Cooking Actions Inference based on Ingredient's Physical Features

Ixchel G. Ramirez-Alpizar[1,2], Ryosuke Hiraki[2] and Kensuke Harada[2,1]

***Abstract*— Most of the cooking recipes available on the internet describe only major cooking steps, since the detailed actions are considered to be common knowledge. However, when we want a robot to cook a meal based on a recipe, we have to give to the robot a step by step plan of each of the tasks needed to execute one recipe step. In this paper, we developed a framework for inferring the executable cooking actions of ingredients, in order to compensate for the common knowledge of humans. We tuned the existing VGG16 Convolutional Neural Network (CNN) to learn the physical features of ingredients. Then, we built an inference model for six different cooking actions based on the learnt physical features of ingredients. The resultant inferred action(s) represents the next possible action(s) that can be executed. As there can be more than one executable action for the same ingredient state, we prioritize the cooking actions considering the previously executed action, for which kind of people the meal is being prepared for and the cooking time allowed. We show experimental results on five different types of ingredients that are not contained in the training dataset of the CNN.**

I. Introduction

In recent years, the use of robots in different areas has broaden due to technological advances in both hardware and software. Among others, the demand of home robots able to clean, do the laundry, cook meals, etc., has increased due to population aging, decreased birth rates, etc. There are few works discussing robotic cooking tasks. Bollini et al. [1] proposed a robot system called "BakeBot", which is able of following simple textual recipes for baking. Yamazaki et al. [2] proposed a system able to recognize vegetables, a cutting board and containers, the system is also able to manipulate cooking tools such as a knife, a peeler, etc. Mu et al. [3] analyzed the mechanics of a knife cutting vegetables and experimentally validated the proposed control strategy for slicing two different vegetables.

Unlike related work, in this paper we focus on the inference of cooking actions towards the automation of robotic cooking tasks, i.e. robots able to autonomously prepare any kind of meal with the minimum of instructions from the user (e.g. a recipe). Most of the cooking recipes available on the internet describe only major cooking steps, since the detailed actions are considered to be common knowledge. However, when we want a robot to cook a meal based on a recipe, we have to give to the robot a step by step plan of each of the tasks needed to accomplish a single recipe's instruction. Also, as the variation of ingredients in types, size and shapes is too large to be hard coded for manipulation or vision recognition, a generalized framework able to deal with any kind of ingredients is needed. Motivated by this, we developed a framework for inferring cooking actions of ingredients based on their physical features and based on the user demands of cooking time and eating habits, as illustrated in Fig. 1.

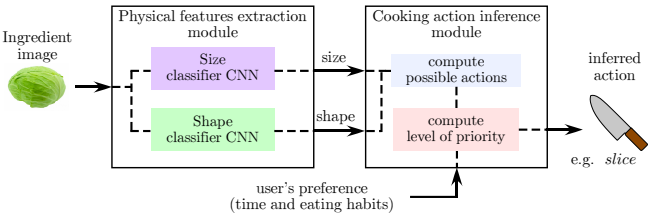


Fig. 1: Proposed framework for the inference of cooking actions.

Similar works have focused only on the state recognition of cooking objects. Paul [4] utilized and tuned the VGG16 [5] Convolutional Neural Network (CNN) model for the classification of cooking objects into 7 different states, such as whole, diced, julienne, etc., achieving a 77% of accuracy. Jelodar et al. [6] proposed a Resnet-based [7] deep model to classify cooking objects into 11 states, such as whole, peeled, juiced, etc., and also fine tuned an individual model for each cooking object (18 objects) in their dataset. In contrast, the goal of this work is to determine which cooking action should be executed next, based on the ingredient physical features without the necessity of knowing its proper name. Furthermore, the physical features can be employed to make decisions on the type/size of gripper or tool needed to manipulate the ingredient.

In this paper, we propose a framework for inferring the executable cooking actions of ingredients, in order to compensate for the common knowledge of humans. Our framework is basically composed of two modules: the physical features extraction module and the cooking action inference module. The physical features extraction module uses two VGG16 networks fine-tuned to learn the size and shape of ingredients. The cooking action inference module is composed by a "possibility model" for six different cooking actions based on the learnt physical features of ingredients, and an action priority level cost that infers one cooking action. To estimate which cooking action should be carried out next, the action priority level cost takes into account the following conditions: the previously executed action, a cooking time constraint (e.g. plenty, normal or in a hurry) and eating habits (e.g. kid, adult, elder adult), where the

[1]Automation Research Team, Industrial Cyber-Physical Systems Research Center, National Institute of Advanced Industrial Science and Technology (AIST), 2-4-7 Aomi, Koto-ku, Tokyo, 135-0064, Japan. ixchel-ramirezalpizar@aist.go.jp

[2]Department of Systems Innovation, Graduate School of Engineering Science, Osaka University, 1-3 Machikaneyama-cho, Toyonaka, 560-8531, Japan {hiraki, harada}@hlab.sys.es.osaka-u.ac.jp

Fig. 2: Labels used in the size classifier CNN.

last two conditions are user imposed. The resultant inferred action represents the next action to be executed by the robot. To verify the validity of the proposed framework, we tested our framework with five different types of ingredient that are not in the training dataset of the CNNs. The obtained results demonstrate the validity of the proposed framework.

This paper is organized as follows: in section II, we describe the physical features extraction module and show experimental results of the trained CNNs. In III-B, we explain the cooking action inference module. Next, in section IV, we show the experimental results of the inferred cooking actions of different ingredients. Finally, in section V, we give the conclusions of this work.

II. Physical Features Extraction of Ingredients

In this section we describe the physical features extraction module of Fig. 1. This module is composed by two VGG16 [5] CNNs fine-tuned to learn the size and shape of ingredients. The VGG16 used in this work is composed of 13 convolutional layers and 3 fully connected layers. In this work, we only tune the weights of the fully connected layers, the convolutional layers' weights are kept as the original trained VGG16 network [5].

The first CNN will learn the size of the ingredient. Considering possible future developments, we decided to classify the ingredient's size into 3 possible states: big (not graspable by one hand), medium (graspable by one hand) and small (difficult to grasp without using cutlery or some similar tool), as illustrated in Fig. 2. It should be noted that the input images are considered to be taken from a constant distance from the ingredients, so that the predicted size agrees with its real size.

The second CNN will learn the shape of the ingredient. Taking into consideration the different types of cutting styles, we decided to classify the ingredient's shape into 6 possible states: round, half-cut, long, slice, julienne and grain, as illustrated in Fig. 3.

The data set for fine tuning both CNNs is composed of images collected from the Web, and also from one portion of the data set kindly provided by A. B. Jelodar et al. [6]. In total we gathered $2,700$ images, from these 90% was used for fine tuning and the rest for verifying the performance of the trained CNNs. The images used for fine tuning were increased to approximately $20,000$ images through flipping, rotation, change of contrast, etc., from which 90% was used for training the CNNs (parameter update) and the rest for tuning the hyper-parameters of the CNNs. Table I shows the micro and macro average results of the shape classifier (using 273 test images). Table II shows the average accuracy of the

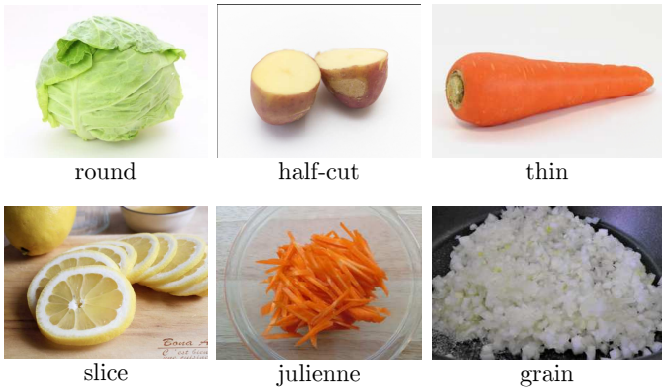


Fig. 3: Labels used in the shape classifier CNN.

TABLE I: Shape classifier results (using 273 test images).

Index	Micro-average	Macro-average
Precision	0.912	0.917
Recall	0.912	0.918
F-measure	0.912	0.918
Accuracy	0.912	0.918

TABLE II: Shape classifier accuracy average per label (using 273 test images).

Label	Accuracy
Round	0.925
Half-cut	1.0
Thin	0.885
Slice	0.917
Julienne	0.820
Grain	0.967

TABLE III: Size classifier results (using 301 test images).

Index	Micro-average	Macro-average
Precision	0.906	0.895
Recall	0.906	0.891
F-measure	0.906	0.888
Accuracy	0.906	0.888

shape classifier per label. Most of the mistakenly classified images had the labels slice, julienne or thin, among which the julienne label yielded the lowest accuracy. This means that it is easily to confuse the ingredient's shape when it is finely cut.

Table III shows the micro and macro average results of the size classifier (using 301 test images). Table IV shows the average accuracy of the size classifier per label. Most of the mistakenly classified images are those where the ingredient's real size is different from the size they apparent in the image (taken from a long distance or too close), and also when the ingredient was sliced. Additionally, there is an unbalanced between the number of medium labeled and small labeled images on the dataset, which led to the medium label accuracy to be the lowest.

TABLE IV: Size classifier accuracy average per label (using 301 test images).

Label	Accuracy
Big	0.871
Medium	0.835
Small	0.957

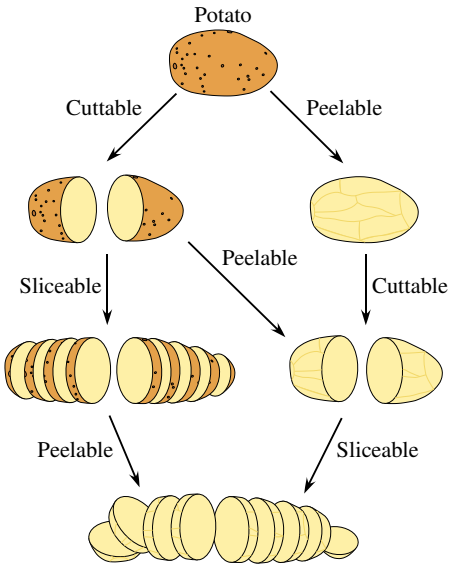


Fig. 4: Flow of possible cooking actions of a potato.

III. Inference of Cooking Actions

In this section we describe the cooking action inference module, which is composed of a possibility model and an action priority level cost function. The possibility model is based on the notion of Affordances as defined by Gibson [8]. The action priority level cost function is determined based on the user's preferences (cooking time, eating habits).

A. Possibility model

The possibility model represents the possible cooking actions (in this work, we consider only six) that can be executed on the ingredient in its present state. For example, as shown in Fig. 4, for a just harvested potato, we could either cut it in half or peel it, if we cut it in half first, then we can either slice it or peel it, etc. Most of the times, we have more than one action that can be executed on the ingredient. But also, based on the type of cooking, or eating habits, we could determine which is the next action to be followed.

In this work, we consider the following six cooking actions: cut in half, slice, rangiri (rough cut [9]), julienne, chop, and pour. We define a possibility model between the ingredient's physical state and the cooking actions as the grade of possibility for an action to happen given a physical state, as shown in Table V. We discretized the grade of possibility in five levels, HP (high probability), MP (medium probability), LP (low probability), NP (not possible), and NI (no influence). The first three represent how probable is the action to be executed next. NP means that in the present physical state of the ingredient the action cannot be executed.

TABLE V: Possibility model

Physical feature	Action					
	Cut in half	Slice	Rangiri	Julienne	Chop	Pour
Big	HP	LP	NP	NP	NP	NP
Medium	MP	HP	HP	LP	NP	NI
Small	NP	NP	NP	HP	HP	HP
Round	HP	LP	NP	NP	NP	NI
Half-cut	MP	MP	NP	NP	NP	NI
Thin	LP	HP	HP	NP	NP	NI
Slice	NP	NP	NP	HP	NP	HP
Julienne	NP	NP	NP	NP	HP	HP
Grain	NP	NP	NP	NP	NP	HP

Finally, NI refers to a physical feature that do not influence the next action that can be executed. For example, for a big ingredient the next highly probable action should be "cut in half" (according to usual cooking methodologies), therefore, for this action the "big" physical feature label represents a higher priority than the "medium" label. On the other hand if the ingredient is thin like a carrot, for the "thin" label the highest priority is set to the "slice" action rather than "cut in half". Also, in the case of a half cut ingredient (e.g. potato), the action could be "pour" however according to usual cooking methodologies it should be first cut in smaller pieces, therefore the grade given is NI.

Based on the probability model, the priority $S(a)$ for a given cooking action $a \in A = \{\text{cut in half, ..., pour}\}$ is computed as:

$$S(a) = \sum_{l} \lambda_{d(l,a)} P_l, \tag{1}$$

where $\lambda_{d(l,a)}$ is the weight representing the grade of possibility $d \in \{\text{HP, ..., NI}\}$ of the physical feature $l \in \{\text{big, medium, ..., grain}\}$ with respect to action a, and P_l represents the probability of the label l being the physical feature that best describes the input image (the output of the size and shape classifiers, described in the previous section). Using equation 1 we can define the probability of execution of every cooking action for a given input image. Although, this is not enough to determine one single action to be executed next, if we take into account the cooking time allowed and the user's eating habits we are able to prioritized the order of execution of the cooking actions.

B. Action priority level cost function

As mentioned in the previous section, by taking into account the allowed cooking time and/or the user's eating habits we can prioritize the order of execution of the cooking actions. For this purpose, we define an action priority level cost function f_p that takes into account three things: 1) the previously executed action, 2) for whom is the meal prepared for and 3) the allowed (desired) cooking time. This function is defined as:

$$f_p(a) = w_{\text{pre}} X(a_1, a_2) + w_{\text{who}} Y(a) + w_{time} Z(a), \tag{2}$$

where w_{pre}, w_{who}, and w_{time} represent the weights of considering the previous action, for whom the meal is prepared

for, and the allowed (desired) time, respectively. The function $X(a_{-1}, a_{-2})$ represents a constant whose value depends on the two previously executed actions (a_{-1}, a_{-2}), and it is defined as:

$$X(a_{-1}, a_{-2}) = \begin{cases} m_2, & \text{if } a = a_{-1} = a_{-2} \\ m_1, & \text{if } a = a_{-1} \\ 0, & \text{otherwise} \end{cases},$$

where m_1 and m_2 are negative constants. The function $Y(a)$ is defined as:

$$Y(a) = \begin{cases} n, & \text{if slice or julienne or chop} \\ -n, & \text{if rangiri or pour} \\ 0, & \text{otherwise} \end{cases},$$

where n represents a positive constant which value depends on who is going to eat the meal (kid/senior or adult). The function $Z(a)$ is defined as:

$$Z(a) = \begin{cases} t, & \text{if julienne or chop} \\ -t, & \text{if rangiri or pour} \\ 0, & \text{otherwise} \end{cases},$$

where t is a negative constant if the meal should be prepare as fast as possible and positive if there is no cooking time limitation.

Like this, the inferred cooking action i is computed through the next equation:

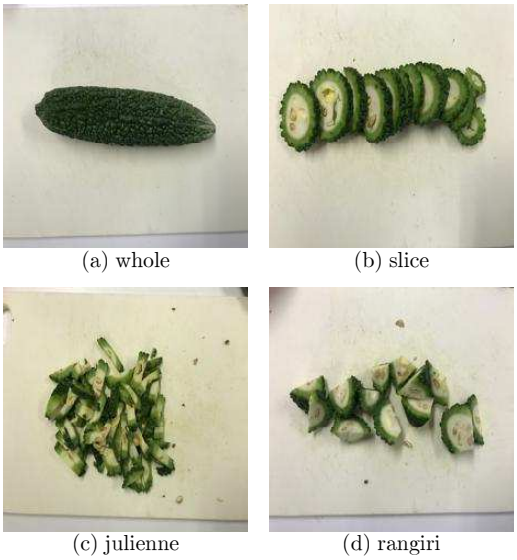
$$i = \underset{a \in A}{\operatorname{argmax}}(S(a) + f_p(a)), \tag{3}$$

where $S(a)$ is computed using equation (1) and $f_p(a)$ using equation (2).

IV. Experimental Results

To verify the validity of the proposed inference model, we use five different ingredients (bitter melon, apple, Japanese persimmon, sweet potato, sausage) which are not included in the data set used for training the shape and size classifiers. We cut the ingredients using different cooking actions and took a picture in each different physical state. We inferred the next cooking action for each combination available of user's preference of time and who will eat the meal. The constants involved in the inference model were empirically determined. In the results presented in this section, the constants are set as, $\lambda_{\text{HP}} = 100$, $\lambda_{\text{MP}} = 50$, $\lambda_{\text{LP}} = 30$, $\lambda_{\text{NP}} = -70$, $\lambda_{\text{NI}} = 0$, $w_{\text{pre}} = 1$, $w_{\text{who}} = 2$, $w_{\text{time}} = 2$, $m_1 = -10$, $m_2 = -80$. When the user wants the meal to be prepare as fast as possible $t = -10$, when the user has no time constraint $t = 10$, otherwise when there is no time consideration by the user t is set to zero. Also, when the user indicates that the meal is for kids or seniors $n = 10$, otherwise $n = 0$.

The first ingredient we use is a bitter melon, the input images used are shown in Fig. 5. The correspondent results to these images are shown in Table VI. In the who column, "Yes" means the user indicated the meal is for kids or seniors, and "No" means there was no input by the user regarding who is going to eat the meal. For the time column, NC means no time constraint, "-" means there was no input by the user, and F means fast (the user requires the meal to be ready as fast as possible). The physical features listed are those obtained from the size and shape classifiers. The inferred actions were computed considering the previous action as the one to its left column (for the first action, X(a) is set to zero). It can be seen that when the user requires the preparation to be faster, instead of starting slicing, the proposed framework determines that doing a rangiri cut will finish faster the cutting process, even though common knowledge dictates that a bitter melon is usually sliced and then cut in juliennes (as is the case when the user has no time constraints). It can also be seen, that although the time is required to be fast, as the who is set to be kids or seniors, the framework will keep the normal cutting style since using the rangiri (Fig. 5(d)) yields bigger pieces that are harder to eat by kids or seniors.



(a) whole (b) slice (c) julienne (d) rangiri

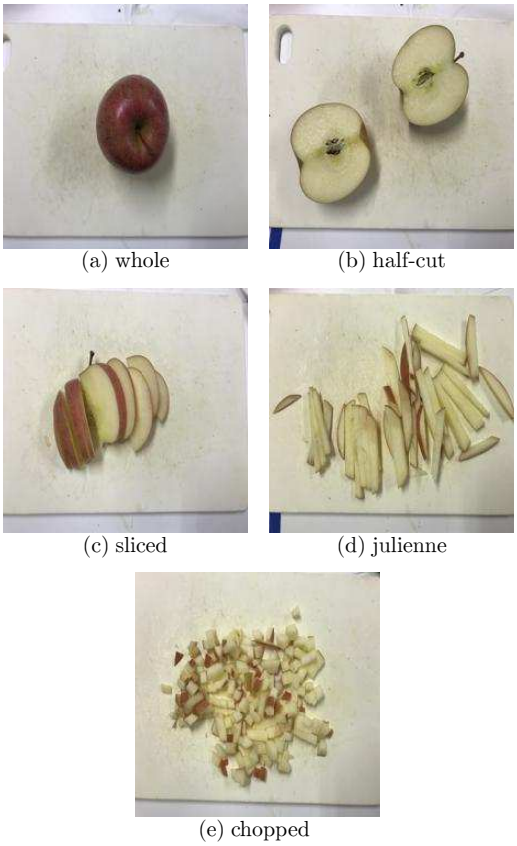
Fig. 5: Input images of a bitter melon

TABLE VI: Action inference result for the images of a bitter melon, shown in Fig. 5

Who	Time	Physical features (shape/size)		
		thin / medium	slice / small	julienne / small
N	NC	slice	julienne	pour
N	-	slice	julienne	pour
N	F	rangiri	pour	
Y	NC	slice	julienne	pour
Y	-	slice	julienne	pour
Y	F	slice	julienne	pour

In Table VII the results of the inferred actions for an apple are shown. The input images are shown in Fig. 6. It can be seen that for the image shown in Fig. 6(b), the shape classifier mistakenly output as shape "slice", when the correct output was "half-cut", since the shape feature was incorrectly classify, the inferred actions based on this image are also incorrect. However, we must point out that despite of this, the following actions (of the following images) were correctly inferred (all but one).

In Table VIII the results of the inferred actions for a Japanese persimmon are shown. The input images are shown

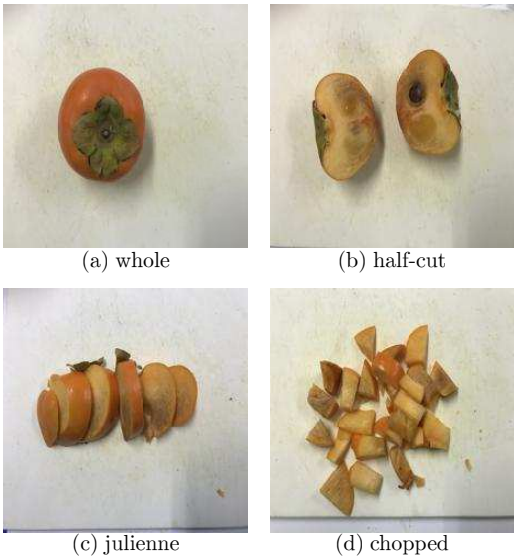


(a) whole (b) half-cut (c) sliced (d) julienne (e) chopped

Fig. 6: Input images of an apple.

TABLE VII: Action inference result for the images of an apple, as shown in Fig. 6

Who	Time	Physical features (shape/size)				
		round/ medium	slice/ medium	slice/ small	julienne/ small	grain/ small
N	NC	cut in half	julienne	julienne	chop	pour
N	-	cut in half	slice	pour		
N	F	cut in half	pour	pour		
Y	NC	cut in half	julienne	julienne	chop	pour
Y	-	cut in half	julienne	julienne	chop	pour
Y	F	cut in half	julienne	pour		



(a) whole (b) half-cut (c) julienne (d) chopped

Fig. 7: Input images of a Japanese persimmon.

TABLE VIII: Action inference result for the images of a Japanese persimmon, as shown in Fig. 7

Who	Time	Physical features (shape/size)			
		round / small	slice / medium	slice small	grain / small
N	NC	cut in half	julienne	julienne	pour
N	-	pour	pour	julienne	pour
N	F	pour	pour	julienne	pour
Y	NC	cut in half	julienne	julienne	pour
Y	-	cut in half	julienne	julienne	pour
Y	F	pour	julienne	julienne	pour

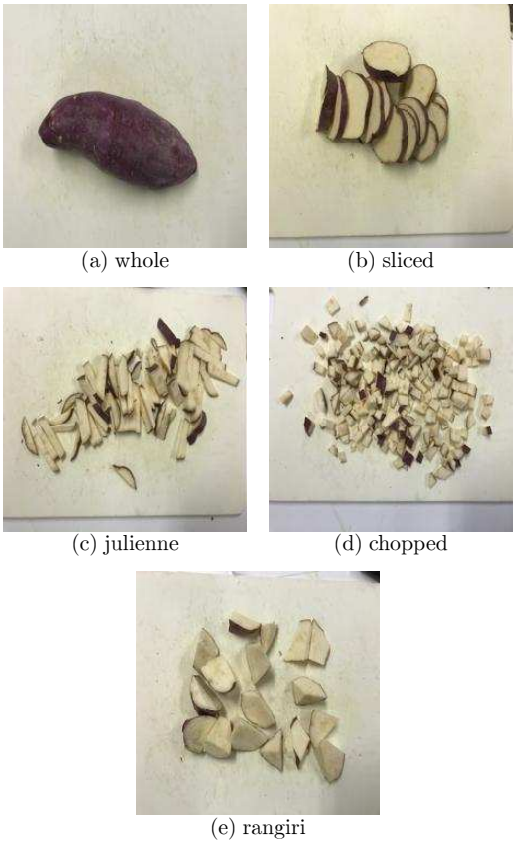
TABLE IX: Action inference result for the images of a sweet potato, as shown in Fig. 8

Who	Time	Physical features (shape/size)			
		thin/ medium	slice/ small	julienne/ small	grain/ small
N	NC	slice	julienne	chop	pour
N	-	slice	julienne	chop	pour
N	F	rangiri	pour		
Y	NC	slice	julienne	chop	pour
Y	-	slice	julienne	chop	pour
Y	F	slice	julienne	chop	pour

in Fig. 7. It can be seen that for the images shown in Fig. 7(b) and (c), the size classifier and the shape classifier, respectively, mistakenly output the size as “small” and the shape as “slice”, when the correct output was “medium” and “half-cut”. In this case, some of the inferred actions based on these images were incorrect.

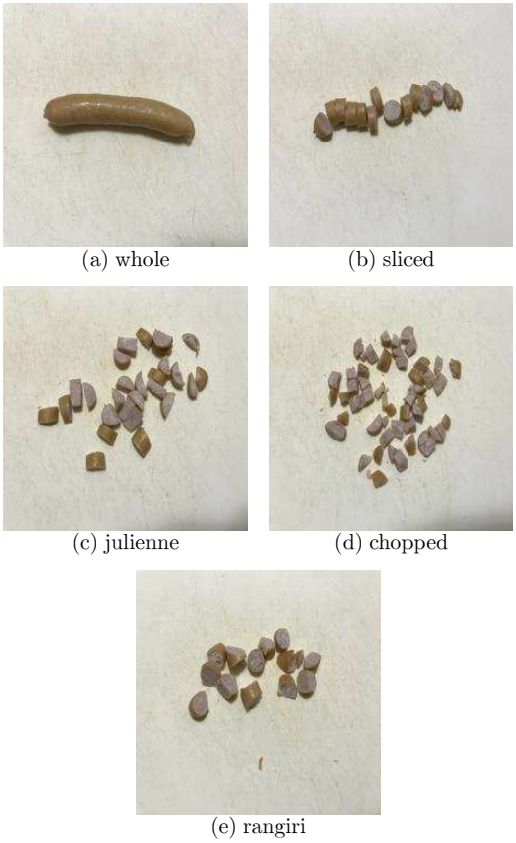
In Table IX the results of the inferred actions for a sweet potato are shown. The input images are shown in Fig. 8. Finally, in Table X the results of the inferred actions for a sausage are shown. The input images are shown in Fig. 9. In the last two ingredients, although their shape is similar, as their size is different, for the sausage it is difficult for the shape classifier to determine that the ingredient is already in juliennes in Fig. 9(c), and it is classified as “slice”, which leads the system to generate again ”julienne” as the next possible action.

The performance of the proposed inference model is 94.4% (only when the physical features were correctly identified), and the overall performance of the proposed framework is about 82.2%. This means that we need a more robust size and shape classifier to improve the overall performance of the system.



(a) whole (b) sliced (c) julienne (d) chopped (e) rangiri

Fig. 8: Input images of a sweet potato.



(a) whole (b) sliced (c) julienne (d) chopped (e) rangiri

Fig. 9: Input images of a sausage.

TABLE X: Action inference result for the images of a sausage, as shown in Fig. 9

Who	Time	Physical features (shape/size)			
		thin/ medium	slice/ small	slice/ small	grain/ small
N	NC	slice	julienne	julienne	pour
N	-	slice	julienne	pour	
N	F	rangiri	pour		
Y	NC	slice	julienne	julienne	pour
Y	-	slice	julienne	julienne	pour
Y	F	slice	julienne	pour	

V. Conclusions

This paper proposed a framework to estimate cooking affordances based on the physical features of ingredients for its use in robot task planning. We proposed a probability model based on the notion of Affordances, where based on the physical features of a given ingredient we define the probability of an action to be executed next. We define a priority level cost function that takes into account user's desired time for cooking and whose going to eat the prepared meal. We showed experimental results for five different ingredients (not included in the data set for training the classifiers of size and shape), which verified the validity of the proposed framework.

As a next step, we would like to improve the performance of the size and shape classifiers and contemplate adding other physical features. Also, increase the number of cooking actions that the system can infer and improve the priority level cost function (to be independent of many parameters). We would also like to extend our system and use recipes taken from the internet and complement these instructions with our system.

References

[1] M. Bollini, S. Tellex, T. Thompson, N. Roy, and D. Rus, "Interpreting and executing recipes with a cooking robot," in *Experimental Robotics*. Springer, 2013, pp. 481–495.

[2] K. Yamazaki, Y. Watanabe, K. Nagahama, K. Okada, and M. Inaba, "Recognition and manipulation integration for a daily assistive robot working on kitchen environments," in *IEEE International Conference on Robotics and Biomimetics*, 2010, pp. 196–201.

[3] X. Mu, Y. Xue, and Y.-B. Jia, "Robotic cutting: Mechanics and control of knife motion," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 3066–3072.

[4] R. Paul, "Classifying cooking object's state using a tuned VGG convolutional neural network," *CoRR*, vol. abs/1805.09391, 2018. [Online]. Available: http://arxiv.org/abs/1805.09391

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[6] A. B. Jelodar, M. S. Salekin, and Y. Sun, "Identifying object states in cooking-related images," *CoRR*, vol. abs/1805.06956, 2018. [Online]. Available: http://arxiv.org/abs/1805.06956

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[8] J. J. Gibson, *The theory of affordances, in Perceiving, Acting, and Knowing. Towards an Ecological Psychology.* Hoboken, NJ: John Wiley & Sons Inc., 1977, no. eds Shaw R., Bransford J.

[9] Kikkoman Corp., "Japanese Cutting Techniques," https://www.kikkoman.com/en/cookbook/jstyle/japanesecutting.html, (accessed on September 14th, 2019).