# Image Recognition Approach for Expediting Chinese Cafeteria Checkout Process

Bo-Ting Wu
*Department of Innovative Information and Technology*
*Tamkang University*
New Taipei City, Taiwan
andywu0913@gms.tku.edu.tw

Ya-Wei Tsou
*Department of Innovative Information and Technology*
*Tamkang University*
New Taipei City, Taiwan
winnie22935@gmail.com

Emerson Tan
*Department of Innovative Information and Technology*
*Tamkang University*
New Taipei City, Taiwan
etan4067@gmail.com

Feng-Cheng Chang
*Department of Innovative Information and Technology*
*Tamkang University*
New Taipei City, Taiwan
135170@mail.tku.edu.tw

***Abstract*—One of the common running themes in modern-day Chinese cafeterias is the hold up in foot traffic in queueing due to checkout. We find out that this bottleneck is caused by the staff requiring extra time to look up the prices of those miscellaneous entrees and calculating the total due amount during checkout. In this paper, this issue is addressed by introducing real-time image recognition techniques into this process. By using a webcam taking live video feed at the checkout desk with the image recognition model outputs the total due amount simultaneously, we are able to eliminate the need to perform manual price calculations. Additionally, the nutrition facts of the meal can also be calculated and displayed to the customers based on the detected entrees. The image recognition model is based on YOLOv3 with 575 entree-catered plate images involved in model training, validation, and testing. The transfer learning technique is also incorporated to speed up the training process. Experimental results show that the recognition accuracy of individual entree is around 70% and that of the entire plate is roughly 63%. With the advanced training with a larger dataset, we believe that the accuracy can be increased, and applying the approach during the checkout will become more practicable.***

***Keywords*—food recognition, automatic price calculation, nutrition facts calculation, object detection, YOLOv3, image recognition, machine learning, transfer learning**

## I. Introduction

Chinese cafeterias offer a variety of selection of entrees during mealtimes. However, with so many selections to choose from comes the downside of having longer queues brought by the checkout staff that takes extra time to look up the prices of those miscellaneous entrees and calculates the total due amount during checkout. The typical process is to identify the entrees on a plate; to look up the price of each entree; to calculate the total price of them. The first step requires identification of food patterns on a plate, thus, manual process is conventionally involved. During rush hours, it limits the processing speed and causes a long waiting queue.

With the advances in intelligent image recognition techniques, perceptual recognition has been improved to a practical implementable level. T. Joutou and K. Yanai proposed the MKL (Multiple Kernel Learning) method [1] to perform classification on food images, achieving a 61.34% classification rate for 50 kinds of foods. H. Kagaya et al. applied CNN (Convolutional Neural Network) to detect and recognize food images and concluded that the performance [2] is much better than traditional methods. These outcomes show that it is feasible to apply image recognition to food pictures.

In this paper, we propose an image recognition approach by implementing a real-time object detection model during the checkout desk to assist the checkout process. The application also provides a supplementary function that displays the overall nutrition facts of the plate for the customers who wish to keep track of their diet. By fusing objection detection into the checkout process, we expect expediting the Chinese cafeteria checkout process is achievable.

This paper is organized as follows: the object detection model used for expediting the checkout queue is elaborated in Sec. II. The experimental results are collected and discussed in Sec. III. Finally, we conclude the work and the future direction in Sec IV.

## II. Processing Architecture

The target prototyping application contains a webcam arranged above the checkout desk, pointing downwards to take video feeds of placed meals from overhead. The video feed will then be delivered as an input of the well-trained object detection model that intelligently detects each corresponding entree on the plate. Eventually, the total due amount and nutrition facts for that plate will be automatically calculated and displayed.

In the following subsections, we expound the details of constructing the entree detection model as well as the price and nutrition facts calculation.

### *A. Entree Detection*

Entree detection model is the core in the approach, which a live video feed is incessantly input to the model and the model is expected to recognize all the entrees shown on the plate. To establish an entree detection model, there have been some prominent object detection model structures as the design references. Below, we compared and adopted one of them as our entree detection model structure, collected and pre-processed the entree images, and lastly trained the model to learn these entrees.

#### *1) The object detection model*

To be a practical functional unit at the cafeteria checkout desk, the recognition accuracy and inference speed of the object detection model should be balanced. Three kinds of object detection models (Faster R-CNN [3], SSD [4], and YOLO [5]) were compared and literature reviewed by us since

they are the three renowned models currently. When it comes to accuracy, Faster R-CNN performs the best, followed by SSD and YOLO close behind [6]. However, the inference speed of YOLO outperforms Faster R-CNN and SSD greatly. Therefore, in order to perform object detection on non-GPU devices with a reasonable inference speed, the third version of YOLO, abbreviated as YOLOv3, was chosen to be our model structure.

Almost all object detection models consist of a base network and a detection network. The purpose of the base network is to extract the features from a given input, while the detection network is to produce the prediction output based on the extracted features from the base network. Darknet-53 is designed as the base network in YOLOv3 with its advantage of solving vanishing gradient problem [7] during back-propagation. As for the detection network in YOLOv3, a feature pyramid network [8] is adopted to improve the prediction efficiency of small objects.

*2) Prepare data for training*

After the object detection model was decided, the next step was to prepare some data for training the model. Images with different kinds of entrees catered on a plate were the data in our case. Due to development time constraints and for experimental purposes, we limited the kinds of entrees down to twelve most popular ones in the school restaurant and made categories out of them (Fig. 1).

The images were collected by routinely taking sample pictures of the plate of food from the aforementioned twelve categories. We took pictures from the plate's four sides and the top whenever we visited the student cafeteria. In the effort to simulate different lighting conditions for the dataset, we also took pictures with differing levels of brightness and flashlight settings.

Data pre-processing took place after the data were collected. In this phase, all the collected data were manually pre-processed by locating and labeling the displayed entrees. The coordinate information x-min, y-min, x-max, y-max from the bounding box and the category of the entree were recorded in the parameter file. The pre-processed information was used in back-propagation during the training.

In the last step of data preparation, the data were split into three piles: the training dataset, validation dataset, and testing dataset. The training dataset was used for training purpose that allowed the model to learn the patterns of entrees from this pile; the validation dataset was used for examining whether the model was well-trained or should proceed further training after each epoch in training; the testing dataset was used as a blind testing which tested the accuracy of a model after it was believed well-trained verified by the validation dataset.

*3) Training the model*

"To stand on the shoulders of giants," the transfer learning technique was applied to the training instead of training from scratch. This took a well-trained YOLOv3 model which was capable of recognizing objects in some other dataset and used it as a starting point to train on recognizing our entrees. The benefit of it is that the training process can be sped up manifolds as the well-trained filters in the convolutional layers in the model are reused.

By transfer learning from others well-trained YOLOv3 model, the three output layers from the model were rebuilt with the exclusive shape of those twelve categories, so that the

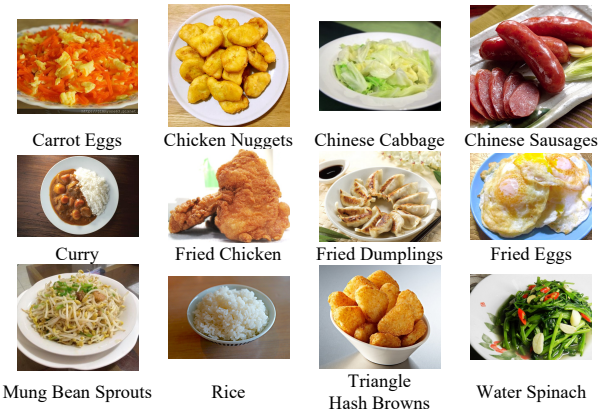


Fig. 1. The twelve categories (Chinese entrees) in the dataset.

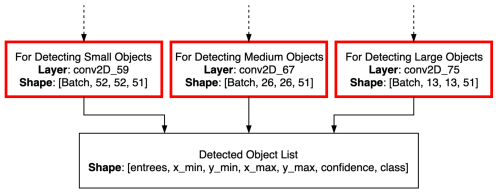


Fig. 2. The three rebuilt output layers.

output shapes of the modified model will fit with the twelve entrees. The adjusted three output shapes, in the format 'Shape: [Batch, x, y, z]', are shown in Fig. 2. The parameters x and y, noted by 52, 26 and 13, represent the number of grids contained in the layers respectively. The first output layer contains the most amount of grids at 52x52, the second contains the half that at 26x26, and the third output layer contains half of the second at 13x13 grids. This means that given the number of grids in each layer, the coverage of each grid per layer becomes increasingly larger. In other words, this means that the first layer has the least amount of coverage in the grids which is ideal and meant for detecting smaller objects whereas the third layer with the most amount of coverage is ideal and meant for detecting larger objects.

To explain the z parameter, note that in each batch, each x and y contains 3 bounding boxes with 17 values, this being 3x17 in short explaining the number 51. Bounding boxes are the boxes encompassing the recognized objects drawn by the model. There are 5 out of 17 values representing the coordinate information x-min, y-min, x-max, y-max from the bounding box and the confidence for detecting an object. The remaining 12 values indicate the confidence of the object belonging to the 12 categories respectively.

The entire training process was carried out by first splitting the training dataset into 5 piles with 100 images per pile, next feeding the first pile in smaller batches into the model for training. The training was set to terminate itself once the latest five validation loss rate starts rising. After the training being terminated, the rest of the piles were incrementally added to the previous pile used in training. The reason for this method is because we were still incessantly collecting new data while the experiment was ongoing. This process was repeated until all five piles had been trained by the model.

### B. *Price And Nutrition Facts Calculation*

Prices and nutrition facts can be calculated after the model has been well-trained and able to detect entrees. For the price calculation, simply sum up all the prices of the detected entrees assuming all the detected entrees come with per serving. For nutrition facts calculation, the fat, carbohydrates, protein and calories from each entree are pre-calculated and summed up based on the detected entrees.

There are two steps to pre-calculate the nutrition facts of each entree. First, look up the recipe and obtain the nutrition facts of a particular weight or piece of an entree. Next, measure the average weight or pieces per serving of that entree catered from student restaurant. Finally, the total amount of nutrition facts of the plate can be obtained by multiplying them together.

## III. Results and Discussions

In our experiment, a total of 575 images were taken in our daily meals. These images were pre-processed and divided into three piles: 500 images for training purpose, 40 images for validating whether the model was well-trained, and the rest of 35 images were for blind testing after the model was believed well-trained.

During the training process, 'loss' was used as an indicator for measuring the detection failure rate of the model. It should continue descending until it started to plateau, which meant the learning of the model was saturated. The training was considered progressing while the training loss and the validation loss were both decreasing. Finally, the training terminated itself after 97 epochs and the model was considered well-trained with the losses both seem not decreasing anymore (Fig. 3).

To evaluate the accuracy of the model after training, the mean average precision (mAP) was used in evaluating the blind testing over the testing dataset. The indicator mAP is calculated by averaging the precision of each category with the range from 0 to 1. The higher the value is, the more extraordinary the model is in recognizing the entrees from the images that it has never seen before. Table I shows the average precision with each category calculated from the testing dataset, and the mAP is approximately 70%, which is not bad.

TABLE I
MAP CALCULATED FROM THE TESTING DATASET

| Category | Average Precision |
|---|---|
| Carrot Eggs | 0.820 |
| Chicken Nuggets | 0.200 |
| Chinese Cabbage | 0.285 |
| Chinese Sausages | 0.750 |
| Curry | 0.800 |
| Fried Chicken | 0.902 |
| Fried Dumplings | 0.696 |
| Fried Egg | 0.905 |
| Mung Bean Sprouts | 0.722 |
| Rice | 0.735 |
| Triangle Hash Browns | 0.551 |
| Water Spinach | 1.000 |
| mAP | 0.697 |

We also evaluated the model by the overall correctness of detecting every entree on a plate. The reason for this is to simulate the circumstances in the actual checkout process that whenever any entree is falsely identified, the total due amount

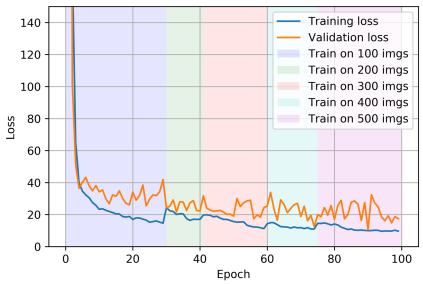


Fig. 3. The losses during training.

Fig. 4. Entrees detected on the test dataset.

of the meal and nutrition facts are undoubtedly being calculated incorrectly. There are 35 images in the testing dataset shown in Fig. 4, with 22 plates being correctly detected and 13 plates did not. The correctness of overall detection on an entire plate is approximately 63% (22/35). Looking into details of these falsely detected plates, most of them are lacking one or two entrees undetected rather than entrees being misclassified. Therefore, we believe that if we gather more and more data into the training, along with setting a higher number of the pile of images to be added incrementally to the training (e.g. from 100 images currently to 200 images), this issue can be resolved.

After having the model well-trained and price and nutrition facts calculation prepared, we performed several trial runs with our laptop and webcam connected at the student restaurant, placing a freshly catered food plate underneath. Fig. 5 demonstrates a screenshot of live video feed taken by the webcam being fed to the object detection program in real-time. Fig. 6 shows the prices and the nutrition facts of each detected entree displayed. With our approach, customers can easily learn the total due amount as well as how many fat, carbohydrates, protein and calories they will ingest.

In the first few experimental training, Chinese sausages were labeled as their individual pieces, thus result in an interesting outcome that the model drew bounding boxes around

Fig. 5. The setup of the application simulating the checkout scenario.

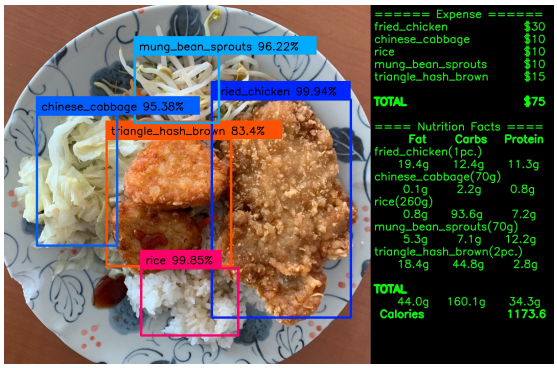


Fig. 6. A screenshot of showing prices and nutrition facts.

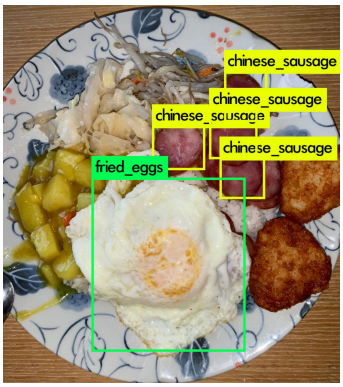


Fig. 7. Chinese sausages are predicted individually.

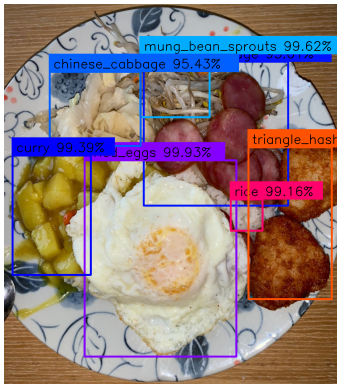


Fig. 8. Chinese sausages are predicted as per serving.

the single pieces even though they were adjacent or overlapped to each other (Fig. 7). As we already assume all the entrees on the plate are catered per serving, drawing one bounding box around a pile of Chinese sausages is more preferred. For this reason, we changed the way how entrees were pre-processed by re-labeling all the sausage piles in the dataset with a singular bounding box respectively. So they will all be labeled together as long as they are adjoint. The testing outcomes were as expected with the model only drew one bounding box around a serving of Chinese sausages (Fig. 8).

## IV. Conclusion

In this paper, we proposed an image recognition approach by implementing a real-time object detection model based in YOLOv3. The application is deployed at a Chinese cafeteria checkout desk to assist the checkout process, along with a supplementary function that displays the overall nutrition facts of the plate. It looks promising by some preliminary testing with the trained model. In the experiment, the mAP with the testing dataset is 69.7%, which means there are about 70% of the entrees are detected correctly. To further assess the feasibility of the approach, we also evaluated the overall correctness of detecting every entree on a plate. Results show that 63% of the catered plates from the testing dataset can be successfully recognized. With the advanced training with a larger dataset, we believe that the accuracy can be increased, and the approach will be more practicable. At that time, checkout duties would be prompter and more assured, and even can be passed off to minor staff members. This has the benefit of giving senior staff members the time to focus on actual kitchen-related tasks, which in turn will increase overall productivity.

The object detection model used in the paper is based on the YOLOv3 structure. We are also exploring alternative approaches to recognize entrees on a plate faster and more accurately. Currently, collecting more data of those lower precision categories into training is the most direct way to enhance the accuracy of the model to a real-world applicable level.

## References


[1] T. Joutou and K. Yanai, "A food image recognition system with multiple kernel learning," in *Proc. Int. Conf. Image Process.*, pp. 285-288, 2009.

[2] H. Kagaya, K. Aizawa, and M. Ogawa, "Food detection and recognition using convolutional neural network," in *Proc. ACM Int. Conf. Multimedia*, pp. 1085-1088, 2014.

[3] S. Ren et al., "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, pp. 91-99, 2015.

[4] W. Liu et al., "SSD: Single shot multibox detector," in *European conference on computer vision*, pp. 21-37, 2016.

[5] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv 2018 arXiv:1804.02767.

[6] J. Hui, *Object detection: speed and accuracy comparison (Faster R-CNN, R-FCN, SSD, FPN, RetinaNet and YOLOv3)*, Mar. 28, 2018. Accessed on: Feb. 1, 2020. [online] Available: https://medium.com/@jonathan_hui/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359

[7] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertainity Fuzziness Knowl.-Based Syst.*, vol. 6, no. 2, pp. 107-116, Apr. 1998.

[8] T.-Y. Lin et al., "Feature Pyramid Networks for Object Detection," in *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2117-2125.