Multiple Pregrasping Poses Prediction Using Combining DCNN and MDN

문성필

Introduction

- We address the problem of **multiple pregrasp poses regression** of a object based on **deep convolutional neural network (DCNN)**

- **Grasping is a multi-valued function** in the sense that a specific pose of an object can be grasped with different finger configurations

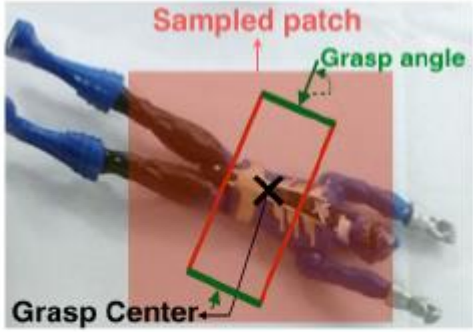
- Standard regression models fail in this case

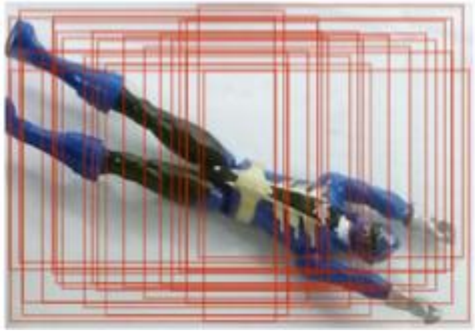
- A **pregrasp pose** as the configuration where closing the fingers until resistance is encountered can leads a proper grasp pose

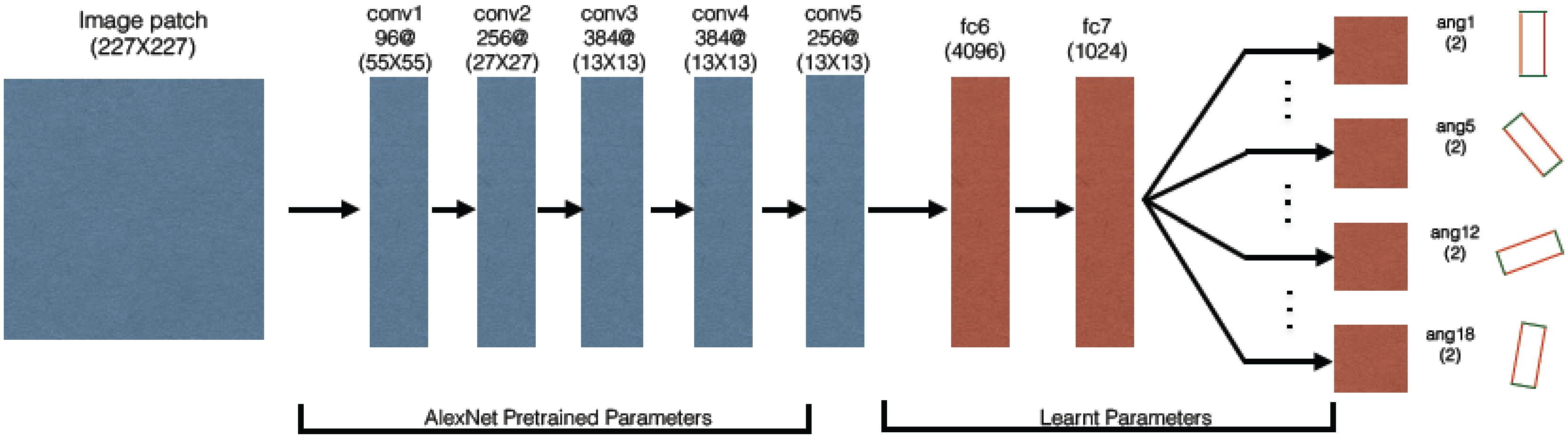
- In this work, a single RGB-D image is used to determine multiple 3D positions of three fingers

Related Works

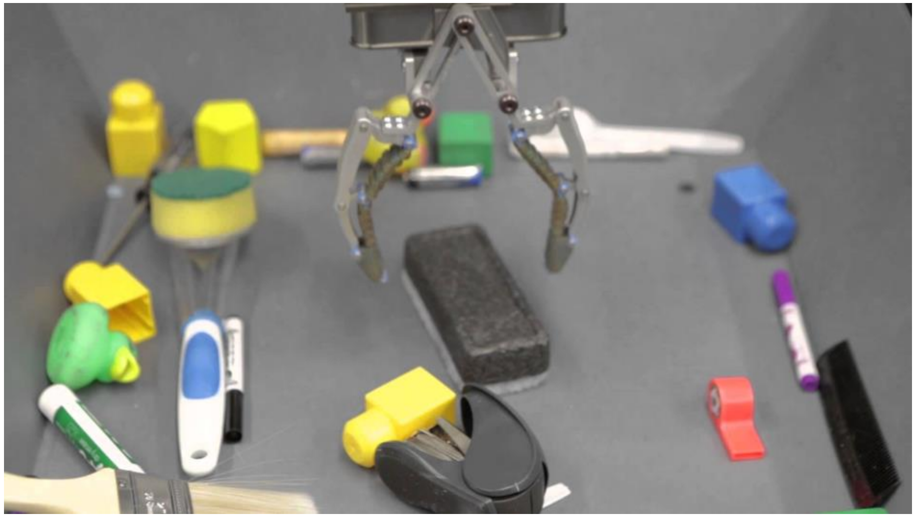
Supersizing Self-supervision: Learning to Grasp from 50K Tries and 700 Robot Hours
Abhinav Gupta, Carnegie Mellon University (2015)

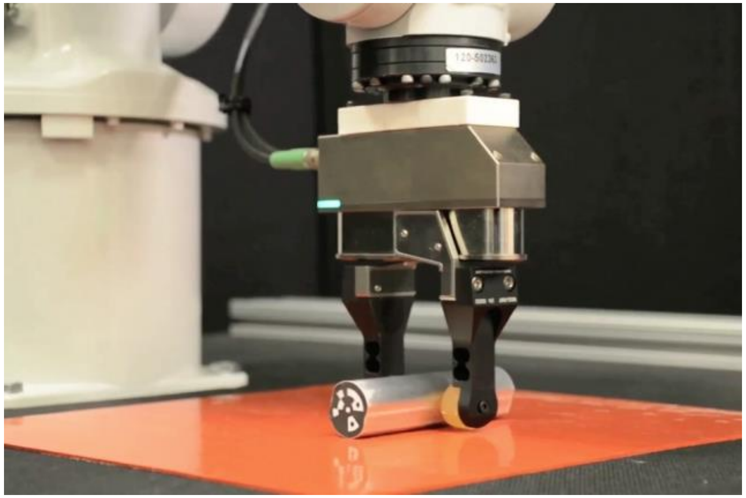






Related Works

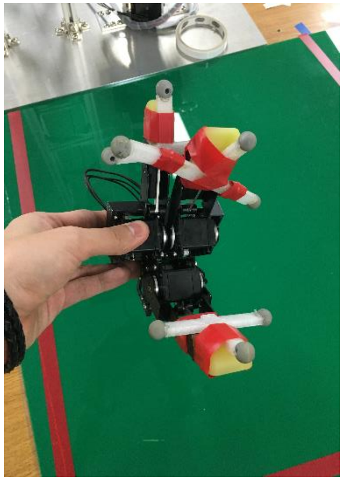




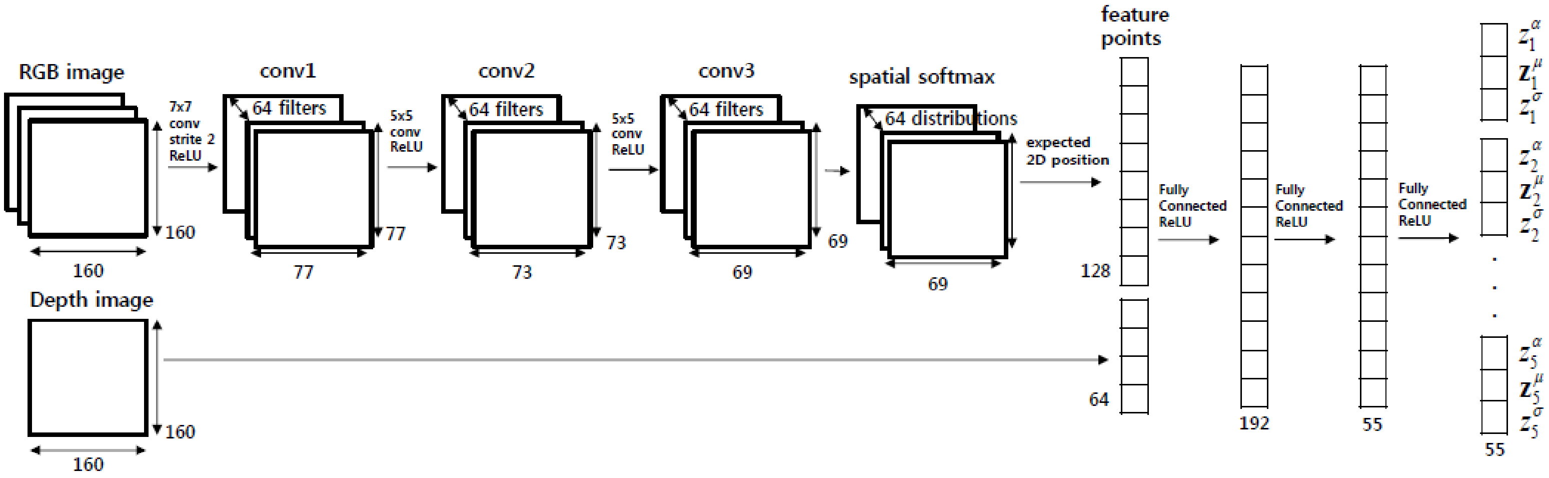
Pre-grasping pose

- It is extremely hard to collect a great amount of training data using traditional kinesthetic teaching procedure, where the human teacher directly moves the robotic arm to make the robot performs pregrasp

- To overcome this problem, we detached robotic hand from the robot arm and attached optical markers to ends of three fingers to track 3D positions of the fingers using optical motion capture system



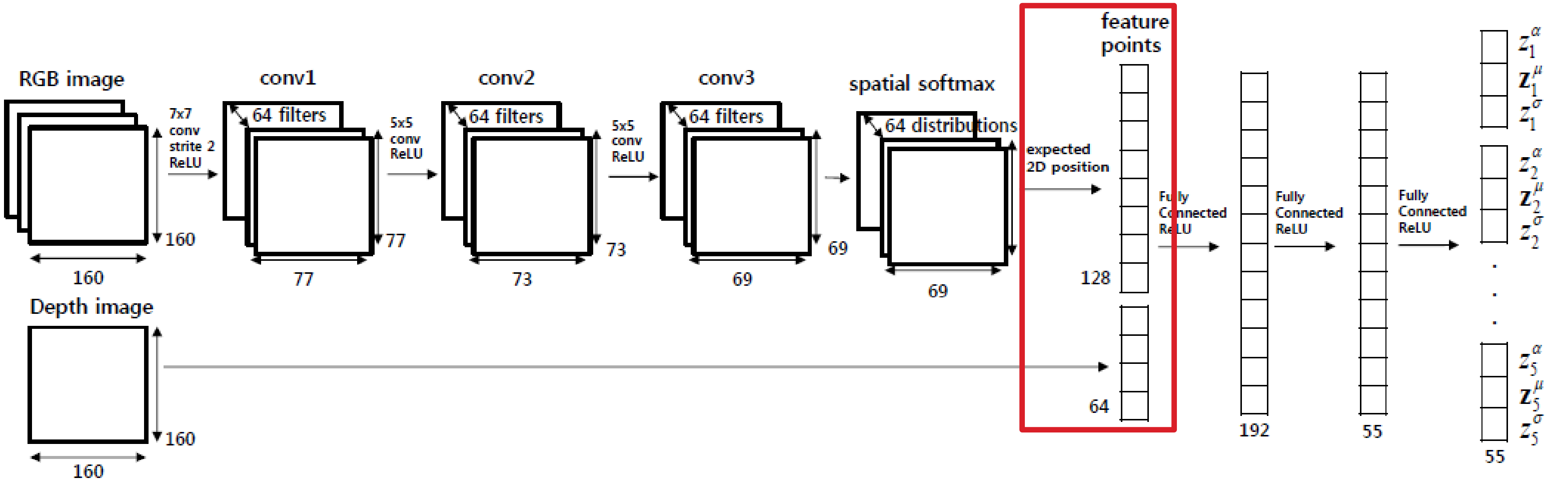
The Proposed Neural Network Architecture

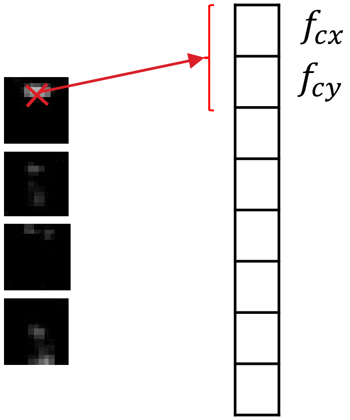


- Combination of a variant of traditional deep convolutional neural network and mixture density network (MDN)

- A supervised learning technique to pretrain DCNN

Spatial Softmax



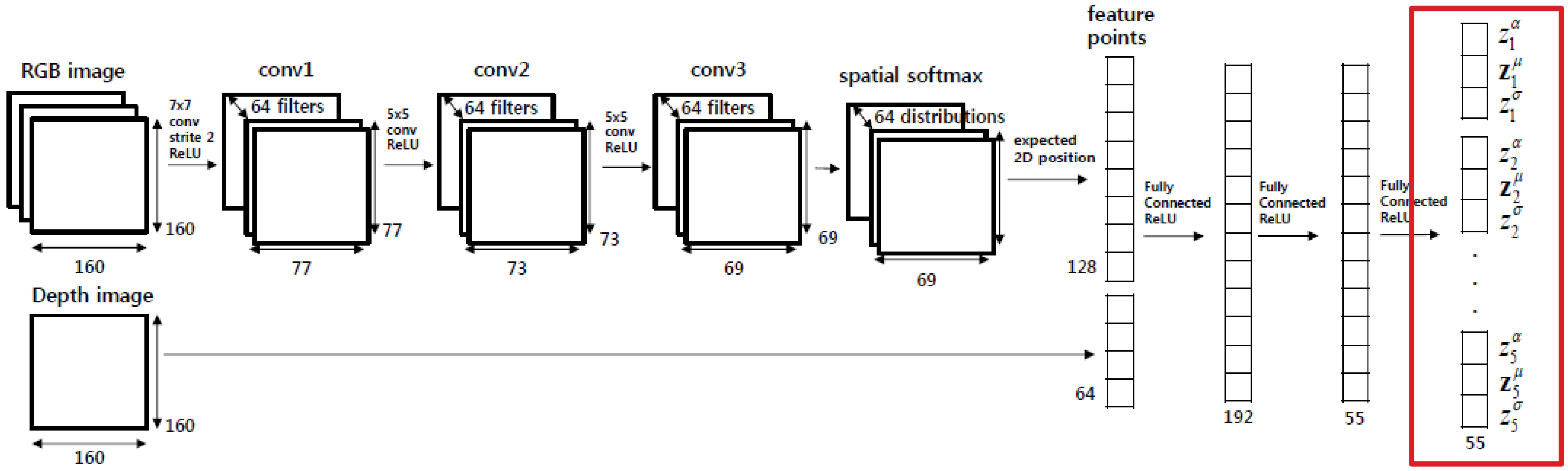


$$s_{cij} = e^{a_{cij}} / \sum_{i'j'} e^{a_{ci'j'}}$$

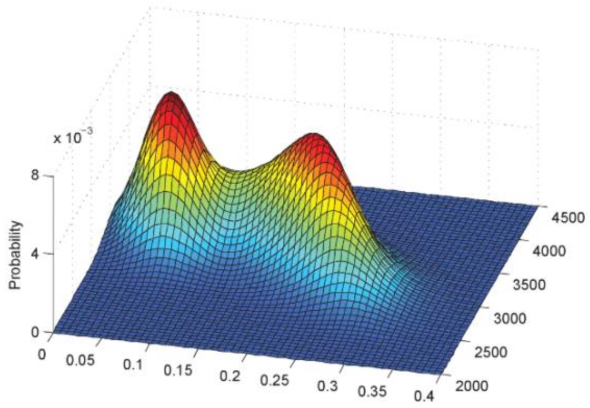
$$f_{cx} = \sum_{ij} s_{cij} x_{ij}$$

$$f_{cy} = \sum_{ij} s_{cij} y_{ij}$$

Mixture Density Network

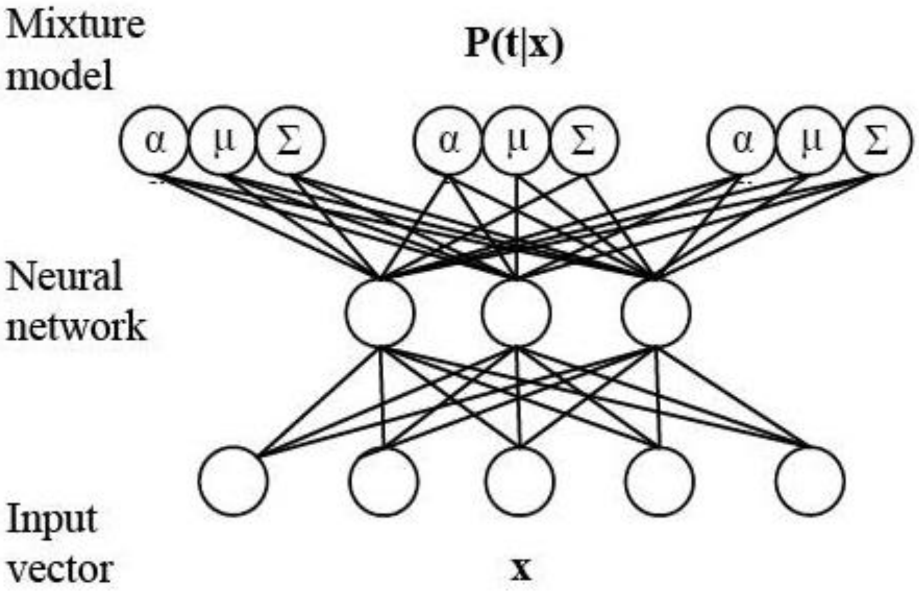


$$\alpha_i = \frac{exp(z_i^\alpha)}{\sum_{j=1}^{m} exp(z_j^\alpha)}, \quad \mu_{ik} = z_{ik}^\mu, \quad \sigma_i = exp(z_i^\sigma)$$



Mixture Density Network

Christopher M. Bishop, 1994

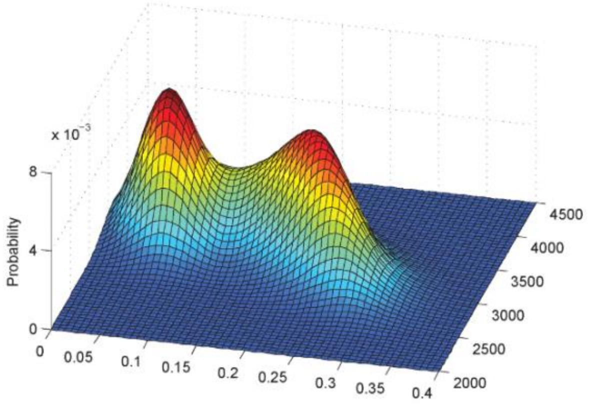


$$p(\mathbf{t}|\mathbf{x}) = \sum_{i=1}^{m} \alpha_i(\mathbf{x})\phi_i(\mathbf{t}|\mathbf{x})$$

$$\phi_i(\mathbf{t}|\mathbf{x}) = \frac{1}{(2\pi)^{c/2}\sigma_i(\mathbf{x})^c} exp\{-\frac{\|\mathbf{t}-\boldsymbol{\mu}_i(\mathbf{x})\|^2}{2\sigma_i(\mathbf{x})^2}\}$$

$$\alpha_i = \frac{exp(z_i^\alpha)}{\sum_{j=1}^{m} exp(z_j^\alpha)}, \quad \mu_{ik} = z_{ik}^\mu, \quad \sigma_i = exp(z_i^\sigma)$$

$$\ell = \sum_{j=1}^{n} \left[-ln\{\sum_{i=1}^{m} \alpha_i(\mathbf{x}^j)\phi_i(\mathbf{t}^j|\mathbf{x}^j)\} \right]$$



Mixture Density Network

$$\alpha_i = \frac{exp(z_i^{\alpha})}{\sum_{j=1}^{m} exp(z_j^{\alpha})}, \quad \mu_{ik} = z_{ik}^{\mu}, \quad \sigma_i = exp(z_i^{\sigma})$$

Parameters for the i-th Gaussian

$$\phi_i(\mathbf{t}|\mathbf{x}) = \frac{1}{(2\pi)^{c/2}\sigma_i(\mathbf{x})^c} exp\{-\frac{\|\mathbf{t}-\boldsymbol{\mu}_i(\mathbf{x})\|^2}{2\sigma_i(\mathbf{x})^2}\}$$

The number of training data

The number of kernel in GMM

$$\ell = \sum_{j=1}^{n} \left[-ln\{\sum_{i=1}^{m} \alpha_i(\mathbf{x}^j)\phi_i(\mathbf{t}^j|\mathbf{x}^j)\} \right]$$

Mixture Density Network

*Backpropagation

$$\pi_i(x,t) = \frac{\alpha_i \Phi_i}{\sum_{j=1}^{m} \alpha_j \Phi_j}$$

$$\frac{\partial \ell}{\partial z_k^{\alpha}} = \alpha_k - \pi_k$$

$$\frac{\partial \ell}{\partial z_i^{\sigma}} = -\pi_i \left\{ \frac{\| t - \mu_i \|^2}{\sigma_i^2} - c \right\}$$

$$\frac{\partial \ell}{\partial z_{ik}^{\mu}} = \pi_i \left\{ \frac{(\mu_{ik} - t_k)}{\sigma_i^2} \right\}$$

Mixture Density Network

To overcome overflow

$$\alpha_i = \frac{exp(z_i^{\alpha} - k)}{\sum_{j=1}^{m} exp\left(z_j^{\alpha} - k\right)}, \qquad where\ k = Max_{i=1}^{m}(z_i^{\alpha})$$

*Log-Sum-Exp trick

$$\log \sum_{i=1}^{n} e^{x_i} = a + \log \sum_{i=1}^{n} e^{x_i - a}$$

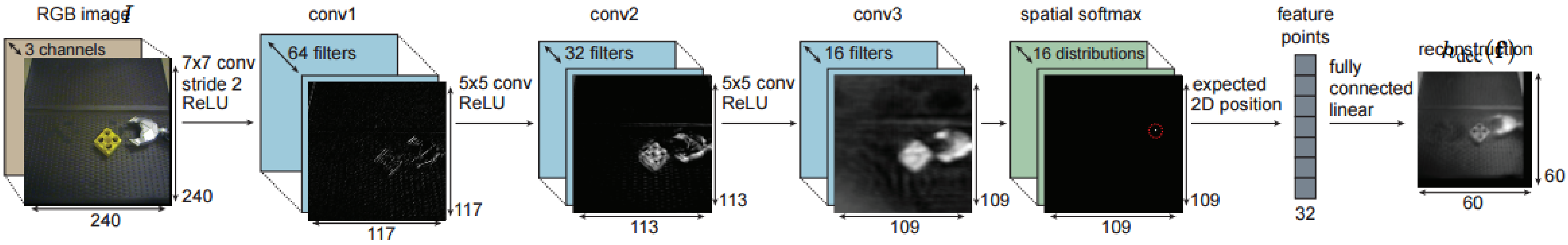
$$\ell = \sum_{j=1}^{n} \left[-ln\{\sum_{i=1}^{m} \alpha_i(\mathbf{x}^j)\phi_i(\mathbf{t}^j|\mathbf{x}^j)\} \right] \qquad \phi_i(\mathbf{t}|\mathbf{x}) = \frac{1}{(2\pi)^{c/2}\sigma_i(\mathbf{x})^c} \boxed{exp\{-\frac{\|\mathbf{t} - \boldsymbol{\mu}_i(\mathbf{x})\|^2}{2\sigma_i(\mathbf{x})^2}}\}$$

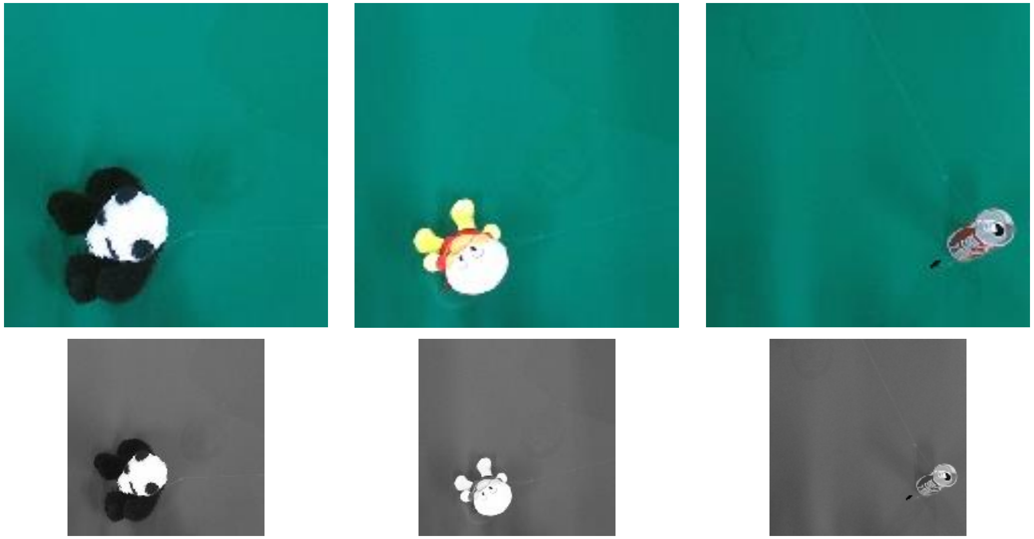
*Gradient clipping

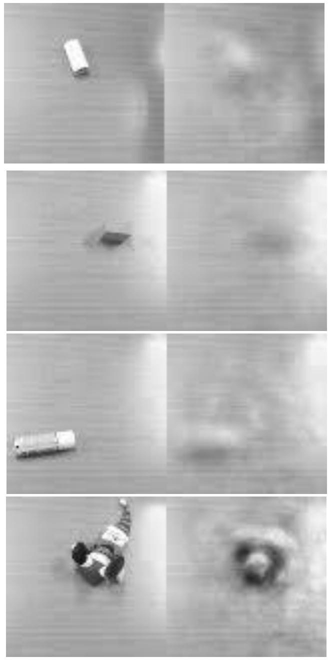
*Small learning rate

```
if $|\delta_w| > threshold$
        $\delta_w^{new} = \delta_w \times \frac{threshold}{|\delta_w|}$
```

Pretraining (*Autoencoder)

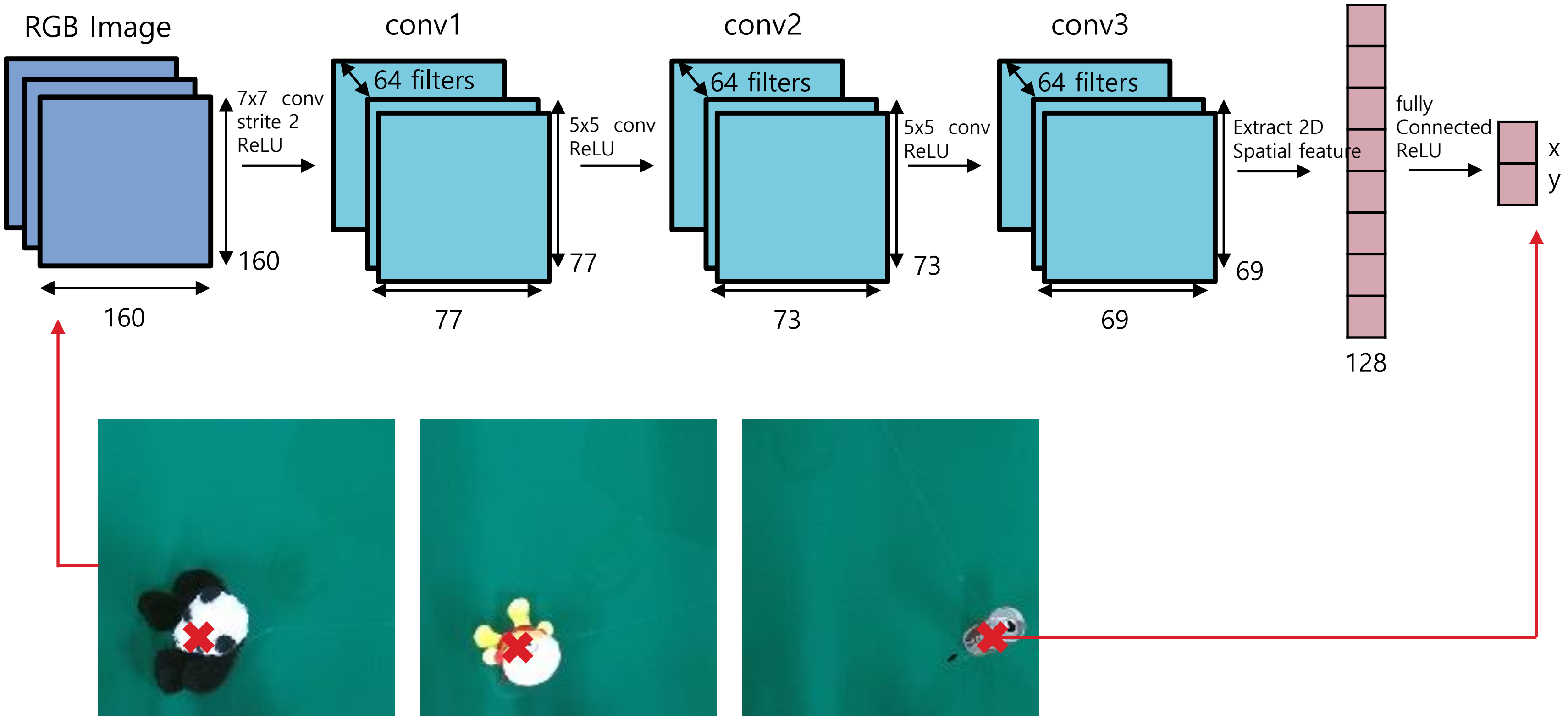






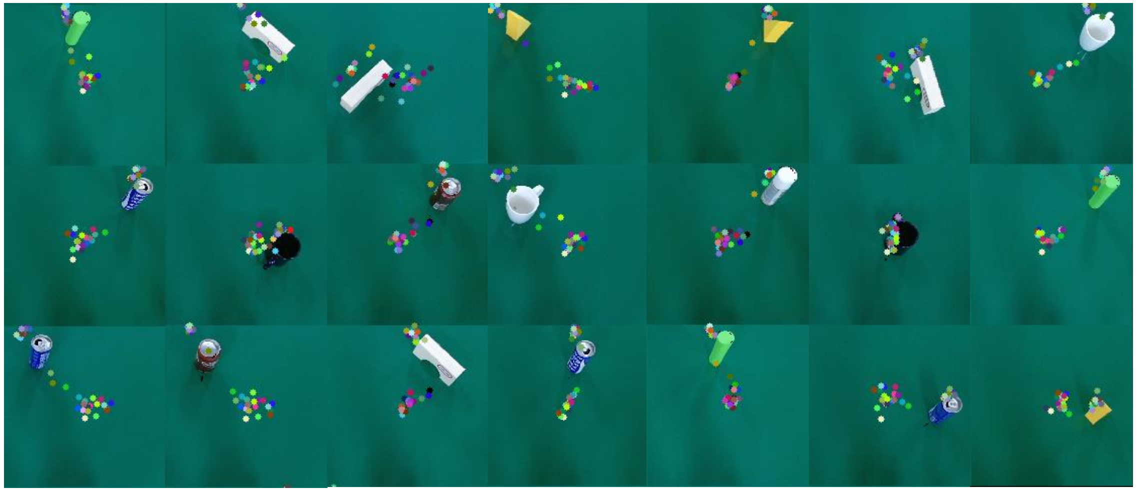
*Deep Spatial Autoencoder for Visuomotor Learning – Chelsea Finn etc, ICRA 2016

Pretraining (Proposed)

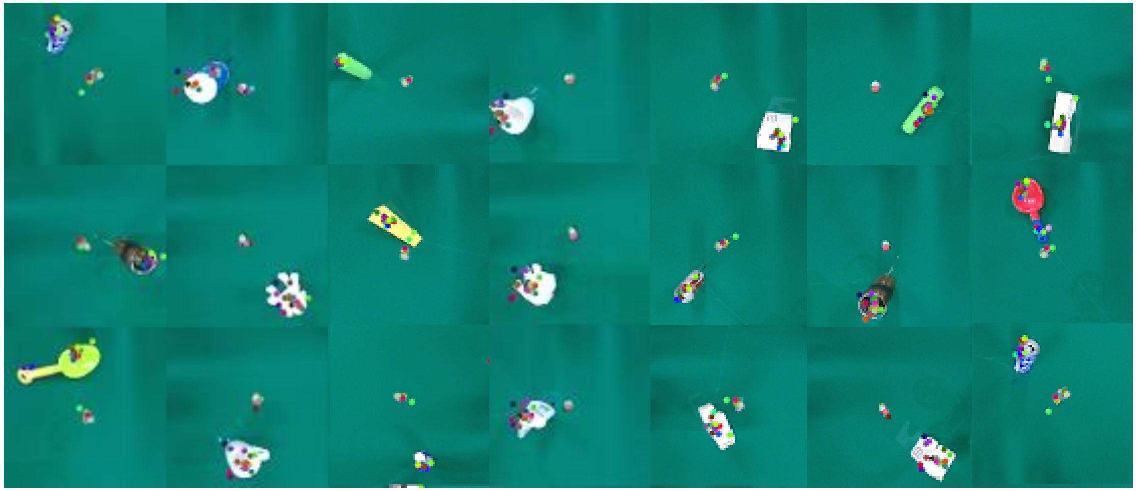


Pretraining

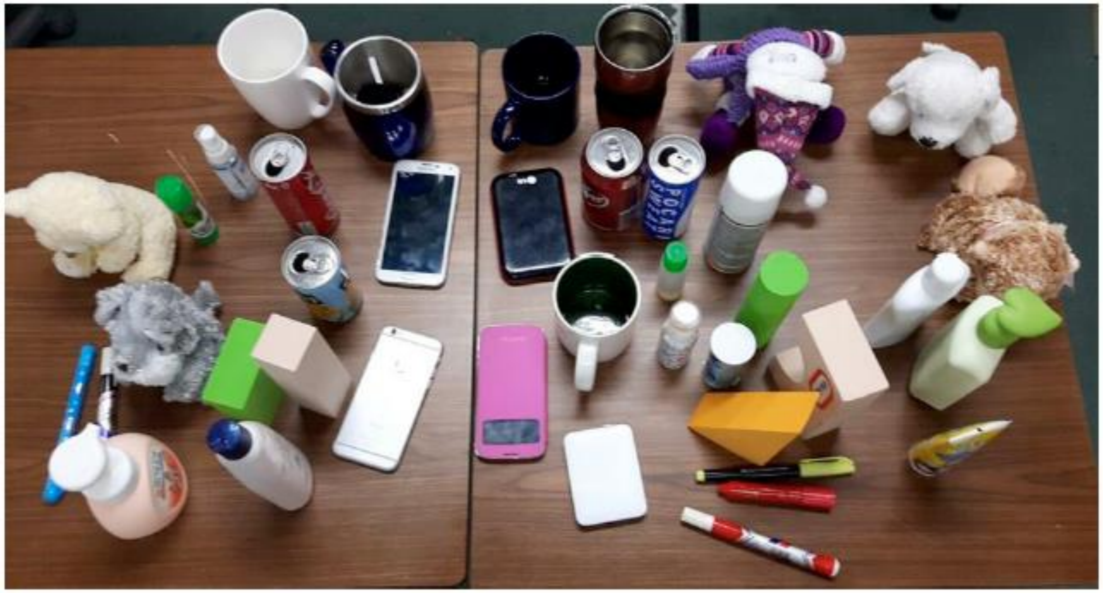
***Autoencoder**



***Proposed**



Experiment

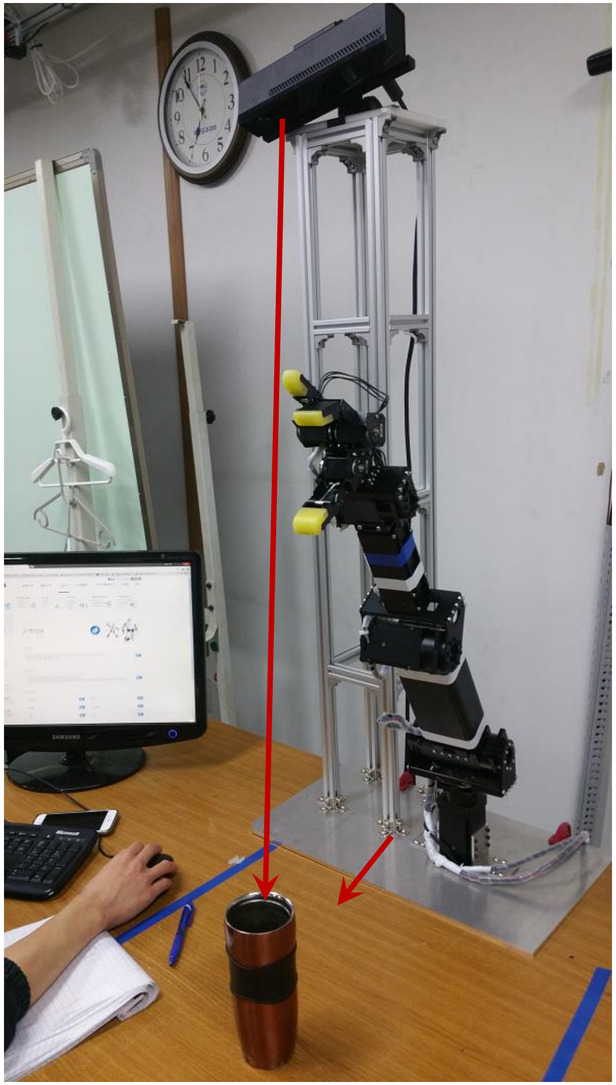


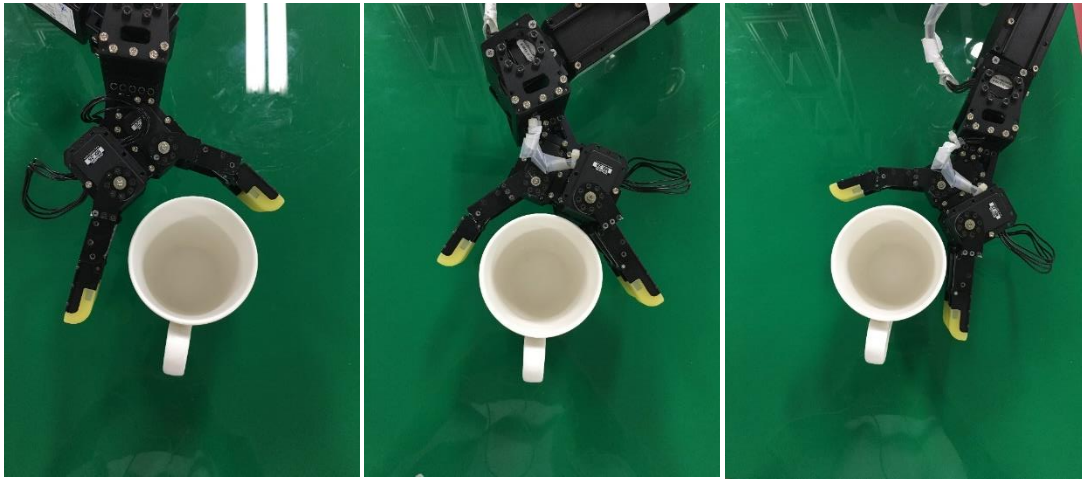
- Dataset consists of 8 categories of objects (cup, cellphone, pen, doll, lotion, can, small cylinder, toy block)
- The size of workspace is 1m x 1m
- The number of input images where each of them includes only an object is 180 and the number of target pregrasp poses for the inputs is 54,000

Experiment

- 150~300 human supervised pregrasp poses were labeled for an input

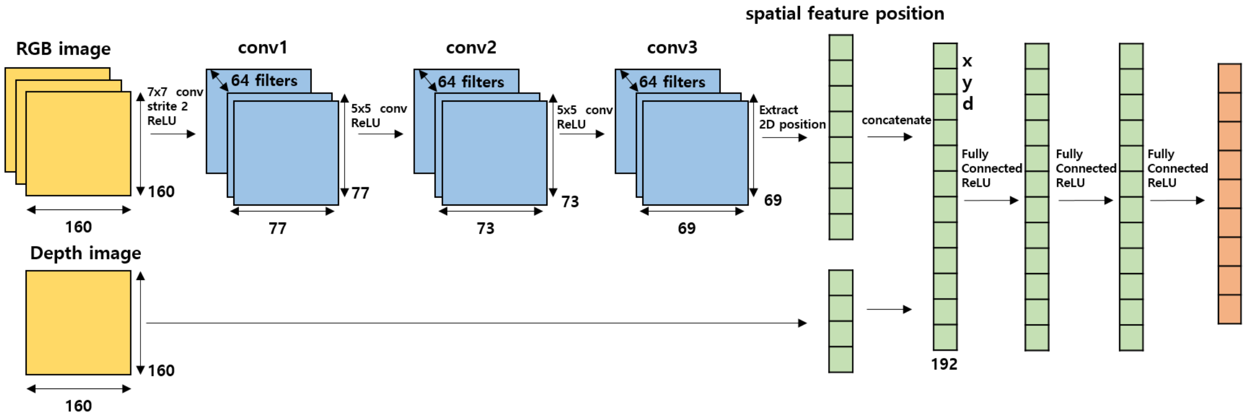
- The number of input images where each of them includes only an object is 550, pregrasp pose for the input is 119,243





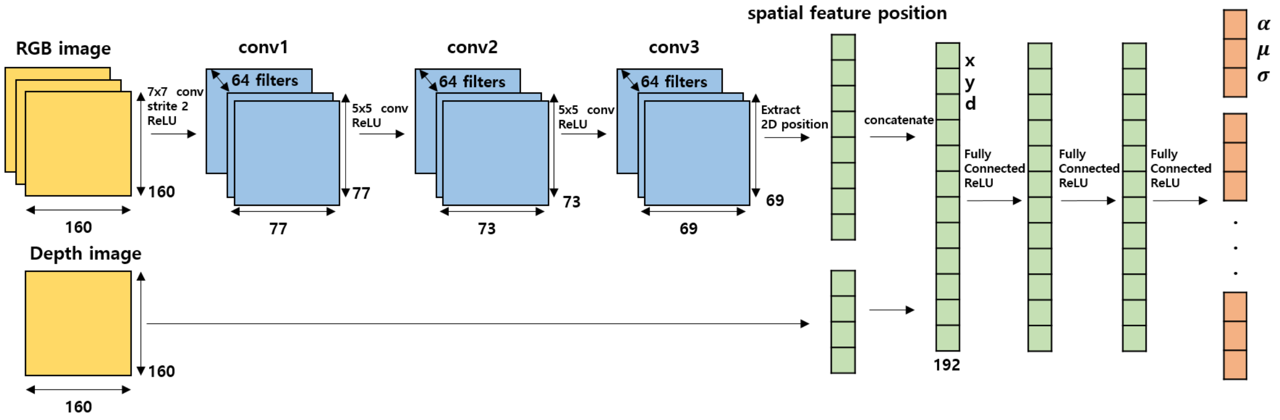
Experiment

1.**DCNN + SP** (Pretraining **DCNN** using the proposed supervised learning method)



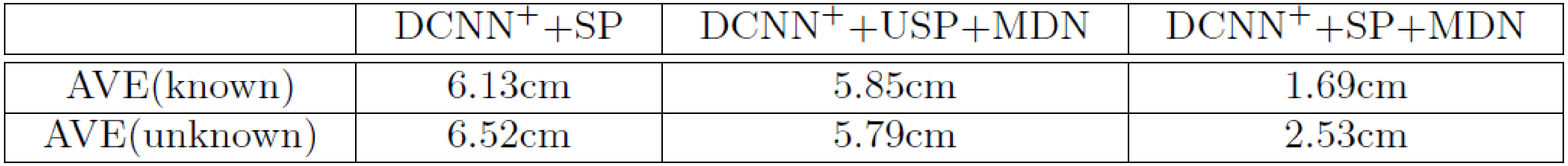
2.**DCNN + USP + MDN** (Pretraining **DCNN** using unsupervised learning method and combining **MDN**)

3.**DCNN + SP + MDN** (Pretraining **DCNN** using the proposed supervised learning method and combining DCNN and **MDN**)



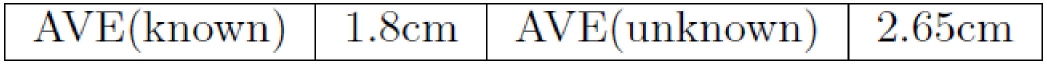
Result

- Average pregrasp pose prediction error. **DCNN+USP+MDN** and **DCNN+SP+MDN** select the mean of the largest Gaussian as the prediction



	DCNN^{+}+SP	DCNN^{+}+USP+MDN	DCNN^{+}+SP+MDN
AVE(known)	6.13cm	5.85cm	1.69cm
AVE(unknown)	6.52cm	5.79cm	2.53cm

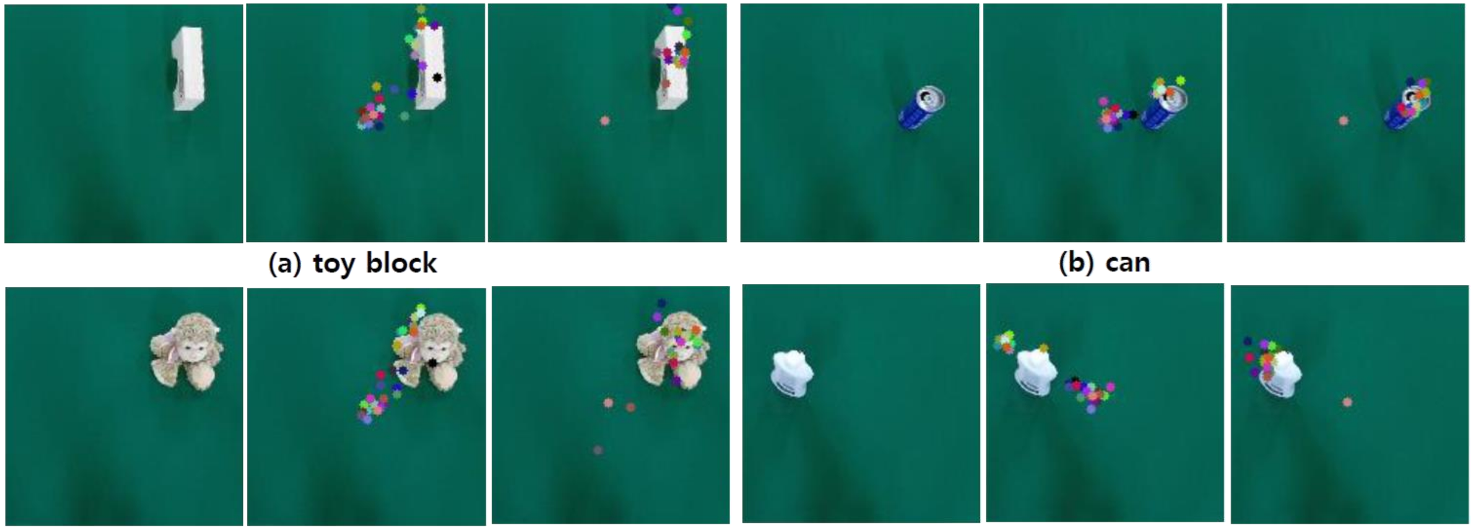
- Average pregrasp pose prediction error. **DCNN+SP+MDN** select the mean of the second largest Gaussian as the prediction



AVE(known)	1.8cm	AVE(unknown)	2.65cm

Result

- Advantage of the proposed supervised pretraining



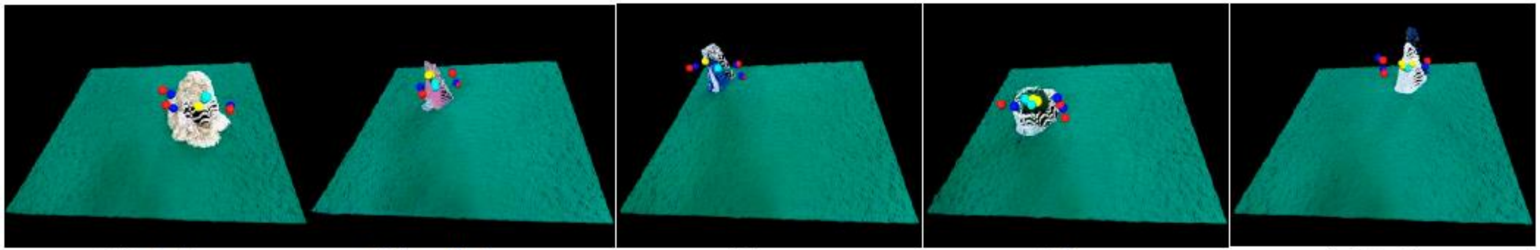
(a) toy block

(b) can

(c) doll

(d) lotion

- Advantage of the use of MDN and proposed supervised pretraining

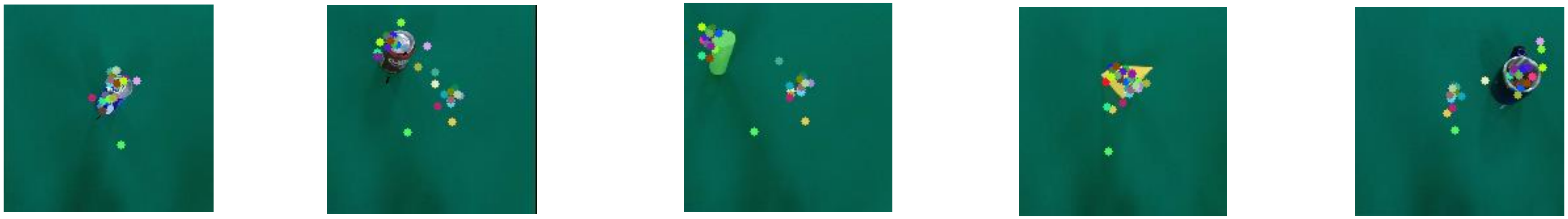


(a) doll (b) cellphone (c) can (d) cup (e) lotion

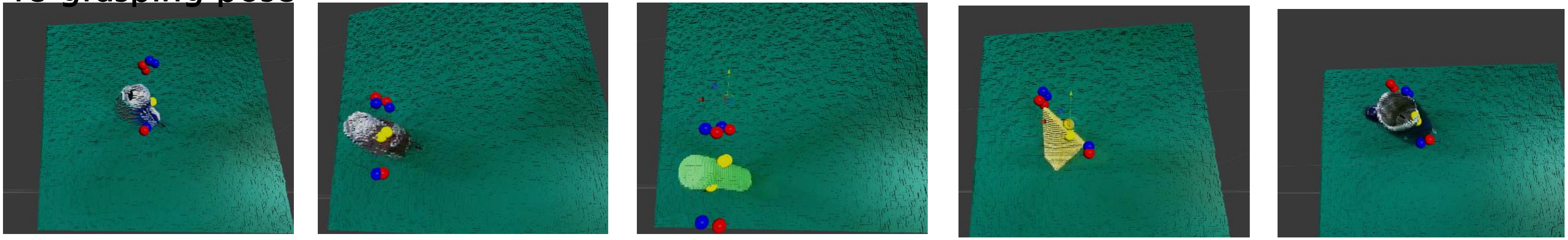
Result

Test data Image

2D Spatial Feature



Pre-grasping pose



감사합니다

INCORAL Since 1985
Intelligence and Control for Robots Laboratory

Hanyang University

Deep Learning based on Human Supervision for Robotic Grasping

Sungphill Moon, Youngbin Park and Il Hong Suh
Hanyang University, Korea.

What

Robotic grasping in the presence of various poses for seen and unseen objects

Related works

Google

CMU

- Self-supervised learning is used to train deep neural network
- Both mothods achieve outstanding performance but they require a lot of time to collect traning due to random trials in self-supervised learning
- Google : 14 robots, 2 months, 800,000 grasping attemps
- CMU : 1 robot (two arms), 700 hours, 50,000 grasping attemps

How

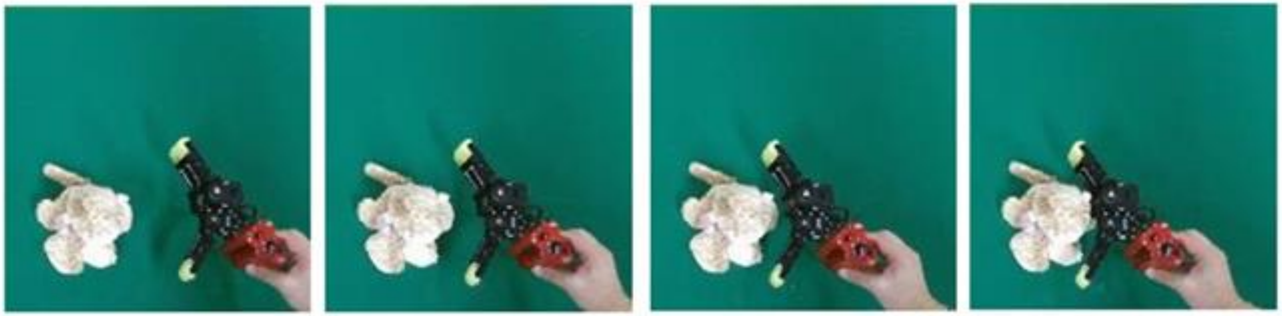


Fig. 1. Demonstration of a grasp trajectory with detached hand.

- To address this problem, we detached the robotic hand from the manipulator. The human teacher then grabbed the hand and demonstrated possible grasp poses as shown in Figure 1. At the moment, a camera captures the demonstrations.
- This significantly reduces the time and human efforts but obvious drawback is that we can not record joint angle trajectories exactly as we can do during typical kinesthetic teaching.

Visual inverse kinematics (VIKi) network

- A novel deep neural network to predict joint angle configuration from a given segmented robotic hand image

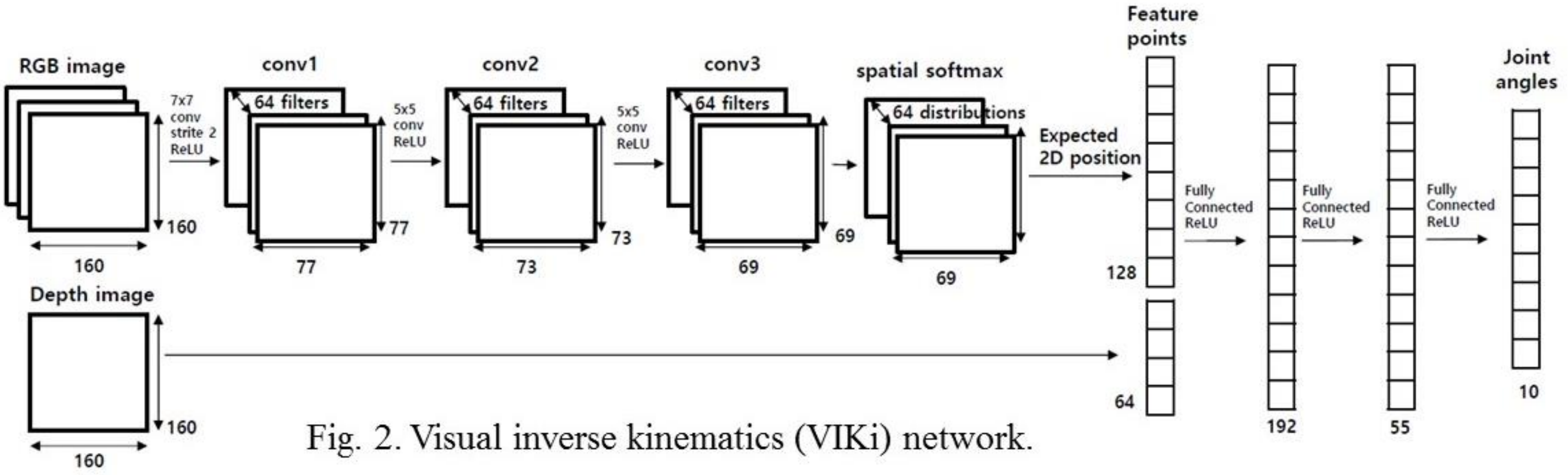


Fig. 2. Visual inverse kinematics (VIKi) network.

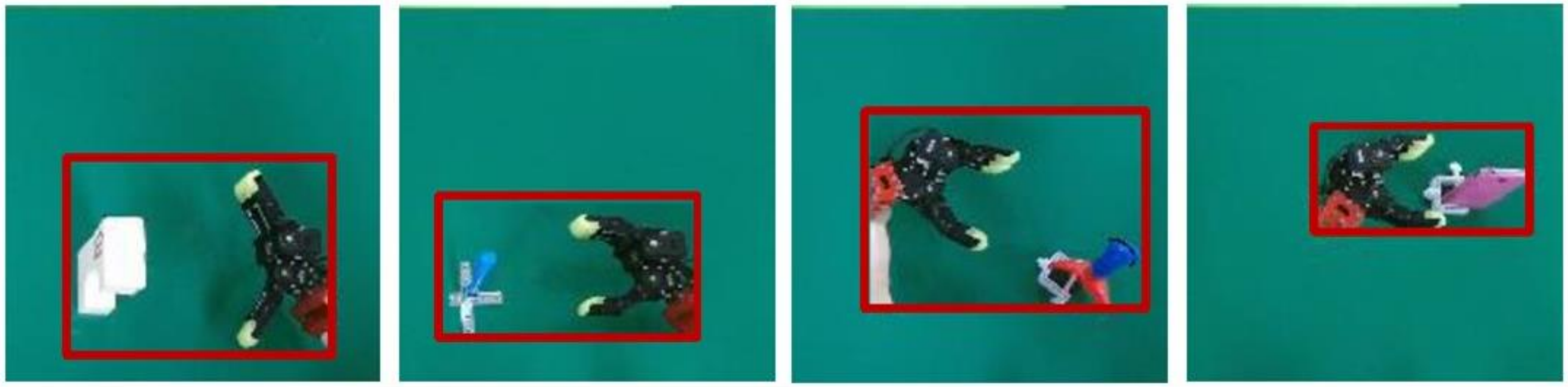
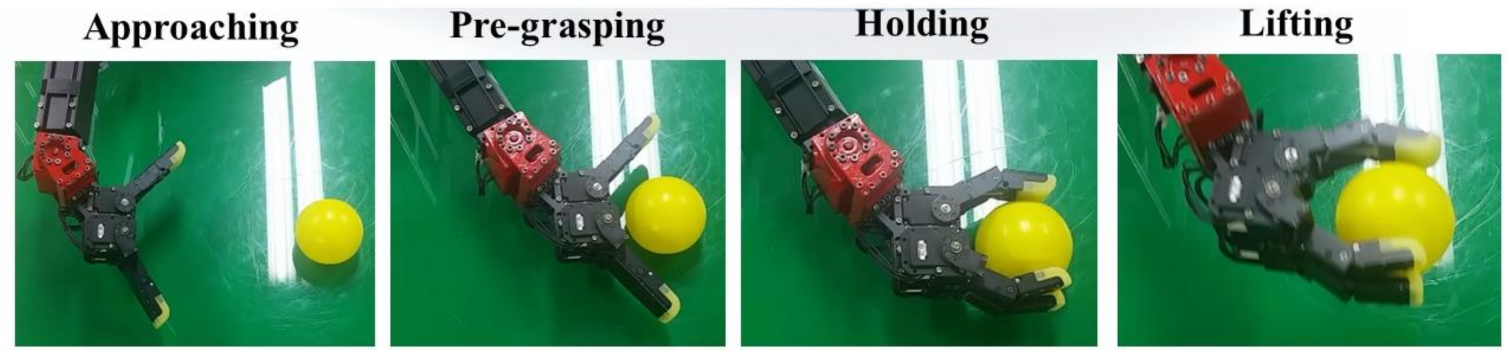


Fig. 3. Examples of the input for VIKi network.

Two Deep Neural Networks for Grasping



Approaching Pre-grasping Holding Lifting

Fig. 4. Four steps for grasp tasks.

- We assume that grasp task is composed of 4 consecutive trajectories. As shown in Figure 4, they are approaching, pre-grasping, holding and lifting.
- Among 4 steps, we developed two network for approaching and pre-grasping, respectively. The implementation of holding and lifting can only be trivial.

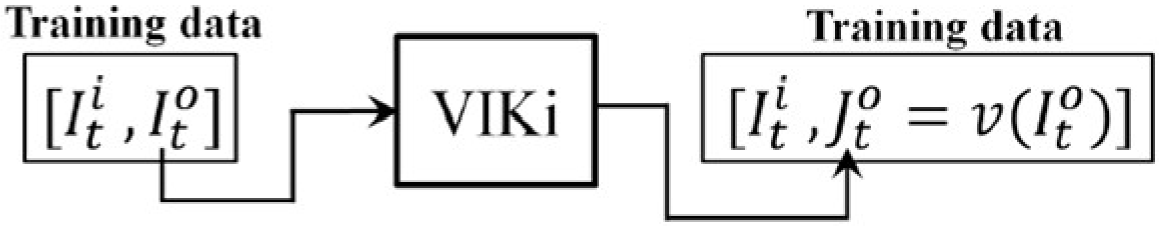
Approaching network

- The structure of the proposed neural network for approaching is identical to VIKi network. The training input image contains an isolated object and the training output is the final pose of approaching.

Pre-grasping network

- The proposed neural network architecture for pre-grasping is identical to VIKi network as well. The input of the network is an image containing an object and the robotic hand and the training output is the subsequent hand pose for eventually making the last pose of pre-grasping.

The role of VIKi network



- VIKi can be considered as a function to transform an training output image to joint angles. Original training data for approaching and pre-grasping networks is composed of a pair of an input image I_t^i and an output image I_t^o . Io t is converted into joint angles J_t^o via VIKi function v

Experiments

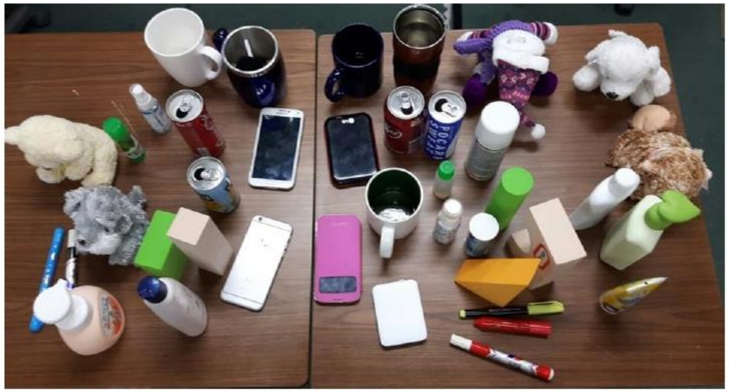


Fig. 5. All objects used for our training and test data.