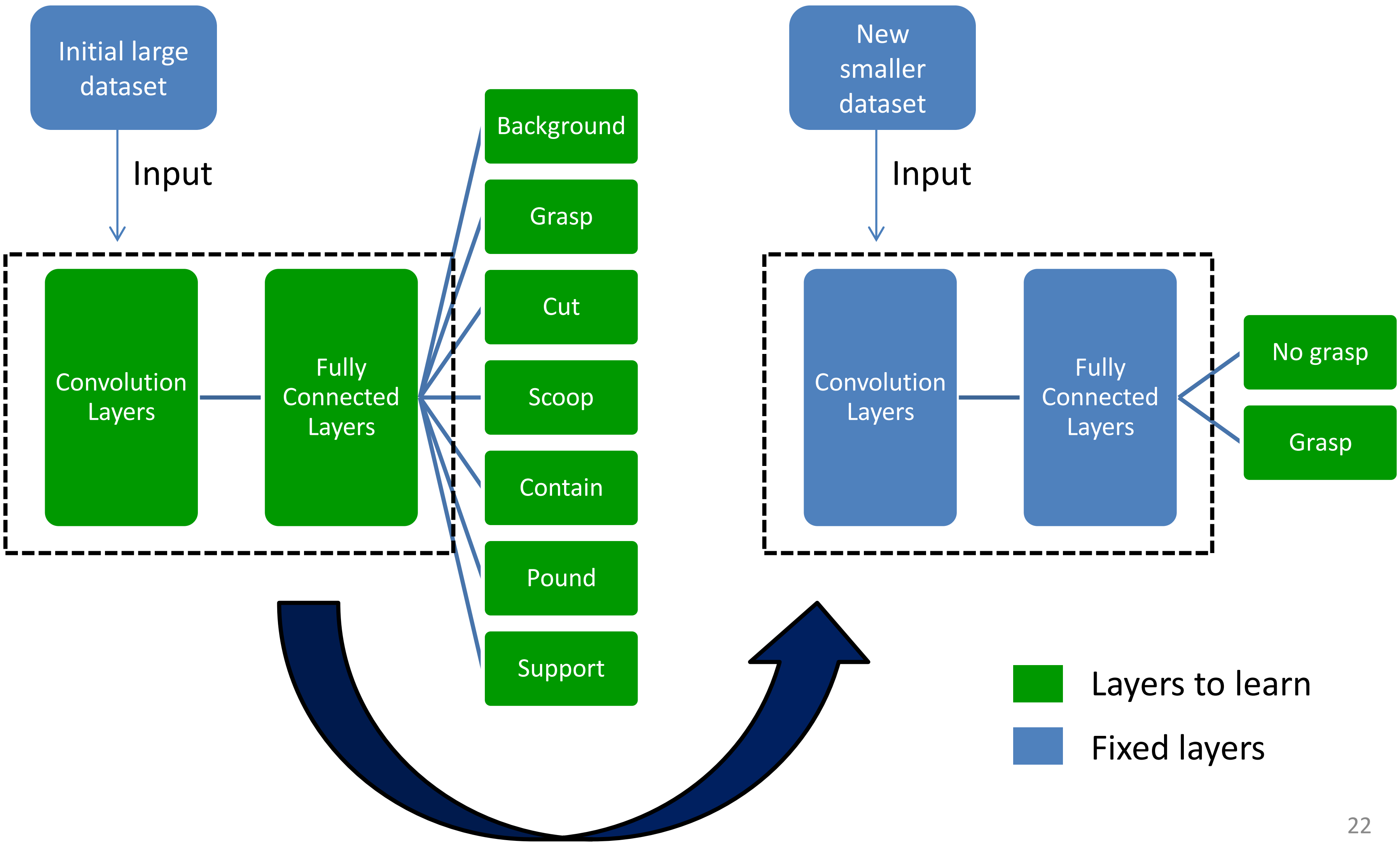
Target dataset from robot

- Manually label each attempt as successful or otherwise
- Each attempt yielded a single patch
- Total number of patches: 250
- Training set: 200
- Test set: 50

Transfer Learned CNN Features

- Use the trained CNN as a feature extractor
- Extract the values after the second last layer, Fully Connected Layer 2, outputs 64 features
- Now, each image is represented as a 64 feature vector
- Train a Softmax layer over these features to learn the robotic affordance – grasp.

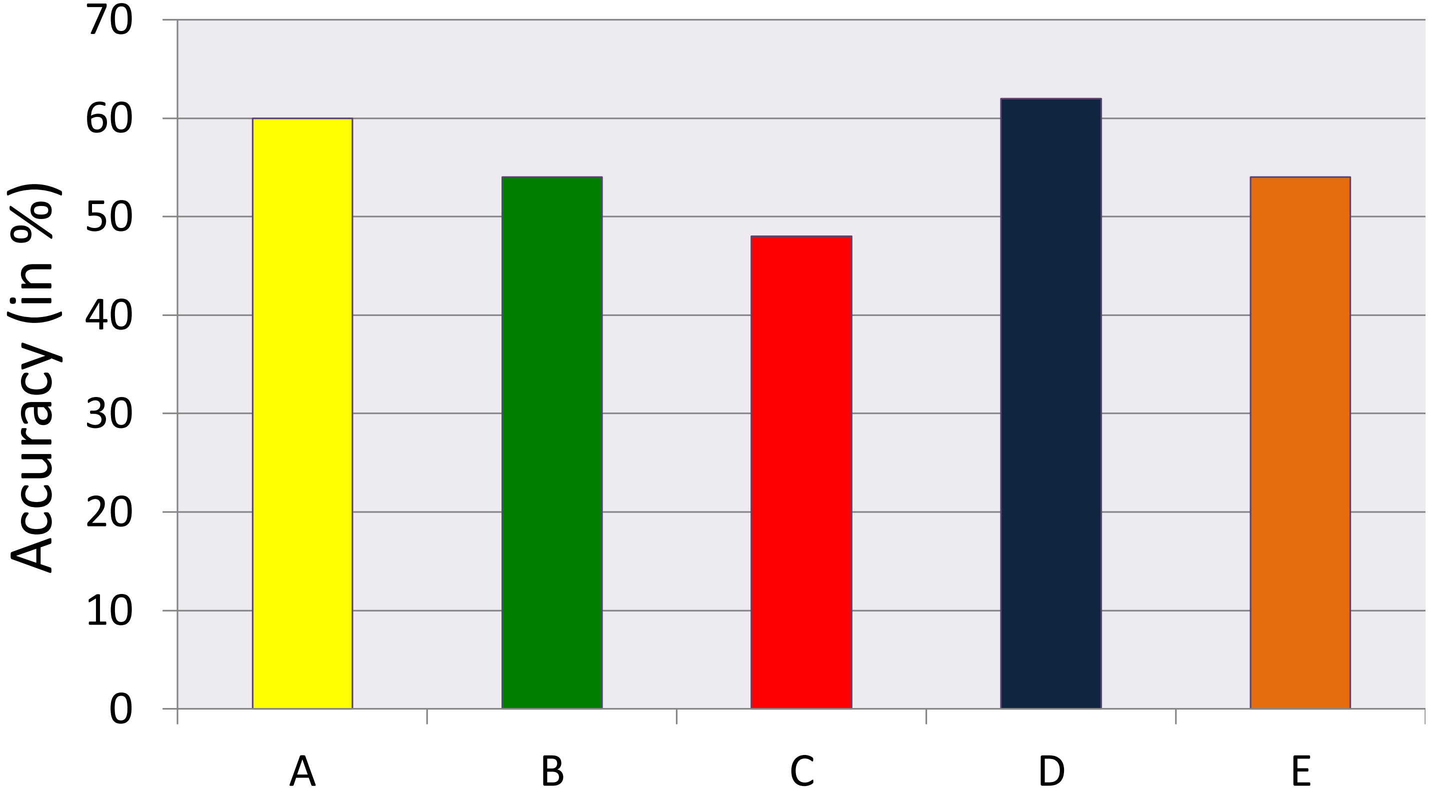
Transfer Learned CNN Features



Experiments

A. Train CNN on all 6 affordances,
   Retrain output layer on target affordance
B. Train CNN on 5 non-grasping source affordances,
   Retrain output layer on target affordance
C. Train CNN on grasping source affordance,
   Retrain output layer on target affordance
D. Train CNN on grasping source affordance,
   NO retraining on target affordance
E. NO training on source affordances,
   Train CNN on target affordance directly

Experiments



Things to do for ICRA

- Modify network to learn better features
- Add push affordance to the target dataset
- Add data for different scales
- Evaluate
  - Changing the number of source affordances
  - Changing the size of the patch
  - Exploring the features learned with different input affordances