Master's Thesis Project Report

Efficient Robot Grasp Learning by Demonstration

1610406 GAO ZIYAN

Supervisor	Chong NakYoung
Main Examiner	Chong NakYoung
Examiners	Iida Hiroyuki
	Nguyen Minh Le
	Okada Shogo

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
(Information Science)

February 2019

Abstract

In this paper, a Learning from Demonstration approach was proposed for robotic grasping with compliant arms. The compliance in the robot arm for safety often causes a problem in grasping. Recurrent neural network has been widely applied for modeling the sequential data such as sound, because recurrent neural network can leverage their internal state to process input sequences, this allows it to show the temporal dynamic behavior of time series. In this research, a recurrent neural network was constructed, given the estimation of the target object position and random initial joint angles of the robot arm, the recurrent model can produce the whole trajectories for grasping the target object.

For demonstrating the behavior to robot, there are several interfaces that can be choose such like demonstrating the behavior by using vision, motion sensor, teleoperation, and directly guiding the robot to demonstrate the behavior by its own. Directly guiding the robot and demonstrating the behavior by its own has several benefits, the first one is that there is no correspondence problem met, also human do not need to wear the motion sensor for showing the behavior. In this research, directly guiding the robot by human is used for demonstrating the grasp behavior.

In order to generate smooth and stable trajectories and to release the human labors, a transform model was proposed that can transform the example trajectory to adapt to the new situations, that is, given the example trajectory, which is formed by a series of discrete point, and the new initial and final discrete point, this transform model will output a new trajectory which satisfies two requirements, the first one is that the new trajectory must pass through the new initial and final discrete point, the second one is that the new trajectory must have the same tendency and same number of discrete points compared with the example trajectory. Then human do not need to record corresponded trajectory of the behavior, only specifying the target joint configurations and initial joint configurations are needed. Dataset can be enlarged by adding multiple initial joint configurations, that is, for one target, there will be multiple trajectories generated. Specifically, the two arms of the robot are trained respectively, and a support vector machine is used to decide which arm needs to be used for grasping the target object.

The evaluation results show that our recurrent model not only has a good prediction for the final joint configurations, but also generates smooth and stable trajectory. Moreover, the model is robust to the changes in the initial joint state which means that even though the initial joint configuration is

affected by disturbances, the model can still generate trajectories leading to the final joint configurations for grasping the object.

Finally, we tested the proposed learning method on the Pepper robot which can successfully grasp randomly placed object on the workbench. Compared to traditional methods which need to avoid singular configurations as well as to secure accurate localization, our method turns out to be robust and efficient and can be applied to cluttered environment.

Contents

List of Figures

Chapter 1

Introduction

Recent advances in robotic grasping have shown promising results. However, to make robots see, perceive, decide, and act in a way a human or a primate does, many challenges still need to be addressed [1]. In recent years, Learning from Demonstration (LfD) was successfully used in the field of robotics for applications such as playing table tennis [2], object manipulation [3], making coffee [4], grasping novel objects [5], carrot grating [6], etc. Since robots must operate in real environments and make decisions based on noisy sensory information and incomplete models of the environment, deep learning methods that directly model the relationship between the available sensory input and the desired output have become more popular [7]. In order to generate smooth trajectories and to decrease the number of human demonstrations, a data augmentation method was proposed to increase the training data. For human-like dual-arm robots, it also needs to make decision for which arm needs to be used for grasping the object. A support vector machine classifier was implemented for the arm selection problem.

1.1 Motivations

In the real environment, robot often has the problem of in-accurate sensing, in the current research, many researchers tried to avoid this problem by carefully design the sensor's position and orientation as well as the experiment environment in order to get high quality sensing data, for instance, researchers might mount the camera on the top of the table and perpendicular to it in order to get high quality depth image, but in the case that the depth camera is mounted on the robot and is not perpendicular to the image plane, then robot cannot sense well.

In the context of humanoid robot grasping, accurate sensing input is es-

sential for successfully grasping the object. On the other hand, RGB camera offers abundant information but without depth information and many object detection deep learning models are also based on RGB image input to detect the object's location. If the height of table is fixed, that is, we add a constrain to the environment, the robot could successfully grasp the object given the pixel location in the RGB image by a handful of human demonstrations.

1.2 Context of the Study

In this research, a social robot named "pepper robot" is used for testing our system. The task is to teach pepper robot to grasp a cubic block random placed on a workbench of fixed height. This task is divided into two sub-goals, this first one is to reproduce the similar behavior compared with humans. The second one is to classify which hand is needed to be used for performing this task. This system consists several modules, in order to generate large dataset, a transform model was derived, for generating the trajectories, recurrent Neural Network, specifically, two Gated Recurrent Unit layers and one fully connected layer are used, Support Vector Machine with None-Linear Kernel was used for inferring which hand is preferred for performing the task.

1.3 Objectives and Contributions

The objective is to develop a system that can accurately judge which arm is needed for performing the task and can generating human-like smooth and accurate grasping behavior. In this study, the main contribution are as follows: the first one is that a effective data augmentation method was found to enlarge the dataset for training the neural network, this augmentation method contributes to the robustness of the system for the initial robot arm configurations and the better convergence of the RNN. The second one is that the head movement was took into consideration, that is, this system is not only based on the pixel location of the object but also consider the robot neck angles for locating the object.

1.4 Neural Network

1.4.1 Simplest Neural Network

This simplest neural network model only contains one neuron. A neuron (node) can be treated as a logistic unit with Sigmoid (logistic) Activation

Function, which outputs a computation value based on sigmoid activation function. This neural network can be applied to logistic regression. The Fig 1.1 shown bellow illustrates the structure of this simplest neural network. Layer 1 is called Input Layer that inputs features, the input layer embedded

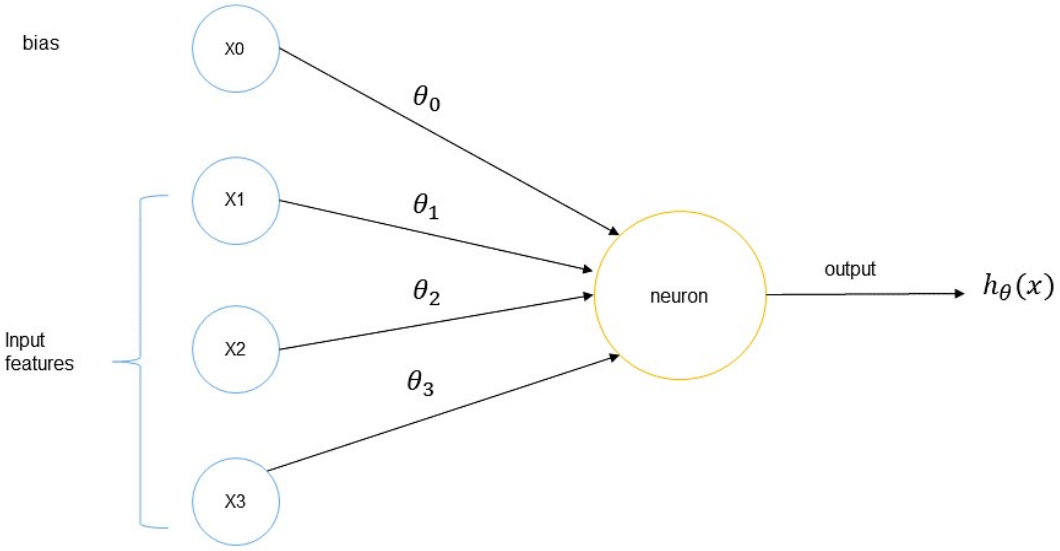

Figure 1.1: Simplest Neural Network

with the input features and bias, bias a trainable parameter. last Layer is called Output Layer that outputs the final value computed by hypothesis $h_\theta(x)$. Each arrow attached between the input layer and the neuron represents a weight. There are two operations inside the neuron, the operations is shown bellow.

$$Z = \theta^T X$$

$$h_\theta = \sigma(Z)$$

The first operation is inner product of the weights and input features, θ are trained parameters, the second is non-linear activation operation, here the activation function could be sigmoid, tanh etc. For the sigmoid function, the formula is:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{1.1}$$

sigmoid activation function can map the value to the range from 0 to 1, it is commonly used in binary classification problem. For the tanh activation function, the formula is

$$f(x) = \frac{e^x - e^{-1}}{e^x + e^{-x}} \tag{1.2}$$

tanh activation function can map the value to the range from -1 to 1, also it is commonly used as the gate activation function in gate mechanism.

1.4.2 Recurrent Neural Network

Recursive Neural Network (RNN) is an artificial neural network with tree-like hierarchical structure and network nodes recursively input information according to their connection order. The idea behind RNNs is to make use of sequential information. In a traditional neural network, all inputs (and outputs) are independent of each other is assumed. RNNs are called recurrent because they perform the same task for every element of a sequence, with the output being depended on the previous computations. Another way to think about RNNs is that they have a memory which captures information about what has been calculated so far. In theory RNNs can make use of information in arbitrarily long sequences, but in practice they are limited to looking back only a few steps (more on this later). The unrolled structure of recurrent neural network are illustrated in Fig 1.2.

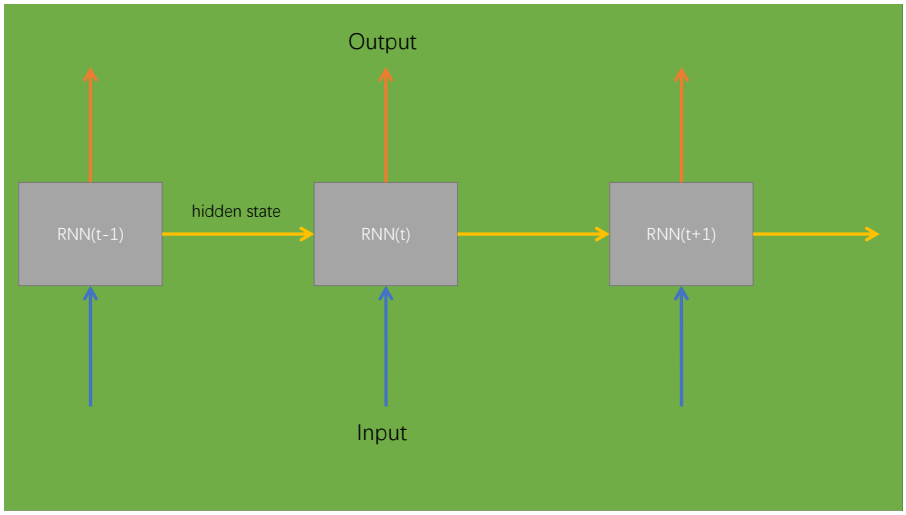

Figure 1.2: Recurrent Neural Network

Under this context, a model based on recurrent neural network is developed, that is, given the environment representation, in this case, the pixel location of the red mark attached on the cubic block and neck angles of the robot as well as the initial joint angles of the robot arm, this model can

generate a series of joint values which forms several trajectories correspond to each joint of the robot. In the experiment, we found that the output trajectories overlapped well at the previous time step but diverged at the last several time steps. The reason is that the error amplified when propagating to the next time step. In order to solve this problem, a one to many training fashion was used which will be explained in the Chapter 3.

1.4.3 Long Short Term Memory

Long Short Term Memory was proposed in 1997 by Sepp Hochreiter and Jrgen Schmidhuber[25]. Long Short Term memory (LSTM) is a variant of the recurrent neural network (RNN). An RNN composed of LSTM units is commonly referred to as an LSTM network (or simply LSTM). The public LSTM unit consists of a cell, an input gate, an output gate, and a forget gate. The cell remembers the values in any time interval and the three gates control the flow of information into and out of the unit. Long short term memory has been widely used in deep learning models and fieds of Natural Language Processing, Music Generation etc.

1.4.4 Gated Recurrent Unit

Gated recurrent units (GRUs) are a gating mechanism in recurrent neural networks, introduced in 2014 by Kyunghyun Cho et al[19]. Their performance on polyphonic music modeling and speech signal modeling was found to be similar to that of long short-term memory (LSTM). However, GRUs have been shown to exhibit better performance on smaller datasets.[2] In this research, GRU was used as the recurrent layer. Fig 1.3 shown bellow illustrates the structure of Gated Recurrent Unit. There are several operations inside shown below.

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{1.3}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{1.4}$$

$$h_t = (1 - z_t)h_{t-1} + z_t \sigma(W_h x_t + U_h(r_t h_{t-1}) + b_h) \tag{1.5}$$

$x_t$is the input vector, h_t is the output vector, z_t is the update gate vector, r_t is the reset gate vector,W, U, b are parameter matrices and vector. GRU has less parameters needed to learn compared with Long Short Term Memory and faster convergence during the training process under this context which had been verified in the experiment. In this research, we use 2 Gated Recurrent Unit and one output layer as our trajectory generator, The input is the initial

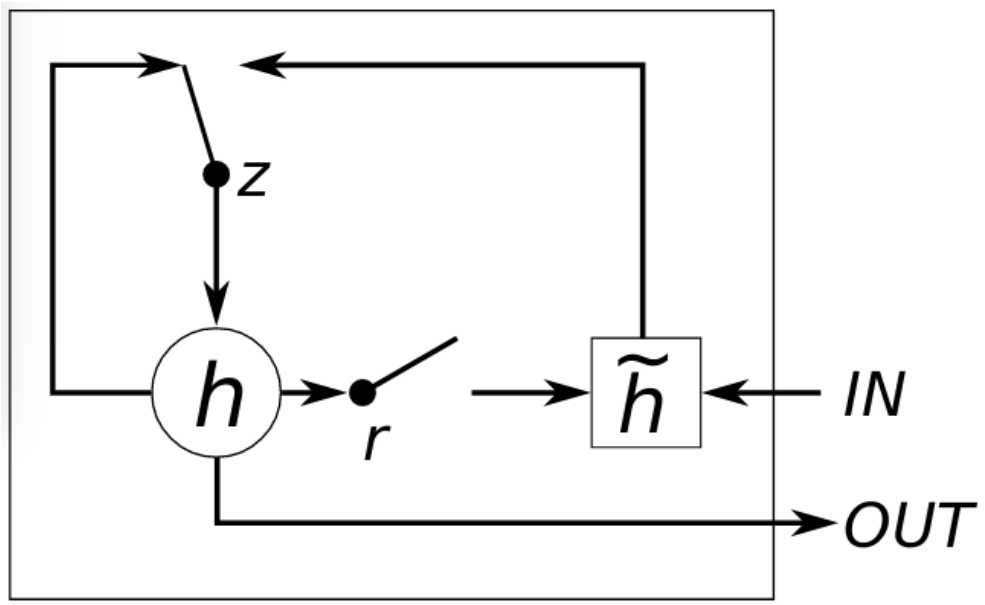

Figure 1.3: Gated Recurrent Unit by Kyunghyun et al

joint configurations of the experimented robot and the pixel location of the red mark attached on the cubic block as well as the neck angles, the output is a sequence of joint configurations for picking up the object. Considering the neck angles can contribute to get rid of the head orientation constrains, that is, if the neck angles were not considered, the robot must need to keep the required head orientation to detect the cubic block, that is not natural compared with human being.

1.4.5 Support Vector Machine

Support Vector Machine (SVM) is a supervised classification method derived from statistical learning theory that often yields good classification results from complex and noisy data. For humanoid robot, it is necessary to find a solution for making decision on which hand is needed to be used for executing the task. In this context, a support vector machine which non-linear kernel was implemented. The input is the pixel location of the red mark attached on the object and the neck configurations. The kernel is radial based function. it turns out that the classifier have affordable performance when we tested on the real robot.

1.5 Thesis Organization

This document will consist of five sections.

Chapter 1 - Introduction

This chapter will conclude the motivations and objective as well as all the concept contained in this thesis.

Chapter 2 - Background

This chapter describes the related work with this thesis.

chapter 3 - Methodology

This chapter provides all of the method used in this thesis for achieve the objective.

chapter 4 - Experiment

This chapter contains all the experiment details, which includes data pre-processing and data generation, recurrent neural network training and evaluation etc. and experiment results.

chapter 5 - Evaluation

This chapter contains the evaluation for the trained model, specifically, the trajectory generator model and hand selection model.

Chapter 6 - Conclusion and Future Work This chapter will contain the conclusion to all the work that has been done in this thesis as well as the future work.

Chapter 2

Background

So far, it is generally accepted that learning a task from scratch, that is, without any prior knowledge, is a difficult task. However, humans seldom try to learn from scratch. They summarize strategies for dealing with learning problems from other people's instructions or demonstrations.

There exists numerous tasks performed by humans in the field of industry due to the requirement of high dexterity such as force control or precise position. The goal of Learning from Demonstrations is to transfer the skills to robot from human which took years to master, also make the robot adapt to the changes can be done easily. Learn from demonstration is attractive because it bootstraps learning by starting from a good example to reduce search space for a good solution unlike reinforcement learning which need to carefully design the reward function and train from scratch. Learning from demonstration offers a user friendly approach of teaching.

A major challenge in LfD is to extend these demonstrations to unseen situations [8]. LfD is not just record and replay the demonstrated behavior but generalize the behavior to the new scenes which have not encountered before. The hypotheses is that human is a good example. One obvious way to mitigate this problem is by acquiring a large number of demonstrations covering as many situations as possible [9]. Some researchers proposed cloud based and crowd-sourced data collection techniques [10],[11],[12] or the use of simulation environments [13]. Another direction is to use smaller number of demonstrations, and update the learning model for better generalization. One possible technique is to hand-engineer task-specific features [14],[15]. [16] uses a large amount of synthesized images for training a model for position detection and transfers to the real physical environment images using a handful of images collected in the real physical world. Our method, in contrast to the previous approach, augments the data based on the demonstrated data. [17] uses a recurrent model to pick and place an object in a

virtual environment and deals with the pick and place task both by recurrent neural network (RNN) and reinforcement learning. [18] uses a deep spatial auto-encoder to acquire a set of feature points that describe the environment for the task. In our approach, we estimate the location only by the robot head orientation and object location in an RGB image.

2.1 Demonstration Interface

Demonstration Interface plays an important role in the way of how the information obtained and transmitted(see Fig 2.1). There are several demonstration interfaces which also affect the choice of policy derivation approaches.

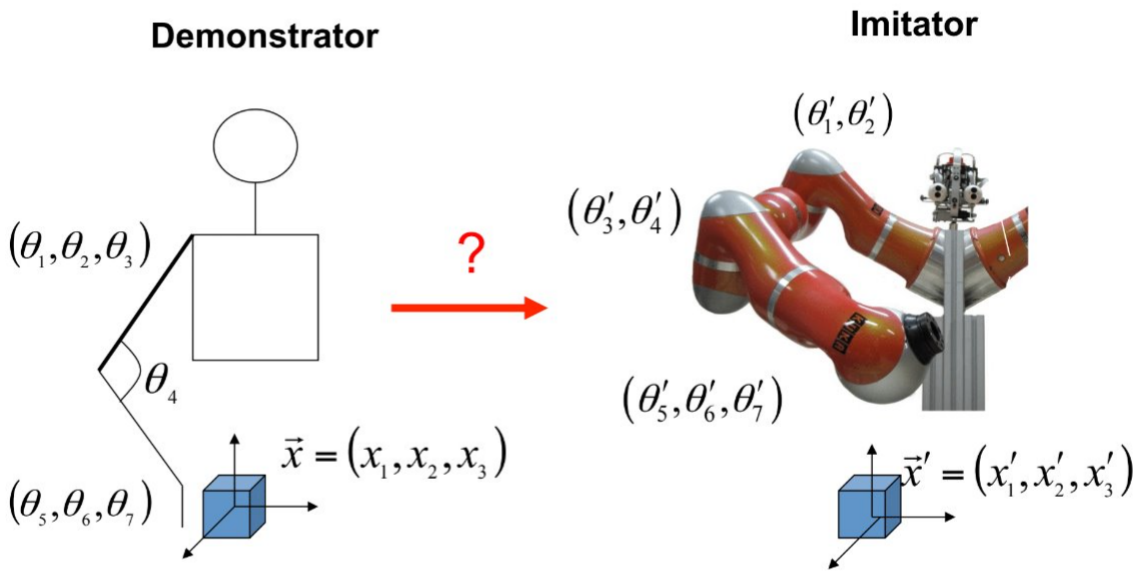

Figure 2.1: Correspondence between demonstrator and imitator.

Correspondence Problem

Correspondence problems involve the identification of mappings between teachers and learners, allowing information to be passed from one to another[8]. In [8], the author defined the correspondence between the two maps, record mapping and embodiment mapping which are shown in Figure 2.2 Record mapping refers to whether the exact state/action experienced by the teacher during the execution of the presentation is recorded in the data

set. Embodiment mapping refers to whether the state/action recorded in the data set happens to be the state/action that the learner will observe/execute. Correspondence plays an important role in how the raw collected data trans-

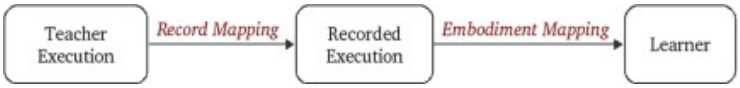

Figure 2.2: Record Mapping and Embodiment Mapping by Brenna D. Argall et al.

mit to the robot and how the robot execute the task.

2.1.1 Demonstrating behavior by using vision

The first one is to record human motions by using vision. This interface almost applied in the context of solely in kinematics teaching. one may use any of various motion tracking systems based on vision. Fig. 2.3 shows an example of whole body motion tracking using visual walking done by [21]. The human body model is first used to extract the motion of the human body from the background and then map it to the humanoid robot.

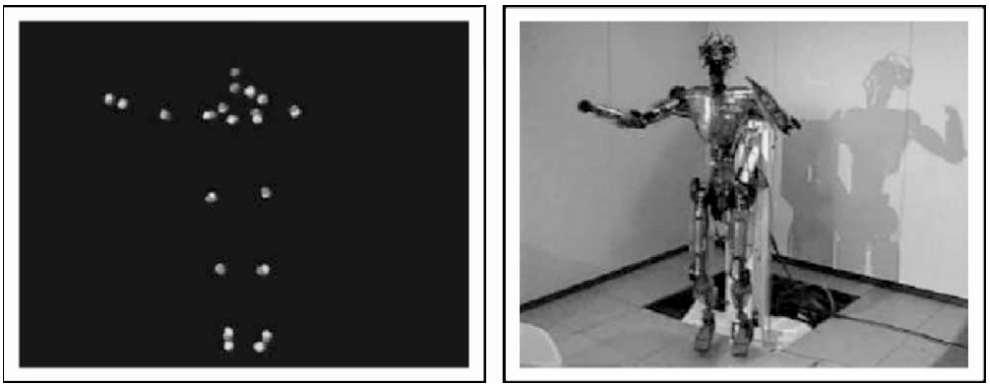

Figure 2.3: Demonstrating behaviors by using vision by Ude et al.

These external devices that track human motion return accurate measurements of the angular displacement of the joint. They have been used for a variety of Learn from Demonstration for body movements. The advantage of this method is that they allow humans to move freely, but it needs a good solution to the correspondence problem. Usually, this is done through a clear mapping between human and robotic joints, but it can be very difficult if the robot (for example, a hexapod robot) is very different from humans.

2.1.2 demonstrating behavior by using motion sensor

The second demonstration interface is the motion sensors [22],see Fig 2.4 Basically the human wear the motion sensors and then demonstrate the desired behavior, at the same time all the joint values changes are recorded. One of the benefit is that there is no correspondence problem exist, the problem is that haptic information cannot be record, Fig 2.4 illustrates an example of this demonstration interface.

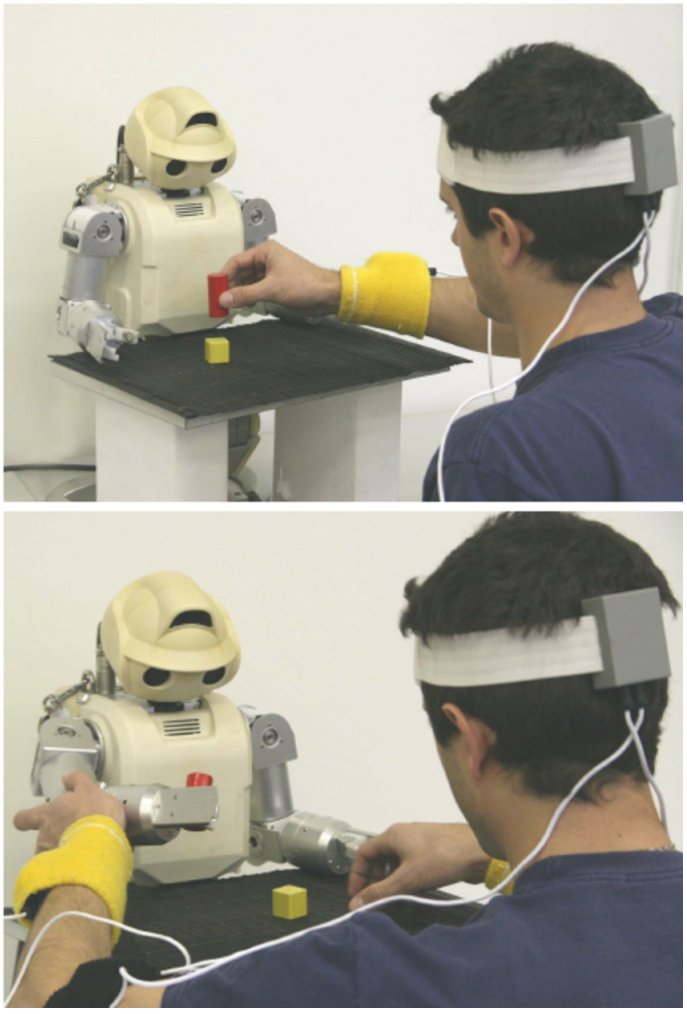

Figure 2.4: Demonstrating behavior by using motion sensor by Calinon et al.

2.1.3 Demonstrating behavior by teleoperation

For demonstrating the behavior, many researches make use of teleoperation technique the perform the task by robot itself. In the period of teleoperation, the robot is controlled by the robotic teacher when recording from robot own sensors. In [9] the author use a Playstation Move controller to control a non-expensive robot perform multitask, see Fig 2.5. Teleoperation provides the most straightforward way to demonstrate the transfer of information in learning. However, remote operation requires the operating robot to be manageable, so not all systems are suitable for this technology. For example, on systems with complex motor control for high degree of freedom manipulator.

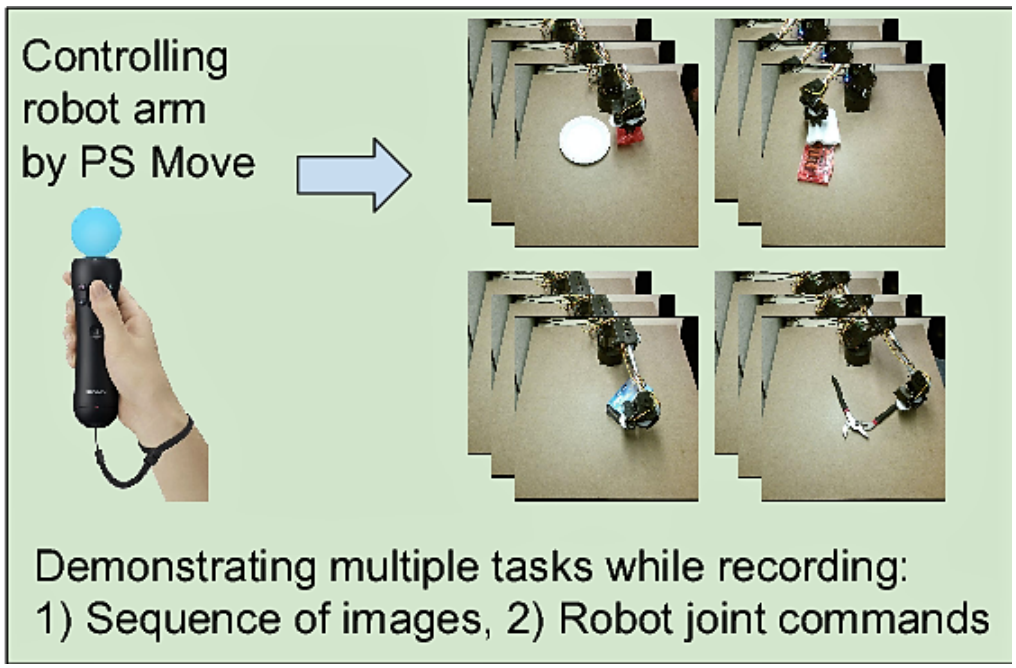

Figure 2.5: Demonstrating behavior by using joystick by Rahmatizadeh et al.

2.1.4 Demonstrating behavior by directly guiding robot

There is another straightforward way to demonstrate the behavior which is directly guiding robot manually to perform the task. Under this circumstances, this method avoids the correspondence problem and it do not require the demonstrator to wear the motion sensors or to use teleoperator to control the robot. Fig 2.6 shows an example of demonstrating behaviors by directly

guiding robot arm, the robotic teacher demonstrate manipulation action to the PR2 robot, he shows the action by directly guiding PR2 robots own end effector to perform the task.

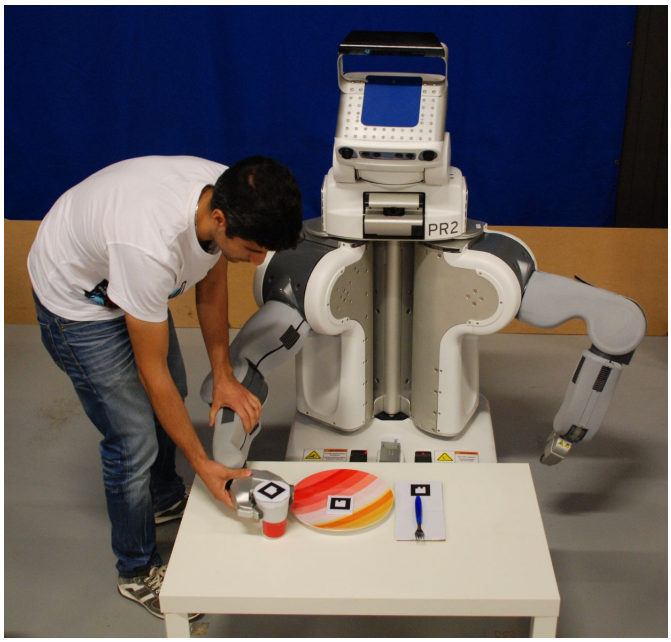

Figure 2.6: Demonstrating behaviors by directly guiding robot by Nichola Abdo et al.

2.2 Policy Derivation

[17] used a recurrent neural network to perform the pick and place task in the simulated environment. in simulation, the exact position and the target is known. a picking and placing behavior generator is trained separately, see Fig 2.7. Given the position of the target in cartesian space as well as the initial joint value of the robot. the recurrent neural network predicts several following states. For connecting two neighboring states, the author used a interpolator to do interpolation between two state. This research aims at automatically perform tasks in the field of industry. Given a Finite State Machine, the robot learns the relationship between the joint angles and object position to perform the task. The problem is that this model has not pay attention to the trajectory generated.

[9] used Variational Autoencoder-Generative Adversarial Network to perform multi-task learning. in this research, the author tested the model on

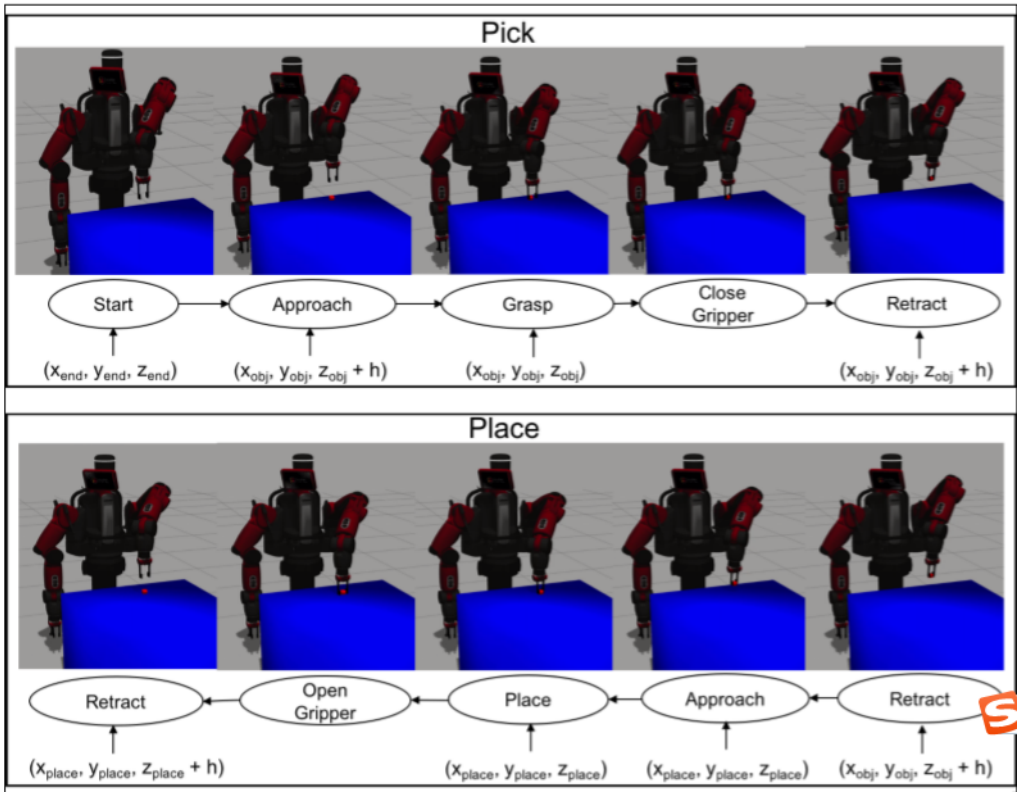

Figure 2.7: Pick and Place behavior generation by using recurrent neural network by Giovanni De Magistris et al.

an in-expensive robot to perform multiple task. Multiple task learning is achieved by a task selector, which is a one-hot vector to encode the category of the task. Firstly, they used a convolutional neural network to encode the instant image to a latent vector, they used the encoded latent vector to reconstruct the image by the Generative Adversarial Network, Generative Adversarial Network has two part: Generator and Discriminator. Discriminator is a binary classifier which classify whether the input image is the image generated by generator or not, instead the generator networks tries to 'cheat' the discriminator. The controller network, however, will pay more attention to some relative features from the image such like the pose of the gripper and relative objects. This pattern lead to encode regularized and useful visual feature extractor. The output of the controller network is the joint configuration which is directly sent to the hardware controller to control the joint angle, see Fig 2.8. They had shown that reconstruction based regularization can significantly improve generalization and robustness, while

training multiple tasks can increase the success rate of all tasks.

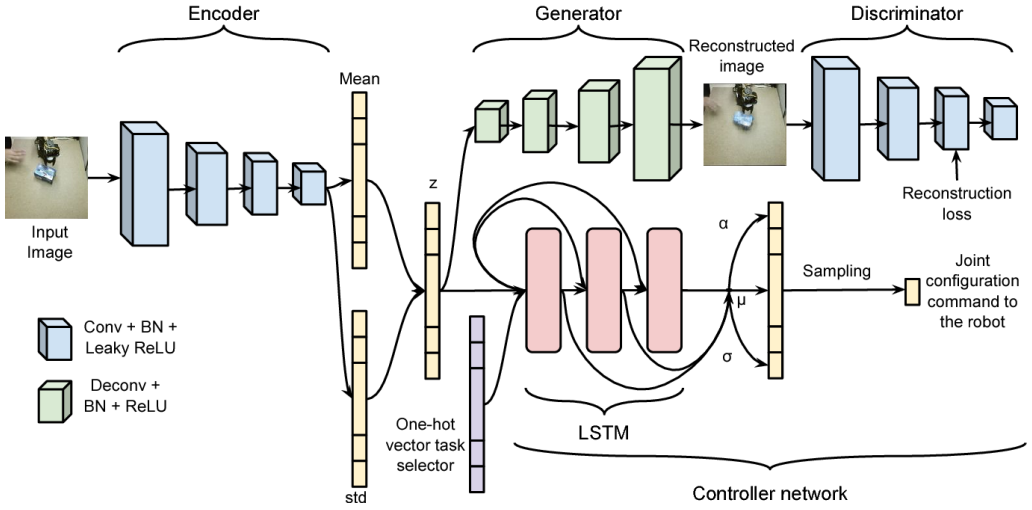

Figure 2.8: Multi-task Learning by Rahmatizadeh et al.

Dynamic Time Warping (DTW)[20] is a method to measure the similarity between two time series. It is mainly used in the field of speech recognition to identify whether two speeches represent the same word. In the time series, the lengths of the two time series that need to compare similarities may not be equal. In the field of speech recognition, the speech speeds of different people are different. And the pronunciation of different phonemes in the same word is different. In these complex cases, the distance (or similarity) between two time series that cannot be effectively solved using traditional Euclidean distances. DTW has been applied in LfD because human cannot demonstrate a set of completely trajectories aligned due to the time, velocity, acceleration variance along with the time. DTW calculates the similarity between two time series by extending and shortening the time series, see Fig 2.9.

[24] proposed a simple but effective approach for solving policy deviation problem. The idea behind this approach is to the variations and similarities between demonstrations to extract what is important to imitate. For instance, the robot performs the same task which grasp a target object in random place and the initial joint configuration is random, under this circumstances, the demonstrated trajectories will different from each other dramatically but for the approached joint configuration, the position of the joint arm should be similar when grasping the object. They call the part of trajectories with low variance a constraint because of the similarity between repeated demonstrations.(see Fig 2.10)

For the same position of the target, multiple trajectories are demonstrated

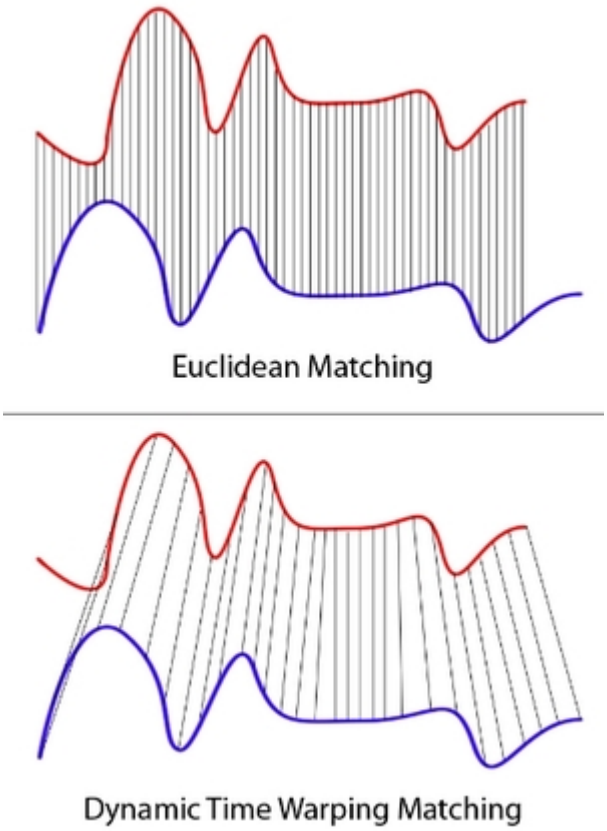

Figure 2.9: Dynamic Time Warping

and recorded in terms of trajectory level, then dynamic time warping is used to align multiple trajectories, then the trajectories is transformed to the space of the target object and the mean and deviation of the multiple trajectories are computed. In the end, the relative object trajectories is joined for satisfying all constraints by multiplication. The limitation, however, is that if paralleling placing the objects during the phase of demonstration, the resulting trajectory cannot perform the task successfully see Fig 2.11. The other demerit is that for improving the performance of the model, human teacher must very carefully demonstrate the behavior or increase the number of demonstrations, instead, in this research, a transform model is proposed to increase the dataset and can contribute to the release of human labor.

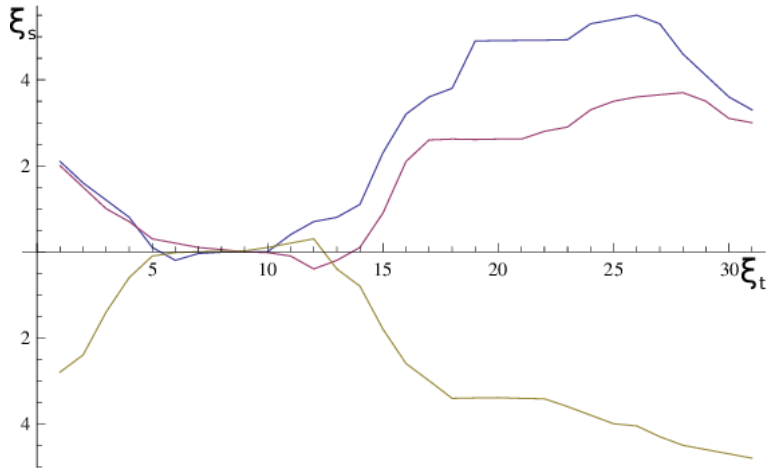

Figure 2.10: Variations and similarities between demonstrations by B. Reiner et al.

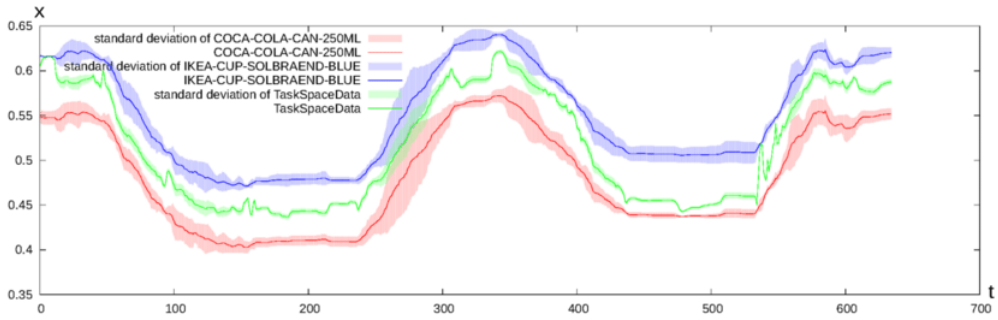

Figure 2.11: Paralleling placing lead to failure of learning by B. Reiner et al.

Chapter 3

Proposed Model

Summary

In this study, at first, in order to avoid the correspondence problem, the interface, which is directly guiding the robot arm to perform the task, is choose. For obtaining a behavior pattern about grasping, we demonstrated the behavior of grasping a cubic block multiple times for the same target by guiding robot hand to perform this task, simultaneously we recorded the arm joint changes in the frequency of 50 HZ, the discrete sampling joint values form a trajectory for each joint of the robot. After obtaining the trajectories of each joint angles changes along with time, An averaged trajectory is obtained by averaging the values at the same time step for each joint of robot. After that, only the target joint configurations for approaching to cubic block placed in different place are demonstrated and the pixel location of the red mark attached on the cubic block as well as the neck angles. In order to transform the example trajectories to the new situation, a transform model, which can transform the example trajectory to the new situation, was derived the input of this transform model is the new initial joint configuration and target joint configuration as well as the example trajectory, the output is a new trajectory which satisfies two requirement, the first one is that the new trajectory must pass through the new initial and target values, the second one is that the new trajectory must have the same tendency compared with the example trajectory. In order to augmenting the dataset, we randomly set multiple initial joint configurations, that is, for one target joint configurations, there will be many trajectories generated. After that, we trained a 3 layer neural network which consists of two Gated Recurrent Unit layers and one output layer for outputting the trajectories for picking up the cubic block, the input is the current joint angles and the pixel location as well as

the current neck angles. Finally a support vector machine with non-linear kernel is used for classifying which hand is preferred for executing the task. Fig 3.1 shows an overview of this research.

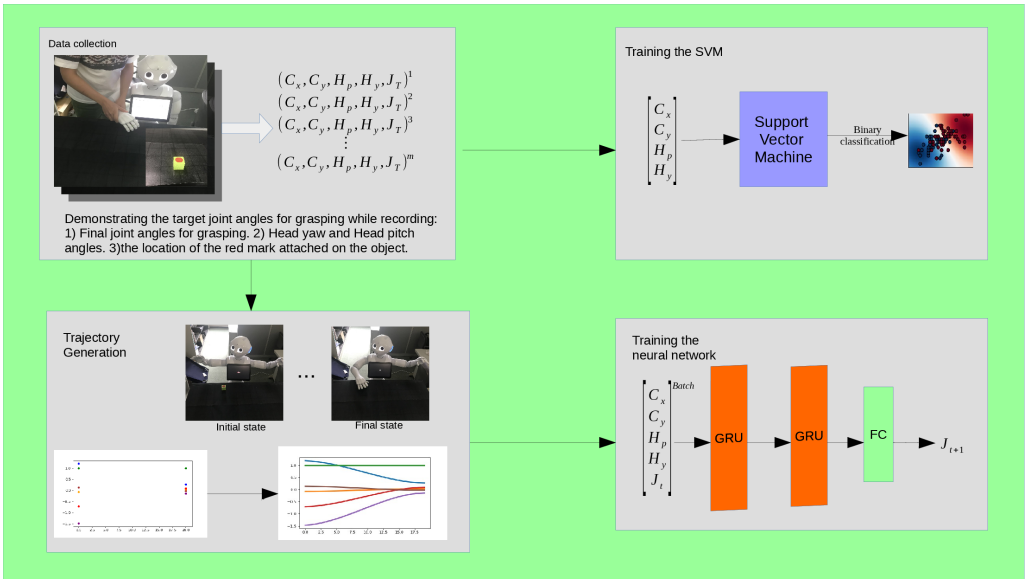

Figure 3.1: An overview of this research.

3.1 Data Collection

In the phase of data collection, there are two stages. In the first stage, demonstrating the behavior to the robot is needed. In order to remove bias and noise, the grasp behaviors were demonstrated multiple times given the same target object and record all the joint values changes along with time, after obtaining the trajectories of each joint angles changes along with time, An averaged trajectory is obtained by averaging the values at the same time step for each joint changes of robot. In the second stage a set of $\{C_x, C_y, H_p, H_y, J_T\}$ are recorded, C_x, C_y refers to the pixel location of the red mark attached on the target object, the location of the red mark is extracted by masking the irrelevant region of the image and compute the center of the red mass. H_p, H_y refers to the head orientations represented by two intersected joint values, this can be differ from different robot configurations. J_T refers to the target joint configurations. a single J_T have multiple $\{C_x, C_y, H_p, H_y\}$ to correspond. These data will be used as the labeled training data and test data for training or evaluating the recurrent neural network and support vector machine.

3.2 Data Generation

In order to release the human labor for demonstrating the behavior, a transform model was proposed for transforming the approximated trajectory to other situation. The objective of the transform model is to transform a known trajectory to other situations given two constrains: initial and final values of the new trajectory. the input of this transform model are a known trajectory, which consists of a series of sampling points, and the new initial and final values of the new trajectory, the output is a new trajectory which satisfies the following conditions: similar shape compared with the original trajectory; pass through the given initial and final points.There are three situations: The new trajectory shares the same initial point but different final point; The new trajectory shares the same final point but different initial point. Both the new initial and final points are different.we use $(j_i, i \in t)$ to represent a single joint value at a certain time step. The transform equation is shown bellow.

For the first situation, because the initial value must keep invariant after transformation, so the formula can be derived as:

$$j_i^{new} = \alpha(j_i - j_1) + j_1 \tag{3.1}$$

we can see that if j_i is j_1, j_1^{new} equals j_1, which satisfies the conditions above. Because the final value derived must be equal to the final value given, so the scale α can be derived as:

$$\alpha(j_t - j_1) + j_1 = j_t^{new} \tag{3.2}$$

$$\alpha = \frac{j_t^{new} - j_1}{j_t - j_1} \tag{3.3}$$

Similarly, the second situation can be written as:

$$j_i^{new} = \alpha(j_i - j_t) + j_t \tag{3.4}$$

$$\alpha = \frac{j_1^{new} - j_t}{j_1 - j_t} \tag{3.5}$$

j_i^{new} is the new joint value, j_i represent the approximated joint value, j_t represents the last joint value of the approximated sequence. Based on the transform equation above, we can transform the behavior to the new situation. For training the Recurrent Neural Network, large data is required to be generated, under this context, the initial joint configuration was change in a certain range in order to enlarge the data set, for instance, if there were one hundred initial joint configurations, for a single target joint configuration, there will be one hundred trajectories generated, this idea was be applied for data augmentation. The figure show bellow illustrates the structure of this transform model, see Fig 3.2.

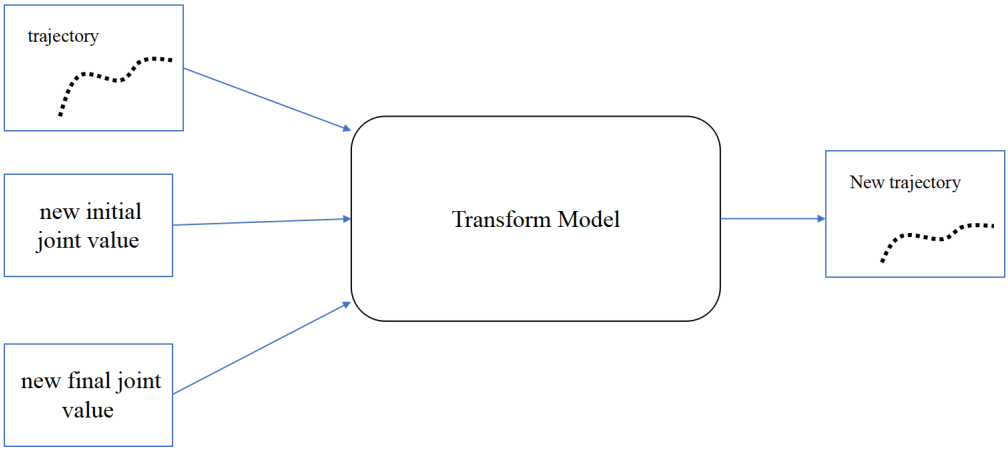

Figure 3.2: Transform model

3.3 Training of recurrent neural network

In this research, 2 Gated Recurrent Unit layers and one output layer was developed as our trajectory generator, The input is the initial joint configurations of the experimented robot and the pixel location of the red mark attached on the cubic block as well as the neck angles, the output is a sequence of joint configurations for approaching to the object. Considering the neck angles can contribute to get rid of the head orientation constrains, that is, if not considering the neck angles, the robot must need to keep the required head orientation to detect the cubic block, that is not natural compared with human being. In the experiment, we found that the output trajectories keep same with the training label at the previous several time step but diverged at the last time steps. The reason is that the error amplified when propagating to the next time step. It is widely accepted that different training fashion lead to different performance even for the same structure of the model. For training the recurrent neural network. Under this context, there are two ways to train the model. The first one is trained by a sequence to sequence fashion, see Fig 3.3. The input are a series sampled value except the last one, the output are a series sampled value except the first one. In other words, the output shifts backward by one time step compared to the input. The second one is trained in a one to many fashion, see Fig 3.4. The input is the first trajectory point, the output are remained sampling points. Once fitting the first trajectory point into the model, then the model needs to generate the whole trajectory. During the training phase, the first one converged faster than the second one , but in the test phase, the model trained in the first

fashion tends to result in non-suitable trajectories, while the model trained in the second fashion can generate smooth as well as accurate trajectories even though it is difficult to converge during training.

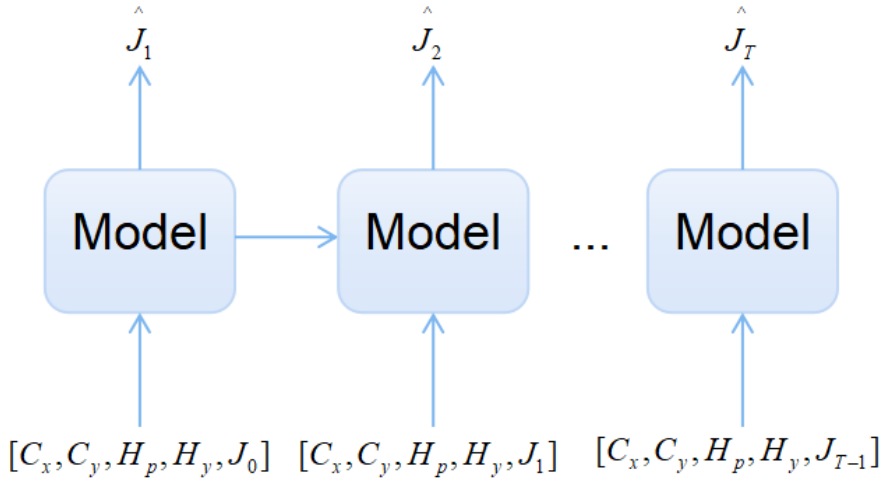

Figure 3.3: Many to many fashion for training RNN

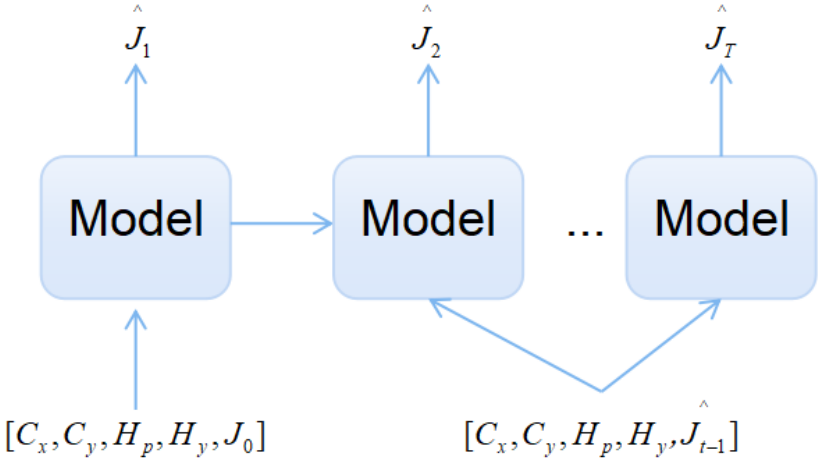

Figure 3.4: One to many fashion for training RNN.

3.4 Training of Support Vector Machine

a Support Vector Machine classifier for hand selection is implemented in this research. The input features are selected as $\{C_x, C_y, H_p, H_y\}$, and the output is a binary signal which inferred to use the left or right arm. Radial Basis Function kernel SVM was used given by

$$K\left(x^{(i)}, x^{(j)}\right) = \phi(x^{(i)})^T \phi(x^{(j)}) = \exp(-\gamma \|x^{(i)} - x^{(j)}\|^2) \tag{3.6}$$

where γ defines how far the influence of a single training example reaches.

Chapter 4

Experimentation

4.1 Physical Environment

In order to simplify the task for object localization, a black colored workbench was made whose height is 83 centimeters from the floor. A 3.5 centimeter cubic block (see Fig. 4.1) was used and attached a red mark on top of the surface of the block to be the target. For convenience, 24 positions were marked on the workbench with the same intervals of 5 centimeters.

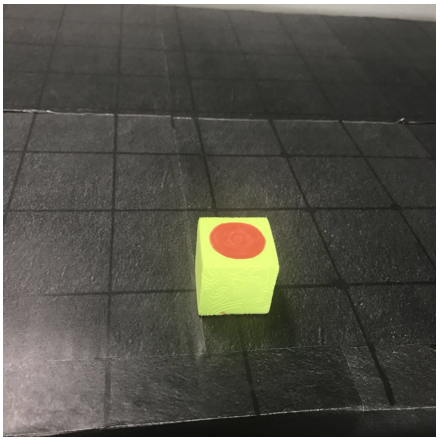

Figure 4.1: Cubic block

In the experiment, the pepper robot (see Fig. 4.2) was used to collect the data as well as to test the proposed recurrent model. Both the left and right arm of pepper robot have five degree of freedom(DOF): two intersect joints at shoulder and elbow and a single joint at the wrist joint. Also there is a intersect joint at robot neck.

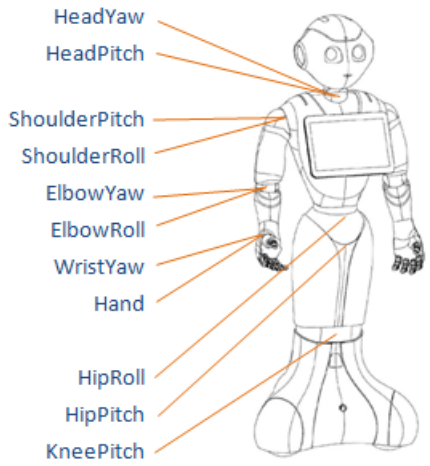

Figure 4.2: pepper robot

The figure (see Fig. 4.3) describes the actuator range for each of the joint, the right arm have the symmetrical property with the left arm. There are two cameras mounted on the robot head, one is mounted on the forehead of the robot, the other one is mounted on the mouth, see Fig 4.4. The RGB camera mounted on the mouth of the Pepper robot is used for recording the instant picture.

The instant joint angles of the robot's left or right arm are also recorded. During the phase of collecting data, the block was placed on the workbench, and then the robot grabs pictures with different head orientations. After finishing taking picture and recording the head orientation, the robot's arm was guided by human to the desired position and recorded the joint angles of robot's right or left arm.(see Fig. 4.5) In this context, the joints changing bellow the robot torso were not considered , during the phase of data collection, we keep the waist and hip joint fixed. the sampling frequency is set to be 50 HZ, the trajectories were downsampled to the length of 25 in order to reduce the noise and we found that downsampling the trajectory can help to make the recurrent neural network more easier to be trained.

4.2 Data collection

Firstly, the robot and target fixed were kept fixed and demonstrated the grasp behavior multiple times to get multiple trajectories, the trajectories were averaged to obtain a general pattern of this behavior, the reason for repeating the behavior multiple times is that a single trial always contains

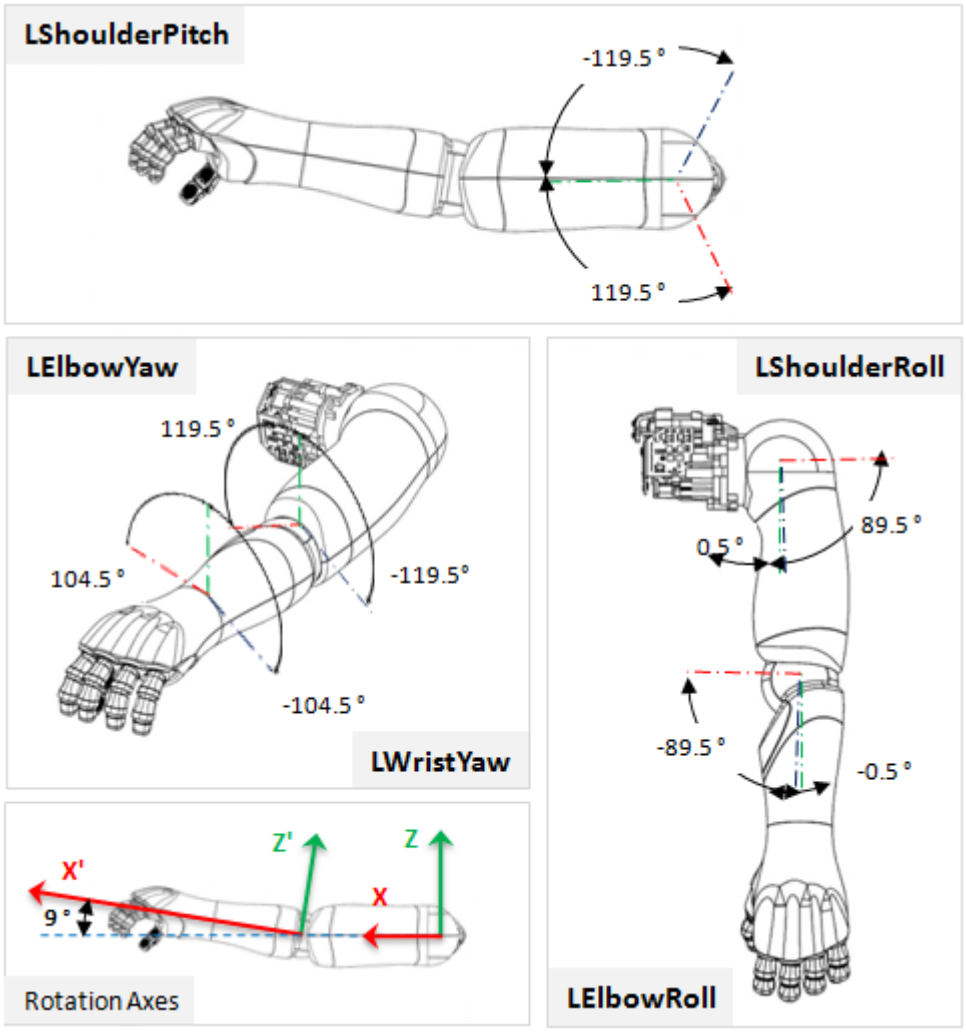

Figure 4.3: Actuator range of the robot arm

bias, averaging process can remove the bias to get a more general pattern. Finally, the transform model derived in Chapter3 was used to generalize this pattern to other cases. The assumption is that grasp behaviors shares same motion pattern, in order to verify this assumption, two different trajectories were collected and repeated the behavior multiple times and averaged it for each of them. The figures bellow illustrates the trajectory(see Fig 4.6), after that, one of them was transformed by using the transform model and put the original trajectory and the transformed trajectory together.(see Fig 4.7) We can see that the transformed trajectory has the similar tendency compared with the other trajectories.

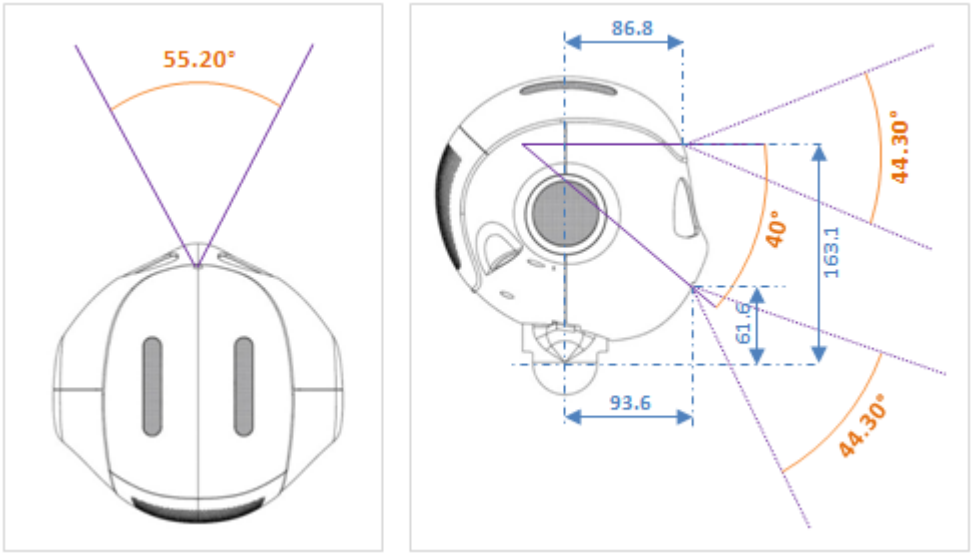

Figure 4.4: Sensing range of pepper robot

Figure 4.5: Human demonstration

In this phase, 12 target joint configurations for grasping the object were demonstrated, at the same time, instant neck angle values and the pixel location of the red mark were also recorded. There were 49 instant picture taken and the corresponded instant neck values. In order to enlarge the dataset, the initial joint value was randomly set in a certain range and use

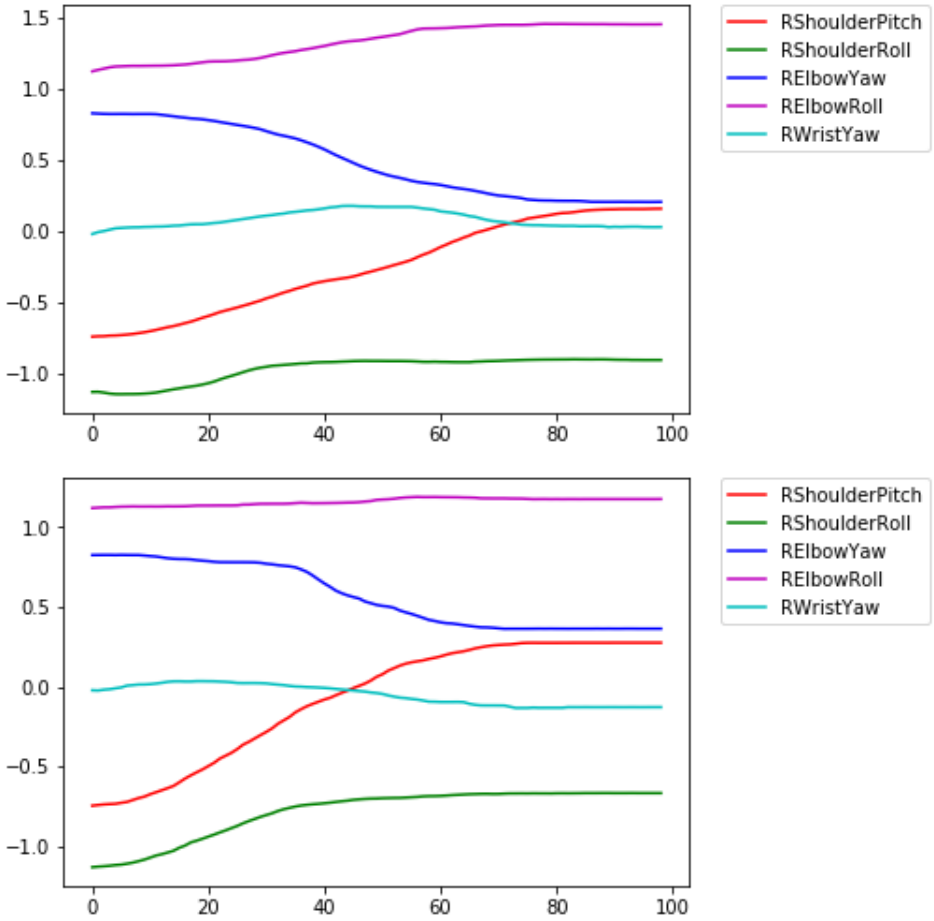

Figure 4.6: Demonstrated trajectory

the transform model to generate the trajectories. Finally, 150528 trajectory sets were collected for each of the robot arm, each trajectory set contains 5 trajectories which represent the joint of the robot arm. For training the trajectory generator, there was ninety percent of the dataset used for training, ten percent of the dataset used for testing. For training the hand selection model, ten-fold cross validation is used.

4.3 Recurrent Neural Network

Based on [17], initially 3 LSTM layers and 1 output layer were used as the model of trajectory generator, each hidden layer has 40 neurons,the size of input is 9, the size of the output is 5. Due to the range of actuator, there

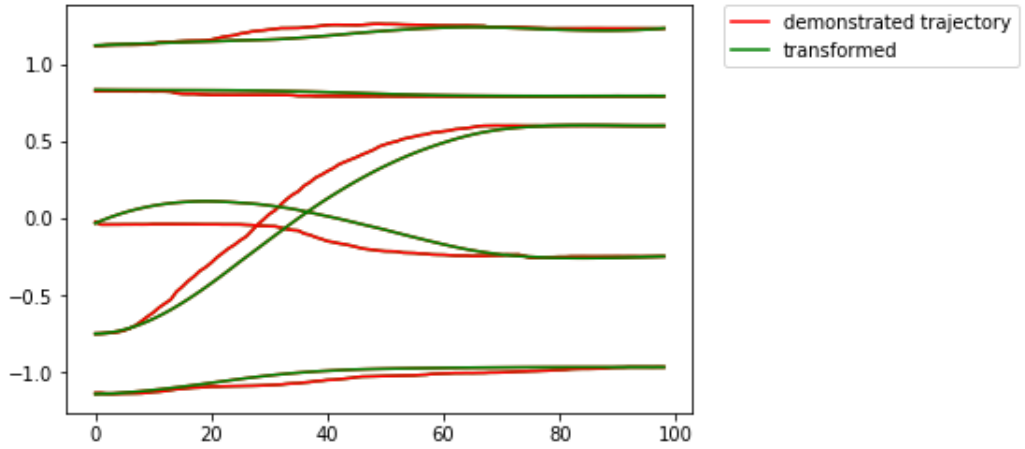

Figure 4.7: Visualization of the transformed trajectory

are no activation function in the output layer, but the problem is that this model needs long time to train, then the LSTM layers were replaced by GRU layer and found that the model converged faster and have the very same performance compared with the original model, to make the model more efficient, wonly two Gate Recurrent Unit layers and one fully connected layer were used. this model achieve the similar performance on the test set. The robot controller takes $\{C_x, C_y, H_p, H_y, J_0\}$ as input and outputs the whole trajectories leading to the final joint angle. During the training phase, the first one converged faster than the second one and the loss function also decreased to less than 10^{-6}. But in the test phase, the model trained in the first fashion tends to result in non-suitable trajectories, while the model trained in the second fashion can generate smooth as well as accurate trajectories even though it spent more time to converge. The Adam optimizer was used for optimizing the model and learning rate was set to be 0.001, the training process was stopped at 200 iterations.

4.4 Support Vector Machine

A Support Vector Machine was developed for hand decision problem. There are two hyper-parameters needed to be set: C and γ, The C parameter tells the SVM optimization how much you want to avoid misclassifying each training example. For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even

if that hyperplane mis-classifies more points. γ can be seen as the inverse of the radius of the influence of samples selected by the model as support vectors.

4.5 Data Flow

The flow chart shown bellow illustrate the processing for grasping the target, see Fig 4.8.

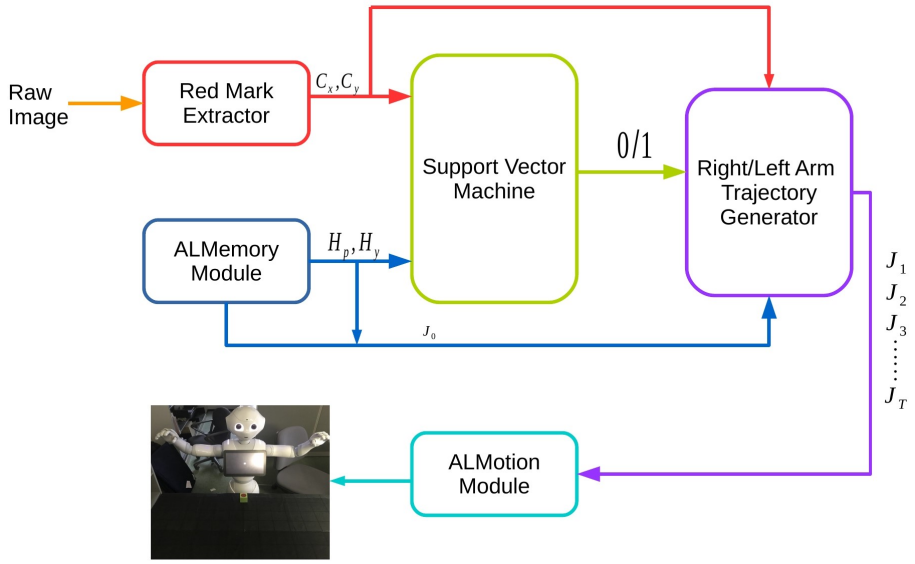

Figure 4.8: Data flow of the research

Firstly, the location of red mark attached on the cubic block was extracted. The location of the red mark can be represented as $[C_x, C_y]$ respective to the left top corner of the image. The resolution of the image captured by the camera mounted on the mouth of the pepper robot is 340x420, then $[C_x, C_y]$ were scaled to the range of $(0, 1)$.

ALMemory is a centralized memory used to store all key information related to the hardware configuration of the robot. More specifically, ALMemory provides information about the current state of the actuators and the sensors. The current neck angles which consist of head pitch and head yaw angles are extracted by using ALMemory API. After that, a binary classifier, more specifically, a support vector machine with a non-linear kernel called radial based function is used to infer the relative position of the robot torso and the target object, 0 refers to the left arm and 1 refers to the right arm respectively.

After determining which arm need to be used for grasping the target object, a trajectory generator, a recurrent neural network is used for generating a series of joint values which form five trajectories corresponding to each joint of the robot arm. The input of this model is the location of the red mark and the current neck angles as well as the left/right arm current joint values. the output is a matrix with the shape of 25×5, 25 means there are 25 time steps, for each time step, this model output a 5 dimensional vector referring to the joint value of the robot.

The ALMotion module provides methods which facilitate making the robot move. It contains four major groups of methods for controlling the joint stiffness or joint position. Under this context, ALMotion module is used for controlling in the joint space. In ALMotion, every time you call a public method to request a motion, a motion task is created to handle the job. During the phase of demonstrating the target joint configurations, we set the stiffnesses of the joints of robot to be zero in order to move the arm of the robot freely.

Chapter 5

Evaluation

5.1 Evaluation of trajectory generator

For the evaluation of the recurrent model for trajectroy generation, the test set for testing trajectory generator which is the recurrent neural network is used. first the input of the test set was fed into trajectory generator, a sequence of joint angles are obtained, Then the forward kinematics was used to obtain the position of end effector at the last time step output relative to the robot torso. The Pepper robot has 5 degrees of freedom for each arm, and the position of cubic block with respect to the robot torso was calculated by multiplying five transformation matrix. the test labels are also a sequence of joint values corresponding to each robot arm joint, the target joint configuration is the last time step value of the five trajectories corresponding to each joint, the position of end effector given the target joint configuration can also be obtained by calculating forward kinematics. once both of the end effector positions are calculated, one is obtained by recurrent neural network, one is obtained by test label, the mean squared error is computed, the result are shown in Fig 5.1 and Fig 5.2 The model is non-sensitive to the initial joint configurations, that is because the transformed model is used to augment the training dataset by randomly set multiple initial joint configurations for a single target joint configurations, also large dataset can effectively avoid the problem of overfitting. Fig 5.3 shown that even though the initial joint configurations are different for the same target, the trajectories of same color converged in the end. the models were compared which are trained by different training sets: one is the augmented by the proposed method, the other one is the origin dataset. We also use forward kinematics to compute the hand position relative to the robot torso and then use mean squared error to calculate the error between the taught position

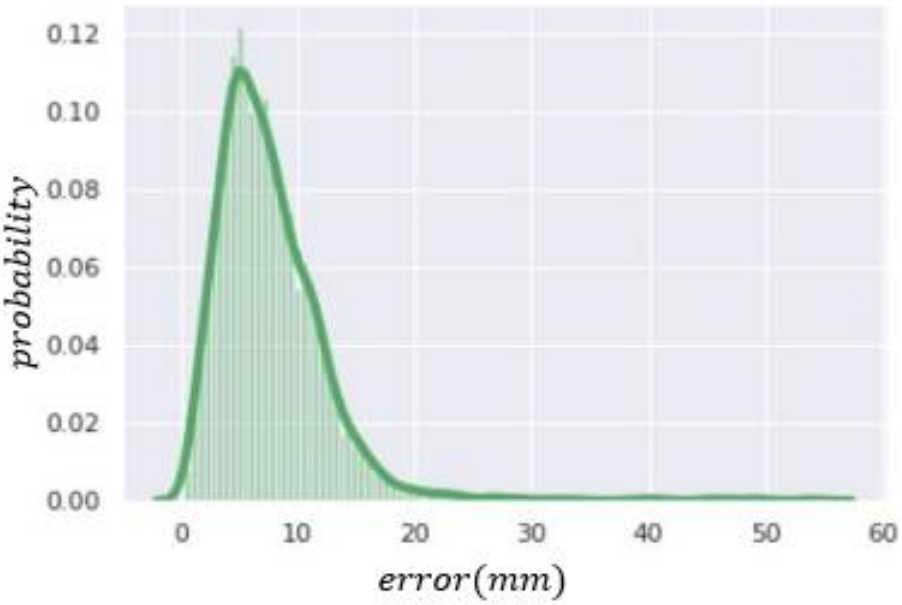

Figure 5.1: Error distribution evaluated by Forward Kinematics for left robot arm.

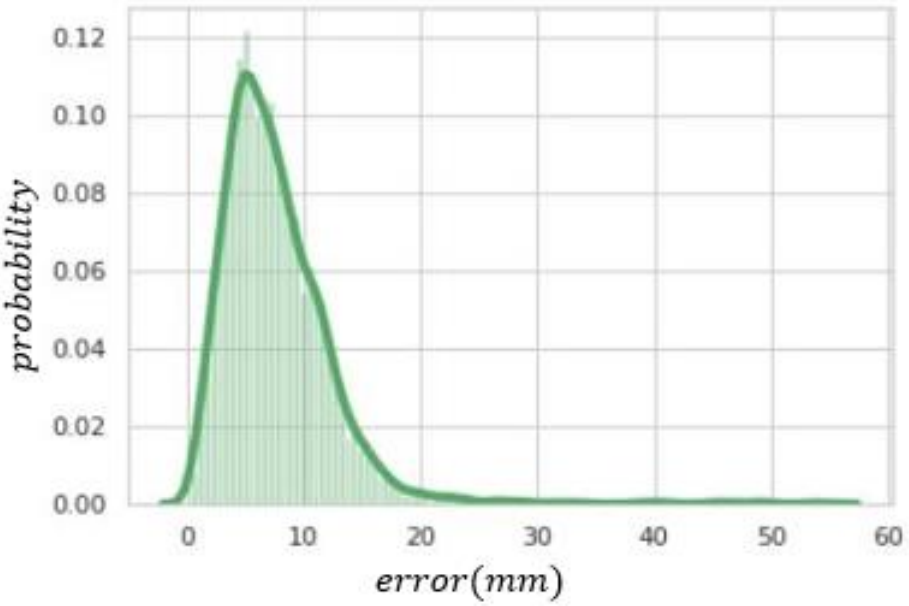

Figure 5.2: Error distribution evaluated by Forward Kinematics for right robot arm.

and the calculated position. The result is shown in Fig. 5.4. It shows that

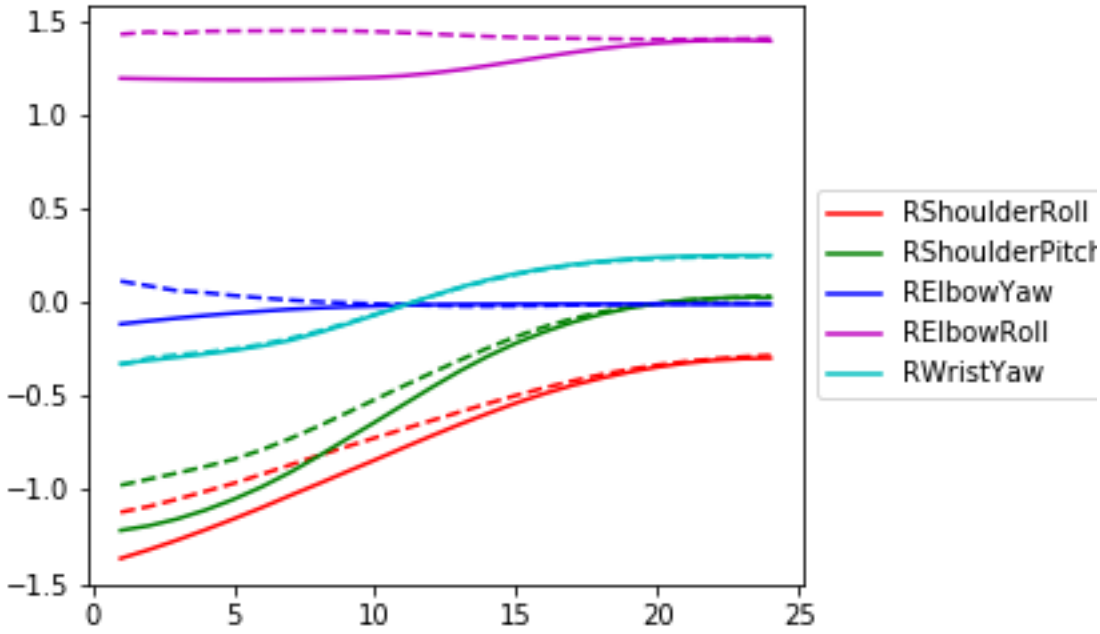

Figure 5.3: Grasping the same target with different initial joint configurations

the model trained by augmented data outperforms the model trained by the original demonstrated data. During the data augmentation section, we augmented the data by randomly changing the initial state within a small range, which is 0.2 radians for each of the joint. After finishing training the model, we random selected a test data from the test set, and generated 1,000 initial states by adding a small perturbation to each of the joint values. Then we input these initial states to our model and pick up the final states. We used forward kinematics to calculate the hand position relative to robot torso and compared with the demonstrated position. We use mean squared error to calculate the distance between them. We found that the perturbed initial joint angles can be tolerated up to a maximum of 0.2 radians, and clearly do not lead to the prediction error. Fig. 5.5 shows the error distribution due to initial perturbations.

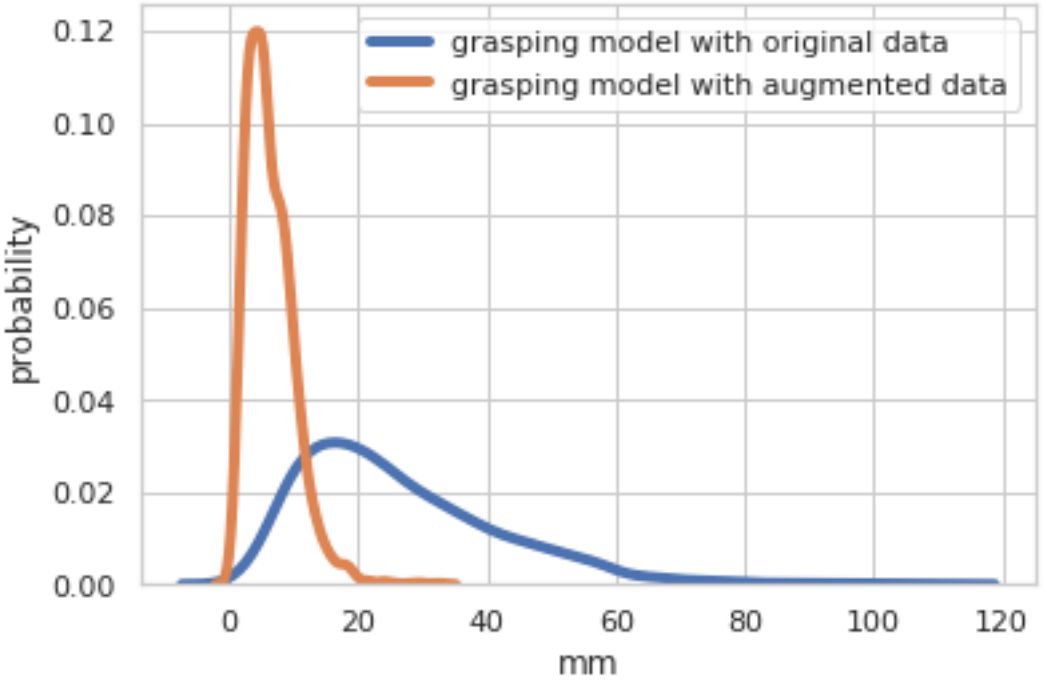

Figure 5.4: Comparision between models with different training data

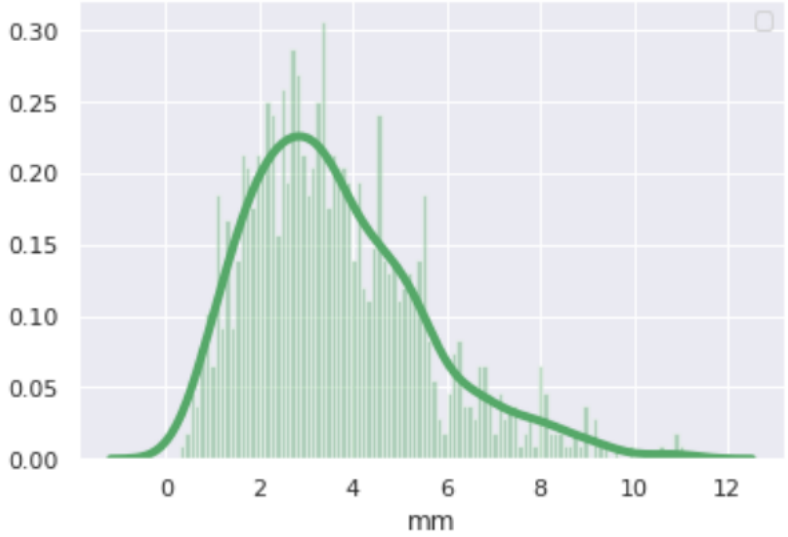

Figure 5.5: Disturbition to the initial joint configurations.

5.2 Evaluation of the hand selection model

We use $\{C_x, H_p\}$ as the feature for visualization. Fig. 5.6 shows that when C equals to 100 and γ equals to 0.1, the SVM exhibit the best classification performance on the test set. For visualizing the result, $[C_x, H_p]$ is selected as the features.

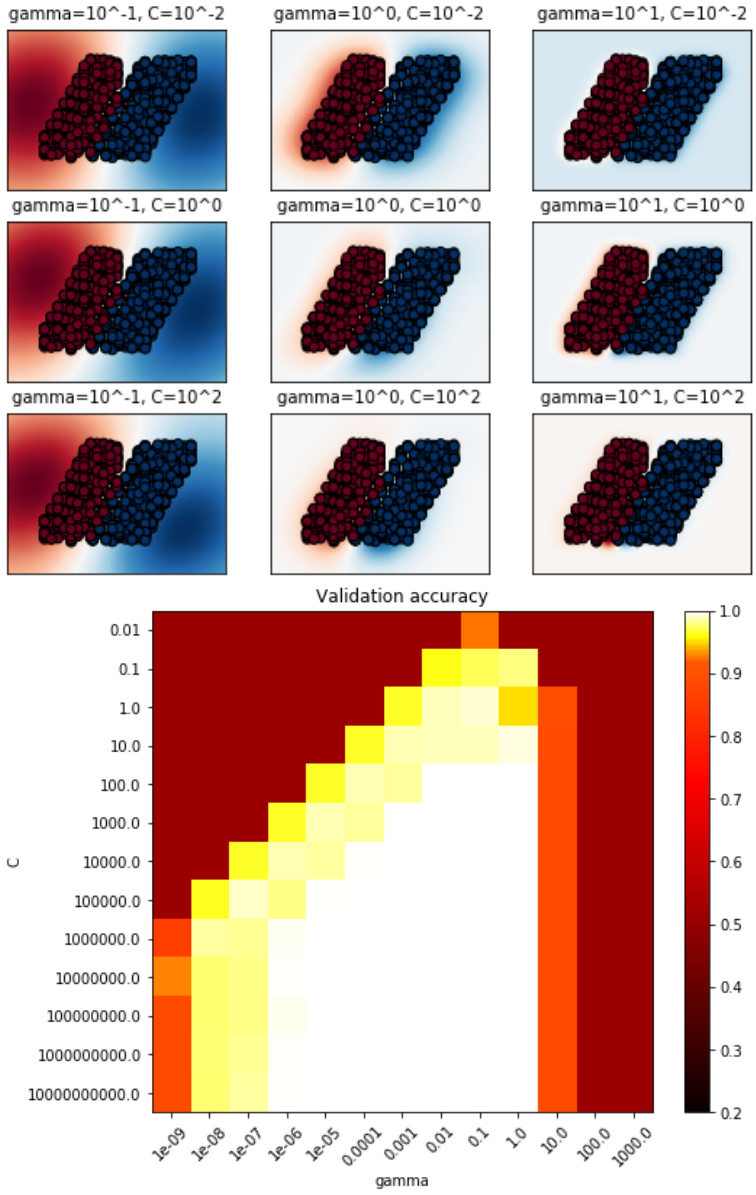

Figure 5.6: Visualization of SVM with different hyperparameters.

Chapter 6

Conclusion and Future work

6.1 Conclusion

a new grasp learning by demonstration algorithm was proposed for a dual arm humanoid robot with joint compliance. the recurrent model can generate stable and smooth trajectories for grasping the object and this model is robust to the changes in the initial state of robot arm joints. the neck angle changes were also considered which make the robot behavior more natural and enlarge the sensing range for the camera. the proposed data augmentation method was very successful in improving the convergence of the recurrent neural network and the smoothness of the trajectory, also the augmentation method make the recurrent neural network avoid overfitting which is caused by bad data or small dataset. The support vector machine classifier with non-linear kernel was capable of deciding which arm needs to be used for grasping based on the head orientation and object location in RGB image features.

The proposed model has some limitations: first of all, it cannot accurately generate trajectories when the object is placed on a different height workbench. Inspired by humans, the camera was used twice to see the object with different head orientations, that is, the input is two locations of the red mark subject to the neck changes and the initial joint configuration as well as the current neck angles, and then used it to train the same model with different input sizes. The result was encouraging, but still needs to be improved. The depth camera mounted on the robot's right eye was also considered to measure the distance between robot and target object, however, due to the measurable range limitations, it cannot sometimes detect the object.

Compared with [24] which needs to demonstrate the task multiple tasks manually, our proposed data augmentation method can reduce the human

labor dramatically and enlarge the dataset for training.

6.2 Future Work

In the future, the parallel camera will be considered which could infer the depth information to sense the position of the object. the waist joint and mobility of the robot will be took into consideration for enlarging grasping range and capability. More complex features such as the shape of the object or category of the object will be considered.

6.3 Acknowledgement

I would like to thank my advisor Prof. Nak Young Chong, for his kindly guidance during the whole period of master study in JAIST. This thesis could not be done without his insightful thinking.

I would like to thank my parents, they always support behind me and encourage me to do my best.

I would like to thank my girl friend Lu Hanzi, she always comforts me when I was depressed.

I also want to thank Mr. Tuyen for his kindly help during the master study.

Bibliography

[1] Task-Informed Grasping (TIG) for rigid and deformable object manipulation. https://www.birmingham.ac.uk/research/activity/metallurgy-materials/robotics/workshops/task-informed-grasping-objects-manipulation.aspx.

[2] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell and A. G. Billard.: Learning and Reproduction of Gestures by Imitation. IEEE Robotics and Automation Magazine, vol. 17, no. 2, pp. 44-54, June 2010.

[3] P. Pastor, H. Hoffmann, T. Asfour and S. Schaal.: Learning and generalization of motor skills by learning from demonstration. 2009 IEEE International Conference on Robotics and Automation, Kobe, 2009, pp. 763-768. doi: 10.1109/ROBOT.2009.5152385

[4] J.Sung, S.H.Jin, and A.Saxena.: Robobarista: Object part-based transfer of manipulation trajectories from crowd-sourcing in 3d point-clouds. International Symposium on Robotics Research (ISRR),2015.

[5] M.Kopicki, R.Detry, M.Adjigble, R.Stolkin, A.Leonardis, and J.L.Wyatt.: One-shot learning and generation of dexterous grasps for novel objects.The International Journal of Robotics Research.vol.35, no.8,pp.959976,2016.

[6] A. L. P. Ureche, K. Umezawa, Y. Nakamura and A. Billard.: Task Parameterization Using Continuous Constraints Extracted From Human Demonstrations. IEEE Transactions on Robotics, vol. 31, no. 6, pp. 1458-1471, Dec. 2015.doi: 10.1109/TRO.2015.2495003.

[7] Rok Pahic.: Deep learning in robotics.

[8] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning.: A survey of robot learning from demonstration. Robot. Auton. Syst. 57, 5 (May 2009), 469-483. DOI=10.1016/j.robot.2008.10.024. http://dx.doi.org/10.1016/j.robot.2008.10.024.

[9] Rahmatizadeh, Rouhollah and Abolghasemi, Pooya and Blni, Ladislau and Levine, Sergey. (2017). Vision-Based Multi-Task Manipulation for Inexpensive Robots Using End-To-End Learning from Demonstration.

[10] B Kehoe, A. Matsukawa, S. Candido, J. Kuffner, and K. Goldberg, Cloud-based robot grasping with the Google object recognition en- gine, in IEEE International Conference on Robotics and Automation (ICRA) , pp. 42634270, 2013.

[11] M. Forbes, M. J.-Y. Chung, M. Cakmak, and R. P. Rao, Robot programming by demonstration with crowdsourced action fixes, in Second AAAI Conference on Human Computation and Crowdsourcing , 2014.

[12] C. Crick, S. Osentoski, G. Jay, and O. C. Jenkins, Human and robot perception in large-scale learning from demonstration, in Interna- tional conference on Human-robot interaction , pp. 339346, ACM, 2011.

[13] Z. Fang, G. Bartels, and M. Beetz, Learning models for constraint- based motion parameterization from interactive physics-based simula- tion, in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) , pp. 40054012, 2016.

[14] S. Calinon, F. Guenter, and A. Billard, On learning, representing, and generalizing a task in a humanoid robot, IEEE Transactions on Systems, Man, and Cybernetics , vol. 37, no. 2, pp. 286–298, 2007.

[15] S. Calinon, F. Dhalluin, D. G. Caldwell, and A. Billard, Handling of multiple constraints and motion alternatives in a robot programming by demonstration framework., in IEEE International Conference on Humanoid Robots (Humanoids) , pp. 582–588, Citeseer, 2009.

[16] Tadanobu Inoue , Subhajit Chaudhury , Giovanni De Magistris and Sakyasingha Dasgupta.: Transfer learning from synthetic to real images using variational autoencoders for robotic applications.(2017)

[17] Giovanni De Magistris, Asim Munawar, Phongtharin Vinayavekhin.: Teaching a Robot Pick and Place Task using Recurrent Neural Network. ViEW2016, Dec 2016, Yokohama, Japan. $< hal - 01426846 >$

[18] Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, Pieter Abbeel.: Deep Spatial Autoencoders for Visuomotor Learning. arXiv:1509.06113 (2015)

[19] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio.: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv:1412.3555(2014)

[20] Silva, D. F., Batista, G. E. A. P. A. (2015). Speeding Up All-Pairwise Dynamic Time Warping Matrix Calculation.

[21] Ale Ude, Christopher G. Atkeson, Marcia Riley, Programming full-body movements for humanoid robots by observation,Robotics and Autonomous Systems,Volume 47, Issues 23,2004,Pages 93-108.

[22] Calinon, Sylvain, Billard, Aude. (2008). A framework integrating statistical and social cues to teach a humanoid robot new skills.

[23] Nichola Abdo, Luciano Spinello, Wolfram Burgard, and Cyrill Stachniss Inferring What to Imitate in Manipulation Actions by Using a Recommender System IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 2014.

[24] B. Reiner, W. Ertel, H. Posenauer and M. Schneider, "LAT: A simple Learning from Demonstration method," 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, 2014, pp. 4436–4441.

[25] Hochreiter Sepp, Schmidhuber Jrgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.