Cooking Ingredient's States Recognition using Fine Tuning of Inception Model and Snapshot Ensemble

Saeed Alahamri[1]

***Abstract*— We have done an experimental study of cooking ingredients recognition using deep learning. In particular, we used the transfer learning approach and fine-tuning the Inception model trained on ImageNet to our dataset. Additionally, we used the snapshot ensemble to get more robust results by accounting for more variations in training. The highest results we achieved on an unseen test set is 87.79%,0.878,0.878, and 0.877 for accuracy, precision, recall, and f1-score respectively.**

I. INTRODUCTION

Food preparation requires many manipulations of ingredients of a recipe which is some time is easy for a human to do and to recognize, however, for robotics cooking tool is still challenging and an active research.

Deep learning is a representation learning technique that can learn to perform specific tasks (i.e., detection, classification, and segmentation) from images, sound, text, or numbers [1]. This technique does not require handcrafting features but instead learns discriminant and other powerful features automatically. Initially, deep learning was inspired by the biological brain functions such as communication and processing of tasks on biological neurons [2]. An example of deep learning is the common CNN, a learning algorithm that consists of one or more convolution layers that serves as the primary building block of a CNN for learning to recognize features of images. Although a version of CNN has existed since 1980, computation power and data availability had been an obstacle until recently [3].

Deep learning has shown great success in object recognition tasks [1], and image segmenation such as [4], which is attributed to a large amount of labeled data such as ImageNet [5] and the recent advances in deep learning. Although object recognition is important, the state of the object is also important which could help the robot to perform well on a certain task. Robotic cooking is considered a helping alternative for human especially for elderly people and people with disability. However, many challenges still remaining unfixed such as detecting the state of objects during an interactive environment such as current kitchen setting [6] and grasping [7].

Human decision agreement on labeling a particular object is generally better in performance than a single human because human decision is subjective. Likewise, machine learning and deep learning models ensemble is known for the robust performance since it capture the variance in the performance. In this paper, we used the Snapshot ensemble approach that mitigates the problem of training multiple neural networks [8]. Snapshot ensemble effective approach relies on training a single neural network for some epochs while using a checkpoint to save the model every certain number of epochs. In our experiment we saved the model every 5 epochs starting with the 10^{th} epoch.

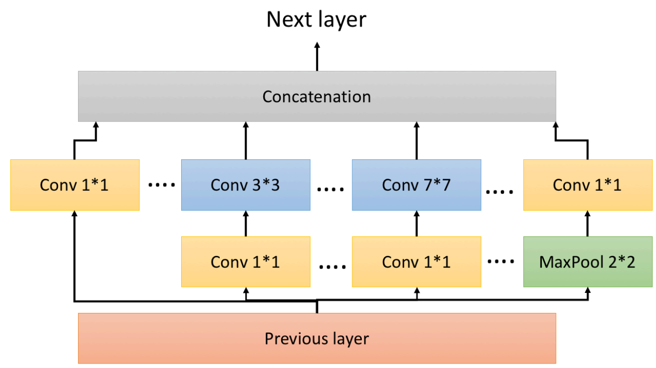

Fig. 1: Inception block

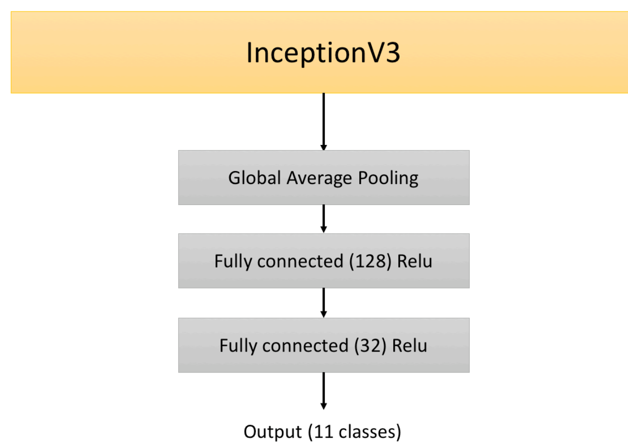

Fig. 2: InceptionV3 model after adding top layers for our classification task

In this paper, we investigate the utility of fine-tuning of previously existing neural network called Inception to identify the object states of cooking ingredients. Using the dataset (discussed later on this paper), we downloaded and fine-tuned an InceptionV3 deep neural network model with initial weights trained on ImageNet [9]. Additionally, we evaluated the performance using an Snapshot ensemble en-

[1]Saeed Alahmari is Ph.D candidate at University of South Florida, Tampa, 33647. United States of America saeed3@mail.usf.edu

semble approach using majority voting for different models saved during training.

II. Dataset

The dataset was provided by the instructor of Deep Learning class at the University of South Florida for a project assignment. This dataset has 17 different most common cooking ingredients collected from more than 250 online cooking videos from the two popular datasets [10][11][12]. The cooking ingredients include the following:

- Chicken/turkey
- beef/pork
- tomato
- onion
- bread
- pepper
- cheese
- strawberry
- milk
- potato
- garlic
- egg
- carrot
- butter
- mushroom
- orange
- cheese

Each object was labeled with a state where the state describe the ingredient during cooking process. The ingredient states includes: whole, peeled, sliced, floured, grated, julienne, diced, juiced, creamy, mixed, and other. Fig shows examples from each class as shown in Fig 3. The training set has 6348 images, validation set has 1377 images, and test set has 680 images. More information of the dataset and the labeling process is provided on [10].

III. Deep learning fine tuning

In our experiment, we chose transfer learning and fine-tuning of InceptionV3 neural network. This neural network can learn better than other neural networks due to the different sizes of the receptive field of convolution layers which is called Inception blocks. However, this is a very deep neural network, which requires time and computing resources for training. We transfer the model previously trained on ImageNet for image recognition as our initial weights. This approach helps our model converge better when fine-tuning. We have also tried to freeze earlier inception blocks while training some of the final inception blocks. We added two fully connected layers at the top of the neural network for classification. Fig 1 shows a single inception block . Fig 2 shows the Inception model after adding layers for classification.

IV. Snapshot ensemble

Our ensemble approach is inspired by [8]. This approach helps in utilizing the variation of training where a model is saved every x epoch (in our experiment $x = 5$). However, we did not use the cyclic annealing scheduling as discussed in [8][13], but rather we used the Adam optimizer [14] with a fix learning rate chosen by repetitive attempts. we used the hyper-parameters that yielded the best performance on our experiment discussed in III. We started saving the models from 10^{th} till 100^{th} epoch (for total of 100 epochs). Then we used the 19 models weights to test on an unseen test set. Each model outputs the probability for each example in the test set where the prospect represents the likelihood of being assigned to a class (i.e., a label from 11 labels on our dataset). We averaged the probabilities by added up the probability from the 19 models test results for each class, then divide the sum of probabilities by the number of models we have (i.e., 19 models). Then, the maximum likelihood is used to get the corresponding class labels.

V. Results and discussion

Fine-tuning of inception neural network trained on ImageNet with 100 epochs and early stopping with learning rate $lr = 0.00001$ showed the highest accuracy of 86.2%, precision 0.861, recall 0.862, f1-score 0.861 on the validation set. Where are the results on test set are 87.79%, 0.878, 0.878, and 0.877 for accuracy, precision, recall, and f1-score respectively. We have tried other learning rates (i.e., 0.001 to 0.00001), adding multiple convolution layers on top of the neural network, freezing different numbers of inception blocks while training the final layers, and scaling the dataset. We used Keras deep learning framework with Tensorflow [15] [16]. The results of fine tuned models testing on the validation set is shown in Table I.

Snapshot ensemble showed results that are not higher than some of the earlier epochs. However, these results confirm that the model with fine-tuning does not have many local minima; therefore, the variation is low during training. This may be attributed to the type of images we have, and the ImageNet images are closely related and partially overlapped. The results of the snapshot learning on validation data (including epoch based models results) are shown in Table II, where the accuracy is 86.56%, precision is 0.865, recall 0.866, and f1-score is 0.865. The test results of snapshot ensemble (including epochs based model results) are shown in table III, where accuracy is 86.61%, precision is 0.867, recall is 0.866, and f1-score is 0.865.

VI. Conclusion

Although cooking has become a hobby, a need for a robot to perform the task is important for needed people how cannot cook by themselves. In this paper, we experimented with transfer learning approach of Inception model trained on ImageNet and fine-tune it to our dataset to recognize the state of ingredients during the cooking process. The results showed about 87.79% accuracy on an unseen test set.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, p. 436, 2015.

Fig. 3: Examples from the dataset for 11 classes [10]

TABLE I: Inception model results for different hyper-parameters tuning

Model	Learning rate	Scaling	Freezing blocks	Validation set				Test set			
				Accuracy (%)	Precision	Recall	F1-score	Accuracy (%)	Precision	Recall	F1-score
Inception	0.00001	Yes	No	86.2	0.861	0.862	0.861	87.79	0.878	0.878	0.877
Inception	0.00001	No	All except last two inception blocks	79.23	0.805	0.792	0.792	77.2	0.79	0.772	0.775
Inception	0.0001	Yes	No	82.2	0.822	0.822	0.82	86.61	0.867	0.866	0.866

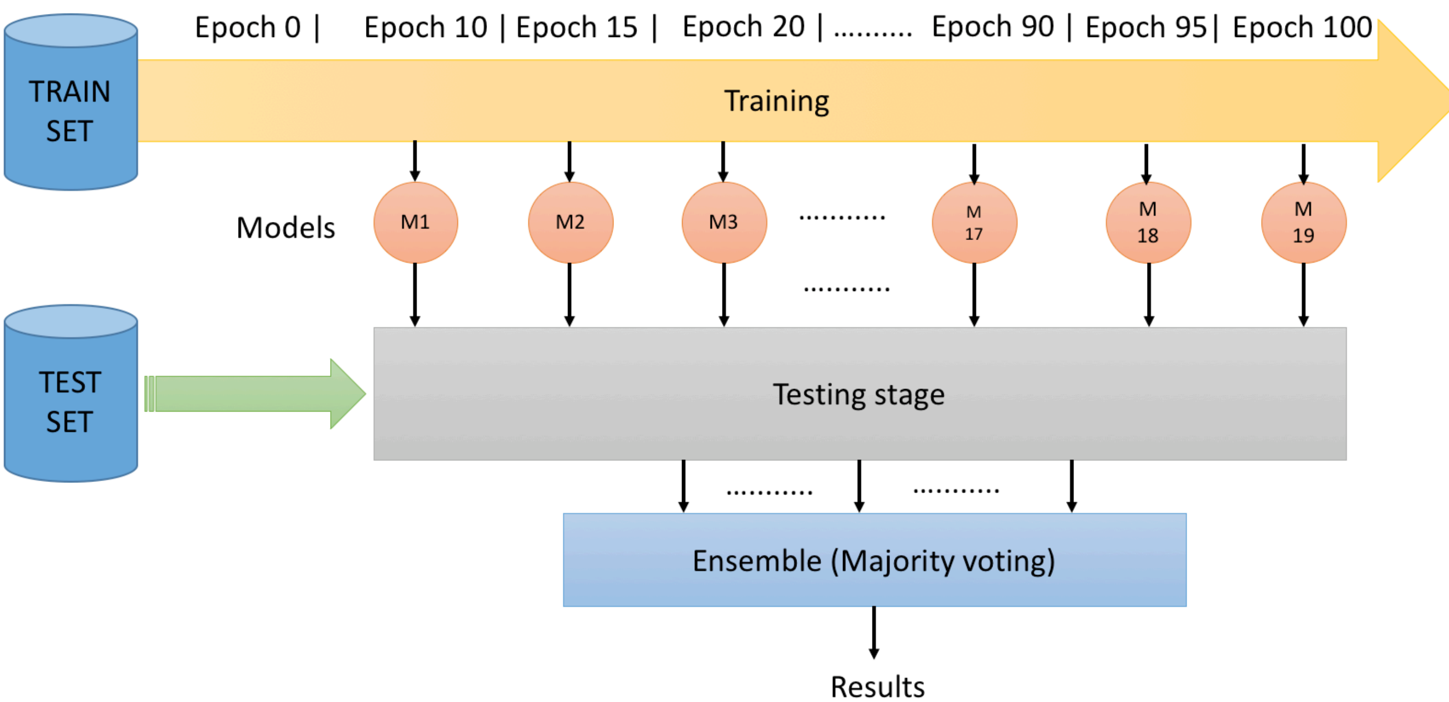

Fig. 4: Epoch based ensemble approach

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[3] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[4] S. S. Alahmari, D. Goldgof, L. Hall, H. A. Phoulady, R. H. Patel, and P. R. Mouton, "Automated cell counts on tissue sections by deep learning and unbiased stereology," *Journal of chemical neuroanatomy*, vol. 96, pp. 94–101, 2019.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

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

[7] Y. Sun, Y. Lin, and Y. Huang, "Robotic grasping for instrument manipulations," in *2016 13th International Conference on Ubiquitous*

TABLE II: Snapshot ensemble results on validation set of fine-tuned Inception neural network

Epoch number	Accuracy (%)	Precision	Recall	F1-score
10	85.476	0.853	0.855	0.853
15	85.403	0.855	0.854	0.854
20	86.129	0.859	0.861	0.859
25	85.766	0.857	0.858	0.857
30	85.185	0.851	0.852	0.851
35	85.694	0.858	0.857	0.857
40	85.113	0.852	0.851	0.852
45	85.621	0.86	0.856	0.857
50	85.839	0.859	0.858	0.858
55	85.621	0.856	0.856	0.856
60	86.783	0.866	0.868	0.867
65	85.33	0.853	0.853	0.853
70	86.347	0.861	0.863	0.862
75	86.71	0.867	0.867	0.867
80	86.783	0.868	0.868	0.868
85	85.911	0.86	0.859	0.859
90	82.716	0.831	0.827	0.828
95	85.766	0.859	0.858	0.858
100	86.638	0.865	0.866	0.865
Ensemble	86.565	0.865	0.866	0.865

TABLE III: Snapshot ensemble results on test set of fine-tuned Inception neural network

Epoch number	Accuracy (%)	Precision	Recall	F1-score
10	86.029	0.862	0.86	0.86
15	85.147	0.855	0.851	0.852
20	85.294	0.855	0.853	0.852
25	87.059	0.871	0.871	0.87
30	86.324	0.865	0.863	0.862
35	85.588	0.858	0.856	0.856
40	83.676	0.839	0.837	0.837
45	85	0.853	0.85	0.851
50	86.324	0.867	0.863	0.863
55	86.176	0.865	0.862	0.861
60	86.029	0.861	0.86	0.859
65	86.176	0.863	0.862	0.861
70	86.029	0.859	0.86	0.858
75	85.588	0.857	0.856	0.855
80	86.029	0.863	0.86	0.86
85	86.324	0.866	0.863	0.864
90	83.235	0.835	0.832	0.833
95	86.324	0.867	0.863	0.864
100	85.882	0.859	0.859	0.857
Ensemble	86.618	0.867	0.866	0.865

Robots and Ambient Intelligence (URAI). IEEE, 2016, pp. 302–304.

[8] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, "Snapshot ensembles: Train 1, get m for free," *arXiv preprint arXiv:1704.00109*, 2017.

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

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

[11] 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.

[12] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele, "A database for fine grained activity detection of cooking activities," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1194–1201.

[13] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

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

[15] F. Chollet *et al.*, "Keras," 2015.

[16] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.