State Recognition of Cooking Objects using Convolution Neural Network

Md Imran Hossain, USF, CSE(93491952)

***Abstract*— Coking object's state recognition is an important task and at the same time it is challenging due to mixture of different items in the dataset. The dataset contains various cooking items and the items is in different state. In this paper, a CNN model have been introduced to classify various states of the images. The proposed CNN model has five convolution layer and two fully connected layer. The model was trained on around 9000 input images and validation accuracy obtained is 54%.**

Keywords*— Convolutional Neural Network, Deep Learning, Tuning, cooking object classification, state classification

I. INTRODUCTION

Nowadays Deep Learning has applications everywhere from kitchen to the astrophysics. And in the field of image and video classification, deep learning is showing significant results. So researcher are now working on more complicated dataset like cooking state or future action prediction in video dataset.

Robot assisted cooking is very helpful for disabled persons or old people or even for people who are really busy in their daily life. To assist in cooking process, a robot need to know both the cooking processes and relationship between various cooking items and ingredients[6,7].

Once, robot recognize the state, it can then take the decision how the food item can be processed. And to find the relationship with cooking utensils and ingredients, cooking related video from various sources like YouTube or website can be helpful[8]. Moreover, the knowledge obtained from those sources in represented by a functional object-oriented network (FOON) [9, 10] which connect objects/states and actions.

Identifying state of an cooking object can help a robot to cutting the object or even cooking the food. For that, robot first need to identify in what state the food item is. But fine-grained object state identification is essential for robotic cooking.

For instance, a robot should distinguish if a potato is a full or a half potato since the robot requires to comprehend the potato differently. The robot should be able to recognize if the potato is peeled or unpeeled so it can process the food items differently.

Several works have already been done to identify cooking procedure[1,2] and predicting cooking task[3-5] and cooking activity classification. Various convolution neural networks model could be used to tackle object/state recognition problem. A Resnet [11], VGG Nets [12], GoogleNet [13], Inception V3 [14] model can be used to get better results. But as course requirement, I have designed a customized CNN model which shows significant results on validation dataset. Also various augmentation method has been tried on the dataset to obtain better accuracy which will be stated later section.

II. DATA AND PREPROCESSING

The dataset contains 17 cooking objects (chicken/turkey, beef/pork, potato, tomato, onion, milk, bread, pepper, cheese, strawberry) with 11 different states (whole, julienne, sliced, chopped, grated, paste, floured, peeled, juice, mixed, other). The dataset contains 9309 images.

Figure 1. Sample image from the input dataset.

In the first step of the project, data annotation has been assigned. As, part of data annotation I have labeled the state of the milk which is either in liquid state or creamy state. We are assigned to draw a bounding box in the region of interest. An image may contain various of object but only the targeted part of image needed to be selected. For my case, first I have selected milk in the images and then I have defined if the milk is liquid or creamy. After annotation, I have dumped the annotation results into PASCAL VOC format. Then all the annotation images have been collected for further analysis.

As the image dataset contain 9309 images, to get a better results I have applied several data processing techniques such as cropping, rotation, affine, flipping vertically and horizontally, tuning by different value of brightness. A table is shown where the augmentation is factor is specifically mentioned after tuning the various value of it. Affine transformation preserves collinearity (i.e., all points lying on a line initially still lie on a line after transformation) and ratios of distances (e.g., the midpoint of a line segment remains the midpoint after transformation)[24], hence it improves the input image quality.

TABLE I. OVERVIEW OF DATA AUGMENTATION

Name	Factors/size
Center Crop	224
Random Rotation	15 degree
Color Jitter	brightness=0.4, contrast=0.4, saturation=0.4, hue=0.2
Random Affine	degrees=5, translate=(0.05, 0.05)
Horizontal Flipping	True
Vertical Flipping	True

Fig02 shows a sample image with different augmentation method. Data augmentation is the process of adding the new data derived from the given data which has proved to be great beneficiary to classify or predict the image/state/video.

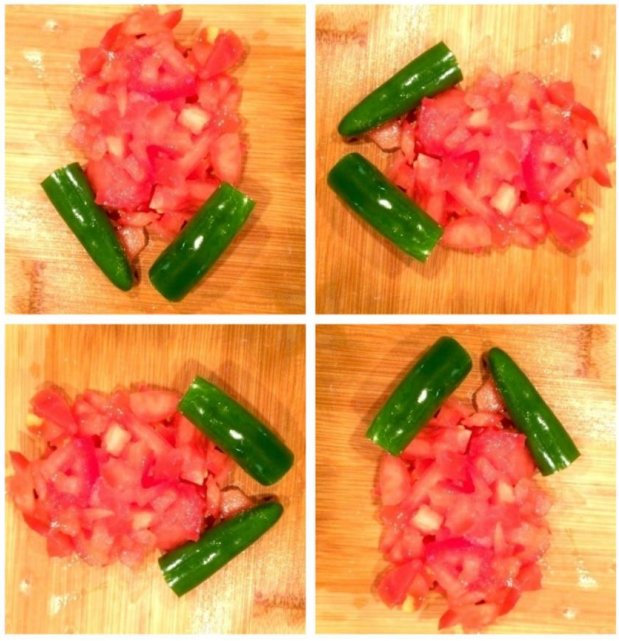

Figure 2. Data Augmentation(Horizontal flip, Vertical flip, Rorate Image)

Also ColorJitter is used to give the image different perspective like brightness, contrast, saturation ,hue which also helps to predict the results. For example, if a vegatable is in dark image, increasing the brightness may help to increase the quality of the dataset. Simialry, Hue and saturation also significaanlty improve the the dataset.

III. METHODOLOGY

Though Transfer Learning is more effective to get a good accuracy on this dataset, but the requirement is to design and implement our own CNN model. Various aspects of the Network has been investigates while designing the convolution layers. For example the main impactor is filter size, output size of convolution layers, dropout factor, learning rate, regularization factor. The architecture is shown in the figure and also the details description is given below.

TABLE II. OVERVIEW OF CNN ARCHITECTURE

Input 224x224 image	
Conv1	Conv2d (3,16,3) Maxpool2d(2,2)
Conv2	Conv2d (16,32,3) BatchNorm2d(32) Maxpool2d(2,2)
Conv3	Conv2d (32,64,3) BatchNorm2d(64) Maxpool2d(2,2)
Conv4	Conv2d (64,64,3) BatchNorm2d(64) Maxpool2d(2,2)
Conv5	Conv2d (64,128,3) BatchNorm2d(128) Maxpool2d(2,2)
Fully-Connected	fc1 = nn.Linear(128x5x5, 256) BatchNorm1d(256) fc2 = nn.Linear(256, num_classes=11)

A. *Convolution Layers*

First layer should be a convolution layer .Output is generated in the form of a Tensor as the input convolved with convolution kernel[16]. The next layer is Max pooling layer . And the kernel size is 2 and a stride value of 2 is used. To do down sampling of the data, or to sat to reduce the overfitting by reducing the dimensionality of data max pooling is necessary. Another benefit of max pooling is that it reduces computational cost by decreasing number of parameters [17]. In my model Max pooling is used five times to keep the dimensionality low.

In the second convolution layers, Batch normalization has been introduced. It helps to shifts all the values in the input

array closer to mean or zero. This method helps faster learning and getting better accuracy and it can be used anywhere in the model. In CNN, weights and parameter become large by accumulating the value, so computation can become harder. This problem can be avoided by normalizing data in each mini batch [18].

Five convolution layers have been used in the model. Initially I used 4 layers but adding one layer gives the better results. And, the filter size is selected as 3x3 also different filter size have been tried. And at the end of the convolution layers, I got the total weights (128x5x5x256).

In the last layer, flattening is used before the flows goes into fully connected layer. Flattening converts the previous layers output into a single continuous linear vector.

A dense layer is just an artificial neural network (ANN) classifier and requires individual features. Flattening is needed to convert the input array into a feature vector [19]. Dense layer is a fully connected layer where each input and output node is connected to each other. Dense layers has the flowing advantages "It solves the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters" [20]. Two fully connected layer have been used in my model.

Rectified Linear Unit (ReLU) as the activation function has become very popular and it is proved that it become 6 times more accurate in determining results than the tanh function[21]. The equation is given below.

$$R(x) = max(0,x)$$
$$i.e\ if\ x < 0\ ,\ R(x) = 0$$
$$and\ if\ x >= 0\ ,\ R(x) = x$$

ReLu is greatly successful in image or video classification problem, that's the another reason to use ReLU in my model too.

B. *Dropout Factor*

The Dropout is applied in the model which is used for regularization to reduce the risk of overfitting. As the name suggests, during training a certain number of neurons in the hidden layer is randomly dropped. We can use it in input layers too. And the factor was 0.5 in this particular CNN model. As the Dropout is introduces, it makes the model more robust[21,22].

C. *Learning Rate*

Learning rate has significant impact in model accuracy. By varying different learning rate, finally the value was chosen 0.001 in this case.

D. *Optimizer*

Adam optimizer has been selected as it is easy to implement, Computationally cost-effective, little memory needs, appropriate for challenges with very noisy/or sparse gradients

E. *Softmax*

Softmax activation function is used in the model in the output layer. Outputs are normalized to sum up to 1, so the probability remain between 0 to 1. Usually it is suggested to use softmax in the last layer of the network for classification problem[25].

IV. Evaluation and Results

Various experiment was done to get the best results like training with three convolution layers, four convolution layer and finally five convolution layers. I have shown the results of epoch vs accuracy and epoch vs loss graph for the above mentioned model.

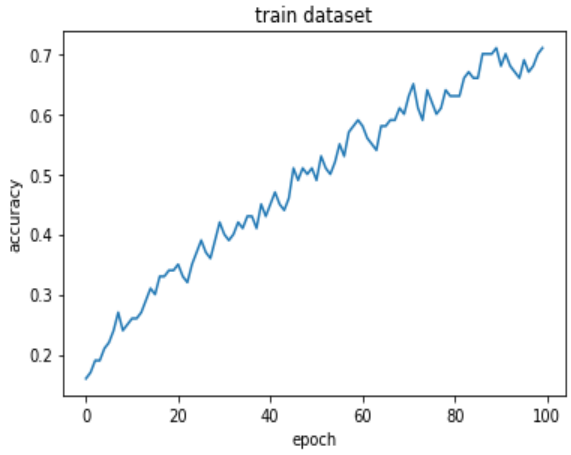

Figure 3. Epoch vs Accuracy

From the *Fig3*, it can be shown, the accuracy is increasing proportionally with respect to epoch. From the result, it can be said that the model is not overfitting. If the model is overfitting, then the accuracy should stop to increase and sometimes the accuracy starts decreasing. The training accuracy obtained with the designed model is 71%.

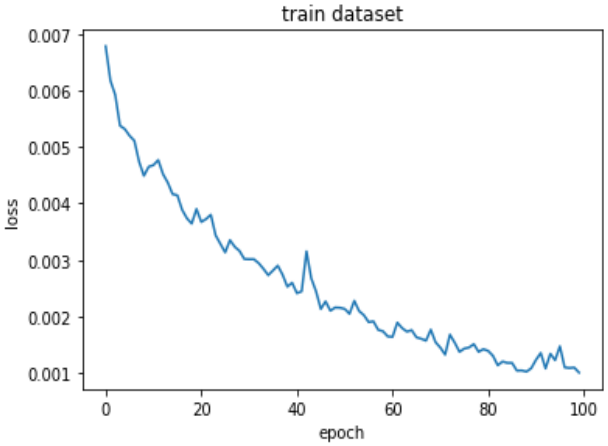

Figure 4. Epoch vs Loss

From the figure, the loss is decreasing as expected as the number of epoch is increasing. And we know, when the loss decreases, the traning process is working fine on the datstet.

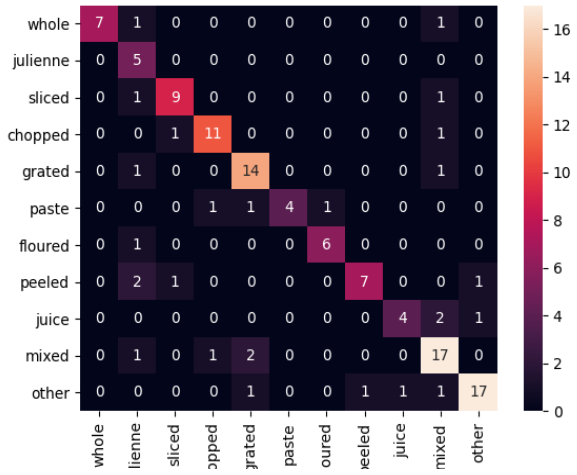

Figure 5. Confusion Matrix

Fig 05. Illustrates the confusion matrix on validation dataset on in which the x-axis is the predicted label and in the y-axis the true label. In case of 'grated' state, the model most of the time predict grated but 2 times it predicts as mixed, 1 time paste and 1 time other.

V. Conclusion

In this project, a CNN model has been designed and implemented to detect the state of cooking object. The dataset was complicated as it contains some confusing objects in the same images, also some of the images is noisy so its very possible to mislabeled them. However, my designed model was successfully able to work on validation data set and the accuracy was about 55%. The state recognition can play a vital role in robotic cooking. As the deep learning is data hungry, so higher the dataset, higher the accuracy of trained model[15]. The designed model can be improved large number of dataset with different number of instances with changing values.

References

[1] Meyers, A., Johnston, N., Rathod, V., Korattikara, A., Gorban, A., Silberman, N., Guadarrama, S., Papandreou, G., Huang, J. and Murphy, K.P., 2015. Im2Calories: towards an automated mobile vision food diary. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1233-1241)

[2] Malmaud, J., Huang, J., Rathod, V., Johnston, N., Rabinovich, A. and Murphy, K., 2015. What's cookin'? interpreting cooking videos using text, speech and vision. arXiv preprint arXiv:1503.01558

[3] Lade, P., Krishnan, N.C. and Panchanathan, S., 2010, December. Task prediction in cooking activities using hierarchical state space markov chain and object based task grouping. In Multimedia (ISM), 2010 IEEE International Symposium on (pp. 284-289). IEEE

[4] Bossard, L., Guillaumin, M. and Van Gool, L., 2014, September. Food101–mining discriminative components with random forests. In European Conference on Computer Vision(pp. 446-461). Springer, Cham.

[5] Sun, Yu. "AI Meets Physical World--Exploring Robot Cooking." arXiv preprint arXiv:1804.07974 (2018)

[6] Sun, Yu, and Yun Lin. "Modeling paired objects and their interaction." New Development in Robot Vision. Springer, Berlin, Heidelberg, 2015. 73-87

[7] Ren, Shaogang, and Yu Sun. "Human-object-object-interaction affordance." Robot Vision (WORV), 2013 IEEE Workshop on. IEEE, 2013

[8] 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

[9] Paulius, David, et al. "Functional object-oriented network for manipulation learning." Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on. IEEE, 2016

[10] 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

[11] 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

[12] 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

[13] 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

[14] 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

[15] LeCun, Yann, Bengio, Yoshua, Hinton, Geoffrey, Deep learning (2015), Nature Publishing Group, a division of Macmillan Publishers Limited

[16] Vincent Dumoulin 1 F and Francesco Visin 2 F † FMILA, Université de Montréal †AIRLab, Politecnico di Milano January 12, 2018, A guide to convolution arithmetic for deep learning

[17] A. Giusti, D. C. Cireşan, J. Masci, L. M. Gambardella and J. Schmidhuber, "Fast image scanning with deep max-pooling convolutional neural networks," 2013 IEEE International Conference on Image Processing, Melbourne, VIC, 2013, pp. 4034-4038

[18] Sergey Ioffe Google Inc., Christian Szegedy Google Inc, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, arXiv:1502.03167v3 [cs.LG] 2 Mar 2015

[19] Jin, Jonghoon & Dundar, Aysegul & Culurciello, Eugenio. (2014). Flattened Convolutional Neural Networks for Feedforward Acceleration

[20] Gao Huang∗ Cornell University, Zhuang Liu∗ Tsinghua University, Laurens van der Maaten Facebook AI Research, Kilian Q. Weinberger Cornell University, Densely Connected Convolutional Networks, arXiv:1608.06993v5 [cs.CV] 28 Jan 2018

[21]] G. E. Dahl, T. N. Sainath and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, 2013, pp. 8609-8613.

[22] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, Department of Computer Science University of Toronto 10 Kings College Road, Rm 3302 Toronto, Ontario, M5S3G4, Canada, Dropout: A Simple Way to Prevent Neural Networks from Overfitting, 2014

[23] G. E. Dahl, T. N. Sainath and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, 2013, pp. 8609-8613.

[24] Croft, H. T.; Falconer, K. J.; and Guy, R. K. Unsolved Problems in Geometry. New York: Springer-Verlag, p. 3, 1991.

[25] Alex Krizhevsky University of Toronto, Ilya Sutskever University of Toronto, Geoffrey E. Hinton University of Toronto, ImageNet Classification with Deep Convolutional Neural Networks, 2012