Abstract—In this paper, we present a cognitive learning framework for the humanoid robot to acquire cutting skill. Our framework consists of two main modules: 1) A visual module that is trained using the deep feature to visually estimate object firmness level, and 2) A motor skill learning module that adopts reinforcement learning method to generate optimized control parameters for cutting. We propose a method that is based on the consideration of energy efficiency for the cutting motion. Experiments conducted on both tasks show that 1) the visual module is able to estimate firmness level of a novel object with state-of-the-art performance, and 2) the motor learning module, given the estimated firmness level from the visual module, is able to improve the actual performance of cutting on a humanoid robot (Baxter). Also, we show that the energy efficiency measure we proposed is a proper criterion for evaluating the cutting skill on cognitive robots.

I. INTRODUCTION

Home assistant robots, which are aiming to help human beings with daily routine tasks, need to be able to master the skill of cutting. Here we consider the cutting action that is aiming to divide the target object into pieces with a knife or other sharp implement. At the first glance, it seems that the skill of cutting is a relatively easy one to acquire. However, it is widely considered as a problem with extreme difficulty to become accurate with the knife work using minimum energy in order to create dishes that cook evenly and look truly professional, for our human beings. The question becomes: how to design a learning mechanism to equip a humanoid robot with the skill of cutting? In this paper, we argue that to achieve the skill of cutting requires both 1) visual recognition of object’s attribute (such as firmness), and 2) fine level robot manipulator control.

See and Estimate: before actually applying a cutting action, as human beings, we evaluate the physical attribute of the target object, especially the level of firmness. This is mainly for the purpose of estimating how much force is needed to cut through the target object while the energy spent is minimized. In the field of Computer Vision, the methods of estimating physical attributes from visual input have been studied [4], [6], [22], however, these works are mainly concerned with either binary or relative attributes. For the purpose of estimating object firmness, we need a multiple scales measure.

Act and Fine-tune: given a visual estimation of firmness measure, we generate a coarse cutting policy. After that, through repetitive trials of cutting this specific category of objects, we, as human beings, gradually learn to adapt our policy parameters to master the cutting of the specific target object. This process is generally referred to as “reinforcement learning” in the field of Robotics. Fig. 1 depicts how human learns to cut new object.

Inspired by these observations and intuitions, in this work, the first implementation of a vision-guided approach to help a humanoid robot master the skill of cutting is presented. It consists of two modules: 1) A CNN based visual recognition module to first estimate the level of firmness given an arbitrary target object; and 2) a reinforcement paradigm is implemented to further refine the cutting policy with regard to the target object.

Moreover, cutting is an action that has to be defined from the perspective of applied forces, due to the inevitable interaction between the sharp implement and the target object. In the literature of robotics, there is a vast amount of actions that has been studied for humanoids and most of them are trajectory-based analysis [15], [20], [30], [31]. In this paper, we are looking for a more generative representation of the action, and we take into account the forces involved, which may be introduced by the agent, or by the object which the agent is interacting with, or from the environment. In this work, we use the force pattern to define cutting policy and present a force-based controller to generate cutting action on a humanoid platform.

The paper is organized in the following manner: We first summarize works in the fields of Computer Vision, Robotics and Cognitive Systems that are closely related in Sec. II; Then we present our novel cognitive and developmental framework for cutting skill learning in Sec. III. In Sec. IV, three experiments are conducted to validate our proposed system, and we also present a new object firmness data-set.
for future research.

II. RELATED WORK

A. Studies on manipulation skill acquisition

Previous works have described robot systems that are capable of doing manipulative actions and skills in the kitchen environment. Lenz et al. [19] presented a model-predictive control approach using deep learning techniques for food-cutting. Yamaguchi et al. [31] developed a pouring behavior model that consists of a library of small skills and using learning and planning methods for generalizing in terms of different materials, container shapes, contexts, container locations and target amounts. Bollini et al. [2] designed a chef robot that is able to parse a recipe, turns it into a high-level plan that involves actions like pour, mix and bake and executes it. Beetz et al. [1] demonstrates a robot with the skill of cooking a pancake. The robot has the perception routine for object detection and localization as well as checking the estimated result. It also performs force-adaptive control to flip the pancake using a spatula. Kormushev et al. [15] also studied the pancake flipping skill. They focused on modulating the demonstrated motion primitives using the Reinforcement Learning technique. Another important domestic application of autonomous robot is the laundry tasks. Van der Burg et al. [30] presented an algorithm for the robot to detect the geometry of the cloth and determine the motion of the gripper to achieve the desired folding configuration. Lee et al. [18] proposed a method of generalizing the demonstrated skill with both kinematic and driven phases and they demonstrated on the task of tying a knot. The cutting action is rarely studied as a skill for the humanoid robot. Our work presents a framework including both visual and motor learning for robots learning a cutting action, which is novel.

B. Skill acquisition based on physical force

There are studies that focus on enabling the robot to generalize the force related skills to different situations. Gams et al. [5] proposed a method for the robot’s motion trajectory to adapt to different surfaces while still keep the same force profile using DMP. Kalakrishnan et al. [10] developed an approach that enables the robot to acquire contact interaction skills through the reinforcement learning. Our work is similar to that approach but we also have a vision system to generalize the policy for different object conditions. Most of those studies require human demonstration. [16] uses a haptic device for the force demonstration. [23], [21], [11] demonstrates force and position profile simultaneously through kinesthetic teaching. Our system does not require a human demonstration. The initial parameters are generated randomly. Our reward feedback after every trial of execution is able to guide the policy search to converge toward a reasonable configuration.

C. Motor skill learning through policy search

Reinforcement learning problems in robotics are different from typical benchmark problems for reinforcement learning. The problems are usually defined with high dimensional and continuous states and actions. There are also challenges in obtaining the true states from the environment, the high cost of each roll-out and under-modeling the details from the real world system [12]. Policy search methods are more suitable for robotics reinforcement problems than most other RL approaches since they provide better integration of domain-specific knowledge and capability of dealing with parameterized policies and they have been proved successfully on robotics [25], [27], [14], [13]. Another category of methods for policy improvement is called black-box optimization. Some recent researches have compare the BBO method with other RL methods and found it to have equal or better performance despite of its simplicity [8], [29]. In this work, we used CMA-ES algorithm [7], which is considered to be a standard in BBO [28].

III. OUR FRAMEWORK

In this section, we present a framework for the cutting skill acquisition. Figure. 2 is a flowchart that illustrates the overall framework. It consists of two main modules: observation and execution.

The vision module takes an image of the target object as input, and is trained to predict an initial estimation of the target object’s firmness level. The prediction model here is trained using a supervised learning method, and we extract the visual features using a pre-trained Convolution Neural Network (CNN).

The motor control module, takes the belief of the firmness level to initialize the parameters for the controller to execute a cutting action. However, as human beings learning a new skill, given the visually recognized physical attribute, we further fine-tune our motor skill through interaction with the physical world and observed feedback. This observation suggests a reinforcement learning paradigm to improve the control parameters. We formulate the robot cutting action as a Markov Decision Process (MDP). In each time step, the amount of force that is exerted on the knife from the robot’s end effector is estimated based on the state of the knife. This MDP is the inner loop of the execution for each trial of cutting as shown in Fig. 3. After one trial of cutting is executed, a cutting quality score is generated with respect to the actual execution of the cutting. The score is then used to update the action parameters. This makes the outer loop as shown in Fig. 3. This iterative learning process continues until the score of the action converges to a local maximum.

We then store this trained set of cutting parameters as the learned knowledge with regard to the firmness level of the object.

A. Visual learning module for predicting object firmness

As a human being, when we are given an everyday object to cut, we are able to first visually estimate the firmness of the object from its appearance. People use tactile sense to get haptic feedback to confirm how hard the object is (by pressing its surface), but this is not always used. Visual perception stays the main resource of information for us to
estimate object firmness. Moreover, when we are given a material (or object) that we have never encountered before, we are able to infer a rough firmness estimation from our previous life experience. Given these observations and intuition, we formulate a learning problem here to empirically study the visually firmness level estimation module.

First, we compiled a testing-bed, which consists of everyday fruits and vegetables. The dataset contains the images of a variety of categories, and the objects considered in our experiments have a large variation of firmness level. Without loss of generality, we constrain our firmness level as an integer parameter \( Y_f \), whose value ranges from 1 to 10, where 1 is the softest and 10 is the hardest.

In this work, we use a pre-trained CNN from [17]. The convolutional layers contain 96 to 1024 kernels of size 3x3 to 7x7. Max pooling kernels of size 3x3 and 5x5 are used at different layers to build robustness to intra-class deformations. For each image in our testing-bed, we use this pre-trained model to extract a 4096 dimensional feature vector using [3], [17]. We then trained a support vector machine (SVM) classifier to do firmness level recognition using the deep features. For each image \( I \) in the training set, we have an annotated pair of training data \( CNN(I), Y_f \).

A training model can be represented by:

\[
Y_f = SVM_\theta(CNN(I)),
\]

where SVM stands for support vector machine; \( \theta \) is the SVM parameter to be optimized; CNN stands for pre-trained net model for feature extraction.

For each testing image \( I \) in our dataset we apply the trained model \( SVM_\theta \) and convert the classification scores to \( P_f(Y_f | I) \). During execution time before cutting, the humanoid robot acquires an image of the object from its camera and then attends to the object area through attention mechanism, and then predicts the firmness of the object using the trained model. In the motor learning module, we use the prior belief, of how hard the object is, to initialize the parameters of the controller for initial cutting action.

**B. Reinforcing the cutting skill**

We formulate the cutting process as a Markov Decision Process (MDP). A MDP consists of a set of states \( S \), a set of actions \( A \), the transition probability \( P \) and the reward function \( R \). A fundamental assumption under MDP is that the current state \((s_t, a_t)\) for time step only depends on the state \((s_{t-1})\) and action \((a_{t-1})\) in the previous time step, which could be formulated by:

\[
s_t \sim p(s_t | s_{t-1}, a_{t-1}).
\]

And, an action at time step \( t (a_t) \) is generated by a policy \( \pi_\theta \) based on the current state, which could be formulated as,

\[
a_t \sim \pi_\theta(a_t | s_t).
\]

An episode is defined as a sequence of states and actions and an episode ends when a certain terminal state is reached. The expected reward of an episode could be formulated as:

\[
J(\theta) = \frac{1}{\alpha} E \left\{ \sum_{t=0}^{T} r_t \right\}
\]

where \( \theta \in \mathbb{R} \) are the policy parameter, \( \alpha \) is the normalization factor. The goal of the reinforcement learning is to find a set of policy parameters \( \theta \) so that the expected reward \( J(\theta) \) is maximized. One category of methods that have been applied to solve this problem is called the policy gradient approach. It follows the steepest gradient of the expected reward to search for the optimized parameters. In each time step \( t \), the update of the policy parameters could be expressed as,

\[
\theta_{t+1} = \theta_t + \alpha_t \nabla J(\theta)_{\theta=\theta_t},
\]

In this work, we adopt the covariance matrix adaption evolution strategy (CMA-ES) to search for the optimized policy parameters for cutting skill learning. Similar to the quasi-Newton methods, it is a second order method estimating a covariance matrix through an iterative procedure. The main advantage of the algorithm is that it does not approximate the derivatives of the function. Thus, it can be applied to non-smooth and noisy problems. It has been widely considered to be one of the most robust algorithms of direct policy search for reinforcement learning problems [8], [29].
C. Evaluating a cutting action

In order to improve the cutting skill, we first need a way to evaluate the cutting result. We base our evaluation metric design on following observations: (1) A cutting action using an unnecessary large amount of force could lead to a separation of the target object, but it would be a waste of energy and is not generally considered as a safe operation when a human is executing a cutting action; (2) A cutting action exerting small force would be inefficient in terms of time; (3) A good cutting action requires not only separating the target object but also balancing well between the exerted force and the amount of time for execution. Thus, in this paper we propose to formulate the cutting energy as the evaluation measure for how good a generated cutting action is.

Here, we use $E$ to denote the cutting energy. It could be expressed in terms of the integration of the force exerted over the path of the full motion sequence, which could be formulated as:

$$ E = \int_C F ds. \tag{6} $$

To give an example of the principle of minimizing energy, we take saw as an example of balancing between distance and force applied. While a saw is used during a cutting task, the distance of the cutting path is prolonged but the amount of force applied in the cutting process has been reduced. In this study, we are fixing the cutting path and the parameter for the evaluation function is the force profile. The force profile is generated based on the state of the robot’s end effector and the policy parameters.

D. Generalization of the optimized parameters through the firmness level

As mentioned in Sec. III-A, a visual module is trained to predict a novel object’s firmness level ($Y_f^I$) given its visual appearance ($I$). In order to link $Y_f^I$ with different policy parameters $\theta$, we trained several sets of policy parameters $\theta$ for different firmness levels $Y_f$. We then create a hash-table $\mathcal{H}$ which has firmness level as the hash-key and the set of policy profiles $\theta$ as the table content. Thus for each level of firmness considered in Sec. III-A, we could retrieve a pre-trained set of control policy $\theta^I$ as:

$$ \theta^I = \mathcal{H}(Y_f^I). \tag{7} $$

When the robot is given a novel object to cut, our visual system first predict a firmness level. Then a set of optimized policy parameters is retrieved from the hash-table. In the experiment section, we test two ways of applying the retrieved control policy: 1) directly apply the retrieved policy to the controller. From the experimental result, we can see that it could be already a fairly good policy for cutting the novel object; 2) we can further apply another round of reinforcement learning procedure. Never the less, since the retrieved policy parameters are not randomly initialized, the learning process takes much shorter time for the target reward to converge.

E. A force-based controller for cutting action

It is worth to mention a practical issue when implementing a cutting action on a humanoid robot platform. To execute the cutting action on a humanoid robot, a mere trajectory based motion controller does not suffice. It is mainly because a trajectory based controller only takes a position and orientation target as the movement goal for execution. When a trajectory based controller is applied for the cutting action, a relatively higher resistance from a hard object could stop the robot end effector from moving further. Thus, the execution of the action will halt when the execution time exceeds a certain threshold.

In order to solve the issue, we designed a force based controller for implementing the cutting action on the humanoid robot. The controller essentially creates a new layer on top of the trajectory based controller. We augmented the controller so that at every time step we are not only assigning a target pose for the controller but also how much torque needed to exert on each actuator. The robot joint torque $\tau_q$ at each time step follows

$$ \tau_q = J^T(q) \tau_x \tag{8} $$

$$ \tau_x \propto \Delta x(F, x_t) \tag{9} $$

, where $\tau_x$ is the torque in the end-effector space. It is proportional to the spatial difference $\Delta x$ in terms of the end-effector’s current position $x_t$ and target position. The target position is determined by how much force $F$ the robot would like to impose to the external environment through its end-effector. In the cutting application, it is how much force the robot used to cut the object. $\tau_x$ is converted to $\tau_q$, the torque on every joint on the robot arm, through the inverse of the Jacobians matrix $J^T(q)$. In such a manner, our robot can successfully follow a certain trajectory as well as execute the desired force profile.

IV. Experiments

The theoretical framework we have presented suggests three hypotheses that deserve empirical tests: (a) Firmness can be inferred from visual feature extracted from the CNN based visual module with decent accuracy; (b) Using reinforcement paradigm could improve the cutting skill evaluated by the energy consumption for the cutting motion; (c) On a humanoid robot platform, the optimized control parameters can be used to generalize a good cut for novel objects with similar firmness level.

To test the three hypotheses empirically, we need to define a set of performance variables. The first hypothesis is related to a visual system, we can test the system’s performance by comparing the predicted firmness with the ground truth on the testing split of the dataset. We use the mean absolute error (MSE) as the metric for the evaluation. Then, we use the same metric to compare our system’s performance with a baseline method to support our hypothesis (a). The hypothesis (b) is about validating the reinforcement learning module’s performance. We qualitatively test the hypothesis
by analyzing the reward and the convergence trend of the motor control policy parameters after a number of trials during iterative learning. For the third hypothesis, we empirically test it by having the robot cutting object B with similar firmness level using the same set of policy parameters trained from object A, and observe whether the system could still obtain sufficiently good performance without going through the iterative reinforcement learning phase.

A. Firmness prediction

1) Setup: In order to validate hypothesis (a), we need a data-set to serve as a test-bed. However, there is no public image data-set is available for our purpose. Thus, we compiled the first object firmness data-set for validation and evaluation.

The data-set consists of 44 categories of fruits and vegetables (The category here refers to the type of fruit and vegetable e.g. apple, orange, broccoli etc.). Each category has 10 to 20 images crawled from Google image search. Figure. 4 shows some examples of the images in the data-set. Most of the categories are the same as the object’s name. But, there are a few special instances. For example, we have two categories for the watermelon in the data-set, one is for whole-body watermelon and the other is sliced watermelon. These two categories would have the same object name but apparently different firmness level (sliced watermelon is much more soft than whole-body watermelon).

We follow a pre-defined training and testing protocol for our experiments. The data-set is divided into training, development and testing set. We take 2 instances from each categories as the development data and another 2 instances as the testing data. The total number of instances in the training set, development set and testing set are 678, 88 and 88.

Here, we set the firmness level as an integer value ranges from 1 to 10. For example, coconut has a firmness level towards 10 since it is hard, and strawberry has a firmness level towards 1 since it is soft. Since the firmness level is actually a subjective term, we designed a survey to find the consensus of the ground-truth firmness levels for each instance. A snapshot of the survey is shown in Fig. 5. The human subject is asked to give a number out of the scale from 1 to 10 to describe how much force he/she is going to use to cut the certain fruit or vegetable shown in the picture. The survey is distributed on Mechanical Turk and we collected 100 valid responses on each of the 44 categories (4400 responses in total). The firmness level of each category is then obtained by averaging the 100 responses. In such a manner, every instance in our data set is annotated with a consensus firmness level.

We used a pre-trained Convolutional Neural Network to extract features from the images, and we used Caffe [9], which is an open source deep learning framework. The main advantage of CNN than the traditional hand-crafted feature extraction procedure is that it gives a much more sophisticated representation of the image through its deep hierarchical structure. The network we used was trained with millions of images from ImageNet visual object recognition challenge. For each image instance in our data-set, we extracted the deep feature using this pre-trained network, which has a dimension of 4096.

Since the output of the model for a firmness level is a value instead of a category, we use a regression model to approach this task. The specific model we choose is the support vector machine for regression implemented in the Scikit-learn, a machine learning python package [24]. By using different kernels, we are able to increase the dimension of the input data. Specifically in this paper, we use liner, second order polynomial and radial basis function kernel (as known as rbf kernel). We use the mean absolute error (MSE) as the evaluation metric, which is defined as the absolute difference between the prediction and the ground-truth divided by the number of samples. The hyper-parameter for tuning the performance on the development data is the penalty parameter for the error term in the SVM model. A higher penalty value would lead to over-fitting while a lower one would potentially lead to under-fitting.

2) Results: Figure 6 is the plot showing the mean absolute error for training and development dataset using different SVM kernels as a function of penalty parameter. The mean absolute error is expressed as

\[
MAE = \frac{1}{n} \sum_{i=1}^{n} |f_i - y_i|
\]

where \(y_i\) is the ground truth label and \(f_i\) is the prediction. The horizontal axis is displayed in the log scale. The error for the training data (solid lines) decreases monotonically for all of the three kernels. The error for the development data (dashed lines) decreases at first and then increases. The lowest error value occurs when the penalty parameter is 1.0.
B. Reinforcing the cutting skill

Using the second order polynomial kernel, which we treat as the tuned hyperparameter during testing. The MAEs with the setting we optimized in the development data are summarized in Table I. For the testing data, the mean absolute error is 0.83. As a comparison, we use a constant firmness level labeling method as a naive baseline. Regardless of the image, the baseline method predicts a firmness level of 5.0. Such a naive method yields a MSE of 1.57 on the testing data. Here we want to mention that the baseline method has a decent performance because the firmness level of objects follows a normal distribution, which means most of the fruits or vegetables have a firmness level close to 5.0. However, our method yields a much better performance on predicting the object firmness and achieve only 0.83 MSE.

### TABLE I

<table>
<thead>
<tr>
<th>Method</th>
<th>Training</th>
<th>Development</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN+SVM</td>
<td>0.24</td>
<td>0.67</td>
<td>0.83</td>
</tr>
<tr>
<td>Baseline</td>
<td>/</td>
<td>/</td>
<td>1.57</td>
</tr>
</tbody>
</table>

![Fig. 6. Tuning the hyperparameter on the development data](image)

1) Setup: We would like to demonstrate the reinforcement learning module by training a set of policy parameters for cutting objects on a humanoid robot. Here we use cucumbers as the main experiment target object.

The cutting motion is formulated in 1D space, which is in the direction of the gravity force in the robot coordinate system. The state \( s = [x, \dot{x}, \ddot{x}] \) in which \( x \) is the distance from the knife to the cutting board. In other words, it measures how far in distance is it from the object being cut through. \( \dot{x} \) is the velocity and \( \ddot{x} \) is the acceleration. Action is defined as the increment of force \( \Delta F_z \) applied at the end effector of the robot. The policy parameter \( \theta \in \mathbb{R}^4 \) and the deterministic policy \( \Delta F_z = s \cdot \theta_{(1:3)} + \theta_4 \) is chosen.

An episode ends when 1) the time exceeds 30 seconds, 2) \( x \) is larger than 20cm or 3) \( x \) becomes 0 when the knife reaches the cutting board. The total reward for the entire episode is calculated as \( R = -\int_0^T F_z v z dt \), where \( T \) is the time when the object is successfully cut through. If the episode finishes due to \( x \) is larger than 20cm, it means the knife never touches the object. In such a case, we apply a reward of \(-100\).

We implement the force based cutting controller as a Robot Operating System (ROS) [26] node. For each episode, a set of control parameters is generated by the CMA-ES module and the cutting process is launched with the learned parameters. After the execution, the reward is calculated and is used to update the model for the next round of policy search.

We use the Baxter as the robot platform to carry out our experiments. Its arm has seven degree-of-freedom. The force command on the end effector is converted to torque command on every actuator through inverse kinematics calculations. The state (position, velocity) of the end effector is published by the robot in each time step. This state is then treated as the same as the state of knife assuming a rigid connection.

2) Results: After 130 episodes of execution, the reward converges to a high level compared to the initial episode as shown in Fig. 8. This trend can also be observed in the experiment because a low reward corresponds to cutting using a force that is either too small or too large. At the beginning we could observe a high proportion of the trials in which either the robot performs an intensive cut that hit hard on the cutting board or the knife moves slowly which consumed a lot of time. There were also many trials in which the knife moved upward instead of downward. But these cases occurred less often as the system evolves through the learning process. The four policy parameters also converge after the learning process. Figure 9 shows that the variation for the values of the normalized parameters dampened along with acquiring more cutting experience. It shows from another perspective that the algorithm explores the possible solutions and converges to an optimized one.

Figure 10 shows the force profile using the optimized parameters obtained after the reinforcement learning. It is interesting to point it out that the plateau region in the middle of the profile is learned be an energy efficient way for

![Fig. 7. Experimental setup](image)
penetrating the skin of the cucumber. The temporal duration is about 0.9 second showing that it is a fairly fast cut through, and it is close to how much time it takes for a human being with proficient cutting skill. Overall, it shows that our system is able to converge to a decent policy parameters for robot cutting skill.

C. Cutting performance on objects with similar firmness

1) Setup: We trained another set of parameters for cutting the banana which has a similar firmness level as the cucumber. Then we used the optimized parameters for cutting cucumbers to cut a banana and vise verse. The optimized parameters are the mean values of the parameters used in the last ten episodes of the learning process. Ten trials of cuttings were performed for each case and the result was recorded as the average of their rewards.

2) Results: The reward in average is lower for cutting the bananas than cutting the cucumbers. It is because the banana has a thicker skin than the cucumber and it costs more energy to cut it through. As shown in Table II, the difference in the rewards of cutting using the optimized parameters from the other object with the similar firmness level is about 30%. It validates our hypothesis (c) that the firmness level of the object could be used to generalize the policy parameters for the cutting action on a novel object.

V. Conclusion and Future Work

In this paper, we presented a cognitive learning framework that would enable a humanoid robot to learn a more energy efficient way to execute the cutting action based on the visual feature of the object. We conducted an empirical test on the object firmness data-set and confirmed that the firmness of the object can be inferred from visual features extracted from the CNN based visual module. A set of experiments on the humanoid robot platform validated our hypothesis that 1) using reinforcement paradigm could improve the cutting skill in terms of the energy efficiency 2) the optimized control parameters can be applied to cut novel objects with similar firmness level.

Human beings rely not only on their visual system but also the tactile system to obtain the feedback in the manipulation tasks. We believe that the deployment of a tactile system on the humanoid robot’s manipulator would help the acquisition and the execution of the force-oriented actions.