Project 1: The State Classification Challenge

Introduction:

Object recognition for coarse estimation of the object type has achieved human-like accuracy. However, for robotics, recognizing object type alone is usually not sufficient. When perform a manipulation tasks, a robot would need to know if the desired outcome of the manipulation has been reached or not. This would require fine-grained object state recognition. It is especially important for robotic cooking.

Robotic cook is very useful especially for seniors and people with severe physical disabilities. Since cooking involves many manipulation tasks that requires physical interaction, such as cutting, it is still very challenging for a robot to perform those tasks in a general kitchen setting [1]. It is well-known that robotic cooking faces tremendous challenges in basic physically-interactive manipulations [2-4] and task-oriented grasping [5-11]. However, the semantic flow of the cooking process has not been fully explored [28].

To perform a cooking task, a robot would need to know not only the cooking process such as a recipe, but also the innate relationship between cooking utensils and food ingredients [12,13]. Once the innate relationship between objects are leaned, it could be used to help robots to recognize the right object and carry out the correct manipulation motions [14].

To learn those relationships, instructional cooking videos on YouTube and many cooking sites could be automatically process to extract the relationships between cooking utensils and food ingredients along with cooking actions [15]. The learned knowledge of cooking is represented by a functional object-oriented network (FOON) [16, 17]. Different from other knowledge representations for service robots [18], FOON connects objects/states and motions.

Object states plays an important role in FOON as it does in cooking. For example, a robot should recognize if an onion is a full onion or a half onion since the robot needs to grasp the onion differently. The robot should also recognize if the onion is a peeled onion or an unpeeled one since the processing steps would be different for the two cases.

An initial state recognition for cooking has been carried out in a small scale [19,29]. In the paper, objects and ingredients in cooking videos were explored and the most frequent objects were analyzed. The paper summarized and examined 11 states from the most frequent cooking objects. A dataset of images containing those objects and their states was created [20].

The state recognition could be tackled using many different convolutional neural networks. In [19], A Resnet [21] based deep model was used and initialized with Imagenet weights and trained on the dataset of 11 classes. VGG nets [22], GoogleNet [23], Inception V3 [24] have also been used.

State recognition is related to captions for images and videos. In [25] Yao et al. use attributes and their interactions with deep networks to provide captions. Other work such as [26], and [27] perform multi-label classification on a single image using RNN- and CNN-based deep architectures. Although these papers provide various labels for an image, they do not consider states of objects as another label for the image.

1) Project Goal:

In this project students will be designing a deep convolutional neural network to classify an image of a cooking object to one of its states. For example given an image of a *"sliced tomato" or "sliced bread"*, the network should give as output *"sliced"*.

2) Project Dataset:

The dataset contains 17 cooking objects (chicken/turkey, beef/pork, tomato, onion, bread, pepper, cheese, strawberry, ...) with 11 different states (whole, julienne, sliced, chopped, grated, paste, floured, peeled, juice, mixed, other). This dataset contains 9309 images. For more information about the dataset we refer the students to https://arxiv.org/abs/1805.06956/.

Other food-related datasets:
http://pic2recipe.csail.mit.edu/
http://www.ivl.disco.unimib.it/activities/food-recognition/

3) Project Description:

The project includes three milestones as follows:

Stage	Description	Due
A	Data Annotation and preprocessing	Feb 7
B	Design and training a deep network for state classification.	Feb 21
	leaderboard will be updated	Feb 25
C	Write a technical report.	Feb 28

A) Data Annotation:

Email Ahmad Babaeian Jelodar <ajelodar@usf.edu> to request your dataset for annotation!

Data annotation starts on February 1. The data annotation includes one batch of approximately 200 frames (images). You will be assigned one (or two) objects with a google drive link which contains a directory of frames (images). You have to annotate all frames containing the objects assigned to you. To annotate you need to create a bounding box around the object (e.g. potato) in the frame and also specify the state (e.g. sliced) of that object. You should do this for every frame that contains the assigned object. After annotation you should dump the results into PASCAL VOC format. For annotation you need to create an account in the cvat.org website. A video has been uploaded to canvas that explains how to create an account in cvat.org, how to create a task, how to annotate and finally how to dump the results. You need to create a zip file from the dumped annotations files and upload it on canvas. After annotations are done, each student will be given a random set of videos from other students to check for annotation errors. Students could lose points for poor annotation.

B) Training:

On February 7, a sample Python code for training a neural network for state classification will be given to you. Students should build upon the code and create their own convolutional neural network. Students should use the collected data from milestone A to train the architecture. You should split the data given to you into train and validation data and use train data for training your model. You should validate your model during and after training with your validation data. You should add convolutional layers, pooling layers and test various deep learning techniques to improve your results as much as possible. Failure to add various layers will result in a deduction of your points. ***The code should be implemented in Python. You can use either Tensorflow or PyTorch. No other programming language or platform is acceptable.*** The provided code from the TA is in PyTorch. We will test your code on unpublished test data and report the results to you 2 days after your submission. If you substantially change the code or have your own implementation of the model, you have to provide instructions on how to run your code and also help files (readme.txt and requirements.txt if needed) and a test script to run on the test portion of the dataset. If your program is not easily test-able on the test set, you will loss 10 pts.

C) Report:

After training a deep model and running experiments, the student is required to write a report to explain the model, experiments, and results. The report should be written in the format of a technical paper. You could find an acceptable paper template at http://ras.papercept.net/conferences/support/files/ieeeconf.zip (latex) or http://ras.papercept.net/conferences/support/files/ieeeconf_letter.dot (Word). Your paper should contain the following sections (or similar sections):

Abstract 1. Introduction 2. Data and Preprocessing 3. Methodology 4. Evaluation and Results 5. Discussion Reference

Here are several examples from the dataset's webpage
http://rpal.cse.usf.edu/datasets_cooking_state_recognition.html
Your paper should be around 4-6 pages long.

The report should include evaluations on accuracy and loss during training using an epoch-accuracy and epoch-loss graph. You may include confusion matrices of the correct and incorrect classifications, comparisons of different experiments such as various learning rates, optimizers, or architectures in the report.

4) Grading Rubric:

Points	Section	Description
20 pts	Data Annotation	Accuracy and preciseness of the labels.
40 pts	Network Design & Training	Parameter training, various training options, various architecture design and modifications, Fine-tuning. Data augmentation and preprocessing.
20 pts	Accuracy	Classification accuracy of your trained model on unpublished test data.
20 pts	Report	Structure of the paper should be logically sound, The description of the approach should be clear. The figures should illustrate the network and help in presenting the results. The results should be thorough and clearly presented. Results should be analyzed and discussed through graphs and figures. The paper should be easy to read. 6% for each section.

References:

[1] Sun, Yu. "AI Meets Physical World--Exploring Robot Cooking." *arXiv preprint arXiv:1804.07974* (2018).
[2] Sun, Yu, and Joe Falco. "Robotic Grasping and Manipulation: First Robotic Grasping and Manipulation Challenge, RGMC 2016, Held in Conjunction with IROS 2016, Daejeon." (2018).
[3] Huang, Y. and Sun, Y. (2017). Learning to Pour, IROS, pp 7005-7010

[4] Huang, Y. and Sun, Y. (2018) A Dataset of Daily Interactive Manipulation, International Journal of Robotics Research (Accepted)
[5] Huang, Yongqiang, and Yu Sun. "A Dataset of Daily Interactive Manipulation." *arXiv preprint arXiv:1807.00858*(2018).
[6] Sun, Y., Yun Lin, and Yongqiang Huang (2016) Robotic Grasping for Instrument Manipulations, URAI, 1-3 (Invited)
[7] Lin, Y. and Sun, Y., 2016. Task-oriented grasp planning based on disturbance distribution. In Robotics Research (ISRR). Springer International Publishing, pp. 577-592.
[8] Lin, Y. and Sun, Y., (2015) Grasp Planning to Maximize Task Coverage, Intl. Journal of Robotics Research, 34(9): 1195-1210.
[9] Lin, Y., and Sun, Y. (2015) Robot Grasp Planning Based on Demonstrated Grasp Strategies, Intl. Journal of Robotics Research, 34(1): 26-42.
[10] Lin, Y. and Sun, Y. (2015) Task-Based Grasp Quality Measures for Grasp Synthesis, IROS, 485-490.
[11] Lin, Y., Sun, Y. (2014) Grasp Planning Based on Grasp Strategy Extraction from Demonstration, IROS, pp. 4458-4463.
[12] Sun, Yu, and Yun Lin. "Modeling paired objects and their interaction." *New Development in Robot Vision*. Springer, Berlin, Heidelberg, 2015. 73-87.
[13] Ren, Shaogang, and Yu Sun. "Human-object-object-interaction affordance." *Robot Vision (WORV), 2013 IEEE Workshop on*. IEEE, 2013.
[14] Sun, Yu, Shaogang Ren, and Yun Lin. "Object–object interaction affordance learning." *Robotics and Autonomous Systems* 62.4 (2014): 487-496.
[15] Babaeian, J.A., Paulius, D. and Sun, Y. (2019) Long Activity Video Understanding using Functional Object-Oriented Network, IEEE Transactions on Multimedia, 21(7): 1813-1824.
[16] Paulius, David, et al. "Functional object-oriented network for manipulation learning." *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016.
[17] Paulius, David, Ahmad B. Jelodar, and Yu Sun. "Functional Object-Oriented Network: Construction & Expansion." *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018.
[18] Paulius, David, and Yu Sun. "A Survey of Knowledge Representation and Retrieval for Learning in Service Robotics." *arXiv preprint arXiv:1807.02192* (2018).
[19] Jelodar, Ahmad Babaeian, Ms Sirajus Salekin, and Yu Sun. "Identifying Object States in Cooking-Related Images." *arXiv preprint arXiv:1805.06956* (2018).
[20] http://rpal.cse.usf.edu/datasets_cooking_state_recognition.html
[21] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning.” In AAAI, vol. 4, 2017, p. 12.
[22] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.
[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich et al., “Going deeper with convolutions.” Cvpr, 2015.
[24] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826.

[25] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei. Boosting image captioning with attributes. In 2017 IEEE International Conference on Computer Vision (ICCV), volume 00, pages 4904–4912, Oct. 2018.
[26] S. Venugopalan, L. A. Hendricks, M. Rohrbach, R. Mooney, T. Darrell, and K. Saenko. Captioning images with diverse objects. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1170–1178, July 2017.
[27] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu. Cnn-rnn: A unified framework for multi-label image classification. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2285–2294, June 2016
[28] Paulius, David, and Yu Sun. "A survey of knowledge representation in service robotics." *Robotics and Autonomous Systems* 118 (2019): 13-30.
[29] Jelodar, Ahmad Babaeian, and Yu Sun. "Joint Object and State Recognition Using Language Knowledge." *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019.