ISSN 2347 - 3983
Volume 8. No. 10, October 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter1508102020.pdf
https://doi.org/10.30534/ijeter/2020/1508102020

Evaluation of Convolutional Neural Network Models for Food Images Detection

Muhd Aqmal Afiq Adnan[1], Marina Yusoff[2]
[1] Clazzy Sdn Bhd, Taman perindustrian USJ1,
Subang Jaya, Selangor, aqmalafiq@yahoo.com
[2]Advanced Analytic Engineering Center (AAEC), Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia, marinay@fskm.uitm.edu.my

ABSTRACT

The food price calculation is a process of estimating food volume, usually in a plate, by guessing a countable specific container such as a spoon or mini bowl to determine the pricing of the food taken. It is a hassle to rely on humans for food calculation as humans are biased, unreliable, and inconsistent. Before performing the food calculation, food items have to be detected and recognized. This paper addresses the employment of Convolutional Neural Network (CNN) for image detection and recognition. About 600 food images are collected from local restaurants and cover three main class items; rice, vegetables, and fried chicken. The evaluation was done with three different CNN models; Inception V2, Single Short Detection (SSD) MobileNet V1, and Resnet50. Interestingly, the Inception V2 model is proven to perform with acceptable accuracy for food-related image subjects. The inception V2 is generally outperformed SSD MobileNet V1 and Resnet50 for food image detection. The findings would be beneficial to restaurants owner and customers for an automated image detection of food.**Key words:** Convolution Neural Network, Food Images, Image Detection, Inception V2, SSD MobileNet V1 and Resnet50

1. INTRODUCTION

The self-servicing restaurant is a common sight to see a long queue on its payment counter as customers line up to have their hand-picked food calculated by the staff, and this is a slow and repetitive process that could cause boredom the team and make the situation even worse. Long queues in the restaurant during peak hours are not rare, especially when the restaurant is famous for its delicious food and customers are willing to spend hours queuing. A study made in Britain that a person snaps after waiting more than 24 minutes [1], especially during workday where most customers are there rushing to get their food for lunch, timing themselves to finish their meal and in time get back to their workplace. Thus, a restaurant that fails to entertain its customers in time might lose its customers, which negatively impacts their income.

Restaurants and canteen especially have their food made in large quantities and would be piled up for display on food counter where customers would personally hand-pick their food. The business model is widely adopted as a customer would not have to wait for their food, and less amount of worker are needed when in compared to full-service restaurants, It was reported that 10% of its members had started "self-service", where customers have to get the dishes by themselves due to the shortage of worker [2]. However, many customers can be accommodated by the self-serving model at a time, causing a bottleneck to the payment counter and massive strain to the person responsible for quoting the total food price for each of the queuing customers.

Due to human intervention in price quoting, there is a chance for the customer's price to be inconsistent [3]. The inconsistency is not an isolated case, but it is a well commonly known. Thus, a clear and concise manner on how the food is calculated is essential, and failure to resolve the issue might give a wrong impression to the business name itself. However, before performing an automated price calculation, the images of food have to be detected. Food images of Malaysian food like ‘Nasi Campur’ seem unique as it consists of many types of items on one plate.

In terms of image recognition, many methods were established. Image detection and recognition solution methods have evolved. Much methods and approaches were produced and evaluated with small, medium, and large images datasets and various images decades [4-7]. Some of the popular techniques are neural networks, support vector machines, and deep learning. Recently, deep learning is becoming popular; for instance, Image Net has seen an excellent capability of detecting various type images based on pre-trained images [8]. Generally, deep CNN provides proper image recognition. However, training and testing datasets require for convergence and accuracy measurement, respectively [16]. Besides, CNN model architecture needs a careful determination to ensure an effective performance such as the convolutional layer are computed with three hyperparameters to later deciding on the size of the output in terms of depth, stride, and padding [13]. The same goes for [9] in food detection and recognition with CNN. They use local response normalization for normalization after the pooling layer, dataset image scaled to 64x64, and the dataset divided where 4 set used for training while the remaining 2

used for validation and testing [12]. Currently, many models are developed to improve image detection capability further, and the models are problem-dependent. Therefore, this paper highlights the evaluation of different CNN models to assist in detecting and recognizing 'Nasi Campur' images.

2. CONVOLUTION NEURAL NETWORK IMPLEMENTATION

2.1 Food Images Acquisition and Preparation

Food images were collected from a local restaurant. Three main classes of the food item are inside the image collect: the rice, vegetables, and fried chicken, each of the food dishes grouped to avoid potential image segmentation problem, and the image are color with good lighting to ensure good image quality. About 200 images of data labeling are made by a human expert to classify each of the image's food items. Both of the data and labels would be stored in a database creating a food image dataset. Figure 1.0 shows the sample of datasets that were collected from the selected restaurants.

The input image would have to go through a preprocessing layer where they have to undergo resizing and normalization to ensure the model's input image compatibility. However, the amount of preprocessing keeps minimal as CNN's nature does not require a substantial image preprocessing task [18].

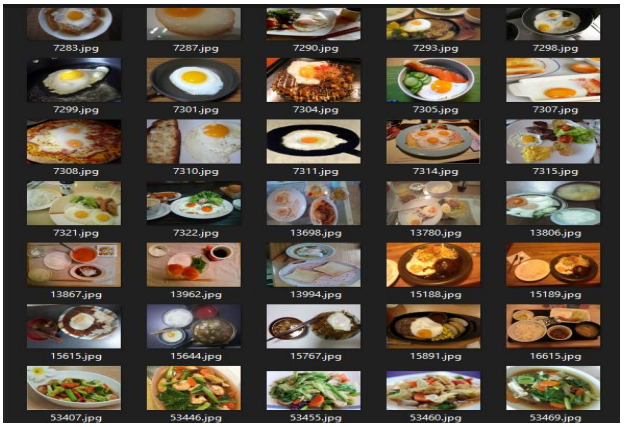

Figure 1: Sample of datasets

2.2 Models of Convolution Neural Network

CNN has excellent capability in detecting patterns with image processing. CNN is different than an Artificial Neural Network. It consists of many layers of neurons, but an additional layer that makes it distinct from the other neural network is the existence of the convolution layer and pooling or subsampling layer. The convolution and pooling layer can be layered multiple times in a typical CNN, the convolution network focuses on the extraction of unique features in the image by using filters, and these filters are called convolution kernels. In contrast, the pooling layer is the activation over a rectangular region in an image resulting in the constant output in terms of position. The final layer is like many neural networks where its output indicates the probability of the predicted class. Hyperparameter consisting of the number and size of kernels and number of the layer is also the factor affecting model accuracy [9] In general, CNN starts with the Convolutional Layer in the network, and it produces a feature map by extracting features of interest or unique features from the input image by using feature detector [18].

This unique feature can be a line or curves that might exist commonly in the same picture label (category) and differentiate the picture label from the rest of the picture label. Moreover, there is a convolution function defined by several parameters such as input size, kernel size, depth of the map stack, zero paddings, and stride in the Convolutional Layer. There is also an activation function where the most commonly used one is the Rectified Linear Units due to its ability to improve the CNN performance [18]. CNN implementation using several models depending's on the datasets. In particular, the models evaluated are the SSD MobileNet [19] Region Convolutional Neural Network (RCNN) Inception V2 [20], and Resnet50 (Yu et al. (2016)). SSD MobileNet. Figure 2 demonstrates the RCNN Inception V2 on a pre-trained Inception V3 CNN architecture [21].

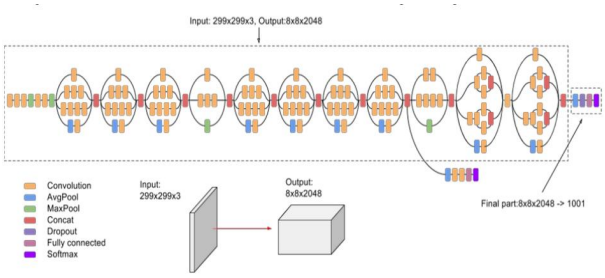

Figure 2: The architecture of RCNN Inception V2

The network architecture of the RCNN Inception V2 produces a good result on the food image dataset [20]. Although its excellent capabilities, a good CNN model is needed to ensure reliable object detection. Our subject is the food, and due to its sophisticated features, such as contour, shape, varied colors, and texture, make it very hard to identify its class, and adding like curry or soup affects and contaminate the subject features significantly.

4. COMPUTATIONAL RESULTS AND DISCUSSION

The results were based on three model Inception V2, SSD MobileNet V1, and RCNN Resnet50 as discussed in the following sections.

4.1 Model I: RCNN Inception V2

For the first try out of using RCNN Inception V2 with the parameters setting set to default, the models have a dynamic

Learning Rate and SOFTMAX converter. The result of the experiment is demonstrated in Figure 3.

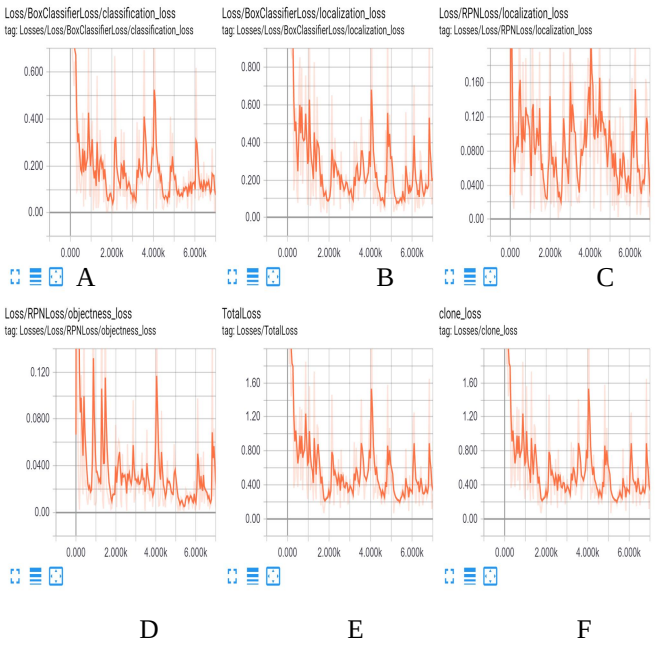

Figure 3: Top from left to right (A) Classification loss, (B) Box Localization Loss, (C) RPN localization loss, (D) objectness loss, (E) Total Loss and (F) clone loss TensorBoard Loss Graph Before Smoothing

Before the result are described in much detail, a sharp point on what does each graph represent will be interpreted, starting from top left most graph is the (A) BoxClassifierLoss/classification loss which represent the l. oss of the classification of the detected object into a various class which in our cases is Rice, Mixed Vegetables, Fried Egg and Fried Chicken. Next to the right is the (B)BoxClassifierLoss/localization loss is the regression loss of the bounding box, practically the box's position. Moreover, to the right, the (C) RPNLoss/localization_loss is the localization loss or the loss of bounding box regressor for the RPN. Hence, the (D) RPNLoss/ objectness loss is the modal's ability to classify the object of interest and differentiate it with the background. While the (E) total loss is the sum of all loss on the modal and make an overall modal performance and lastly, the (F) CloneLoss is for when the system use multiple GPU for training where TensorFlow create a clone of the model to train on each GPU and report the loss of each clone thus it is ignored since the usage of single GPU only for training. Figure 4 shows the training where x plane represents the value of the said loss, and y plane represents the number of epochs gone through. The graph is jumping up and down, which makes it hard to track; thus, a smoothing function is needed so that it would be easy to identify roughly how the training goes.

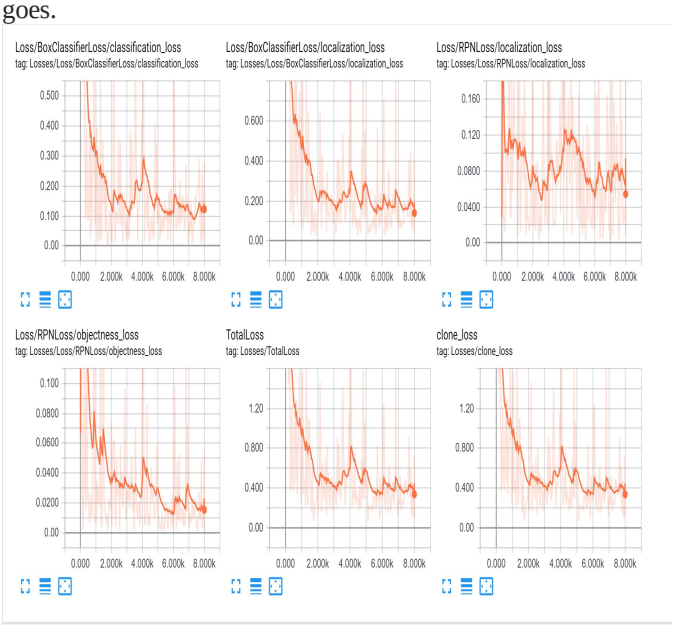

Figure 4: TensorBoard Loss Graph After Smoothing

After the smoothing process, it is clear and easy to identify how the graph patterns go, and the pattern that the classification loss graph is the capability of the modal to classify the object detected correctly. It has a new downward trend signifying a good learning process; however, it starts to have an upward trend after 2000 epochs and worsening at 3000 epochs. It signifies that the modal is most likely to become overfitting. The learning rate had to keep jumping up and down for the entire duration of the training, which might be due to the changing trend of the training graph shown previously in Figure 5.

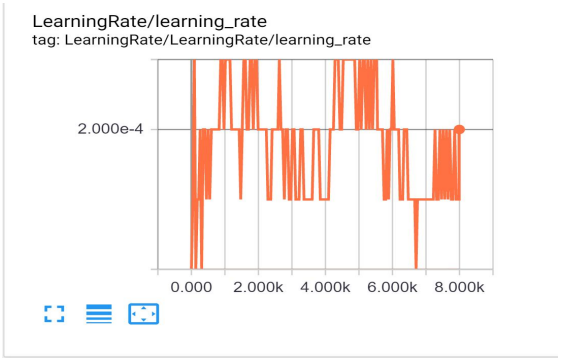

Figure 5: Jumping Learning Rate

Although the last position of TotalLost is at its lowest point of 0.4, the fact that it is an overfitted network makes it useless for real-world usage since it lost its generalization capability. However, an evaluation with a random food picture the modal unable to detect most of the class given and only manage to detect most of the case correctly was rice class might be due to its common characteristics of white in color and its contour

and shape and very much different from the rest of the class label.

4.2 Model II: SSD MobileNet V1

Another experiment made to compare the modal performance between the previous experiment and to ensure the best model take into consideration. This model provides performances and speed in mind. However, Version 1 picks due to its capability of running faster on GPU-based than in comparison Version 2 is faster on mobile devices and slower against its counterpart. This model is much faster in terms of its speed in detecting but at the price of accuracy.

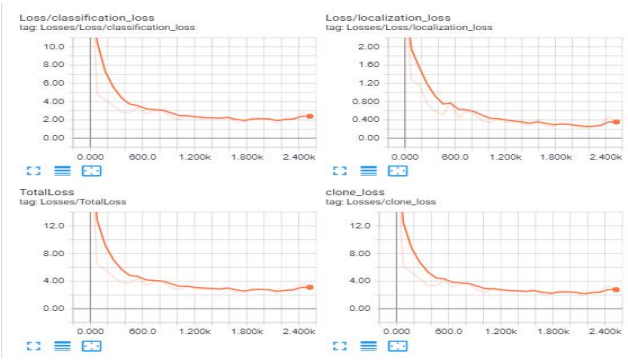

Figure 6: MobileNet Classification Loss

In terms of graph pattern, the graph starts high, and then it starts decreasing before it maintains its position after 18k epochs while barely touching Total Loss of 2. As a whole, the Loss graphs usually are going and have an acceptable downward pattern. However, its regularization loss keeps increase from the start of the training, but paying attention to the Y-axis of the graph, the difference is not that large as it is just a difference of 0.001 between them.

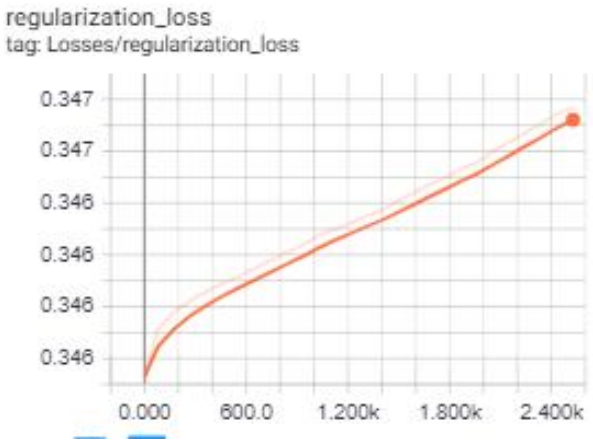

Figure 7: MobileNet Regularization Loss

Based on the graph, the total loss was more than 2. Thus, it signifies a quite bad accuracy of the modal. A physical evaluation made to ensure to check how well the modal work in working dataset and the result is as the picture below where the modal mistakenly identifies fried eggs on the rice might be due for the reason of the color of the rice which has been contaminated by yellow curry which almost similar in color to the yellowness of fried eggs.

4.3 Model III: Resnet50

Continuing to the 3rd model, which is Resnet50, the model uses default configurations and its weight set with pre-trained of 1000 different object categories used for ImageNet Classifications. The localization loss for the model was flawed as it became higher than its starting value. The classification loss was doing quite well, with less than 0.50 in terms of its value. It combined with its region proposal network of having less than $ 0.10 in its value.

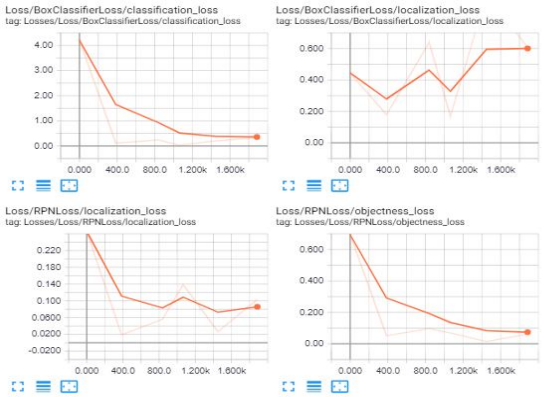

Figure 8: Faster RCNN Resnet50 Performances

Figure 9: Resnet50 Total Loss

4.4 Comparison Results

The comparison made between Perception V2, SSD MobileNet V1, and Resnet50 as tabulated in Table 1. It found that the best suitable for food image recognition is the Inception V2. The total loss and classification loss indicated excellent performance for this model. It is supported by the findings by Yu et al. (2016) were the Inception V2

outperformed ResNet and SSD MobileNet VI for food detections.

Table 1: Performance comparison of Inception V2, SSD MobileNet V1, and Resnet50

Model	Training Time (minutes)	TL	LL	EO	EO
I	38	0.401	0.101	0.069	12,770
II	53	3.10	0.354	2.400	2,528
II	10	1.000	0.09	0.130	7,512

Note:
Total Loss (TL)
Localization Loss (LL)
Classification Loss (CL)
Epoch before Overfitting (EO)

5. Conclusion

This paper presents CNN's performance for the detection and recognition of food images using real datasets from the selected restaurants. It is observed that Inception V2 demonstrates better achievements in comparison with SSD MobileNet V1, and Resnet50. The patterns discovered are deemed to the detection of items on the plate. These could be the initial step and a feasible approach for a restaurant to automate food prices. In the future, more images and consideration of more food items to enable us to see the capability of CNN.

ACKNOWLEDGEMENT

A special thanks to the Faculty of Computer Mathematical Sciences and Universiti Teknologi MARA, Malaysia for providing essential support and knowledge for the work.

REFERENCES

[1] The Telegraph. (2016, January 27). Britons 'lose patience after waiting five minutes to be served at a bar'. Retrieved from The Telegraph: https://www.telegraph.co.uk/news/newstopics/howaboutthat/11373010/Britons-lose-patience-after-waiting-five-minutes-to-be-served-at-a-bar.html

[2] Sivanandam, H. (2017, jun 1). The Star Online. Retrieved from thestar: https://www.thestar.com.my/news/nation/2017/06/01/sorry-no-roti-canai-thosai-restaurants-also-adopting-selfservice-due-to-shortage-of-workers/

[3] Lim, J. (2018). If you get overcharged for nasi campur, can you refuse to pay? Retrieved from Asklegal.my: https://asklegal.my/p/can-i-not-pay-overcharge-food-malaysia-nasi-campur

[4] Yusoff, M., Abdul Rahman, S., Mutalib, S., and Mohamed, A. (2012). Kohonen neural network performance in license plate number identification.

[5] Mezgec, S., & Koroušić Seljak, B. (2017). NutriNet: a deep learning food and drink image recognition system for dietary assessment. *Nutrients*, *9*(7), 657.

[6] McAllister, P., Zheng, H., Bond, R., and Moorhead, A. (2018). Combining deep residual neural network features with supervised machine learning algorithms to classify diverse food image datasets. *Computers in biology and medicine*, *95*, 217-233.

[7] Redzuan, F. I. M., and Yusoff, M. (2019). Knots timber detection and classification with C-Support Vector Machine. *Bulletin of Electrical Engineering and Informatics*, *8*(1), 246-252.

[8] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

[9] Kagaya, H., & Aizawa, K. (2015, September). Highly accurate food/non-food image classification based on a deep convolutional neural network. In *International conference on image analysis and processing* (pp. 350-357). Springer, Cham.

[10] Liu, C., Cao, Y., Luo, Y., Chen, G., Vokkarane, V., and Ma, Y. (2016, May). Deepfood: Deep learning-based food image recognition for computer-aided dietary assessment. In *International Conference on Smart Homes and Health Telematics* (pp. 37-48). Springer, Cham.

[11] Hassannejad, H., Matrella, G., Ciampolini, P., De Munari, I., Mordonini, M., and Cagnoni, S. (2016, October). Food image recognition using very deep convolutional networks. In *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management* (pp. 41-49).

[12] Singla, A., Yuan, L., and Ebrahimi, T. (2016, October). Food/non-food image classification and food categorization using pre-trained googlenet model. In *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management* (pp. 3-11).

[13] Islam, M. T., Karim Siddique, B. M. N., Rahman, S., and Jabid, T. (2018). Food Image Classification with Convolutional Neural Network. *2018 International Conference on Intelligent Informatics and Biomedical Sciences, ICIIBMS 2018*, *3*, 257–262.

[14] Şengür, A., Akbulut, Y., and Budak, Ü. (2019, September). Food Image Classification with Deep Features. In *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)* (pp. 1-6). IEEE.

[15] Pan, L., Qin, J., Chen, H., Xiang, X., Li, C., and Chen, R. (2019). Image augmentation-based food recognition with convolutional neural networks. *Comput. Mater. Continua*, *59*(1), 297-313.

[16] Yoshiyuki Kawano, K. Y. (2014). Automatic Expansion of a Food Image Dataset Leveraging Existing Categories with Domain Adaptation. Automatic Expansion of a Food

Image Dataset Leveraging Existing Categories with Domain Adaptation, 16.

[17] Kagaya, H., Aizawa, K., and Ogawa, M. (2014). *Food Detection and Recognition Using Convolutional Neural Network*. (3), 1085–1088. https://doi.org/10.1145/2647868.2654970

[18] Liu, Y. H. (2018). Feature Extraction and Image Recognition with Convolutional Neural Networks. *Journal of Physics: Conference Series*, *1087*(6). https://doi.org/10.1088/1742-6596/1087/6/062032

[19] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

[20] Yu, Q., Mao, D., and Wang, J. (2016). *Deep Learning Based Food Recognition*.

[21] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2015). *Rethinking the Inception Architecture for Computer Vision*. Retrieved from http://arxiv.org/abs/1512.00567