COMPARATIVE PERFORMANCE ANALYSIS OF SINGLE-SHOT DETECTOR AND FASTER R-CNN FOR OBJECT DETECTION

Jie Zhao and Meng Su

School of Engineering, Penn State University, The Behrend College

ABSTRACT

Object detection is a pivotal technology in computer vision that detects multi-class objects with their localizations in an image. It can untangle the enigma of complicated scenes in the real world. Two main algorithms for implementing object detection are Single-Shot Detector (SSD) and Faster RCNN, which have unique structures of deep learning neural networks. This study compares two prominent object detection algorithms: SSD and Faster R-CNN, focusing on Intersection over Union (IoU) thresholds and runtime efficiency. Using COCO data sets with the validation of 2017, we evaluate the bounding box localization and recognition accuracy of both algorithms. By analyzing *IoU thresholds* and time efficiency, our findings offer insights into selecting optimized algorithms for different object detection tasks.

KEYWORDS

Object Detection, Single Shot Detector, Faster RCNN, Deep Learning, Intersection of Union (IoU), LSVRC (Large Scale Visual Recognition Challenge), COCO Datasets

1. INTRODUCTION

1.1. Differences Between Image Recognition and Object Detection

Image recognition and object detection are key computer vision components with distinct goals and applications. Image recognition focuses on identifying a specific object or class within an image, typically providing a label for the entire image [1].

For instance, an image might be classified as depicting a "cat" or a "dog." In contrast, object detection not only identifies multiple objects within a single image but also pinpoints their exact locations. This is typically accomplished by drawing bounding boxes around each detected object, enabling the analysis of complex scenes with several elements.

The key challenge of object detection lies in its ability to not only classify multiple objects but also accurately localize them. While image recognition is concerned with assigning a label, object detection requires both classification and spatial information, making it more complex. The ability to handle varying object sizes, multiple objects, and occlusions (where objects overlap or are partially hidden) is critical in object detection, bringing this task closer to human-level visual understanding [2].

David C. Wyld et al. (Eds): NWCOM, ASOFT, ITCCMA, CSITY, BDIoT, MLNLP, SIGPRO, AIFZ– 2024
pp. 209-222, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.141919

1.2. Challenging Evaluation Metrics for Object Detection

Evaluating object detection models is inherently more complex than image recognition. In image recognition, success is determined by matching the predicted label to the ground truth. Object detection, however, requires additional precision in the form of localization. It is not enough for a model to predict the correct class label; it must also place a bounding box around each object in the image with a high degree of accuracy [3,4,5].

One example of this challenge is in a scene with five zebras. While a model may correctly identify "zebra" as the class label, if the bounding boxes do not accurately correspond to the location of the zebras, the detection is considered imprecise. Therefore, both class accuracy and spatial accuracy are critical for robust object detection.

Adding another layer of complexity is the need for exact annotations of an object's size and position, which can be challenging to estimate visually. To address this, metrics like Intersection over Union (IoU) are utilized to quantify the overlap between the predicted bounding box and the ground truth, providing a clear measure of localization accuracy.

2. RELATED WORK

In recent years, the field of object detection has advanced significantly, leading to the development of various models designed to enhance accuracy, speed, and applicability across different tasks. In this section, we explore two prominent models, Single Shot Detector (SSD) and Faster R-CNN, which represent the two main approaches to object detection: single-stage and multi-stage pipelines. Additionally, we discuss the importance of utilizing the COCO dataset for training and evaluating these models, as well as the metrics employed to compare their performance.

2.1. Single Shot Detector (SSD)

The Single Shot Detector (SSD) is an object detection model that employs a single-stage architecture. This approach enables the model to perform both object classification and localization simultaneously in one forward pass through the neural network. It is faster than multi-stage models because it avoids the need for a region proposal stage. SSD employs a base network, typically VGG16, which acts as a feature extractor. On top of this, SSD adds several convolutional layers to detect objects at different scales. These layers predict the bounding boxes and class scores for each object directly from the feature maps [6,7,8,9,10,11].

2.1.1. Performance and Applications

SSD is widely known for its balance between speed and accuracy. Thanks to its architectural design, the SSD model is capable of processing images in real time. This makes it ideal for applications that demand rapid detection, such as autonomous driving, surveillance systems, and live video analysis. However, SSD tends to struggle with detecting small objects due to its single-stage structure, where smaller objects may not be detected in lower-resolution feature maps [6,7,8,9,10,11].

2.1.2. Why Pytorch?

The implementation of SSD in PyTorch is advantageous due to the framework’s flexibility and strong support for dynamic computation graphs. PyTorch is widely adopted for research and development because of its ease of use, active community support, and seamless integration with GPU acceleration, which is essential for training large models like SSD on datasets such as COCO.

2.2. Faster-RCNN

Faster R-CNN is a multi-stage object detection model that builds on the earlier R-CNN and Fast R-CNN architectures. A significant innovation in Faster R-CNN is the integration of the Region Proposal Network (RPN). In the initial stage, the RPN generates a set of candidate object proposals, which are then used for further processing. The second stage of Faster R-CNN processes these proposals using a convolutional network to classify objects and refine their bounding boxes. This two-stage process allows for more precise object detection, especially for smaller objects and crowded scenes [12,13,14,15].

2.2.1. Performance and Applications

While Faster R-CNN is slower than SSD, it offers superior accuracy, particularly in cases where fine-grained localization is required. This makes it well-suited for tasks where accuracy is more important than speed, such as medical image analysis, wildlife monitoring, and detailed video analytics. The multi-stage approach enables the model to handle complex scenes with overlapping objects, ensuring higher precision even in challenging environments [12,13,14,15].

2.2.2. Comparison of SSD and Faster-RCNN

The key difference between SSD and Faster R-CNN lies in their architecture. SSD is a single-stage detector, which optimizes speed by detecting objects directly from feature maps without a region proposal step. This makes SSD more efficient but less accurate for small objects and crowded scenes. In contrast, Faster R-CNN is a two-stage detector, first proposing regions and then refining the detection. This results in slower processing but higher accuracy, especially when dealing with complex images or small objects.

2.3. COCO Datasets

The COCO (Common Objects in Context) dataset is a widely used benchmark for object detection, segmentation, and captioning tasks. The COCO 2017 dataset is particularly important for training and evaluating models like SSD and Faster R-CNN due to its diversity and complexity. The dataset includes [16]:

- 118,000 training images
- 5,000 validation images
- 41,000 test images

Each image is annotated with objects from 91 categories, such as "person," "vehicle," and "animal." These categories are further grouped into 12 super-categories, including "furniture," "appliance," and "sports" [16].

2.3.1. Comparison Between COCO Versions

COCO provides several versions (2014, 2015, 2017), with the 2017 version being widely used in current research due to its updated annotations and larger dataset size. The variety of objects, including small, medium, and large objects, as well as the crowded scenes, make COCO an excellent dataset for evaluating object detection models in diverse and realistic contexts [16].

2.3.2. Importance of COCO for Model Evaluation

COCO's detailed annotations, which include bounding boxes, segmentation masks, and keypoints, allow for precise evaluation of both object localization and classification. This comprehensive dataset ensures that models are tested on challenging real-world scenarios, making it a gold standard for object detection research.

2.4. Object Detection Metrics (LSVRC – Large Scale Visual Recognition Challenge)

Two primary metrics are used to evaluate object detection models: runtime efficiency and recognition accuracy [17,18].

2.4.1. Runtime Efficiency

Runtime efficiency evaluates how quickly the model can process images and detect objects. For real-time applications, such as autonomous driving, faster models like SSD are preferable because they can process more frames per second (FPS) compared to models like Faster R-CNN, which are slower due to their multi-stage processing [17,18].

2.4.2. Recognition Accuracy

Recognition accuracy is measured using metrics like Intersection over Union (IoU), which calculates the overlap between predicted bounding boxes and the ground-truth annotations. Higher IoU thresholds indicate more precise localization. While Faster R-CNN typically achieves higher IoU scores, SSD may struggle with small objects due to its single-stage approach. For multi-object detection, achieving high IoU thresholds across multiple objects remains challenging, as it requires precise localization of each object in the image [17,18].

2.4.3. Application to SSD and Faster R-CNN

Both SSD and Faster R-CNN are evaluated based on a balance between speed and accuracy. SSD excels in scenarios requiring real-time detection, where small trade-offs in accuracy are acceptable. With its more intricate architecture, Faster R-CNN is particularly well-suited for tasks where accuracy is of utmost importance, such as detecting small objects or identifying items in crowded environments. These trade-offs between speed and accuracy should be carefully considered when choosing the most appropriate model for specific tasks.

3. METHODOLOGIES

3.1. Object Detection Metric and the IoU Measurements

Object detection metrics serve to figure out two following relations, which are shown below:

- Each bounding box should be mapped to the related class label.

- The bounding box with the class label in ground truth annotation should be mapped to the related one in the model's prediction.

The intersection of union (IoU) is the basis for determining two relations. The IoU is derived from the Jaccard index, which measures the correlation between two areas. The correlation between the two trajectory areas is converted to an overlapping area between the vicinity. The bigger the value of correlation, the larger the overlapping area. The extreme case is if trajectory areas resemble each other, the value of IoU is 1. The other extreme case is that if two areas are independent of each other, the value of IoU is 0. Hence, the range of values in IoU is between 0 and 1 [19,20,21,22,23,24,25].

In turn, the IoU equation between the ground truth annotation and the model's prediction is shown below [19,20]:

$$IoU = \frac{Area_{Intersection}}{Area_{union}}$$

$$IoU = \frac{Area_{Intersection}}{Area_{ground\_truth} + Area_{prediction} - Area_{Intersection}}$$

Even though the constant threshold of IoU is critical to evaluating recognition accuracy in object detection, specific IoU values vary in different literatures. In [26], the constant thresholds of IoU are 0.26, 0.27, 0.49, 0.59, 0.65 and 0.66. In [27], the constant thresholds of IoU are 0 and 0.125. In [28], the constant thresholds of IoU are 0.38, 0.56, 0.67 and 0.76. However, we will propose the suggested constant thresholds for the explanations in the experiment.

3.2. Case Studies of TP, FP, FN, and TN

Another four important indexes of object detection metrics are TP, FP, FN, and TN, with the definition shown below [19,20]:

- TP: Truth Positive
- FP: False Positive
- FN: False Negative
- TN: Truth Negative

Next, by using COCO data sets, four metric indexes are redefined [19,20]:

- TP: The set bounding box makes the overlapping area greater than the iou threshold, and the predictive class label is the same as the COCO annotation.

- FP: The set bounding box makes the overlapping area less than the iou threshold, even though the predictive class label is the same as the COCO annotation.

- FN: the set bounding box in prediction is independent of the bounding box in ground truth, no matter whether the class labels between the prediction and the COCO annotation are the same.

- TN: The set bounding box makes the overlapping area greater than the iou threshold, but the predictive class label differs from the COCO annotation.

As an illustration, the image in the COCO 2017 validation, "000000000885.jpg" [16], is deployed to explain the four redefined metrics, shown in Fig.1-4.

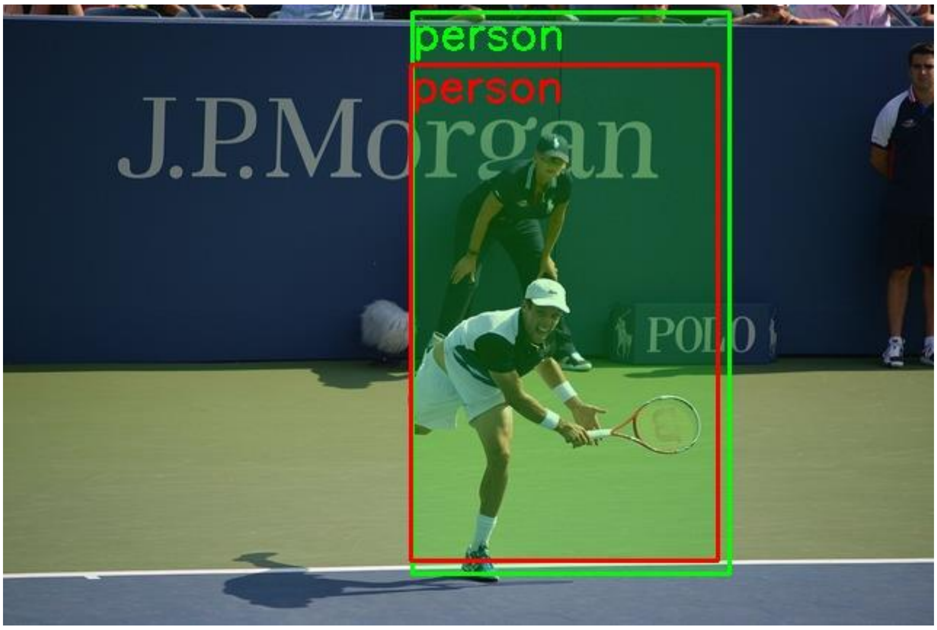

Fig.1 The Case of TP with COCO Image, “000000000885.jpg”

In Fig.1, the red bounding box in the model’s prediction fits well with the COCO ground truth in green. Meanwhile, the red label annotation in the model’s prediction, “person”, is the same as the COCO annotation in green.

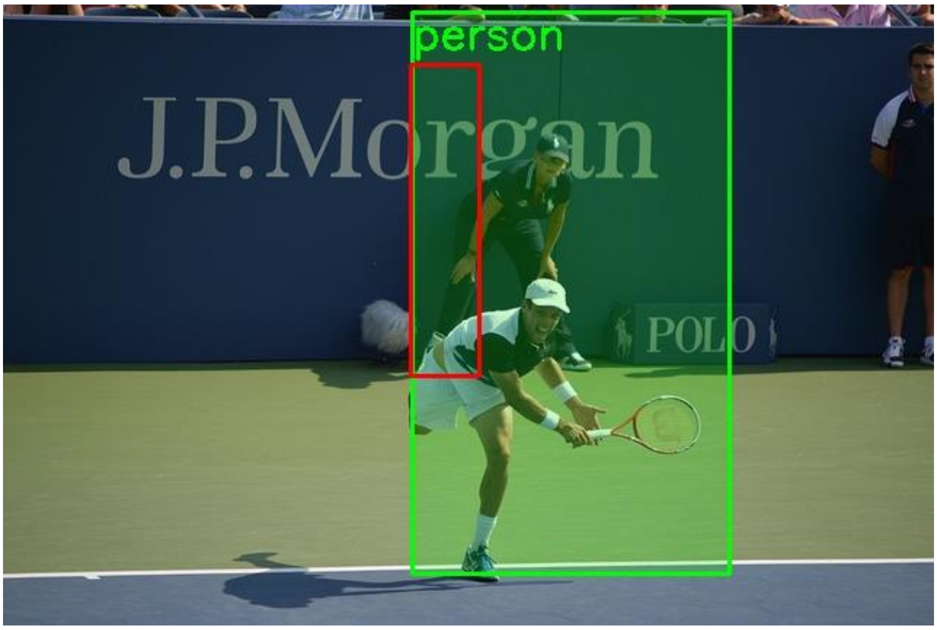

Fig.2 The Case of FP with COCO Image, “000000000885.jpg”

In Fig.2, the red bounding box in the model’s prediction is not big enough to cover the COCO ground truth in green, even though the red label annotation in the model’s prediction, “person”, is the same as the COCO annotation in green. This means that the performance of the model’s prediction is weak.

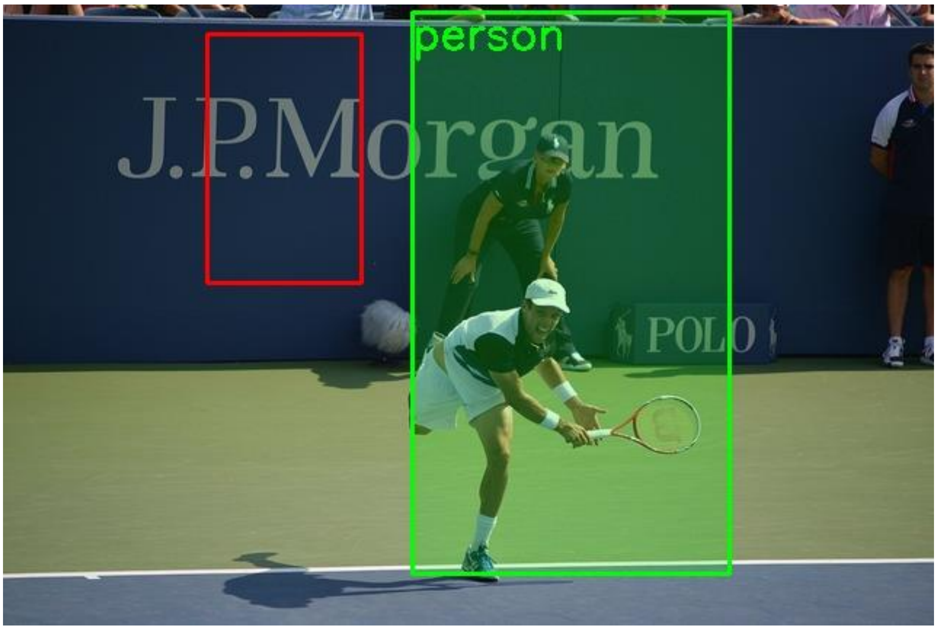

Fig.3 The Case of FN with COCO Image, “000000000885.jpg”

Fig.3 shows no overlapping area between the red bounding box in the model’s prediction and the COCO ground truth in green. There is no correlation between the prediction and the ground truth, no matter whether the red label annotation in the model’s prediction, “person,” is the same as the COCO annotation in green. In this case, we do not need to consider the iou.

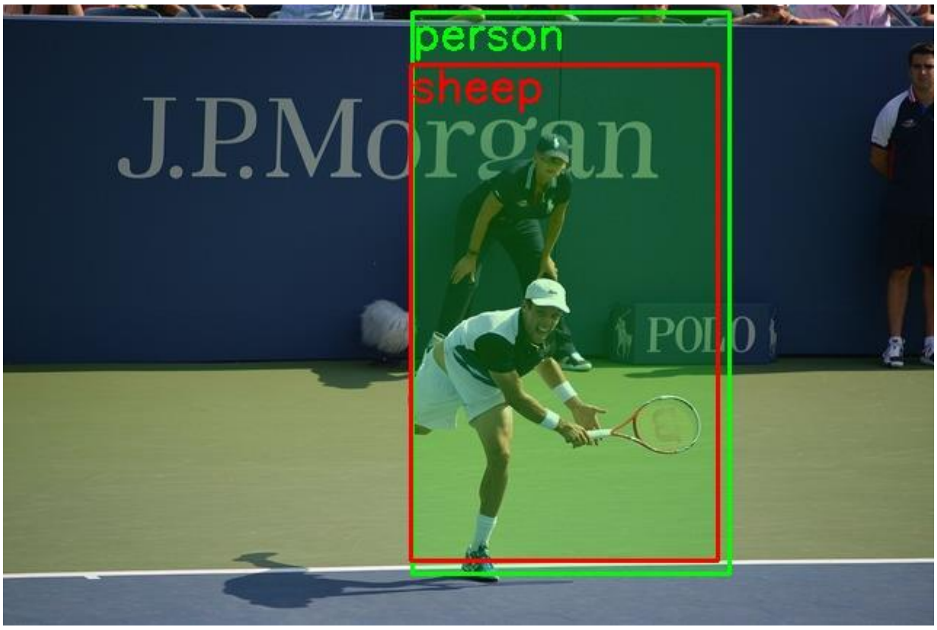

Fig.4 The Case of TN with COCO Image, “000000000885.jpg”

In Fig.4, although the red bounding box in the model’s prediction fits well with the COCO ground truth in green, as the red label annotation in the model’s prediction, “sheep,” is different from the COCO annotation in green, “person,” it loses its meaning when calculating the iou.

3.3. Proposed Recognition Accuracy

The recognition accuracy is proposed and measured, which is shown below.

3.3.1. Proposed_AP(Average Precision)

Given an object in the COCO data set, the total number of bounding boxes in the model's prediction is related to average recognition precision if the IoU is greater than the IoU threshold. The IoU threshold, as a constant between 0 and 1, is empirical data that determines the object detection metrics [19,20].

3.3.2. Proposed_mAP (mean Average Precision)

Based on the proposed_AP, the equation of the proposed mAP is shown below [19,20]:

$$mAP_{proposed} = \frac{\sum num\ of\ bounding\ boxes\ in\ model's\ prediction}{\sum num\ of\ bounding\ boxes\ in\ ground\ truth\ annotation}$$

3.4. Pseudo Algorithm for Single Shot Detector

The pseudo-code for object detection is shown below in Tab.1:

Table 1. The pseudo-code for Single Shot Detector.

```
○ Import libraries to support object detection
○ Get a pre-trained model of Single Shot Detector
○ Initialize class label, iou_threshold, bbox_groundTruth, bbox_prediction,
  counter_groundTruth and counter_prediction
○ Outer loop in different sizes of images
    • Fetch data frame of ground truth annotation in COCO_2017 validation
    • Fetch the total number of bounding boxes per image in ground truth annotation
    • Fetch the total number of bounding boxes per image per class object in ground
      truth annotation
○ Inner loop in the total number of bounding boxes per image in ground truth
    • Get the same number of bounding boxes in the model's prediction
○ The 3rd loop in the total number of bounding boxes per image per class object in
  ground truth
        • Compare class labels between the ground truth annotation and the model's
          prediction
        • If the comparison is the same, calculate the iou between the ground truth
          and the model's prediction
        • If iou is over the iou threshold, counter_prediction increments
○ End loop
○ End loop
○ End loop
○ Calculate the recognition accuracy
  $mAP_{proposed} = \frac{\sum num\ of\ bounding\ boxes\ in\ model's\ prediction}{\sum num\ of\ bounding\ boxes\ in\ ground\ truth\ annotation}$
```

3.5. Pseudo Algorithm for Faster-RCNN

The pseudo-code for Faster RCNN is similar to that for Single Shot Detector. The difference is, however, in the pseudo-code for Faster RCNN, inside the "Outer loop in different sizes of images", the operation of "Fetch the total number of bounding boxes per image per class object in ground truth annotation" is replaced with the operation of using a threshold to get the total number of bounding boxes in the model's prediction. In turn, the inner loop is repeated based on the total number of bounding boxes in the model's prediction, rather than the total number of bounding boxes per image in ground truth.

4. EXPERIMENTS

4.1. Parameters in the Pre-trained Models

The experiment is conducted to leverage the evaluation of object detection between Single-Shot Detector and Faster RCNN. To implement the experiment, two pre-trained models of Single-Shot Detector and Faster RCNN are utilized with the support of Pytorch. As SSD is based on VGG16 with the structure of scale jittering, different scales used for training the SSD are [0.07, 0.15, 0.33, 0.51, 0.69, 0.87, 1.05]. In addition, the total number of parameters, including hyper-parameters, is 35641826. On the other hand, Faster RCNN is based on the Resnet50 as its basic structure, and the total number of parameters used for training the model is 41755286.

4.2. Constant IoU thresholds and COCO 2017 validation

Using COCO data with the 2017 validation as image sets, several constant thresholds of the IoU are trialed for object detection performance. The constant thresholds of the IoU are selected from the evenly normal distribution. Hence, the constant thresholds of the IoU used in the experiment are 0.25, 0.5, 0.75, and 0.9. Besides, eight class categories with high occurrences in COCO data sets are selected for object detection evaluation. The eight class categories in COCO data sets are "person", "car", "horse", "tennis racket", "cup", "pizza", "dining table", and "laptop".

Besides, the configurations of implementing both SSD and Fatser-RCNN for object detection, are the graphic card - RTX 3060, and GPU usage for running the pre-trained model.

5. EXPERIMENTAL RESULTS AND ANALYSES

The experiment results and analyses are shown in Tab.2-5.

5.1. The Experiments with Constant Threshold – 0.25

The experiments with a constant threshold – 0.25 are conducted using both SSD and Faster-RCNN. The results are shown in Tab.2.

In Tab.2, our findings are shown below:

- For running time efficiency, with the unit-second, different categories in Single Shot Detector are in Column Three, and Faster RCNN are in Column Six. On average, the Single Shot Detector is quicker seven times than Faster RCNN.

- In each algorithm, given the class category and the total number of per-class category bounding boxes in ground truth, recognition accuracy results from the total number of bounding boxes in prediction divided by the bounding boxes in the ground truth.

- The red and bold row of the bounding box number of prediction indicates the principle of the pre-trained model is not the case that will occur in the real applications, because the bounding box number of prediction must not be beyond the bounding box number of COCO annotation as the maximum. Therefore, if the bounding box number of prediction is red and bold, the recognition accuracy is not taken into account.

- With the iou_threshold - 0.25, Single Shot Detector outperforms Faster RCNN because Single Shot Detector has fewer rows with red and bold digits than Faster RCNN. However, the row with red and bold digits, which violates the regulation between prediction and the ground truth, indicates incorrect prediction or inappropriate bounding boxes in the prediction. As a result, the Single Shot Detector with fewer rows of red and bold digits is superior to Faster RCNN when iou_threshold is 0.25.

- Consider the cause of the row with the red and bold digit where the total number of bounding boxes in prediction is greater than that in the ground truth annotation. The small value of the iou threshold as a gatekeeper will not be selected out but filtered in a lot of bounding boxes as the model's prediction, where there may include the bounding boxes the ground truth annotation judges as FP. The bounding box's combination of TP and FP in prediction would be greater than sorely TP in ground truth annotation.

Table 2. Performance comparison between SSD and Faster-RCNN with constant threshold - 0.25.

Class Category	**bbox of ground truth**	**Single Shot Detector**			**Faster RCNN**		
		running Time (second)	**bbox Of predict**	**recognition accuracy (%)**	**running Time (second)**	**bbox Of predict**	**recognition accuracy (%)**
person	11004	1117.65	**24383**	**X**	7540.19	**30626**	**X**
car	1932	1039.78	**2546**	**X**	7462.88	**3429**	**X**
horse	273	1031.48	**358**	**X**	7676.19	**425**	**X**
tennis racket	225	1028.81	147	65.33	7367.42	**241**	**X**
cup	899	1039.59	600	66.74	7001.41	**1222**	**X**
pizza	285	1033.72	214	75.09	6988.11	86	30.18
dining table	697	1040.03	532	76.33	6994.75	**713**	**X**
laptop	231	1031.83	160	69.26	9474.77	**244**	**X**

5.2. The Experiments with Constant Threshold – 0.5

The experiments with a constant threshold – 0.5 are conducted using both SSD and Faster-RCNN. The results are shown in Tab.3.

In Tab.3, our findings are shown below:

- Compared with Tab.1, Tab. 2 has fewer red and bold rows, indicating better performance in iou_threshold with 0.5.

- For time efficiency, approximately, running SSD for object detection is seven times quicker than Faster-RCNN.

- As to recognition accuracy, given the class objects – "car," "horse," "dining table," and "laptop," the principle of SSD resembles Faster-RCNN, because their difference in recognition accuracy is no greater than 10%. Given the class objects – “tennis racket” and “cup”, SSD is inferior to Faster-RCNN, because their difference in recognition accuracy is at least greater than 20%. Given the class objects – “pizza”, SSD is superior to Faster-RCNN, because their difference in recognition accuracy is at least greater than 40%.

- Consider the cause of the row with the red and bold digit where the total number of bounding boxes in prediction is greater than that in-ground truth annotation. For object detection, multi-class categories’ small areas may cause the wrong FP bounding boxes. If multiple class categories exist in an image, all areas of the bounding box of prediction, the bounding box of ground truth, and their IoU (if their intersection is not zero) will be very small and sensitive. This means that a small variation from one iou threshold to another will make a huge difference in the ratio of TP bounding boxes to the FP in the model’s prediction, which will cause imprecise recognition accuracy.

Tab.3: performance comparison between SSD and Faster-RCNN with constant threshold - 0.5.

Class Category	**bbox of ground truth**	**Single Shot Detector**			**Faster RCNN**		
		running Time (second)	**bbox Of predict**	**recognition accuracy (%)**	**running Time (second)**	**bbox Of predict**	**recognition accuracy (%)**
person	11004	1094.6	**15752**	**X**	7297.72	**16339**	**X**
car	1932	1041.43	1585	82.04	7115.42	1784	92.34
horse	273	1035.47	270	98.9	7077.43	246	90.11
tennis racket	225	1034.47	115	51.11	7388.36	158	70.22
cup	899	1039.74	472	52.5	6917.97	729	81.09
pizza	285	1032.74	157	55.09	6885.71	38	13.33
dining table	697	1049.46	337	48.35	6907.61	273	39.17
laptop	231	1037.31	122	52.81	6867.72	151	65.37

5.3. The Experiments with Constant Threshold – 0.75

The experiments with a constant threshold – 0.75 are conducted using both SSD and Faster-RCNN. The results are shown in Tab.4.

Tab.4: performance comparison between SSD and Faster-RCNN with constant threshold - 0.75.

Class Category	**bbox of ground truth**	**Single Shot Detector**			**Faster RCNN**		
		running Time (second)	**bbox Of predict**	**recognition accuracy (%)**	**running Time (second)**	**bbox Of predict**	**recognition accuracy (%)**
person	11004	1170.71	10835	98.46	7068.67	10138	92.13
car	1932	1032.27	1030	53.31	6816.29	1016	52.59
horse	273	1026.33	190	69.6	6789.46	170	62.27
tennis racket	225	1044	68	30.22	8771	104	46.22
cup	899	1061.03	346	38.49	6790.68	476	52.95
pizza	285	1034.4	125	43.86	6824.04	23	8.07
dining table	697	1035.14	212	30.42	6803.06	122	17.5
laptop	231	1043.53	105	45.45	6767.64	110	47.62

In Tab.4, our findings are shown below:

- As there is no red and bold row, the constant iou_threshold - 0.75, can guarantee recognition accuracy, avoiding imprecise cases.

- For time efficiency, running SSD for object detection is probably seven times quicker than Faster-RCNN.

- As to recognition accuracy, given the class objects – "person", "car", "horse" and "laptop", the principle of SSD resembles Faster-RCNN, because their difference of recognition accuracy is no greater than 10%. Given the class objects – "tennis racket" and "cup", SSD is inferior to Faster-RCNN, because their difference in recognition accuracy is at least greater than 10%. Given the class objects – "pizza" and "dining table", SSD is superior to Faster-RCNN, because their difference in recognition accuracy is at least greater than 10%.

5.4. The Experiments with Constant Threshold – 0.9

The experiments with a constant threshold – 0.9 are conducted using both SSD and Faster-RCNN. The results are shown in Tab.5.

Tab.5: performance comparison between SSD and Faster-RCNN with constant threshold - 0.9.

Class Category	**bbox of ground truth**	**Single Shot Detector**			**Faster RCNN**		
		running Time (second)	**bbox Of predict**	**recognition accuracy (%)**	**running Time (second)**	**bbox Of predict**	**recognition accuracy (%)**
person	11004	1116.82	7627	69.31	8429.98	5762	52.36
car	1932	1011.38	668	34.58	7431.84	518	26.81
horse	273	1015.4	112	41.03	6914.48	74	27.11
tennis racket	225	927.07	27	12	7017.67	35	15.56
cup	899	948.39	210	23.36	7888.9	246	27.36
pizza	285	940.65	72	25.26	7677.99	4	1.4
dining table	697	926.87	122	17.5	6768.92	56	8.03
laptop	231	945.24	38	16.45	6728.89	53	22.94

In Tab.5, our findings are shown below:

- As there is no red and bold row, the constant iou_threshold – 0.9 can guarantee recognition accuracy, with avoidance of imprecise cases.

- For time efficiency, running SSD for object detection is statistically seven times quicker than Faster-RCNN.

- As to recognition accuracy, given the class objects – "car", "tennis racket", "cup", "dining table" and "laptop", the principle of SSD resembles Faster-RCNN, because their difference in recognition accuracy is no greater than 10%. Given the class objects – "person", "horse" and "pizza", SSD is superior to Faster-RCNN, because their difference in recognition accuracy is at least greater than 10%.

- Compared with Tab.2-4, the recognition accuracy of both algorithms in Tab.5 is lower. This means that, when iou_threshold increases, the recognition accuracy of both algorithms decreases.

CONCLUSIONS

The conclusions are drawn, which are shown below:

- The advantage of choosing the constant iou_thresholds is simplifying the computation in implementing both algorithms.

- The selection of constant thresholds is tricky for object detection performance, because the iou_threshold determines both performance evaluation stability and recognition accuracy. In the first placement, the small value of the iou_threshold, which filters in more predictive cases, will intrigue the unstabilized performance evaluation because the predictive cases may include both TP and FP cases. In the second placement, the large iou_threshold will ensure stability in the performance evaluation. Nonetheless, it narrows down the possibility of recognition accuracy because the iou_threshold is inversely proportional to recognition accuracy.

- For object detection, using COCO 2017 validation, SSD is proven to be more optimized than Faster-RCNN. It is because that SSD is testified to be more efficient than Faster-RCNN and that SSD provides more performance evaluation stability than Faster-RCNN.

- When the data sets are scalable, like COCO data with 2017 training sets, the question of which algorithm, either SSN or Faster-RCNN, is optimized for object detection still awaits the answer.

REFERENCES

[1] Zhao, Jie, and Meng Su (2024) "An Evaluation of Neural Network Efficacies for Image Recognition on Edge-AI Computer Vision Platform." International Journal of Electrical and Computer Engineering 18.1 10-15.

[2] Jason Brownlee (2019) “A Gentle Introduction to Object Recognition With Deep Learning”, https://machinelearningmastery.com/object-recognition-with-deep-learning.

[3] Padilla, Rafael, Sergio L. Netto, and Eduardo AB Da Silva (2020) "A survey on performance metrics for object-detection algorithms." 2020 international conference on systems, signals and image processing (IWSSIP). IEEE.

[4] Sanchez, S. A., H. J. Romero, and A. D. Morales (2020) "A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework." IOP conference series: materials science and engineering. Vol. 844. No. 1. IOP Publishing.

[5] Oksuz, Kemal, et al (2018) "Localization recall precision (LRP): A new performance metric for object detection." Proceedings of the European conference on computer vision (ECCV).

[6] Liu, Wei, et al (2016) "Ssd: Single shot multibox detector." Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing.

[7] Simonyan, Karen, and Andrew Zisserman (2014) "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556.

[8] Howard, Andrew G (2017) "MobileNets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861.

[9] Sandler, Mark, et al (2018) "Mobilenetv2: Inverted residuals and linear bottlenecks." Proceedings of the IEEE conference on computer vision and pattern recognition.

[10] Howard, Andrew, et al (2019) "Searching for mobilenetv3." Proceedings of the IEEE/CVF international conference on computer vision.

[11] Redmon, J. (2016) "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition.

[12] He, Kaiming, et al (2016) "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition.

[13] Girshick, Ross, et al (2014) "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition.

[14] Girshick, Ross (2015) "Fast r-cnn." Proceedings of the IEEE international conference on computer vision.

[15] Ren, Shaoqing, et al (2016) "Faster R-CNN: Towards real-time object detection with region proposal networks." IEEE transactions on pattern analysis and machine intelligence 39.6: 1137-1149.

[16] Lin, Tsung-Yi, et al (2014) "Microsoft coco: Common objects in context." Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer International Publishing.

[17] Russakovsky, Olga, et al (2015) "Imagenet large scale visual recognition challenge." International journal of computer vision 115: 211-252.

[18] Everingham, Mark, et al (2010) "The pascal visual object classes (voc) challenge." International journal of computer vision 88: 303-338.

[19] Erhan, Dumitru, et al (2014) "Scalable object detection using deep neural networks." Proceedings of the IEEE conference on computer vision and pattern recognition.

[20] Vedaldi, Andrea, et al., eds (2020) Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II. Vol. 12347. Springer Nature.

[21] Qin, Zekui, et al (2019) "Advanced intersection over union loss for visual tracking." 2019 Chinese Automation Congress (CAC). IEEE.

[22] Tong, Chenghao, et al (2022) "NGIoU Loss: Generalized intersection over union loss based on a new bounding box regression." Applied Sciences 12.24: 12785.

[23] Wang, Xufei, and Jeongyoung Song (2021) "ICIoU: Improved loss based on complete intersection over union for bounding box regression." IEEE Access 9: 105686-105695.

[24] Gajjar, Amitkumar N., and Jigneshkumar Jethva (2022) "Intersection over Union based analysis of Image detection/segmentation using CNN model." 2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T). IEEE.

[25] Chen, Peng, et al (2021) "Shape similarity intersection-over-union loss hybrid model for detection of synthetic aperture radar small ship objects in complex scenes." IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 14: 9518-9529.

[26] Rezatofighi, Hamid, et al (2019) "Generalized intersection over union: A metric and a loss for bounding box regression." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.

[27] Hou, Feifei, et al (2021) "Improved Mask R-CNN with distance guided intersection over union for GPR signature detection and segmentation." Automation in Construction 121: 103414.

[28] Huang, Zhihui, et al (2022) "A multivariate intersection over union of SiamRPN network for visual tracking." The Visual Computer 38.8: 2739-2750.

AUTHORS

Jie Zhao is an assistant professor of Computer Science and Software Engineering at Penn State University. She received her PhD in ECE at Texas Tech University in 2019 and worked as a Postdoc at Texas A&M University the next year. Her research interests include Computer Vision with Artificial Intelligence and Big Data Analytics with Machine Learning.

Meng Su is an associate professor of Computer Science and Software Engineering at Penn State University. He holds a B.S. and M.S. from Nanjing University and Ph.D. from Southern Illinois University at Carbondale. His research interests encompass manifold learning, diffusion equations, and applications in data science.