Cooking State Recognition from Images via Fine-Tuned VGG19 Model

Md Taufeeq Uddin
University of South Florida
mdtaufeeq@mail.usf.edu

***Abstract*—In order to perform automated tasks such as cooking, an agent (robot) must have a good understanding of cooking objects, and interaction between numerous cooking objects to carry out a set of actions such as grasping, manipulation. In this work, we focus on training computer (robot) how to recognize different cooking objects with a specific cooking state from cooking instructional videos using convolutional neural networks (fine-tuned VGG19 based model). Our experimental results show that our model recognizes cooking states with an accuracy of 71.4% and 68.4% on validation and test dataset, respectively.**

***Index Terms*—Cooking State Recognition, Transfer Learning - VGG19.**

I. Introduction

The application of robotic systems (e.g., cooking, cleaning) has a crucial impact on our society - assisting disable or senior citizens to perform daily activities such as cooking, cleaning. In order to perform this sort of tasks, a robotic system needs to perform a set of complicated tasks such as grasping, motion planning, control, and perception. In the case of cooking, given a cooking instruction at high level via text or audio, a robot should be able to prepare a meal all by itself. To do so, a robot must be able to recognize different cooking objects, different states of an object, the interplay between different objects, the interplay between different states the objects' are in. Fortunately, we can teach a robot to do all of these distinct tasks and cooking dynamics using machine learning and multimodal data e.g., videos, motion data.

In this work, we aim to teach a robot different states of a given cooking object from raw images using state-of-the-art transfer learning approach [2]. First, we extracted raw images from Youtube videos and then annotated the cooking object with the state using a bounding box to create labeled data. Second, we fine-tuned the VGG19 model [3] using labeled (cooking state recognition) dataset. In our final model, we included a few new layers on top of fine-tuned VGG19 model including one global max-pooling layer, two fully-connected layers, two drop-out layers and finally one softmax layer. We explored learning rate, optimization algorithms such as RMSProp, Adam to figure out the best possible cooking state recognition model. Our (final) submitted model obtained recognition accuracy of 71.4% and 68.4% on validation and test dataset, respectively.

II. Related Work

Over the last decade, object recognition [4] from stationary and dynamic data in generic and wild settings becomes one of the crucial topics of study to the researchers given the rise of deep learning, the ubiquity of data, and computing power. There are some application-specific studies available as well. For example, Sun et al. [5] provided a broader overview of the robotic system for cooking. The author demonstrated how a robotic chef will take a text level command about a recipe, and then perform a set of tasks such as perception, manipulation and grasping in a dynamic environment to prepare a recipe. The author also developed a computation pipeline named FOON (functional object-oriented network) [6] in which they used convolutional neural network to recognize states of objects from images and deep recurrent network for motion generation.

There are few studies available in the literature which mainly focused on the object (and its state) recognition in the context of cooking. For instance, Jelodar et al. [1] introduced the cooking state recognition task from a robotics point of view. They modeled the state recognition task as multiclass (11 classes) classification task. The extracted raw images from Youtube cooking instruction videos and annotated the images using bounding boxes to label objects and states a given object is in. Finally, for recognition task, they used a ResNet based fine-tuned classification model which obtained decent accuracy in benchmark dataset. There are some other studies [7]–[9] available in the literature that also use transfer learning approach [2] for cooking state recognition, in which researchers modeled the cooking state classification as a seven class classification problem to recognize the state of a given cooking object using fine-tuned VGG16, Inception V3 architectures [10] with some additional layers.

III. Dataset

Cooking state recognition dataset is available at `http://rpal.cse.usf.edu/datasets_cooking_state_recognition.html`. This dataset consists of images of cooking objects such as onion, tomato etc, with different states such as sliced, diced, etc. There are 11 different cooking states in total including creamy-paste, diced, floured, grated, juiced, julienne, mixed, other, peeled, sliced, and whole states. Annotated images were extracted from cooking instruction videos (in which a subject performs a cooking task deploying a camera on his/her forehead). The training and validation dataset has 6348 and 1377 images, respectively. A sample of the dataset representing 11 cooking states is shown in Figure 2.

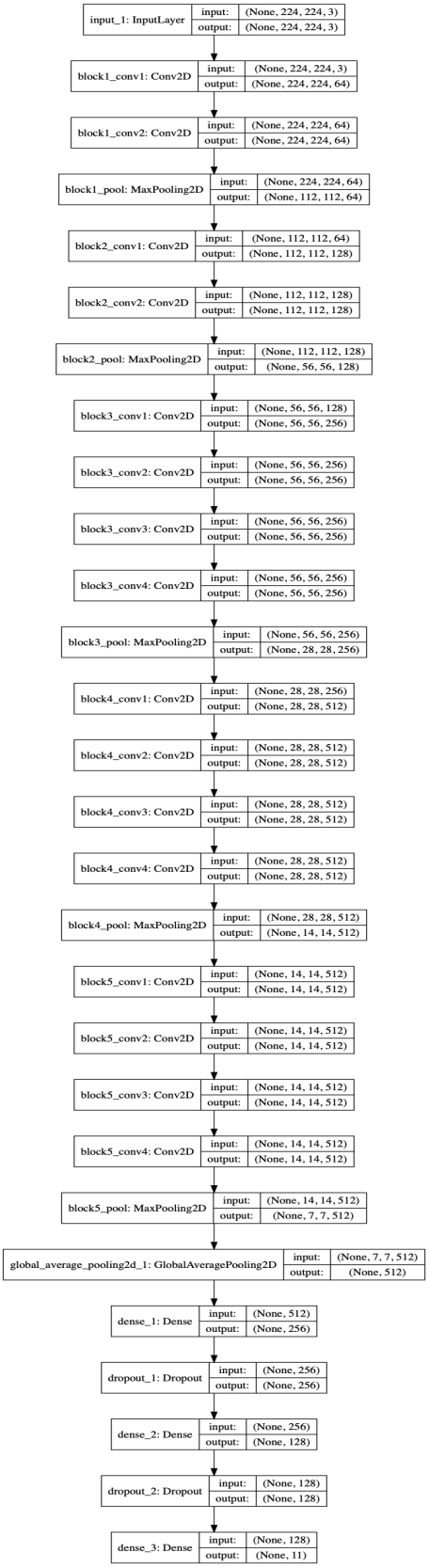


Fig. 1: Cooking state recognition model.

IV. Method

A. Data processing

In this work, we augmented the training dataset by performing horizontal flip operation. All images in training, validation and test dataset are resized to 224X224 (i.e., the number of pixels in horizontal and vertical directions are 224 and 224, respectively).

B. VGG19 based Cooking State Recognition Model

VGG19 is a deep convolutional neural network which was trained on ImageNet dataset [11]. VGG19 contains five different blocks; each block contains multiple convolution layers and one maxpooling layer. It also contains three fully connected layers on top of the convolutional layers. In this work, we fine-tuned our cooking state recognition model by freezing all layers of VGG19 model. We then included one global maxpooling layer, two fully connected layers, two drop-out layers, and one softmax classification layers on top of VGG19 model. A pictorial representation of the cooking state recognition model is shown in Figure 1. The total number of trainable parameters is 165,643 and non-trainable parameters is 20,024,384.

C. Parameters Tuning

To find out optimal model, we experimented with batch size of 32, 64, and 128. We finally chose the batch size of 128 for our final submission. The learning rate has experimented with value: 0.01, 0.001, and 0.0001; finally, we used a learning rate of 0.001 since it provides a good balance in terms of training time and accuracy. We experimented with different number of hidden units for our fully connected layers, e.g., 256, 128, 32, 16. We also experimented with two optimization algorithms such as RMSProp [12], and Adam [13]. Since the size of the cooking state recognition dataset is comparatively small to train a deep model, overfitting was a concern; as a consequence, we experimented with drop-out [14] as a regularizer. The impact of hidden units, optimizer, and drop-out in the model building process is demonstrated in Section V.

V. Experimental Results

In our experiment, we used a separate training dataset to train the recognition model and validate the learning on a separated validation dataset. Finally, we tested our final trained model on the unknown test dataset. To evaluate the generalization of the trained model, we plotted loss per epochs using training and validation dataset. To evaluate the recognition performance, we use several evaluation metrics such as accuracy, confusion matrix, precision, recall, and f1-score [15].

A subset of the designed and experimented models are included in Table II. The training and validation loss per epochs for each of the model is shown in Figure 3. From Table II and Figure 3, we can observe that when the number of hidden units is higher, the difference between training loss and validation loss is higher; due to overfitting. As a consequence,

(a) Creamy paste (b) Diced (c) Floured (d) Grated (e) Juiced (f) Julienne

(g) Mixed (h) Other (i) Peeled (j) Sliced (k) Whole

Fig. 2: Sample images from the cooking state recognition dataset [1].

TABLE I: Performance evaluation on validation dataset using model 3. Here, in our confusion matrix, N represents the total number of images in the validation dataset; horizontal direction indicates classified cooking states, and vertical direction indicates original cooking states; Pr., Re. and F1 indicate precision, recall, and f1-score, respectively.

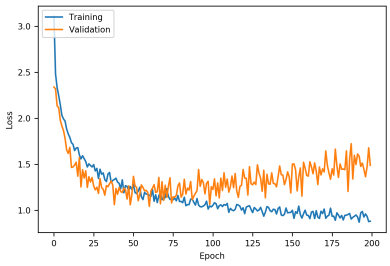
N = 1377	**Cr. Paste**	**Diced**	**Floured**	**Grated**	**Juiced**	**Jullienne**	**Mixed**	**Other**	**Peeled**	**Sliced**	**Whole**	**Pr.**	**Re.**	**F1**
Cr. Paste	78	2	3	4	4	1	1	7	1	3	1	0.66	0.74	0.70
Diced	3	84	2	4	1	2	4	8	0	3	1	0.78	0.75	0.76
Floured	1	2	73	1	1	4	0	8	0	14	6	0.80	0.66	0.73
Grated	16	4	2	72	4	11	0	3	0	3	1	0.77	0.62	0.69
Juiced	5	0	2	1	89	0	0	1	2	1	0	0.84	0.88	0.86
Jullienne	1	0	1	3	0	86	3	5	2	6	1	0.75	0.80	0.77
Mixed	0	0	0	0	0	5	88	6	0	0	0	0.80	0.89	0.84
Other	6	7	2	6	3	2	13	73	5	19	7	0.47	0.51	0.49
Peeled	3	3	0	0	0	2	0	3	73	6	11	0.74	0.72	0.73
Sliced	5	7	1	2	0	2	0	26	2	161	9	0.69	0.75	0.72
Whole	1	1	3	0	4	0	1	15	13	16	113	0.75	0.68	0.71

TABLE II: Influence of hyper-parameters. Here, each row of the table represents one specific model.

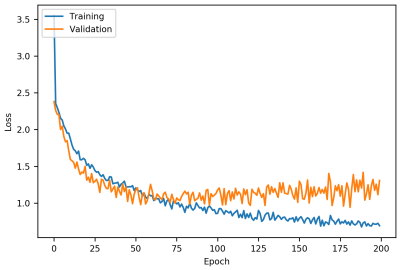
Model	**Hidden Units**	**Optimizers**	**Drop-Out**
1	(32, 16)	RMSProp	(0.2, 0.2)
2	(32, 16)	Adam	(0.2, 0.2)
3	(256, 128)	RMSProp	(0.2, 0.2)
4	(256, 128)	RMSProp	(0.5, 0.5)
5	(256, 128)	Adam	(0.5, 0.5)

we changed the drop-out rate to a higher value (0.2 to 0.5), and we observed a decent improvement in terms of generalization. We can infer that drop-out could decrease the bad influence of overfitting. According to Figure 3 and 4, we report that the fluctuation in validation loss and accuracy from one epoch to another is significantly higher than training loss and accuracy. One of the reasons behind that might be the size of the validation dataset (1377 images).

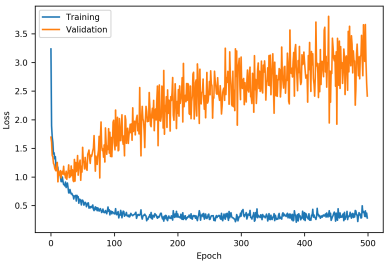
From Figure 4, we can see that deep model leads to better recognition performance. However, the deeper model also incorporates overfitting as we can see in Figure 4(c). Note that using the model 1, model 2, model 3, model 4 and model 5, we obtained cooking state recognition accuracy of 64.7%, 67.8%, 71.9%, 67.4%, 71.3%, respectively on validation dataset. Recall that model 3 was our final submission; we achieved 68.4% accuracy on the test dataset using model 3. From Table I, we observe that the recognition model fails to recognize ”other” state with decent F1-score (0.49) compared to rest of the cooking states. The reason behind that could be internal variability in ”other” state since in our experimental design, we put all possible states in ”other” state category, aside from the states (e.g., diced, juiced, sliced, etc) available in the dataset. We can also observe that model 3 misclassified most of the cooking states (especially sliced, whole) as ”other” state in a significant number of times. Model 3 also misclassified grated state as creamy paste and julienne states, floured state as sliced state. Finally, based on the lower right segment of confusion matrix in Table I, we can infer that inter-state similarity between other, peeled, sliced, and whole is higher which leads to higher misclassification rate in that specific



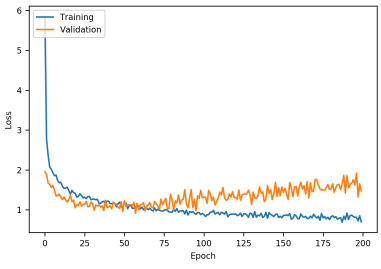
(a) Model 1



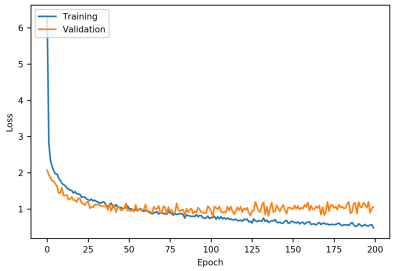
(b) Model 2



(c) Model 3

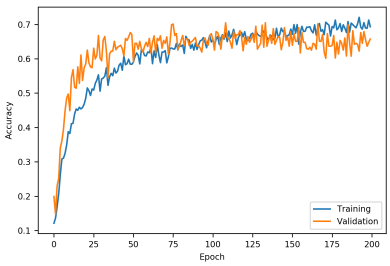


(d) Model 4

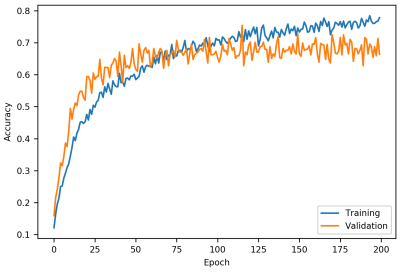


(e) Model 5

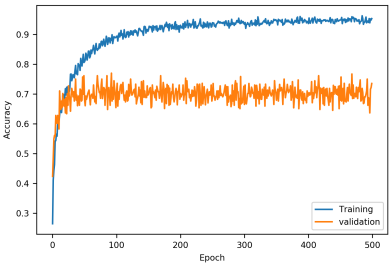
Fig. 3: The depiction of generalization of cooking state recognition model on training and validation dataset.



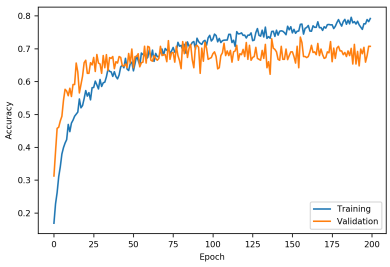
(a) Model 1



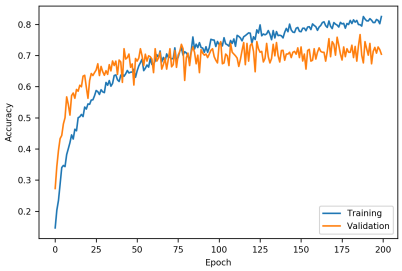
(b) Model 2



(c) Model 3



(d) Model 4



(e) Model 5

Fig. 4: The depiction of recognition accuracy of cooking state recognition model on training and validation dataset.

segment of the confusion matrix.

VI. Conclusion

In this paper, we recognized cooking states from raw images using VGG19 based fine-tuned model. Based on our experiments, we reported that deeper models (model 3, model 5) tend to recognize the cooking states more accurately than a less deep model (model 1, model 2). However, the deep model tends to overfit; hence, the usage of regularizer could be beneficial. The intra-state variability and inter-states similarity could lead to poor recognition performance. Finally, the dataset is comparatively small towards building robust cooking state recognition model. As a consequence, in our future work, we will focus on collecting more data to reduce overfit, to reduce intra-state variability and to reduce inter-state similarity.

References

[1] A. B. Jelodar, M. S. Salekin, and Y. Sun, "Identifying object states in cooking-related images," *arXiv preprint arXiv:1805.06956*, 2018.

[2] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[4] Z. Zhao, P. Zheng, S. Xu, and X. Wu, "Object detection with deep learning: A review," *CoRR*, vol. abs/1807.05511, 2018. [Online]. Available: http://arxiv.org/abs/1807.05511

[5] Y. Sun, "Ai meets physical world–exploring robot cooking," *arXiv preprint arXiv:1804.07974*, 2018.

[6] D. Paulius, Y. Huang, R. Milton, W. D. Buchanan, J. Sam, and Y. Sun, "Functional object-oriented network for manipulation learning," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2655–2662.

[7] R. Paul, "Classifying cooking object's state using a tuned vgg convolutional neural network," *arXiv preprint arXiv:1805.09391*, 2018.

[8] A. Sharma, "State classification with CNN," *CoRR*, vol. abs/1806.03973, 2018. [Online]. Available: http://arxiv.org/abs/1806.03973

[9] M. S. Salekin and A. B. Jelodar, "Cooking state recognition from images using inception architecture," *CoRR*, vol. abs/1805.09967, 2018. [Online]. Available: http://arxiv.org/abs/1805.09967

[10] 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.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," 2009.

[12] O. Wichrowska, N. Maheswaranathan, M. W. Hoffman, S. G. Colmenarejo, M. Denil, N. de Freitas, and J. Sohl-Dickstein, "Learned optimizers that scale and generalize," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 3751–3760.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[15] M. Hossin and M. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 2, p. 1, 2015.