ISSN 2278-3091
Volume 9, No.4, July – August 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse308942020.pdf
https://doi.org/10.30534/ijatcse/2020/308942020

Improved Deep Learning Framework For Multi Food Instance Segmentations

Suhaila Farhan Ahmad Abuowaida[1], Huah Yong Chan[2]
[1,2]School Of Computer Sciences, Universiti Sains Malaysia,11800, Pulau Pinang, Malaysia
suhilaowida@student.usm.my, hychan@ usm.my

ABSTRACT

This paper presents a novel framework to develop the multi food instance segmentation framework. The proposed framework consists of four steps. Firstly, multi food image enhancement using image processing to handle different sizes of the image. Secondly, improve the recognition algorithm through the proposed novel backbone to extract low and high levels of features. Thirdly, locate the item of food using Region Proposal Network (RPN). Finally, develop a new technique called Max RoI layer, to get the best boundaries for the multi food image in the instance segmentation. The proposed method is validated by an experimental study on creating dataset has 27000 images were collected and annotated to train this deep learning (DL)-based algorithm to identify the multi foods in images, and the results are compared with the latest algorithm in object instance segmentation, they are Mask R-CNN, YOLACT, and CASCADE R-CNN. The proposed framework achieves better performance about the accuracy AP with different thresholds (AP 50 , AP 75 , AP 90), which are all higher than those of three frequently used methods—the Mask R-CNN, YOLACT, and CASCADE R-CNN. The training time proposed method is also slightly lower than the other methods.

Key words: Deep learning, Multi food, Instance segmentation

1. INTRODUCTION

Food recognition based on a single image is a crucial and challenging issue in computer vision, where the process of estimating food calories and dietary assessment is mainly dependent on food recognition. Furthermore, food recognition results in various issues in the recognition system, including the diverse types of food with the same shape and appearance, leading to the difficulty in distinguishing the food through its images. Moreover, some types of item food consist of various shapes, colours, and sizes, which increases the challenge in classifying food type. Provided that the multi food image normally comprises more than one food item, separating the irregular shaped food is a complex process, especially when an occlusion is present in the multi food image. Additionally, the varying amount of food in the image is an important factor influencing the method of food recognition, which is implemented in the development of multi food recognition. Therefore, developing multi food recognition is a complex task. Although various algorithms suggested specific methods to solve food recognition, most of the studies identified multi food based on a single image with one food item [1]. As a result, the management of images with two or more items, such as chicken and rice images, was challenging for these algorithms. Segmentation methods were implemented in food recognition, which managed more than one food item in one image [2][3][4][5]. In this case, the researcher of [4] suggested the deformable part model (DPM) and circle detector handle with multiple items of food, while the researcher of [2] suggested the segmentation of multiple items of food using local variation algorithms. Following the rapid development and wide use of smartphones, which featured a powerful processor, high-resolution cameras, and large memory capacity, the applications for rapid development for food recognition were proposed[3][5]. However, these applications required the user to locate the items of food for recognition and segmentation. In the recent years, the Convolutional Neural Network (CNN) has received significant attention for its ability to handle computer vision applications, such as recognition[6][7][8][9][10], object detection [11][12][13], and segmentation [14][15][16] [17][18]. Therefore, several algorithms were proposed for food recognition based on CNN. Specifically, the authors in [19] applied CNN for the classification of image food, which used SVM in fully connected layers. As a result, it was found that CNN on food detection and recognition was better compared to the traditional methods. The research in [20] utilised the architecture of faster R-CNN to localise various items of food for the detection of multi food, while the authors in [2] proposed the encoder and decoder architecture using CNN and grab cut for the classification and segmentation of food items. In this case, although the studies focused on the classification and segmentation of food items, the sizes of the food items were not quantified due to the challenges of this task. This article aims to build a framework for food instance segmentation to solve the aforementioned issues. Therefore, a novel framework known as multi-food instance segmentation was proposed, which consists of four steps. The first step is the pre-processing image, which assists in reducing the time consumed for the multi food instance segmentation process. The second step is the novel backbone architecture, which extracts the feature maps of food item with more accuracy, and less time and power as shown in the experimental result. The third step is the adoption of Region Proposal Network (RPN) for the localisation of multiple items of food, followed

by the fourth step, which is the improvement in Fully Convolution Network (FCN) to produce instance segmentation for the prevention from the overlapping problem between items food. This prevention is made through a new technique known as Max RoI to manage different sizes of the feature map generated from RPN. Provided that this study aims towards multi food instance segmentation, the dataset of multi food was discussed. The image dataset was created using images consisting of multi food annotated using VGG Image Annotator (VIA) [21].

Two measurements were involved in the evaluation of the multi food instance segmentation framework, namely the average over union section over IoU (AP) with different thresholds and time for training and testing. These measurements were then compared to Mask R-CNN [16], YOLACT [17], and CASCADE R-CNN [18] algorithms. The factors of the different results obtained from these four algorithms were identified. Summary of the results is presented at the end of this article.

1.1 The dataset

The instance segmentation tasks based on deep learning required a large number of datasets and computing power. The computing power issue was solved through the use of Graphics Processing Units (GPUs). However, there are a few datasets that contain well-annotated open source, for instance, segmentation especially multi food instance segmentation. Therefore, it was created dataset containing multi food to achieve multi food instance segmentation. In addition, COCO Dataset [25] containing food was used in the training process. In the process of creating a database, the reasons that affect the accuracy of the results were taken, therefore, the focus was on taking the image of multi food with high accuracy through using iPhone 5 camera (8-megapixel iSight camera, panorama, autofocus and LED flash). The captured image is that the stored have a different resolution so, in this paper adaptation image processing algorithm to handle with a difference in resolution as we will see later. then, the annotated each image in the dataset using VGG Image Annotator (VIA) [21] through drawing a polygon when you get all the points of the polygon. The annotation file of our dataset has the item of food information such as width, length, and other information. Finally, the annotation file was converted into a Microsoft COCO dataset format [25] through the Python program. Then choose the item of food in our database for this paper was not randomly selected. So images that selected contain many challenges such as select apple, tomato, and carrot with different size, selected the lemon and banana that have the same features in colors and shape, select grape and apple with different color, the apple, orange, and lemon come with the same shape and the kiwi and potato have the same porosity, therefore, the detection and instance segmentation of multi food is a harder task. The multi food dataset consists of 27 classes (Banana, Onion, Grape, Pear, Rice, Lemon, Pringles, Tomato, Potato, Cucumber, Roasted Chicken Breast, Apple, Bread, Carrot, Egg, Orange, Cantaloupe, Peach, Plum, Kiwi, Cake, Hotdog, Pizza, Donut, Fig, broccoli, Sandwich). Therefore, the multiple food dataset consists of 27000 multi food images. submission.

2. METHODOLOGY

This article presents notable insights into the multi food instance segmentation through the identification of multi food items tasks, and localisation and segmentation of each food item based on the input image task are challenging tasks due to several challenges, including the multiplicity and the aforementioned difference between the colour and sizes of the food items. Moreover, the novel framework in this study was proposed through the improvement in many methods, including prepossessing, CNN network, RPN, and segmentation algorithms (refer to Figure 1) to obtain the most substantial result through an accurate presentation of the multi food image.

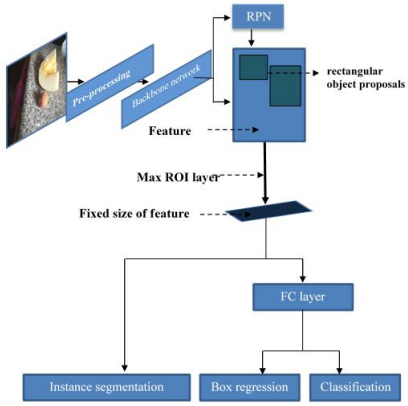


Figure 1: Multi food instance segmentation framework.

The overall loss function of the multi food instance segmentation for each food item was represented by the following formula: $L = L_{classification} + L_{localization} + L_{segmentation}$ (1)

Where, the classification and localisation loss function was according to the formula below:

$$L(p_i.t_i) = \frac{1}{N}\sum_{i=1} |p_i^* - p_i|^2 + \lambda\frac{1}{N}\sum_i p^* L_{loc}(t_i.t_i^*) \quad (2)$$

where p_i is predicted probability of anchor i, p_i^* is ground truth of anchor i, t_i is coordinates predicted, t_i^* is coordinates ground truth, N is a normalization term and = a balancing parameter. The loss function of the instance segmentation was identified using the per-pixel sigmoid and average binary cross-entropy based on the formula in (3) to generate boundaries for each class. As a result, the loss function was found in the input image, and it could be represented through the following formula:

$$L_{segmentation} = -\frac{1}{s^2}\sum_{1\le i,j\le s}\left[y_{ij}\log y_{ij}^k + (1 - y_{ij}\log(1 - y_{ij}^k)\right] (3)$$

Where y_{ij} is the ground truth of boundaries of size region (m^2), y_{ij}^k, k is ground truth class and the predicted value of boundaries.

2.1 Pre-processing

In this section, it is shown that the food image size was changed to 1028 x 1028 pixels, to obtain the most information from the food image. Any image sizes were acceptable in this study through two options, namely:

1. If the image is smaller than the required size, padding techniques are applied.
2. If the image is larger than the required size, the Lanczos resampling [22] is applied for the down-sampling of the input image while preserving the original image feature. Notably, provided that time consumption was an important aspect in this study, JPEG compression was applied to reduce the image size from 4.38 MB and 2.49 MB to (0.196MB and 0.0385MB, respectively. In this case, the dataset was reduced from 2.63 GB to 116 MB to decrease the time consumed in JPEG compression.

2.2. Backbone network

For the use of ResNet as a backbone of this study framework, an improvement was made on it and the backpropagation of ResNet was analysed. Following that, new rules of different parameters of the ResNet were acquired based on the new gradient formula. The layer requiring further training was trained, while the frequency of the training was reduced for some layer, which did not require training based on the formula presented in this section to solve the issues of the vanishing gradient.

2.2.1. Simplify resent

In this section, an analysis of ResNet was performed, followed by the proposal of the network. A novel architecture was proposed by determining a better training for each layer to improve the performance of ResNet and identify a more suitable filter size compared to ResNet [10] to extract high and low levels of feature from the input image. The performance of ResNet was achieved using the deep network, which consisted of a set of blocks to solve the issues of vanishing gradient [10]. The process of establishing the two-layer block is presented in Figure 2.

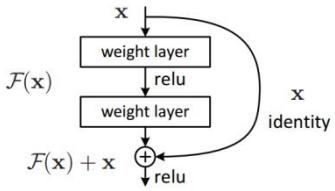


Figure 2: Resnet block[10].

The formula of the established block of two layers can be defined based on the following formula: $H(x) = F(x.\{W_i\}) + x$, (4). Where x is the input of the building block, H(x) is the output vectors of the building block and F(x,W $_i$) is the residual mapping that learned in training process. The ResNet prevented the issues of vanishing gradient through the identity shortcut, which connected different dimensions. However, the explanation on identity shortcut in [10] was not accurate although one of the main issues of ResNet was the neglect of the activation layer in the backpropagation process. To illustrate, the formula describing the changing process in the parameter of ResNet was absent, leading to the low accuracy in the gradient formula. Besides, the formula used in the ResNet did not specify which layers in the training process need more training than another layer.

2.2.2. Analysis of the Backpropagation

The identity shortcut connections were the keys to solving the issues of vanishing gradient through gradient formula [10]. However, formulating a direct inference of the backpropagation of ResNet through gradient formula was highly challenging. Therefore, the ResNet was analysed into a backpropagation network to elaborate on the aforementioned issues. The total loss function in ResNet was the square of the difference between the predicted output and the ground truth, as shown in the following formula:

$$L = \frac{1}{N}\sum_{i=1}^{n} |\hat{y}_i - y_i|^2 = \frac{1}{N}\sum_{i=1}^{n} |e_i|^2, \quad (5)$$

where c is the number of classifications, N the is normalization term, y_i is the ground truth value and $\hat{y}_i$ is the predicted value. The ResNet in the forward propagation refers to a set of the first input layer $[x_1.x_2.x_m]$. Following that, $[s_1^1, s_2^1, .., s_n^1]$ is the first hidden layer connected by weight wi1, w_i^1 where s $_i$= $\theta(s_i)$ through activation function $\theta(.)$ in the first hidden layer. The number of the hidden layer (conv) is represented as follows:

$$s_i^{conv} = \sum_{i=1}^{n} \theta(s_i^{conv-1}) \cdot \omega_i^{conv} + \theta(s_i^{conv-1}) \quad (6)$$

The predicted output of ResNet was obtained from the last layer, and the backpropagation in ResNet was represented through the following formula: $\frac{\partial L}{\partial \omega_i^5} = \frac{\partial L}{\partial s_i^5} \cdot \frac{\partial s_i^5}{\partial \omega_i^5} = \delta_i^5 \cdot \frac{\partial(\sum_{i=1}^{n} \theta(s_i^4)\cdot\omega_i^5)}{\partial \omega_i^5} = \delta_i^5 \cdot \theta(s_i^4)$ (7)

δ_i^5 was defined as follows:

$$\delta_i^5 = \frac{\partial L}{\partial s_i^5} = \frac{\partial \frac{1}{N}\sum_{i=1}^{n} |\hat{y}_i - y_i|^2}{\partial s_i^5} = \frac{2}{N} \cdot |\hat{y}_i - y_i| \cdot \frac{\partial \hat{y}_i}{\partial s_i^5} = \frac{2}{N} \cdot e_i \cdot \theta'(s_i^5) \quad (8)$$

Where, $\hat{y}_i = \theta(s_i^5)$, e $_i$ refers to the standard deviation between the predicted output and the actual output of the last layer. This was followed by the calculation of the gradient of L for weight w $_i^4$ based on the following formula:

$$\frac{\partial L}{\partial \omega_i^4} = \frac{\partial L}{\partial s_i^4} \cdot \frac{\partial s_i^4}{\partial \omega_i^4} = \delta_i^4 \cdot \frac{\partial(\sum_{i=1}^{n} \theta(s_i^3)\cdot\omega_i^4 + \theta(s_i^3))}{\partial \omega_i^4} = \delta_i^4 \cdot \theta(s_i^3) \quad (9)$$

where s_i^4 have two parts, namely $s_i^4 = \sum_{i=1}^{n} \theta(s_i^3) \cdot \omega_i^4$ as the stander part and $\theta(s_i^3)$, which was incorporated through the identity of the shortcut connections. Moreover, the Î´i4 was represented through the following formula:

The δ_i^4 represented as following formul $\delta_i^4 = \frac{\partial L}{\partial s_i^4} = \frac{\partial L}{\partial s_i^5} \cdot \frac{\partial s_i^5}{\partial s_i^4} = \theta'(s_i^4)\sum_{i=1}^{n} \delta_i^5 \omega_{ij}^5$(10) Then, the gradient of L for the w_i^3 was calculated based on the following formula:

$$\frac{\partial L}{\partial \omega_i^3} = \delta_i^3 \cdot \theta(s_i^2) \quad (11)$$

The following formula was used for the representation of s_i^3 :

$$\delta_i^3 = \theta'(s_i^3)\sum_{i=1}^{n}\delta_i^4(\omega_i^4 + 1) \quad (12)$$

Meanwhile, the remaining hidden layer in ResNet incorporated the same gradient formula, which is as follows, including the formula in (14):$\frac{\partial L}{\partial \omega_i^{conv}} = \delta_i^{conv} \cdot \theta(s_i^{conv-1})$ (13)

and

$$\delta_i^{conv} = \theta'(s_i^{conv})\sum_{i=1}^{n}\delta_i^{conv+1}(\omega_i^{conv+1} + 1) \quad (14)$$

Where, conv = 2,.4. Finally, the gradient of L for w_i^1 was calculated based on the following formula in (15) and (16):

$$\frac{\partial L}{\partial \omega_i^1} = \delta_i^1 \cdot x_i \quad (15) \text{ and}$$

$$\delta_i^1 = \theta'(s_i^1)\sum_{i=1}^{n}\delta_i^2(\omega_i^2 + 1) \quad (16)$$

Therefore, the gradient of ResNet (5 layers) connection weight can be represented as the following formula: $\frac{\partial L}{\partial \omega_i^{conv}} = \delta_i^{conv} \cdot \theta(s_i^{conv-1}), \quad conv = 1..5$ (17)

The gradient of the first connected weight was fading with the increase in the number of layers in the network. The use of identity shortcut connection was used in ResNet solved the issues of vanishing gradient. The $\Delta\delta_i^{conv}$.in refers to the gradient increase in the number of layer (conv) in ResNet and solves the aforementioned issue in a deep network. It is represented by the following formula:

$$\Delta\delta_i^{conv} = \theta'(s_i^{conv})\sum_{i=1}^{n}\delta_i^{conv+1}, \quad conv = 1..4 \quad (18)$$

The $\Delta\delta_i^{conv}$ solve the vanishing problem in deep network.

2.2.3. Improvement in the Resnet

Although the main importance of the shortcut connections was to solve the issues of the vanishing gradient, several issues persisted in ResNet, including the inadequate reinforcement in the layers. However, a large filter size was selected in the first convolution when the reinforcement received by the layer was more than its required amount. In this article, specific strategies were proposed to solve the issue of insufficient training and provide an optimal filter size, as shown in Figure3:

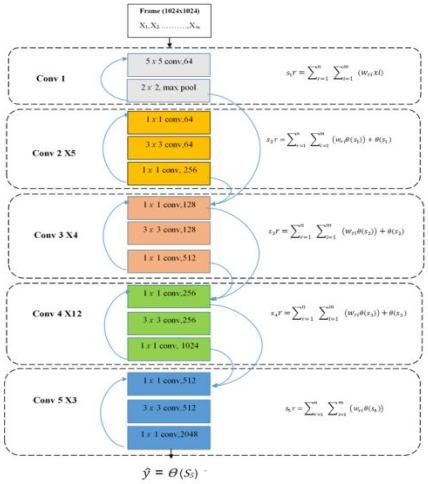


Figure 3: Improvement resnet.

The repeated formula was combined with the forwarded ResNet formula, which is as follows:

$$y = \sum_{r=1}^{n} F(x, \{W_{i,r}\}) + x, \quad (19)$$

Where, r refers to the number of repetitions for each convolution block of ResNet. For the layer with inadequate training, the r should be increased, while the exceeding number of training should be reduced. The backpropagation of the improvement in ResNet was based on the (20), while the size of the selected filter was smaller than ResNet until feature extraction and the reduction in feature size and parameters. As a result, computation efficiency was enhanced.

$$\frac{\partial L}{\partial \omega_i^{conv}} = \delta_i^{conv} \cdot \theta(s_i^{conv-1}) \quad (20)$$

3. LOCALISATION

Provided that the focus of this article included multi food, RPN was adopted in the effectiveness of the multi food to determine the position of multiple objects in the input image [13]. Furthermore, the RPN accepted any sizes of the feature map, which functioned as the output. Meanwhile, the proposed CNN functioned as the input to generate several rectangular object proposals, as shown in Figure 1.

The object was present in the rectangular object proposals, while the sliding window was provided in all the feature maps obtained from the last convolution layer of the proposed CNN. Each sliding window consisted of nine anchors, which were the central points of the sliding window. Notably, provided that the sliding window was different in terms of Aspect Ratio (AR) and Scale (S), the coordinate for each anchor was calculated based on the input image. As a result, the value of $p*$ for each anchor was calculated based on two factors:

1. The anchors with the highest intersection-over-union overlap and a ground truth box.
2. The Overlap Intersection-Over-Union (IoU) for each anchor, which was higher than 0.7.

The IoU could be represented through the following formula:

$$IoU = \frac{Anchor \cap groundtruthbox}{Anchor \cup groundtruthbox} \quad (21)$$

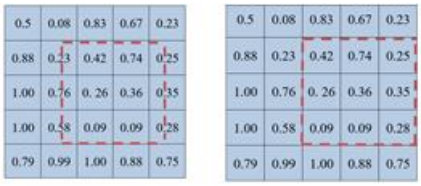
3.1. Max region of interest pooling (max roi)

Several rectangular object proposals were generated from RPN on feature maps, as shown in Figure 1. As a result, different features of map size were produced, leading to an impact on the instance segmentation accuracy. Accordingly, this article proposed a novel technique to manage the diverse sizes of the feature map. Known as Max RoI, the feature map was reduced into a fixed size through the following two stages.

1. The first stage was maintaining the location of feature maps by avoiding the quantisation (refer to Figure 5(a)), which was applied through the RoI Pool [13] for each RoI boundary (refer to Figure 5(b)). However, the poor result was observed in the instance segmentation of the RoI Pool [16] due to the high rate of quantisation (refer to Figure 5(b) and 6(b)). Provided that each pixel value is important and effective in obtaining the optimum results in instance segmentation, this study solved the issue of misalignment was

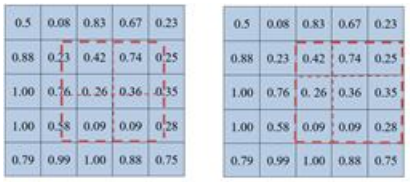
solved by voiding the quantisation.

2. The second stage applied max pooling for each bin (refer to Figure 6).



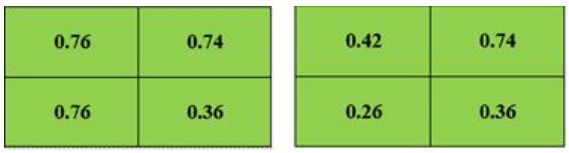
(a) (b)

Figure 4: (a). The avoidance of quantisation by max roi (b). The roipool after the application of quantisation.



(a) (b)

Figure 5: (a). The second avoidance of quantisation by Max RoI (b). The RoIPool after the second application of quantisation.



(a) (b)

Figure 6: (a). The result of Max RoI (b). The result of RoIPool.

3.2. Fully connected layer (fc)

This section presents the classification of the food items and their regression by reshaping the results of the Max ROI, which were transferred to the FC layer (refer to Figure 1). This phase consisted of two branches, namely food classification and the localisation of each food item.

Algorithm	AP_{50}	AP_{60}	AP_{70}	AP_{80}	AP_{90}
Our framework	96.23	95.06	86.82	85.08	72.80

3.3. Instance segmentation

The enhancement of FCN [23] was performed to distinguish between the levels of class, while the instance segmentation returned the boundaries of each food item. This phase consisted of three steps, which are as follows:

1. The first stage was the transfer of the output of Max ROI for food item through a series of 3 x 3 convolutional layers, which were re-applied three times after the application of ReLU to generate boundaries for each region obtained from Max ROI.
2. The second stage involved a 1 x 1 convolutional layer for each feature map obtained from the last convolution.
3. The third stage converted the segmentation size according to the input image through bilinear interpolation, in which different thresholds used in the food paper included (50, 60, 70, 80, 90).

4. EXPERIMENTAL RESULTS

In this section, the experimental results and evaluation of the algorithm from the previous sections are illustrated. The evaluation of the multi food instance segmentation framework was performed using the following two measurements: AP with different thresholds was used compare between the multi-object instance segmentation frameworks, which included the enhanced CNN, RPN, and instance segmentation and average GPU time for training and testing time.

4.1. Experimental specifications

A multi-object instance segmentation framework was implemented using TensorFlow [24]. All methods were tested on Amazon Web Services (AWS) and Amazon Machine Image (AMI), with the GPU-Us-Tesla V100 16 GB and VCPUs-8 cores 61 GB. Furthermore, the optimisation algorithm used in this study framework was gradient descent (SGD). The weight decay amounted to 0.0001, with a learning momentum of 0.9, and a learning rate of 0.001 for 50 epochs. Each of the epochs consisted of 1000 iteration.

4.2. Backbone result

The proposed backbone network was selected for the specific number of duplicates for each convolution block based on the acquired results, The enhancing CNN improved the performance through the selection of more suitable duplicates of the number of layers. As a result, high accuracy and reduction in the number of parameters were obtained, which reduced computation time. The most suitable duplicates for each convolution block are as follows:

1. One-time repetition of the first block convolution.
2. Five-time repetition of the second block convolution.
3. Four-time repetition of the third block convolution.
4. Repetition of the fourth block convolution by 12 times.
5. Three-time repetition of the fifth block convolution.

4.3.Multi food instance segmentation framework result

Table 1 presents the multi food instance segmentation framework performance with different thresholds (50, 60, 70, 80, 90), in which the framework became more selective with the increase in the thresholds. It was found that the accuracies for AP50, AP60, AP70, AP80, AP90 amounted to 96.23, 95.06, 86.82, 85.08, 72.80 sequentially, indicating that this study framework contributes to the near-optimal solution.

Table 1: Multi food instance segmentation framework performance with different thresholds (50,60,70,80,90).

4.4. Comparison with the state-of-the-art algorithms

It could be seen in Table 2 that the frameworks proposed in this study were compared with the state-of-the-art algorithms, namely Mask, R-CNN [16], YOLACT [17], and CASCADE R-CNN [18]. These algorithms were then trained and tested with various thresholds amounting to 0.5, 0.75, 0.9.

Table 2: Evaluation results of multi food instance segmentation framework with different state-of-the-art algorithms.

Algorithm	Training time in second	Testing time in second
Mask R-CNN	2500	31.02
YOLACT	5880	17.20
CASCADE R-CNN	9600	19.43
Our framework	2000	19.21

Table 2 above presents the results of the comparison between the algorithms in the testing process. The performance of the proposed framework was found to exhibit better results with different AP, namely (AP_{50} AP_{75} AP_{90}). The AP accuracy of our framework amounted to 96.23, 85.82, 72.80, which was followed by AP_{50}, $AP_{75}$and AP_{90}. Notably, these values were significantly higher compared to the values of the Mask R-CNN 90.02, 77.02, and 69.17, CASCADE R-CNN 95.01, 83.49, and 70.92, and YOLACT 90.40, 76.51, and 69.03 due to the suggestion of a new backbone in this study, which was essential to achieve important results. The proposed network solved the issue of vanishing gradient through identity shortcut based on a new gradient formula while taking the efficiency training for each convolution block into account. This aspect was considered through a specific duplicate increase or decrease in training and the reduction in the filter size to extract a certain amount of feature from the input image and transfer it to other layers. The benefit obtained from the feature led to positive results compared to other algorithms. Besides the framework, novel techniques were also proposed in this study to manage different sizes of feature map generated from the RPN known as Max RoI. Furthermore, the Max RoI techniques reduced the feature maps into a fixed size while maintaining the location of the map, which was obtained from the previous algorithms by avoiding the quantisation. As a result, significant results were obtained in the instance segmentation due to the impact of every pixel value in the feature maps on the instance segmentation process. Given the significant performance of the proposed framework for the framework in the multi food instance segmentation framework, it was predicted that the proposed framework stored a potential for substantial results.

4.5. Time

Provided that training and testing time is an important factor in the evaluation of the performance of algorithms, a set of techniques were developed to reduce time. .

Table 3: Evaluation training and testing time of multi food instance segmentation framework with different state-of-the-art algorithms.

Algorithm	AP_{50}	AP_{75}	AP_{90}
Mask R-CNN[	90.02	77.02	69.17
YOLACT	90.40	76.51	69.03
CASCADE R-CNN [	95.01	83.49	70.92
Our framework	96.23	85.82	72.80

Based on the evaluation training and testing time of multi food instance segmentation framework with different state-of-the-art algorithms shown in the table above, it could be seen that this study framework required the shortest time in training process due to its dependence on the proposed backbone, which resulted in the avoidance of the issue in ResNet. This phenomenon occurred with the reduction in the filter size and the focus on the layer, which required more training and reduction in the number of unrequired training on layers to achieve improved results in the shortest time. Comparatively, the Mask R-CNN, CASCADE R-CNN, and YOLACT required a longer time in the training process due to the impact of the implementation of ResNet101 as the backbone network. As mentioned previously, ResNet was faced with several issues, including the use of a large filter size affecting the parameters of the increased consumption time. Besides, many layers did not require training, which resulted in a significant consumption of time in the training process. Notably, the CASCADE R-CNN algorithm required a long time in the training process compared to other algorithms due to the repetition of the bounding box by three times by the algorithm to achieve the best result. The proposed framework consumed 19.21 second in the testing process, making it the second algorithm with less testing time after a YOLACT algorithm despite the high-speed characterisation of the YOLACT algorithms in this process. Simultaneously, the accuracy in AP_{50}, $AP_{75}$and AP_{90}.The visual experimental results from this paper are compared to the state of the art algorithms, as presents in Figure. 7.

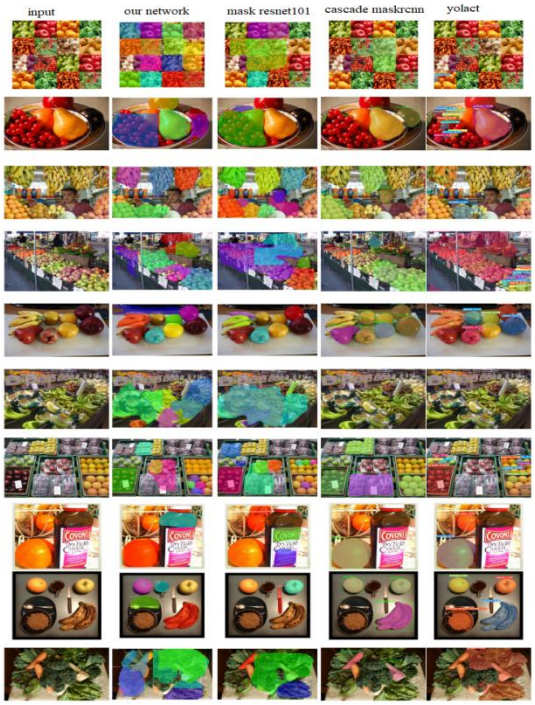


Figure 7: The visual experimental results from this paper are compared to the state of the art algorithms

5. CONCLUSION

This paper proposed a high-performance multi food instance segmentation framework through two main steps. In the first step, the instance segmentation dataset should be created, in which the images used to build the dataset in this study were acquired by iPhone5. The images were annotated by VIA, which was then converted into a COCO dataset format through Python. In the second step, the architecture of multi food instance segmentation was proposed through four steps. Firstly, multi food image was enhanced through an image processing to manage different image sizes, followed by the extraction of low and high levels of features through the proposed backbone. The third step was the adoption of RPN to locate the food item and develop a new technique called Max RoI layer, which managed various features map to obtain the best boundaries for the multi food image. A better performance was observed from the proposed framework in terms of the accuracy AP with different thresholds, namely AP_{50}, AP_{75} and AP_{90}. These thresholds were higher than those of the three instance segmentation algorithms, including Mask R-CNN, YOLACT, and CASCADE R-CNN. Shorter training time was also utilised in the proposed method compared to other methods. Notably, the proposed framework functioned in the rapid and accurate identification, detection, and segmentation of the multi food in terms of accuracy, training, and testing time. However, the results of this study were based on a training and testing dataset with a total of 27000 images for 27 class of food item. Therefore, it is suggested that more images in the database are enlarged to further increase the accuracy and robustness of the proposed method.

REFERENCES

[1] P. Pouladzadeh, P. Kuhad, S. V. B. Peddi, A. Yassine, S. Shirmohammadi, Food calorie measurement using deep learning neural network, in: 2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings, IEEE, 2016, pp. 1–6. https://doi.org/10.1109/I2MTC.2016.7520547

[2] K. J. Pfisterer, R. Amelard, A. G. Chung, B. Syrnyk, A. MacLean, A. Wong, Fully-automatic semantic segmentation for food intake tracking in longterm care homes, arXiv preprint arXiv:1910.11250 (2019).

[3] Y. Kawano, K. Yanai, Real-time mobile food recognition system, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2013, pp. 1–7.

[4] Y. Matsuda, H. Hoashi, K. Yanai, Recognition of multiple-food images by detecting candidate regions, in: 2012 IEEE International Conference on Multimedia and Expo, IEEE, 2012, pp. 25–30.

[5] C. Morikawa, H. Sugiyama, K. Aizawa, Food region segmentation in meal images using touch points, in: Proceedings of the ACM multimedia 2012 workshop on Multimedia for cooking and eating activities, 2012, pp. 7–12. https://doi.org/10.1145/2390776.2390779

[6] S. D. Bimorogo, G. P. Kusuma, A comparative study of pretrained convolutional neural network model to identify plant diseases on android mobile device, In-ternational Journal of Advanced Trends in Computer Science and Engineering 9(2020). https://doi.org/10.30534/ijatcse/2020/53932020

[7] I. P. Aji, G. P. Kusuma, Landmark classification service using convolutional neural network and kubernetes, International Journal of Advanced Trends in Com-puter Science and Engineering 9 (2020) https://doi.org/10.30534/ijatcse/2020/52932020

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

[9] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[10] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[11] M. Gayathri, M. Meghana, M. Trivedh, D. Manju, Suspicious activity detection and tracking through unmanned aerial vehicle using deep learning techniques, International Journal of Advanced Trends in Computer Science and Engineering9 (2020).

[12] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.

[13] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in neural information processing systems, 2015, pp. 91–99.

[14] J. Dai, K. He, J. Sun, Instance-aware semantic segmentation via multi-task network cascades, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3150–3158.

[15] Y. Li, H. Qi, J. Dai, X. Ji, Y. Wei, Fully convolutional instance aware semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2359– 2367. https://doi.org/10.1109/CVPR.2017.472

[16] K. He, G. Gkioxari, P. Dollar, R. Girshick, Mask r-´ cnn, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.

[17] D. Bolya, C. Zhou, F. Xiao, Y. J. Lee, Yolact: realtime instance segmentation, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 9157–9166.

[18] Z. Cai, N. Vasconcelos, Cascade r-cnn: High quality object detection and instance segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence (2019).

[19] H. Kagaya, K. Aizawa, M. Ogawa, Food detection and recognition using convolutional neural network, in: Proceedings of the 22nd ACM international conference on Multimedia, 2014, pp. 1085–1088. https://doi.org/10.1145/2647868.2654970

[20] T. Ege, K. Yanai, Estimating food calories for multiple-dish food photos, in: 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR), IEEE, 2017, pp. 646–651.
[21] A. Dutta, A. Zisserman, The vgg image annotator (via), arXiv preprint arXiv:1904.10699 (2019).
[22] K. Turkowski, Filters for common resampling tasks, in: Graphics gems, Academic Press Professional, Inc., 1990, pp. 147–165.
[23] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.
M. Abadi, A. Agarwal, P. Barham, E. Brevdo,
Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., Tensorflow: Large-scale machine learning on heterogeneous distributed systems, arXiv preprint arXiv:1603.04467 (2016).E. BackProp, Y. LeCun, L. Bottou, G. B. Orr, K. Muller, NEURAL NETWORKS: TRICKS OF THE TRADE, 1998.
[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, DevaRamanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objectsin context. InEuropean conference on computer vision, pages 740–755. Springer, 2014.
HTTPS://DOI.ORG/10.1007/978-3-319-10602-1_48