Open Access **Baghdad Science Journal** **P-ISSN: 2078-8665**
Published Online First: November 2022 **2023, 20(3): 893-903** **E-ISSN: 2411-7986**

DOI: https://dx.doi.org/10.21123/bsj.2022.7243

Comparison of Faster R-CNN and YOLOv5 for Overlapping Objects Recognition

Muhamad Munawar Yusro ***Rozniza Ali**** ***Muhammad Suzuri Hitam***

Department of Computer Science, Universiti Malaysia Terengganu, Malaysia.
*Corresponding author: rozniza@umt.edu.my
E-mail addresses: p4636@pps.umt.edu.my , suzuri@umt.edu.my

Received 28/3/2022, Revised 15/7/2022, Accepted 17/7/2022, Published Online First 20/11/2022
Published 1/6/2023

Abstract:

Classifying an overlapping object is one of the main challenges faced by researchers who work in object detection and recognition. Most of the available algorithms that have been developed are only able to classify or recognize objects which are either individually separated from each other or a single object in a scene(s), but not overlapping kitchen utensil objects. In this project, Faster R-CNN and YOLOv5 algorithms were proposed to detect and classify an overlapping object in a kitchen area. The YOLOv5 and Faster R-CNN were applied to overlapping objects where the filter or kernel that are expected to be able to separate the overlapping object in the dedicated layer of applying models. A kitchen utensil benchmark image database and overlapping kitchen utensils from internet were used as base benchmark objects. The evaluation and training/validation sets are set at 20% and 80% respectively. This project evaluated the performance of these techniques and analyzed their strengths and speeds based on accuracy, precision and F1 score. The analysis results in this project concluded that the YOLOv5 produces accurate bounding boxes whereas the Faster R-CNN detects more objects. In an identical testing environment, YOLOv5 shows the better performance than Faster R-CNN algorithm. After running in the same environment, this project gained the accuracy of 0.8912(89.12%) for YOLOv5 and 0.8392 (83.92%) for Faster R-CNN, while the loss value was 0.1852 for YOLOv5 and 0.2166 for Faster R-CNN. The comparison of these two methods is most current and never been applied in overlapping objects, especially kitchen utensils.

Keywords: Computer vision, Convolutional neural network, Faster r-cnn, Kitchen utensils, Overlapping object recognition, Yolo.

Introduction:

Detecting an overlapping object for a normal person is not difficult, but for some people, such as the impaired visual person, the process of detecting is not easy. They still need special assistant tools for helping them to recognize the objects [1]. In computer vision, the problem of object detection is a problem estimating the class and location of objects contained in picture [2]. With the traditional image processing method approach, object detection can apply various techniques such as optical flow, frame differencing, and background subtraction [3]. Each technique has its own advantages and disadvantages in terms of accuracy and computing time [4]. Some examples of case studies that apply object detection techniques include face detection [5], applications to support smart cities [6], vehicle detection [7], and player and ball detection on tennis broadcast videos [7].

Classifying an overlapping object is one of the main challenges faced by researchers who work in object detection and recognition. Most of the available algorithms that have been developed are only able to classify or recognize objects which are either individually separated from each other or a single object in a scene(s). Khauola Drid et.al (2020) combined YOLOv3 and Faster R-CNN detector to detect small objects and overlapping objects using prepared dataset PASCAL VOC 2007 and 2012. Zhi Tian et al (2019) have proposed a fully convolutional one-stage object detector (FCOS) to build object detection in a per-pixel prediction fashion, analogue to semantic segmentation.

Current approaches to object detection and recognition make essential use of machine learning methods. To learn large amount of objects from image databases, we need a model with large

learning capacity. Convolution Neural Network (CNN) constitutes one such class of model. Their capacity can be controlled by varying their depth and breath. However, CNN today has not been trained to detect and classify an overlapping object. The use of common filter in the available CNN algorithm is not able to classify an overlapping object.

In this project, Faster R-CNN and YOLOv5 were proposed to detect and classify an overlapping object in overlapping kitchen utensils. The model was trained and tested on the collection of two datasets: a collection of single kitchen utensils images from the University of Edinburgh (897 images), and a collection of overlapping kitchen utensils that were collected and downloaded from the internet (3484 images). The analysis compared the YOLOv5 and Faster R-CNN performance in recognizing the overlapping kitchen utensils. In an identical testing environment, YOLOv5 shows a better performance than Faster R-CNN. YOLOv5 has the highest speed with a computing time of about 0.61 seconds per image. Faster R-CNN with InceptionV2 has mAP value of ~63%. The comparison of these two methods is most current and never been applied in overlapping objects, especially kitchen utensils.

Related Studies:

Recently, with the availability of large amounts of data supported by increasingly advanced computer hardware technology, the solution of object detection problems began to shift to a deep learning approach which in various studies proved to produce more promising accuracy [8]. In addition, object detection with deep learning can be done automatically by the machine because it does not pass the stage of handcrafted feature extraction. According to Sultana et al. (2019), there are two approaches in deep learning-based object detection algorithms, namely two-stage (two stages) and one-stage (one stage) as in Fig. 1. In two-stage object detectors, the proposal and classification region stages are carried out on separate networks.

In this one-stage object detection approach, the region proposal method is used to look for regions or parts of an image that may be an object. Then the extraction of features from each region using CNN. The extraction of the feature is used as input for the classifier model for the classification process so that the class of the region is obtained and the regressor model to obtain the bounding box for the object contained in image [9]. Some of the two-stage object detector algorithms include R-CNN [10], Fast R-CNN [11], Faster R-CNN [12], and R-FCN [11]. Algorithms like this have high precision value but have the disadvantage of high computational complexity [6].

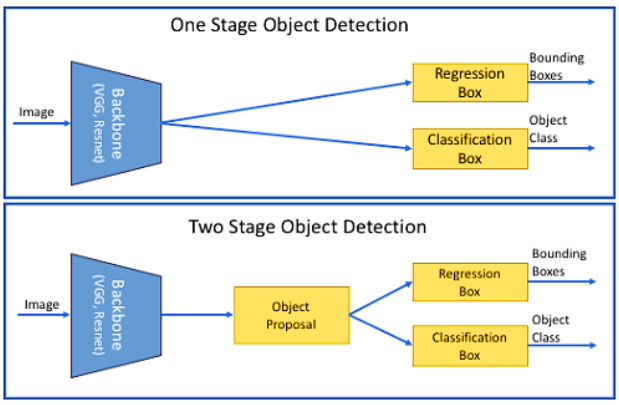

Figure 1. Two stages vs. one stage object detection models.

In one-stage object detectors, proposal and classification region processes are combined in a single network, such as YOLO [13], YOLOv2 [14], YOLOv3 [15, 16], and SSD [14]. The advantage of the one-stage object detector algorithm is that it has a smaller computational time but with a smaller degree of accuracy than the two-stage object detector model [17]. However, some state-of-the-art one-stage detectors such as YOLOv5 and SSD have been able to compete even outperforming the two-stage detector method in terms of accuracy.

Intersection over Union (IoU)

The main usage of IoU in object detection is to evaluate object detectors by measuring the similarity of predicted bounding box (bbox) with ground truth (GT) bbox. Calculation of IoU metrics is done by comparing the area of slices or overlapping regions with the combined area between predicted bbox with ground truth bbox. The IoU between forms (volume) A, B ⊆ S ∈ Rn was obtained by [15]:

$$IoU = \frac{|A \cap B|}{|A \cap B|} \quad 1$$

Faster R-CNN

There is an improvement over Fast R-CNN called Faster R-CNN which is more efficient in aspects of computing time and has almost real-time detection performance. The fundamental change to Faster R-CNN is in the use of the Region Proposal Network (RPN) as a region proposer to resurrect the proposed region in place of the selective search method (Fig. 2).

RPN significantly reduced computing time to resurrect the proposal region due to computation sharing with the Fast R-CNN network. As for

detecting objects used fast R-CNN network structure [9]. In other words, a combination of RPN as a region proposer and Fast R-CNN as an object detector is Faster R-CNN.

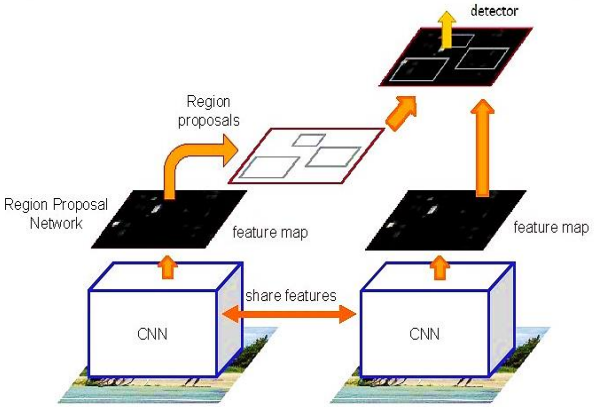

Figure 2. Flow Diagram of Faster R-CNN

YOLO: You Only Look Once

YOLO uses the principle of one stage object detection, has a very high computing speed and can process images in real time. In YOLO, the usually separate components of object detection are put together in one artificial neural network so that YOLO has ability in end-to-end training and high speeds while maintaining high precision [5].

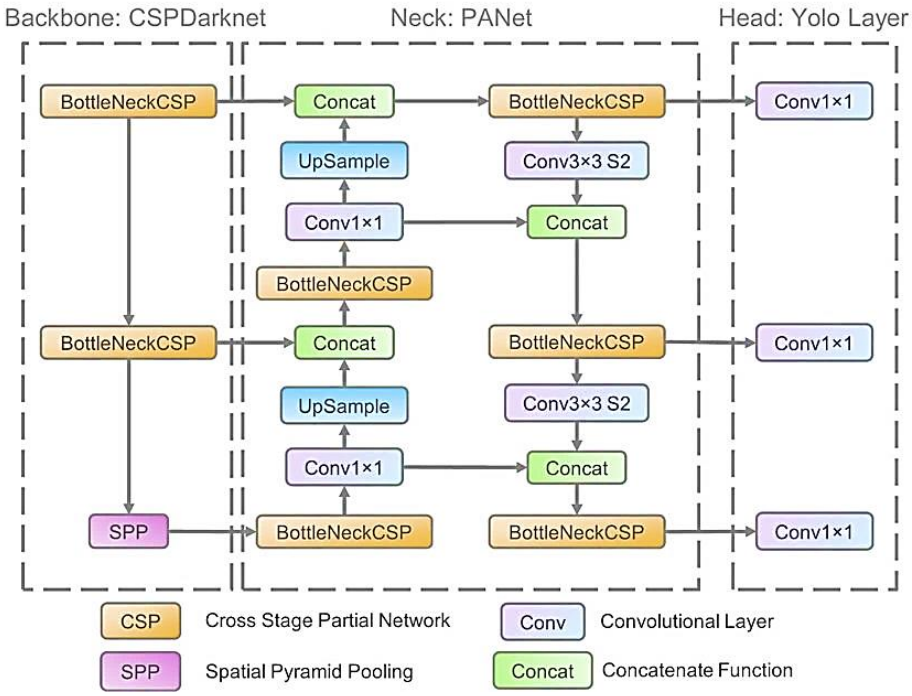

Figure 3. The network architecture of YOLOv5 [18].

As shown in Fig.3, YOLOv5 consists of: (1) Backbone: CSPDarknet, (2) Neck: PANet, and (3) Head: YOLO Layer [18]. The image data is first input to CSPDarknet for step of feature extraction, and after that sent to PANet for feature fusion. Finally, the last part is YOLOv5's head layer outputs detection results (class, score, location, size).

Methodology:

Data and Research Stages

The datasets obtained are used for object detection processes with the Deep Learning Faster R-CNN and YOLOv5. These two methods were chosen because in addition to being well structured, their use is very wide both in the academic field as in research and practically [2, 3]. After the deep learning model is obtained through the training process, then the model evaluation and performance analysis of each algorithm is carried out.

Some of the stages performed in this project are shown in Fig. 4. In general, the research stage consists of data acquisition, data annotation, object detection model training, model evaluation (calculating mAP), model implementation, and

performance comparison analysis of the 2 object detection algorithms.

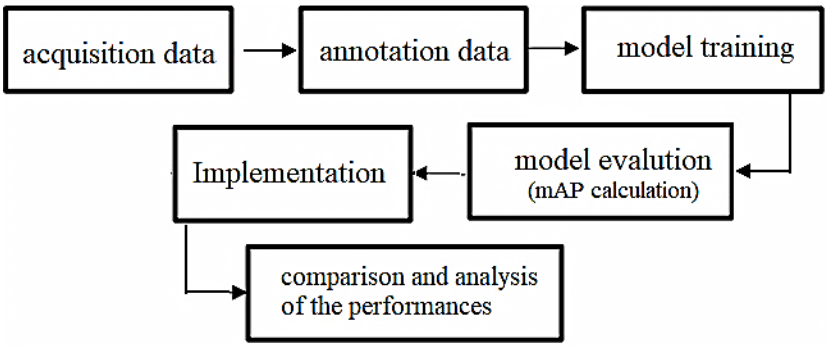

Figure 4. The research stages

Data Acquisition

The original database has 897 images [9]. The kitchen utensils images were collected by visiting charity shops and stores around the City of Edinburgh, and were manually captured for high resolution, proper lighting conditions on a constant background. Additional 3484 overlapping kitchen utensils images were downloaded from the internet and captured using the specific browser's add-ons. The number of images in each class including the images from the Edinburgh Kitchen Utensil Database are shown in Table 1.

Table 1. Name of object classes and their amount in each class

Class	Number of Images	Overlapping Kitchen Utensils Images	Wearable Sensor images
Bottle Opener	30	113	40
Bread Knife	24	240	40
Can Opener	19	192	40
Dessert Spoon	33	187	40
Dinner Fork	59	155	40
Dinner Knife	51	136	40
Fish Slice	82	205	40
Kitchen Knife	39	179	40
Ladle	54	183	40
Masher	38	161	40
Peeler	18	185	40
Pizza Cutter	16	217	40
Potato Peeler	22	168	40
Serving Spoon	84	152	40
Soup Spoon	27	194	40
Spatula	53	183	40
Teaspoon	105	146	40
Tongs	37	190	40
Whisk	44	178	40
Wooden Spoon	62	120	40
TOTAL	**897**	**3484**	**800**

Data Annotation

Annotation of data is done by providing bounding boxes and class labels on each object in the image. In this project, the data was classified into 20 classes, as in Table 1. The process of giving ground truth boxes and labels to images is done using the labelIMG tool and website www.makesense.ai that has a service to create bounding boxes and class labels on the image for image annotation.

Model Training

Object detection algorithms with a deep learning approach are divided into two types,

namely two-stage object detector and one-stage object detector. In two stages object detectors the proposal and classification region process are carried out on a separate network, while in one-stage object detector the proposal and classification region process are combined in one network [19]. Object detectors used in the project include two-stage detectors, Faster R-CNN, and one-stage detector YOLOv5. The training model is carried out using graphics processing unit (GPU) with the following scheme:

a. The ratio of training and testing data on this experiment is 0.8:0.2.
b. For Faster R-CNN used anchor boxes with three scales and three ratios, while the prior box for YOLOv5 was obtained from the results of k-means clustering on data bounding boxes in image datasets.
c. By doing the process of augmentation in the data can increase the amount of data for model training so that it can overcome the problem of overfitting and make the model more robust and have better accuracy [20]. There are various types of augmentation techniques that are often used in deep learning for image processing including random cropping, image mirroring, and multi-scale training. The augmentation technique used in this project is random horizontal flipping [21].
d. Faster R-CNN use three pretrained models, inceptionV2 backbone, ResNet50, and MobileNet. All three models have been trained with the COCO dataset. YOLOv5 used a pretrained model with a Darknet-53 backbone that had been trained on the COCO dataset [22].
e. The framework in this project uses TensorFlow as a machine learning framework for faster R-CNN and YOLOv5 training.

Model Evaluation

Evaluation of the detection model is done by calculating precision and recall. The value of recall and precision will be used to measure how well the recognized object fits into the reference one. Definition of recall the proportion of objects correctly detected among all objects that should be detected [23]. Precision is the proportion among the positive classes detected correctly among the number of positive classes detected [12]. Precision and recall are calculated with Eq.2 and Eq.3 based on Table 2 [5].

Table 2. The confusion matrix compares the actual target values with those predicted by the machine/deep learning model.

	True Positif	**True Negatif**
Predicted Positif	*True Positive* (TP)	*False Positive* (FP)
Predicted Negatif	*False Negative* (FN)	*True Negative* (TN)

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

The IoU threshold can be used to determine whether a detection is correct. For instance, the threshold value of IoU is set to 0.5, then:

- The IoU threshold can be used to determine whether a detection is correct. Suppose the threshold value of IoU 0.5 is set, then:
- If the IoU $\geq$ 0.5, it means true detection and includes True Positive (TP). In other words, the model correctly predicts the object as an object.
- If the IoU < 0.5, it means false detection and includes False Positive (FP), or the model predicts the background as an object when it shouldn't be an object.
- When an object fails to be detected by the model, it includes False Negative (FN), meaning that the model mistook the object for a background.
- When the model does not detect in the background that does not need to be detected, it includes True Negative (TN). For object detection, this FN metric is not used.

Metrics for predictions are calculated for each class. Average Precision (AP) is used as an indicator to evaluate the performance of dataset classes on models and to measure performance in object detection. AP summarizes the shape of the precision-recall curve, and it defines the score based on the precision average of a set of recall values equidistant (0, 0.1, 0.2, ..., 1). AP's calculation follows Eq.4 and Eq.5 [16].

$$AP = \frac{1}{11}\sum_{r(0,0.1,0.2,\ldots,1)} P_{interp}(r) \quad (4)$$

$P_{interp}(r)$ or interpolated precision defined as

$$P_{interp}(r) = \max_{\tilde{r}:\tilde{r}\geq r} P(\tilde{r}) \quad 5$$

where $P(\tilde{r})$ is precision measured on recall ($\tilde{r}$).

Model Implementation

The object detector models that have been trained are then implemented on large computing devices, namely Google COLAB, to detect overlapping objects. At Google Colab, each object detector model was tested using some test data. At this stage of implementation, measurements are taken to inference time and the use of resources (such as CPU and memory) of each algorithm. From the object detection process carried out on TensorFlow's official website, namely Tensorflow.org, it can be understood that inference time is the time that takes the model when determining or jumping conclusions about bounding boxes and class labels on each object detected in an image. Inference time is measured when the model detects an image or during testing.

Algorithmic Performance Comparison Analysis

Each object detection algorithm is compared based on evaluation results that include mAP calculations as well as measurements of computing time and resource usage. Analysis is performed on the performance of each algorithm so the advantages and disadvantages of each algorithm to detect overlapping objects based on trade off accuracy, computational time, and resource usage can be explained.

Results:

Data Annotations

At this stage, each image data is annotated, namely by providing bounding boxes and class labels according to each object in the image. Fig.5 shows examples of kitchen utensils image types in (a) single object and (b) non-single and non-overlapping objects, and (c) overlapping objects.

(a) (b) (c)

Figure 5. example of kitchen utensils image: (a) single object and (b) non-single and non-overlapping objects, and (c) overlapping objects.

Fig.5 shows the process of data annotation using labelImg tool. Objects in the image are bounded and labelled according to the number of kitchen utensils objects in the image. The number of classes in kitchen utensils follows the name of a class from a database of kitchen utensils created by researchers from the university of Edinburgh, as shown in Table 1, which is as many as 20 classes.

The annotation process using labelImg (Fig.6) generates files with *.xml format. For faster R-CNN model training, the *.xml file is then converted into a *.csv file that contains four coordinates (x_min, y_min, x_max, and y_max) and label of classes. After that, the *.csv file along with the image dataset are converted into TFRECORD files with *record format. File *record is the input for training object detection model.

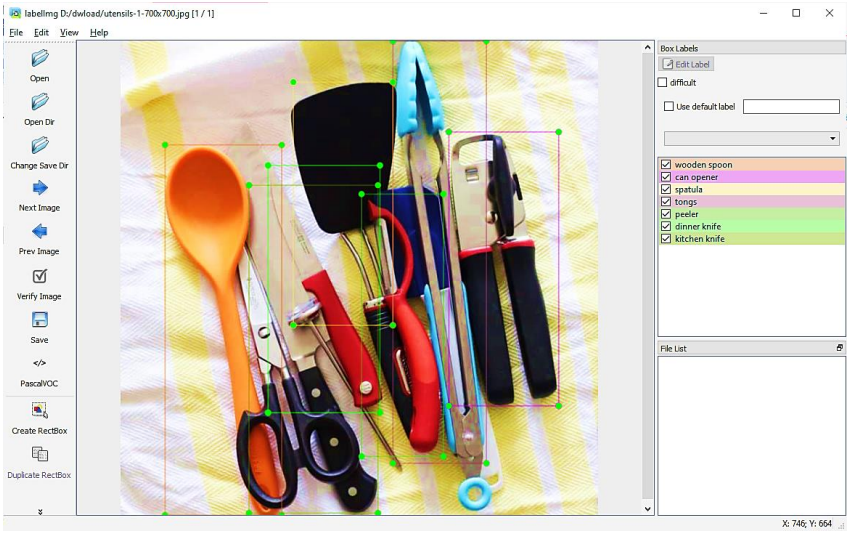

Figure 6. Data annotation process using labelIMG for overlapping kitchen utensils image

For YOLOv5 model's training, the *.xml file is converted into a text file with *.txt format. Each line in the *.txt file contains data about the image with image_index path_of_image image_width image_height box_1 box_2 ... box_n. Each box_x as many as n boxes are written in this format (label_index x_min y_min x_max y_max). The variables named image_index and label_index contain an index of numbers starting from zero (0). The examples of conversion results into *.txt format are as shown Fig. 7.

```
0 /path_to_dataimage/abc1.jpg 2280 1244 0 712 284 1470 522 1 413 1076 942 1374
1 /path_to_dataimage/abc20.jpg 2280 1244 0 1305 2514 2425 1814 0 1570 1037 1685 1302
2 /path_to_dataimage/abc311.jpg 2280 1244 0 129 790 775 1250 0 1366 786 1769 1268
```

Figure 7. Example of the conversion results into a txt file.

Data Augmentation

There are two methods of performing augmentation techniques during model training, namely offline and online (on-the-fly). Offline augmentation methods are carried out separately from the training process. The augmentation technique is first applied to the image data, then the image that has been treated is stored in storage and becomes additional training data. Online or on-the-fly augmentation techniques are carried out as model training progresses. Online methods are more widely used because they are more efficient and do not reduce storage capacity because there is no need to store augmentation image data.

In this project the vertical flipping and rotation (at 90 degrees) augmentation technique was applied to the training of all models (Faster R-CNN and YOLOv5). For both online and offline augmentation are used for Faster R-CNN and YOLOv5 model training, used because TensorFlow's object detection API has provided modules/functions to perform the task. Some results of image augmentation step are shown Fig.8.

Figure 8. The result images after applying horizontal flipping (b), vertical flipping (c), and after rotating 90 degrees right from the original image (a).

Model Evaluation

Evaluation of the object detection model is done by calculating the mean average precision (mAP) value. In this project, the mAP calculation was done with the IoU threshold value for non-max suppression (NMS) process is 0.6. Based on Table 3, YOLOv5 has the highest mAP value compared to Faster R-CNN methods, while the lowest mAP is owned by the Faster R-CNN method with Resnet152 as the backbone. YOLOv5 model with MS COCO and Pytorch backbone becomes the most accurate model in detecting overlapping objects and their class labels with a mAP of 63.54%. This suggests that the two-stage detection model is superior in terms of accuracy than the one-stage model.

Table 3. The value of mAP for each algorithm/method

Algorithm/Method	*Backbone*	mAP (%)	Inference time for each image (second)
Faster R-CNN	InceptionV2	63.46	2.74
	ResNet152	60.47	2.59
YOLOv5	MS COCO and PyTorch	63.54	0.61

Discussion:

Performance Comparison Analysis

The one-stage method (YOLOv5) has advantages in accuracy over the two-stage method (Faster R-CNN), but it has disadvantages in terms of computation time. In this project, the computing time required by the two-stage method to perform detection was longer than the one-stage methods. The two-stage method performs the proposal region stage and classification and regression separately thus increasing the computing time. At each stage, the two-stage method performs classification, the first being to determine the presence of an object and the second to determine the class label. Fig.9 shows the image of the recognition process result.

Figure 9. The example images of the recognition process result.

Table 3 displays a comparison of mAP and inference time of each method. For the one-stage method, the fastest computing time is owned by YOLOv5 with a computing speed of 0.61 seconds, which is five times faster than the Faster R-CNN method. Although YOLOv5 is the fastest, its mAP value is still the smallest. The Faster R-CNN with InceptionV2 or with Resnet152 reached similar computing time but had ~3% higher mAP. So, Faster R-CNN takes the longest computing time compared to other detection algorithms.

In this project, researchers used Tensorboard to visualize the result of 2 methods, YOLOv5 and Faster R-CNN. After running 100 epochs in training, this project gained the accuracy of 0.8912(89.12%) for YOLOv5 and 0.8392 (83.92%) for Faster R-CNN (Fig.11), while the loss value was 0.1852 for YOLOv5 and 0.2166 for Faster R-CNN (Fig.10).

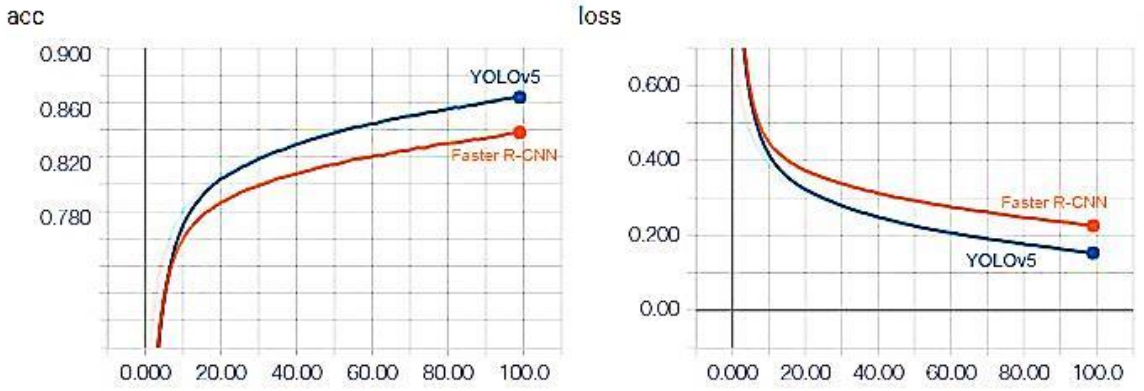

Figure 10. The result of training step in 100 epochs.

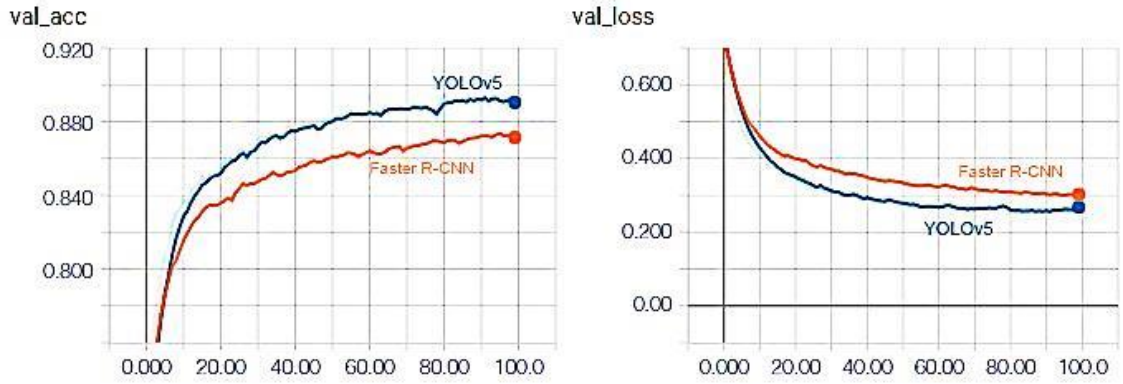

Figure 11. The result of validation step in 100 epochs.

Conclusion:

The value of mAP is directly proportional to the algorithm's inference time, which is the higher the mAP value, the longer the computing time needed. Algorithms that can extract features better then result in better detection performance for complex detection areas and dense. After running in the same environment, this project gained the accuracy of 0.8912(89.12%) for YOLOv5 and 0.8392 (83.92%). YOLOv5 has the highest speed with a computing time of about 0.61 seconds per image, and it has mAP value of 63, 54%. Therefore, based on the previous result, the team chose the algorithm with the highest accuracy, YOLOv5 can be the best alternative option to use in detection of overlapping object. For future work, the combination of YOLOv5 and Faster R-CNN can be applied to gain higher accuracy and higher computing speed.

Acknowledgment:

Gratefully, the authors acknowledge the collaborators; this project was funded by Fundamental Research Grant Scheme (FRGS) RACER/1/2019/ICT02/UMT/1.

Authors' Declaration:

- Conflicts of Interest: None.
- We hereby confirm that all the figures and tables in the manuscript are ours. Besides, the figures and images, which are not ours, have been given permission for re-publication attached with the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee in Malaysia University.

Author's Contributions Statement:

M.M.Y. developed the theory and carried out the experiments, wrote the manuscript with input from all the authors. R.A. and M.S.H. conceived the presented idea, supervised and directed the project, planned the experiments, and provided proofreading of the manuscript. All authors discussed the results and contributed to the final manuscript.

References:

1. Bashiri FS, LaRose E, Badger JC, D'Souza R M, Yu Z, Peissig P. Object Detection to Assist Visually Impaired People: A Deep Neural Network Adventure. Adv. in Vis. Comp., ISVC. 2018: 500-510.
2. Panchal P, Prajapati G, Patel S, Shah H, Nasriwala J. A Review of Object Detection and Tracking

Methods. Int J Res Emerg Sci. Technol. 2015; 2(1): 7-12.
3. Nguyen K, Huynh NT, Nguyen PC, Nguyen KD, Vo ND, Nguyen TV. Detecting Objects from Space: An Evaluation of Deep-Learning Modern Approaches. Electronics. 2020; 9: 583.
4. Alganci U, Soydas M, Sertel E. Comparative Research on Deep Learning Approaches for Airplane Detection from Very High-resolution Satellite Images. Remote Sens. 2020; 12(3): 458.
5. Sharma V. Face Mask Detection Using YOLOv5 for COVID-19. MSc. [thesis]. USA: California State University; 2020.
6. He Y, Zeng H, Fan Y, Ji S, Wu J. Application of Deep Learning in Integrated Pest Management: A Real-Time System for Detection and Diagnosis of Oilseed Rape Pests. Mob. Inf. Syst. 2019; 2019: 1-14.
7. Redmon J, Divvala S, Girshick R, Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2016; 779-788.
8. Arulprakash E, Aruldoss M. A Study on Generic Object Detection with Emphasis on Future Research Directions. J King Saud Univ Comp Info Sci. 2021 Aug 12; Online first.
9. Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Comp Vis Patt Recog. 2016.
10. Cai J, Li Y. Realtime Single-stage Instance Segmentation Network Based on Anchors. Comput Electr Eng. 2021; 95: 107464. https://doi.org/10.1016/j.compeleceng.2021.107464.
11. Granada R, Monteiro J, Barros R C, Meneguzzi F. A Deep Neural Architecture for Kitchen Activity Recognition. The Thirtieth International Flairs Conference. 22-24 May 2017: 56-61.
12. Kim KY, Kim Y, Park J, Kim YS. Real-Time Performance Evaluation Metrics for Object Detection and Tracking of Intelligent Video Surveillance Systems. Asia Pac J Contemp Educ Commun Technol. 2016: 173-179. https://apiar.org.au/journal-paper/real-time-performance-evaluation-metrics-for-object-detection-and-tracking-of-intelligent-video-surveillance-systems/.
13. Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv: Comput Vis Patt Recog. 2020 Apr 23; 1-17. https://arxiv.org/abs/2004.10934.
14. Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger. Proc IEEE Comput. Soc Conf Comput Vis Pattern Recognit. 2018; 1-9.
15. Bernardin K, Elbs Er, Stiefelhagen R. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. Eurasip J Image Video Process. 2008: 246309. https://doi.org/10.1155/2008/246309.
16. Redmon J, Farhadi A. YOLOv3: An Incremental Improvement. arXiv:1804.02767v1 [cs.CV]. 2018 Apr 8: 1-6.
17. Carranza-García M, Torres-Mateo J, Lara-Benítez P, García-Gutiérrez J. On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles using Camera Data. Remote Sens. 2021; 13(1): 89.
18. Renjie X, Haifeng L, Kangjie L, Lin C, Yunfei L. A Forest Fire Detection System Based on Ensemble Learning. Forest. 2021; 12: 217.
19. Tian Z, Shen C, Chen H, He T. FCOS: Fully Convolutional One-Stage Object Detection. Proc. of the IEEE/CVF. Int Conf. Comput. Vis. 2019: 9627-9636.
20. Abdulmunem I A, Harba E S, Harba H S. Advanced Intelligent Data Hiding Using Video Stego and Convolutional Neural Networks. Baghdad Sci J. 2021; 18(4): 1317.
21. Asroni A, Ku-Mahamud K R, Damarjati, C, Slamat H B. Arabic Speech Classification Method Based on Padding and Deep Learning Neural Network. Baghdad Sci J. 2021; 18(2(Suppl.)): 0925-936.
22. Drid K, Allaoui M, Kherfi ML. Object Detector Combination for Increasing Accuracy and Detecting More Overlapping. Objects.Image and Signal Processing. 2020: 290-296. https://doi.org/10.1007%2F978-3-030-51935-3_31.
23. Hassan NF, Abdulrazzaq HI. Pose Invariant Palm Vein Identification System using Convolutional Neural Network. Baghdad Sci J. 2018; 15(4): 0502-509.

Open Access **Baghdad Science Journal** **P-ISSN: 2078-8665**
Published Online First: November 2022 **2023, 20(3): 893-903** **E-ISSN: 2411-7986**

مقارنة بين (Faster R-CNN) و (YOLOv5) للتعرف على الاجسام المتداخلة

محمد منور يسرو روزنيزا علي* محمد سوزوري هيتام

قسم علوم الحاسوب ، جامعة ماليزيا تيرينجانو ، ماليزيا.

الخلاصة:
يعد تصنيف الجسم المتداخل أحد التحديات الرئيسية التي يواجهها الباحثون الذين يعملون في اكتشاف الأشياء والتعرف عليها. معظم الخوارزميات المتاحة التي تم تطويرها قادرة فقط على تصنيف أو التعرف على الأشياء التي تكون إما منفصلة بشكل فردي عن بعضها البعض أوجسم واحد في مشهد (مشاهد) ، ولكن لا تتداخل مع اجسام أدوات المطبخ. في هذا المشروع ، تم اقتراح خوارزميات Faster R-CNN و YOLOv5 لاكتشاف وتصنيف جسم متداخل في منطقة المطبخ. تم تطبيق YOLOv5 و Faster R-CNN على االاجسام المتداخلة حيث من المتوقع أن يتمكن المرشح أو النواة من فصل االجسم المتداخل في الطبقة المخصصة لتطبيق النماذج. تم استخدام قاعدة بيانات الصور المعيارية لأدوات المطبخ وأدوات المطبخ المتداخلة من الإنترنتااجسام مرجعية أساسية. تم تعيين مجموعات التقييم والتدريب / التحقق عند 20٪ و 80٪ على التوالي. قام هذا المشروع بتقييم أداء هذه التقنيات وتحليل نقاط قوتها وسرعاتها بناءً على الدقة والدقة ودرجة.F1 خلصت نتائج التحليل في هذا المشروع إلى أن YOLOv5 ينتج مربعات إحاطة دقيقة بينما يكتشف Faster R-CNN المزيد من االاجسام. في بيئة اختبار مماثلة ، يُظهر YOLOv5 أداءً أفضل من خوارزمية R-CNN الأسرع. بعد التشغيل في نفس البيئة، حصل هذا المشروع على دقة 0.8912 (89.12٪) لـ YOLOv5 و 0.8392 (83.92٪) لـ Faster R-CNN ، بينما كانت قيمة الخسارة 0.1852 لـ YOLOv5 و 0.2166 لأسرع .R-CNN تعد المقارنة بين هاتين الطريقتين هي الأكثر حداثة ولم يتم تطبيقها مطلقًا في الكائنات المتداخلة وخاصة أدوات المطبخ.

الكلمات المفتاحية: رؤية الكمبيوتر، الشبكة العصبية التلافيفية، سرعة r-cnn ، أدوات المطبخ ، التعرف على الأشياء المتداخلة ، Yolo