Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

Simultaneous Multi-task Determination And Manipulation With Vision-based Self-reasoning Network

Zhifeng Qian[1*], Xuanhui Xu[1*], Mingyu You[12&], Hongjun Zhou[1]

***Abstract*— It remains a great challenge for robots to simultaneously determine the task objective and schedule the corresponding manipulation, which requires robots to autonomously perceive and understand the environments, think up a response strategy and plan a set of trajectories or actions. This paper proposes a muti-task self-reasoning neural network (MSRnet), which is an end-to-end imitation learning framework consisting of a self-reasoning module and a control module. The self-reasoning module supports the task objective inference, while the control module predicts the robot motor angles for manipulation. With multi-task learning, MSRnet enables robots to accomplish multiple tasks with one model without updating parameters. This paper takes three tasks as an example, involving pouring water into a cup of coffee powder, taking a spoon and stirring coffee with the spoon, which simulate the application scene of making coffee. We employ a low-cost robotic arm(less than $300) to evaluate our MSRnet on coffee maker(CM) dataset which is collected by ourselves. MSRnet with 640*480 resolution inputs can achieve 83.3% success rate on multi-task test, 76.7% success rate on complex environment test. The result verifies that MSRnet can accurately infer the task objective and generate corresponding task actions simultaneously.**

***Index Terms*— imitation learning, self-reasoning, multi-task manipulation**

I. INTRODUCTION

Plenty of research shows that robot learning of manipulation tasks from observing expert demonstrations is an outstanding approach [1] [2]. Robots can acquire a range of skills such as pushing, reaching [3], grasping [4], picking and placing [5], and complex cleanup tasks [6]. However, most approaches for robot imitation only focus on accomplishing a single task. It remains a challenge to maintain multiple manipulation skills with one model. In addition, autonomously switching task skills in complex environments is even more difficult. Robots need to reason out the task objective with current observations.

Service robots are widely expected in various applications. To be smart, they should autonomously determine the task objective and accomplish multiple tasks. For instance, in order to make a cup of coffee, robot needs at least the ability of pouring water, taking a spoon and stirring coffee. Having these proud potential observations, robot must infer the current task category itself instead of being specified by human. Then the robot generates actions corresponding to the task target. In other words, robots are expected to be conscious of "which task should I do" and obtain the skill of "how to do it" simultaneously based on the environment understanding.

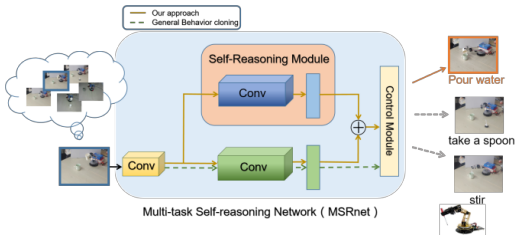


Fig. 1. The goal of our MSRnet is to maintain 3 manipulation skills with visual input. Our model takes the current observations as input and generates the task objective and the joint angles of the robotic arm simultaneously as the output. Our model learn to accomplish 3 tasks without updating model patameters. For example, the end-to-end system catches sight of a cup on the table and the robot also hold another cup, so the system can understand that task objective is "pouring water", and robot motor action of "pouring water" is also generated simultaneously.The green line represents traditional Behaviour Cloning, while the brown line represents our specific approach.

Imitation learning enables robots to efficiently learn the desired behavior from the experts' demonstrations [4] [7] [8]. Traditional robot imitation learning methods usually employ manual programming models to cope with the different scenes without understanding environments. Recently, some research concentrates on Model Agnostic Meta-Learning(MAML) [9] and one-shot or few-shot learning [5] in order to help robots quickly adapt to new tasks. However, these approaches only acquire one specific task skill at a time and forget the previous skills after updating the parameters of the model. Consequently, they have no ability to complete multiple tasks with only one model. Besides the multi-task maintaining, switching tasks is also a challenge for robots. Some approach [10] roughly selects the output network architecture for the specific task. Other approach artificially selects the task objective by a task selector vector without environmental perception and analytical reasoning. The core issue is the ability of reasoning based on the environment understanding. If the target can't be correctly inferred, action execution will be even harder.

In this paper, we propose a vision-based Multi-task Self-

[1]The School of Electronics and Information, Tongji University, Shanghai, China.
[2]Shanghai Institute of Intelligent Science and Technology, Tongji University, Shanghai, China.
*Zhifeng Qian and Xuanhui Xu contribute equally to this work.
&Mingyu You is the corresponding author (myyou@tongji.edu.cn).

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

Reasoning deep neural network(MSRnet) to enable robotic arm to determinate the task objective and perform the corresponding manipulation task simultaneously. An overview of our approach is illustrated in Fig.1. This paper takes three tasks as an example to simulate the application scene of making coffee. Considering the complexity of the scene, imitation learning based on deep convolutional networks is employed to increase the convergence speed and reduce the cost of training [11]. Compared to traditional behavioral cloning methods, we introduce the self-reasoning module to figure out the target task from the three candidates based on the environmental perception. MSRnet can predict joints of the low-cost robotic arm for three tasks without updating parameters of the model. To the end, we evaluate our MSRnet on the dataset of three tasks collected by ourselves and verify its feasibility.

Our main contributions in this paper are as follows:

1) We propose a specific Multi-task Self-Reasoning deep neural network (MSRnet) to enable robots to acquire multiple manipulation skills accurately without updating parameters of the model.
2) We design a self-reasoning module for robot to automatically decide the task objective in a complex scene while executing behavior cloning.
3) We enable robots to simultaneously determinate the task objective and perform multiple manipulation tasks with an end-to-end deep neural network.
4) We show that it is feasible to use a low-cost, sensorless robotic arm to perform multiple complicated manipulation tasks.

The organization of the paper is as follows. First, we review the related work in Section II. Data collection and our neural network architecture are explained in Section III. In Section IV, we discuss our experiments to validate the proposed method. At last, Section V concludes the paper.

II. RELATED WORK

Traditional imitation learning methods usually employ action generation models, such as Dynamic Movement Primitives(DMPs) [12] [13] [14], to generate actions. However, these methods demand researchers to establish a mathematical model of the task and manually program to control robots. As robots move from simple scenes to complex scenes, it brings impossible missions to establish precise models for all possible tasks. Moreover, traditional methods can't infer the task objective.

Along with the development of computer vision and pattern recognition, models' ability of environment understanding has been greatly developed. This lays a theoretical foundation for our work. In 2012, Alex Krizhevsky et al. [15] from University of Toronto presented AlexNet which achieved top-1 error rate of 37.5% and won the championship of the ILSVRC-2012 competition. Since then, a series of well-known basic networks have been gradually proposed. Such as Vgg [16], Inception [17] and DarkNet [18]. Benefit from the advancement of these methods, we can extract features from raw images. Different networks are evaluated in our work for generalization, which is illustrated in section IV.

Recently, Rouhollah Rahmatizadeh et al. [6] proposed a technique which joins a number of modules for vision-based multi-task manipulation, such as VAE-GAN and RNN. Their model takes images as input and the parameters are shared across the tasks, which means their model has the ability to maintain multiple manipulation skills. However, the approach needs an additional one-hot vector to artificially select task. In practice, this method is time consuming and human beings are heavily dependent. Lerrel Pinto et al. [19] from Carnegie Mellon University proposed an end-to-end learning framework for multi-task learning. This framework shares feature extraction network across different tasks and each task has a control module. Although the results indicate that their model achieves high success rate on all tasks, these models don't have the ability to understand the environment or determine the task objective. The performances of these models are far from our goal.

Since 2017, a lot of researchers have paid attention on the intersection of few-shot learning [20] [3] [5] and imitation learning [21] [22] [23]. Imitation learning, which allows agents to learn behavior from expert demonstrations [1], is widely considered to learn autonomous policies in fields such as robot, computer games, and autopilot. Imitative learning has achieved good results in the complex behavior learning. On the intersection of few-shot learning and Imitation learning, Pieter Abbeel who proposed Model Agnostic Meta-Learning (MAML) [9] and his team did numerous works. They aim to complete three tasks: push, reach and pick-and-place. Their methods seem to be able to complete multiple tasks within one frame. Although impressive, those methods can't enable robots to simultaneously maintain multiple manipulation skills. In the wake of the task changing, they have to fine-tune their model with a few demonstrations. Forgetting the previous skill will be inevitable.

Simultaneous multi-task determination and manipulation is still a challenging job. The previous methods could neither simultaneously acquire multiple abilities nor autonomously determine the task objective. Compared with the above works, our approach provides a state-of-art framework in multi-task learning. This framework can simultaneously infer the task objective and predict the trajectories of the robotic arm.

III. METHOD

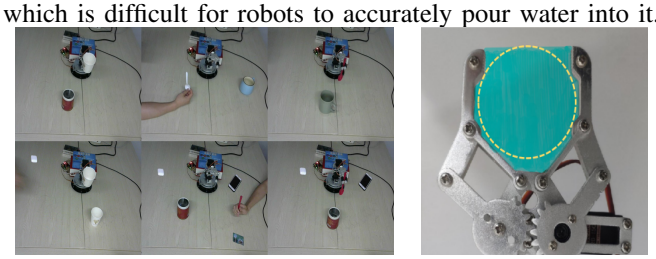
In this section, we introduce the collected dataset containing demonstrations for three tasks with a low-cost robotic arm. We also illustrate the architecture of the proposed Multi-task Self-Reasoning network (MSRnet).

A. Data Collection

In order to emphasize the importance of task objective determination and multi-task learning, this paper takes three tasks as an example to simulate the application scene of making coffee, involving pouring water into a cup of coffee powder (task 1), taking a spoon (task 2) and stirring coffee with the spoon (task 3). These tasks are different and need robots to extract valid features to distinguish the slight different of input and reason out the appropriate task to cope with.

We collect the CM dataset considering that there is no available dataset for these three specific tasks. We assemble a 6-DoF robotic arm which meets the requirements of light weight and low price. To collect demonstrations, we control the robotic arm to complete the task through manual programming. We take images of three tasks with a RGB camera from the third-person perspective. The angle of the camera is fixed in order to make the teachers' perspective consistent with the students' .

The collected demonstrations $\mathcal{D} = \{o_i, u_i\}_i^T$ consist of 640*480 RGB images and the joint angles as labels. u_i is the joint angles of robotic arms and o_i is the observations of the environment. Based on the joint angles, we can calculate the coordinates of the end effector which are approximately at the center of the cup rims. We collect 420 demonstrations for three tasks in total. Each task includes 100 demonstrations in simple scenes and 40 ones in complex scenes. The examples of the dataset are illustrated in Fig.2(a). Only robotic arms and target objects are on the table in the simple scenes while there are other disturbing objects like a phone, a key or a pen on the table in complex scenes. In order to improve the understanding of the scene and prevent the model from overfitting, we use five cups of different colors, sizes and height as target containers and two different colored spoons. The rim of the cups is about 9 to 13 centimeters in diameter, which is difficult for robots to accurately pour water into it.



(a) (b)

Fig. 2. (a) are examples of the dataset. From left to right, it represents task 1, task 2 and task 3. The first line represents demonstrations in the simple scenes while the second line represents demonstrations in the complex scenes. All the images are the initial states of each task. (b) shows that when robot gripes the spoon, the spoon should be in the blue area. And The blue area can be approximated as the yellow circle.

B. Multi-task Self-Reasoning network Architecture

A vision-based Multi-task Self-Reasoning deep neural network (MSRnet) is proposed, which is illustrated in Fig.3.

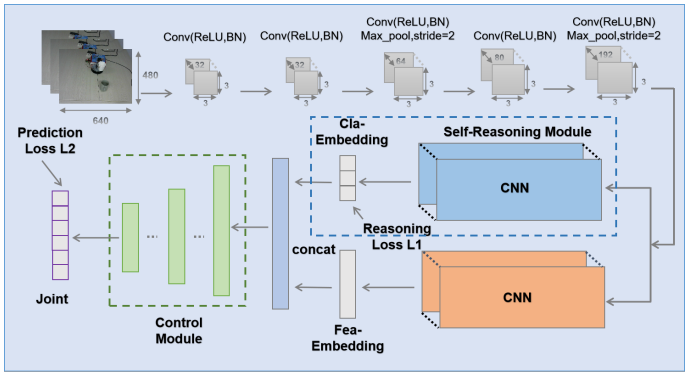


Fig. 3. The architecture of Multi-task Self-Reasoning network(MSRnet). It consists of a self-reasoning module that can reason out the task objective, and a control module that outputs joint angles to control the robotic arm. Multiple tasks share the parameters of the first few convolution layers.

We modify the architecture of Inception_V3 [17] which is balanced in network depth and computational efficiency. Inception_V3 network has five convolutional layers followed by 42-layer inception modules which use dimensionality reduction and re-aggregation to improve the understanding of the environment and achieve superior performance in image classification tasks.

As Fig.3 shows, the first five convolution layers of Inception_V3 with relu activation, max pooling and batch normalization process the input images and output the common feature map which is input to two parallel branches. Both the structure of the two branch networks are the rest of Inception_V3, 42-layer inception modules. The upper branch which we named self-reasoning module, is followed by a softmax function to output a three-dimensional cla-embedding. The lower branch further extracts features from the common feature map and outputs a 2048-dimensional fea-embedding, just like the original Inception_V3 network. Then the embedding of the two branches are concatenated into the control module consisting of five fully connected layers. Finally, the control module predicts six joint angles to control the robotic arm.

In MSRnet, the first five layers are designed to extract the key information of the input image, such as the thing held in the end effector of the robotic arm, the position of the cup on the table and whether someone is handing over the spoon. The self-reasoning module is different from the network architecture of general behavioral cloning methods. Instead of artificially giving a task selection vector, the self-reasoning module learns the difference between tasks from the common feature map, and reasons out the task objective when facing with different scenes. The control module learns the mapping from the extracted feature to the joint angles of the robotic arm.

We introduced a specific loss function to measure the similarity between learned policy and demonstration policy. One component of the total loss function is reasoning loss $\mathcal{L}_1$, which is the cross entropy loss between the cla-embedding

output from the self-reasoning module and the real label

$$\mathcal{L}_1 = -\sum_{i=1}^{M} y_i log\left(\frac{e^(\tilde{y}_i)}{\sum_{j=1}^{M} e^(\tilde{y}_j)}\right) \quad (1)$$

in which y_i is the expected task objective, $\tilde{y}_i$ is the actual task objective and M is 3 in this paper, which represents the number of tasks needed to be infer.

Another component loss is prediction loss $\mathcal{L}_2$, which is the mean square error (MSE) of real labels and the control module output value of the robotic arm motor angles

$$\mathcal{L}_2 = \frac{1}{A}\sum_{i=1}^{A} \mu_i (a_i - \tilde{a}_i)^2 \quad (2)$$

in which a_i is the excepted motor angle. $\tilde{a_i}$ is the actual motor angle and μ is the parameter of each motor angle and it satisfies the condition that $\sum_{i=1}^{N} \mu_i = 1$. A is 6 in this paper, which represents the number of joint angles. What's more, in order to help the network converge more quickly, Inception_V3 itself has an auxiliary loss function $\mathcal{L}_{aux}$ which also predicts task categories. The total training objective $\mathcal{L}$ for the entire model is given by the combined loss function

$$\mathcal{L} = \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_2 + \lambda_3 \mathcal{L}_{aux} \quad (3)$$

with λ_1, λ_2 and λ_{aux} being hyperparameters to make the three loss functions on the same order of magnitude. The algorithm is summarized in Algorithm 1.

Algorithm 1 TRAINING ITERATION PROCEDURE

Require: B is the batch size. L_1 is reasoning loss, L_2 is prediction loss, L_{aux} is the auxiliary loss of Incepiton V3, θ denotes the model parameters, $\tilde{\theta}$ denotes the updated model parameters, $\pi_{Cla}(o_k)$ denotes the self-reasoning module. μ_i is the parameter of each motor angles, α denotes the learning rate.

$\beta = RandomSample(\{\varepsilon_1, \cdots, \varepsilon_N\}, B)$
for $\varepsilon^j \in \beta$ **do**
  $L_1 += \sum_{k=0}^{j} \|\pi_{Cla}(o_k) - y_k\|_2^2$
  $L_2 += \sum_{k=0}^{j} \sum_{i=1}^{6} \mu_i \|\pi(o_k)_i - a_i\|_2^2$
  $L_{aux} += L_{aux}$
end for
$L_{tec} += \lambda_1 L_1 + \lambda_2 L_2 + \lambda_{aux} L_{aux}$
$\tilde{\theta} = \theta - \alpha \nabla_\theta L_{tec}$
return $\tilde{\theta}$

IV. EXPERIMENTS

Through our experiments, we aim to answer the following questions: (1) Can the MSRnet maintains multiple manipulation skills? (2) Is it possible to automatically understand the environment and determine the task objective? (3) For the purpose of further understanding, we additionally evaluate: (a) Does the MSRnet care the basic network architecture? (b) Is this method robust enough to generalize from simple scenes to complex scenes?

We design a few sets of comparative experiments with a low-cost 6-DoF robotic arm. We take 80% of the demonstrations as a training dataset, 10% of the demonstrations as a validation dataset, 10% of the demonstrations as a testing dataset. We use a batch-size of 10 for the training. We employ SGD [24] optimizer to train each network. We initialize weights as pre-trained models on ImageNet [25], and all new layers in the control module are initialized with a zero-mean Gaussian with standard deviation 0.01. Moreover, we also screen some videos to visually show our approach, which is uploaded.

TABLE I
MUTLI-TASK AND TASK-SPECIFIC MODEL

	Pre	Cla	SSP	Pre-rate(4.5)	Pre-rate(2.25)	Rea-rate
task-specific model 1	√			70%	30%	
task-specific model 2	√			60%	50%	
task-specific model 3	√			70%	40%	
MSRnet without SSP	√	√		76.7%	56.7%	**100**%
MSRnet	√	√	√	**83.3**%	**70**%	**100**%

Furthermore, our tasks are different from the previous tasks such as push, reach and pick-and-place [3]. Before evaluation of success from the task executing, we define the criteria of success. As we know, when pour water or stir coffee, the target area is a circle which is centered on the center of the cup rim. When grip the spoon, the spoon should be in the blue area shown in Fig.2(b). And The blue area can be approximated as a circle. So the target of task 2 is still a circle which is centered on a spoon. Based on the above analysis, the criteria is formulated as follows. The grippers' coordinates are read when the robotic arm finished the task. In task 1 and 3, the coordinates should be in the target area which is centered on the center of the cup rim and the grip is higher than the cup rim. And in task 2, the coordinates should be in the target area which is centered on the spoon and the grip is not higher than spoon. In Table I Pre-rate(4.5cm) means that the target area is a circle with a radius of $4.5\ cm$. And Pre-rate(2.25cm) means the radius is $2.25cm$.

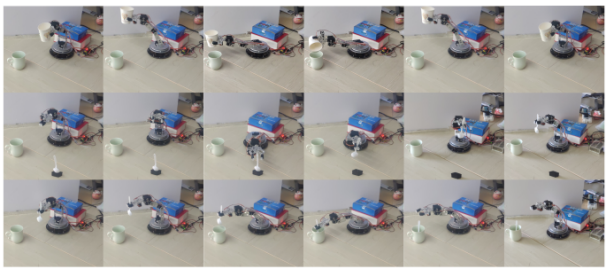


Fig. 4. From the first line to the third one, they represent task 1, task 2 and task 3. Each line is the states when the robot arm we intercepted completes the task in demonstrations.

A. Mutli-task learning

As the first experiment, we aim to demonstrate that MSRnet can determine the task objective and generate the corresponding robot action simultaneously. We design three tasks simulating the application scenario of making coffee, involving pouring water into a cup of coffee powder (task 1),

taking a spoon (task 2), and finally stirring the mixture with the spoon (task 3), illustrated in Fig.4. In order to demonstrate the advancement of our network, we compare its performance with task-specific models. Each task employs it own dataset when the task-specific model is trained for single task. And task-specific models do not have reasoning loss which means it does not employ self-reasoning module, only predict the angle of joints.

The results of the experiments are reported in Table I. Pre refers to the control module, Cla refers to self-reasoning module, SSP refers to sharing parameters of shallow layers, Rea-rate refers to reasoning accuracy. MSRnet without SSP achieves 76.7% Pre-rate(4.5cm) on multi-task test, which is state-of-the-art on the premise of completing multiple tasks with one model. Its Pre-rate(4.5cm) is 6.7% higher than task-specific model 1 and 3, 16.7% higher than task-specific model 2. Obviously, as long as Pre-rate of MSRnet without SSP is the same as task-specific training, it means that our approach is feasible. Moreover, the commonality which our tasks have, improves the performance of approach. For example, the target areas of task 1 and task 3 both are circle which is centered on the center of the cup. Although these are different tasks, there are some common features. So Multi-task training means the amount of training dataset is three times that of single-task training. As we all know, deep learning is significantly data driven. The increased amount of data in the training dataset will naturally improve the performance of network.

Furthermore, The diameter of cup rim could be smaller in daily life. So we conducted a more critical test. In this test, the radius of target area is 2.25cm. MSRnet achieves 70% Pre-rate(2.25cm). MSRnet without SSP achieves 56.7% Pre-rate(2.25cm). And task-specific training averagely achieves 40% Pre-rate(2.25cm).

B. Self-reasoning

Table I also demonstrates that the reasoning success rate is 100% in multi-task learning, which means MSRnet can distinguish different tasks and reason out the task objective with raw input images. And the high Pre-rate represents that MSRnet focus on the right features. Moreover, MSRnet achieves 83.3% Pre-rate($4.5cm$), which has the best performance. This is due to that the self-reasoning branch and the other branch focus on the same area. For example, these two modules both have to focus on the area where has a spoon in task 2. In other words, both reasoning loss and prediction loss optimize feature extraction capabilities of the shallow layers. Sharing parameters of shallow layers significantly improves the convergence speed and the performance of our approach.

However, from Table I we can see that the performance of task 2 is the worst. This is due to that task 2 is more complex than task 1 or task 3. In task 2 robotic arm is asked to take the spoon. In the robots' view, there is a spoon which is the target and a cup which is a interference. Model needs to figure out which one is the correct target. This illustrates that complex scenes would affect the performance of our network.

TABLE II
TEST THE ROBUSTNESS OF MSRNET IN COMPLEX SCENE

	Pre-rate(4.5cm)	Pre-rate(2.25cm)
Simple Scenes	83.3%	70%
Complex Scenes	60%	43.3%
Complex Scenes + fine-tune	76.7%	63.3%

C. Additionally evaluation

In order to further understand the MSRnet, we additionally evaluate its performance. We test our network in complex scenes which have are never shown in the training dataset. This evaluates the generalization of the model. In the daily life, the environment is much more complex. In addition to the cup on the table, there will be phones, keys, pens, etc, which is show in Fig.2(a). So we want to evaluate MSRnet's ability of environment understanding and task determining in complex scenes. All the experiments in Table II employs Inception_V3 with SSP. Our network achieves 83.3% Pre-rate(4.5cm) in simple scenes. Pre-rate(4.5cm) declines to 60% when tested in complex scenes. After fine-tune, the Pre-rate(4.5cm) reaches 76.7%. From the results, although the performance of Pre-rate(4.5cm) slightly declines, our MSRnet is robust enough to complete tasks in complex scenes.

We want to figure out if MSRnet is sensitive to feature extractors. We compare the performance of MSRnet with three different basic networks. The results are reported in table III. All these networks are trained in simple scenes and tested in simple scenes. Obviously, Inception_V3 with SSP achieves the best performance. Inception_V3 without SSP achieves 76.7% Pre-rate(4.5cm), Darknet53 achieves 73.3% Pre-rate(4.5cm) and Vgg16 achieves 53.3% Pre-rate(4.5cm). Inception_V3 and Darknet have similar performance while Vgg16 is the weakest. The results illustrate that Inception_V3 or Darknet53 is more formidable than Vgg16 when extract features. In summary, Inception_V3 with SSP is suitable for our tasks. It has sufficient ability of feature expression and does not cause overfitting.

TABLE III
THE PERFORMANCE OF DIFFERENT PUBLIC FEATURE EXTRACTORS

	Pre-rate(4.5)	Pre-rate(2.25)
Vgg16	**53.3**%	**33.3**%
Darknet53	73.3%	53.3%
InceptionV3	76.7%	56.7%
InceptionV3 + SSP	83.3%	70%

D. Failure case analysis

Fig.5 shows the failed cases. The first line is a failed case, and the second line is a successful case. In these two samples, the spatial locations of cup are different. However,

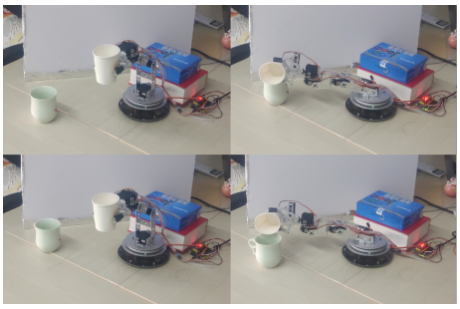


Fig. 5. The first line is a failed case, and the second line is a successful case. These two spatial structures are similar from the current perspective which confuses the network.

two cups seem in the same place in the current perspective, which confuses the network. From this perspective, there is some area in the desk where is difficult for network to locate the cup. In order to deal with the issue, We attempt to shoot dataset from different perspectives. Unfortunately each perspective has this issue. And if we shoot from the top to bottom, we would lose height information and fail to sense the height of the cup. Moreover, in different position the shape of cup rim is visually changed, which makes things more complicated.

In the field of computer vision, Multi-view fusion or RGB-D cameras are widely considered to solve the issues which are difficult for single perspective such as perceiving depth of field. We believe Multi-view fusion and RGB-D cameras can be a choice. And we will explore those approaches in the following work.

V. CONCLUSION

In this paper, we propose an end-to-end muti-task self-reasoning neural network (MSRnet) with combined loss. The approach can enable low-cost robots to simultaneously determinate the task objective from complex scenes and perform multiple manipulation tasks by visual perception. Our experimnts show that both the specific self-reasoning module and multi-task learning can improve the ability of understanding complex scenes and the performance of multiple manipulation tasks. In our future work, we shall concentrate on multi-task learning for more diverse tasks and develop a larger dataset for our methods to acquire more complex manipulation skills.

REFERENCES

[1] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters. *An Algorithmic Perspective on Imitation Learning*. now, 2018.

[2] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics & Autonomous Systems*, 57(5):469–483, 2009.

[3] Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. 2017.

[4] Sergey Levine, Peter Pastor, Alex Krizhevsky, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with large-scale data collection. *International Journal of Robotics Research*, (10), 2016.

[5] Tianhe Yu, Chelsea Finn, Annie Xie, Sudeep Dasari, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot imitation from observing humans via domain-adaptive meta-learning. 2018.

[6] Rouhollah Rahmatizadeh, Pooya Abolghasemi, Ladislau Bölöni, and Sergey Levine. Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration. 2018.

[7] Pierre Sermanet, Corey Lynch, Jasmine Hsu, and Sergey Levine. Time-contrastive networks: Self-supervised learning from multi-view observation. 2017.

[8] X. Wei, F. Sun, Y. Yu, C. Liu, B. Fang, and M. Jing. Robotic skills learning based on dynamical movement primitives using a wearable device. In *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 756–761, Dec 2017.

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. 2017.

[10] Lerrel Pinto and Abhinav Gupta. Learning to push by grasping: Using multiple tasks for effective learning. In *IEEE International Conference on Robotics & Automation*, 2016.

[11] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J. Andrew Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics*, 7(1-2):1–179, 2018.

[12] Stefan Schaal. Dynamic movement primitives-a framework for motor control in humans and humanoid robotics. In *Adaptive motion of animals and machines*, pages 261–280. Springer, 2006.

[13] Aleš Ude, Andrej Gams, Tamim Asfour, and Jun Morimoto. Task-specific generalization of discrete and periodic dynamic movement primitives. *IEEE Transactions on Robotics*, 26(5):800–815, 2010.

[14] Stefan Schaal, Jan Peters, Jun Nakanishi, and Auke Ijspeert. Learning movement primitives. In *Robotics research. the eleventh international symposium*, pages 561–572. Springer, 2005.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[17] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[18] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[19] Lerrel Pinto and Abhinav Gupta. Learning to push by grasping: Using multiple tasks for effective learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2161–2168. IEEE, 2017.

[20] Yan Duan, Marcin Andrychowicz, Bradly C. Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. 2017.

[21] Tianhe Yu, Pieter Abbeel, Sergey Levine, and Chelsea Finn. One-shot hierarchical imitation learning of compound visuomotor tasks. *arXiv preprint arXiv:1810.11043*, 2018.

[22] Bradly C. Stadie, Pieter Abbeel, and Ilya Sutskever. Third-person imitation learning. 2017.

[23] Tianhao Zhang, Zoe Mccarthy, Owen Jow, Dennis Lee, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. 2017.

[24] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.