Comparative Analysis of Classification and Segmentation Performance of Different Dog Breeds Based on Mask R-CNN

Youtian Luo, Dan Li

Jincheng College, Sichuan University, Chengdu 611731, Sichuan, China

Abstract: ***The classification of different image data is a common task in real life, which has been achieved as early as convolutional neural networks. With the emergence of R-CNN (Region CNN), simple classification tasks began to develop towards object detection tasks, followed by various segmentation tasks. After Faster R-CNN [4], Mask R-CNN [3] achieved instance segmentation while detecting objects. This algorithm model needs to distinguish different target objects in the image panel and learn a large number of characteristics to represent the details of each target object. In dog data images, there are differences and similarities in the characteristics of dog species, such as fur color, texture, and body shape. So in this article, the author selected and created two different dog datasets, representing targets with significant differences and similarities in features. Compare and analyze the performance of Mask R-CNN object detection and instance segmentation using these two dog datasets.***

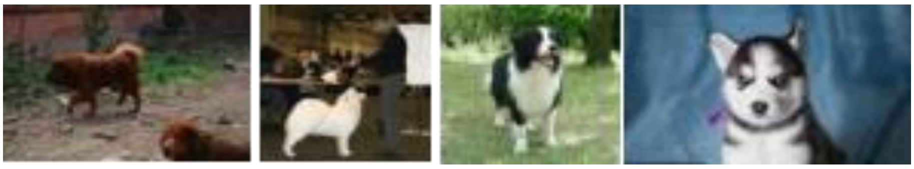
Keywords: Mask R-CNN; Classification; Division; Contrast.

1. INTRODUCTION

The beginning of object detection task is classification task. The ordinary convolutional neural network CNN has significant advantages in feature extraction, so it was originally used for image classification. RCNN introduces the concept of region and combines it with CNN, successfully applying it to object detection problems. Instance segmentation requires correctly identifying all objects of different categories in a graph and segmenting them one by one. Mask R-CNN is based on faster R-CNN and introduces a feature pyramid network [1-3] to extract candidate regions (Region Proposals) and construct a Mask branch parallel to the prediction (category judgment and border offset) branch. It detects the location of the target in the image and predicts a class aware Mask through the Mask branch. This method can extract targets more accurately and has achieved great success in instance segmentation.

To analyze the performance of Mask R-CNN in classification and segmentation on different datasets. In this article, the author selected two datasets of different breeds of dogs, some of which were sourced from the Dogs Dataset at Stanford University [4-6]. The author integrated data from two dissimilar breeds (with significant differences in body size and fur color) and two similar breeds (with high similarity in body size and fur color) in these categories, and expanded the datasets to about 200 images each. These two datasets represent image sets with significant differences in features and image sets with small differences in features. Based on the previously proposed Mask R-CNN, the author used these two datasets for training and compared the classification and segmentation performance of Mask R-CNN on both datasets using the obtained results [7-10].

After experimental comparison, it was found that Mask R-CNN has strong robustness in classification and segmentation performance when dealing with similar feature targets [11-13]. Wu, Z. (2024). presents an innovative integration of the REEGWO algorithm with CNNs and BiLSTM networks, enhancing deep learning model optimization, which can be applied to other areas requiring improved hyperparameter tuning and sequential data prediction [14].



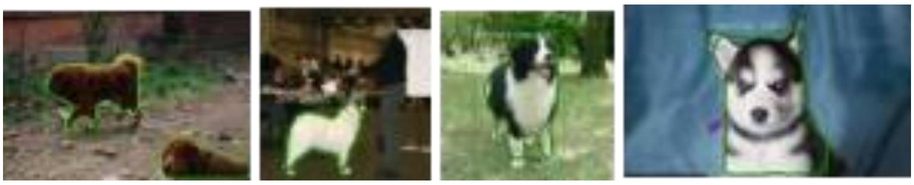


Figure 1: Classification and segmentation results of dogs

Hardware environment: Due to the considerable computational complexity of Mask R-CNN, in the original text, 8 Nvidia Tesla M40 GPUs were used for training the coco dataset, which requires high hardware facilities. Therefore, the author deployed Mask R-CNN on a GeForce RTX 2080 graphics card server to meet the basic computing power requirements of the algorithm [15].

Software environment: Based on the Ubuntu system, the algorithm code is implemented using the pytorch framework, and the corresponding pytorch version and dependencies of CUDA are installed. In addition, due to the use of the Coco dataset format in the production of the dataset, corresponding Coco processing modules need to be installed [16-17].

1.2 Dataset Production

In the production of the dataset, some image data were selected from Stanford University's Dogs Dataset, and the dataset was expanded to around 200 images to increase the image data volume. Using labelme annotation software, manually annotate the images into two categories for each dataset. The datasets with significantly different species features are labeled as Tibetan Mastiff and Samoyed, respectively, referred to as TM&S in the following text. The datasets with similar species features are labeled as Border Collie and Malamute, respectively, referred to as BC&M. After labeling each image, a one-to-one JSON file is generated and integrated. The ratio of training set to testing set is 5:1. Finally, the images were saved in the form of the Coco dataset.

1.3 Comparative Analysis of Data

In order to compare the classification and segmentation performance of Mask R-CNN on two datasets, this paper compares the average accuracy of Bounding Box (the region where the target is located, similar to general object detection) and the average accuracy of segmentation (instance segmentation) separately. The number of categories involved in this article is 3 (the two categories to be classified and the background), and the average accuracy refers to the average accuracy of all categories to be classified. During the training process, after a certain number of iterations, record the average accuracy of the two and observe the changes in their average accuracy over time.

Comparison is not limited to the selection of the same backbone network, and different backbone networks can also be selected and the results can be compared and analyzed after adjusting various parameters.

2. MASK R-CNN

Mask R-CNN is based on the structure of Faster R-CNN and follows its ideas. This algorithm is parallel to the prediction (class discrimination and border offset) module of Faster R-CNN, adding a Mask branch consisting of several convolutional layers that can predict the precise location of the target (target mask). So, based on this newly added branch structure, Mask R-CNN can perform instance segmentation while performing traditional object detection.

2.1 Faster R-CNN

Due to the fact that Mask R-CNN adopts the thinking of Faster R-CNN and uses a two-stage construction, some constructions in Faster R-CNN need to be explored.

RPN (Region Proposal Network): The first stage of the model is to extract ROI using RPN. RPN was proposed in Faster RCNN with the goal of improving the extraction speed of candidate boxes. The idea of RPN is to select several candidate anchors (of different scales and widths) on the feature map through certain rules, and then pass these anchors to the classification branch and the an chor box regression branch respectively. Finally, based on the classification probability of the anchors and the set IoU threshold, select the anchors that meet the criteria as RoI.

Prediction branch: This branch is decoupled into two small branches, one for classification and the other for regression correction of detection boxes. These two small branches are both implemented through fully connected layers to fulfill their respective functions.

2.2 ROI ALIGEN

The earlier RoIPool method generally chose the max pooling method, and in the selection and division of candidate regions, two rounds of rounding were performed for ease of operation. This method is rough in downsampling and may produce some small deviations in the position of candidate boxes. For general classification tasks, the determination of detection boxes often has strong robustness to these small deviations. However, in segmentation tasks, the division of target regions is often at the pixel level, and this small deviation may not have an impact on classification, but it is very negative for the segmentation task of Mask branch. RoIAlign was proposed in Mask RCNN to replace the RoIPool operation. RoIAlign upsamples and downsamples pixels at the decimal level, and this method reduces dimensionality without causing bias in the judgment results of the region, meeting the segmentation requirements.

2.3 Mask Branch

The Mask branch is the difference between Mask R-CNN. Based on the structure of Faster R-CNN, a new Mask branch is constructed parallel to the original prediction branch after RoIAlign operation, which can perform instance segmentation tasks. In this branch, the fully convolutional network FCN [5] is used instead of the fully connected layer. For the input RoI of this branch, the fully connected layer is generally used to integrate the target features extracted in the previous structure. Therefore, the fully connected layer can perform overall target discrimination based on all target features, and is commonly used for classification. The function of convolution is to extract the features of the target. The deeper it is, the more representative it is. Therefore, full convolution is used to replace fully connected layers, which is called convolution. Therefore, full convolution can handle segmentation tasks. At the end of full convolution, in order to obtain the target mask, it is necessary to perform deconvolution to the original image size.

2.4 FPN

The Mask branch also adopts the structure of a feature pyramid network [2]. In the experiments conducted in the original text, it was found that the ResNet backbone architecture using FPN has significant improvements in accuracy and speed compared to the general ResNet backbone architecture.

In the entire network, feature maps at different levels have different strengths and weaknesses in semantic information. Shallow feature maps have higher resolution but weaker semantic information, while deep feature maps have the opposite characteristics. Identifying targets of different sizes is a fundamental task in instance segmentation, but deep coarse resolution feature maps cannot provide more details and require shallow high-resolution detail parts. Therefore, the role of FPN is to fuse semantic information from multiple layers for the detection of targets of different sizes.

3. EXPERIMENTS

In this section, we will train Mask R-CNN on datasets with significant differences in feature similarity and compare its performance in classification and segmentation tasks on the two datasets.

3.1 Network Selection and Evaluation Criteria

The author deployed the Mask R-CNN algorithm on a server with GeForce RTX 2080 GPU. In terms of network selection, both datasets use ResNet networks with a depth of 50 as the backbone for training. In this network, FPN is introduced for feature extraction and named ResNet-50-FPN. In addition, the author also attempted to use the ResNet-50-C4 network as the backbone to perform feature extraction on the last convolutional layer of the fourth stage in the ResNet network. For the experimental results, the Coco standard measurement indicators were used for evaluation, including AP, AP50, and AP75, which respectively represent the comprehensive average accuracy, the average accuracy with an IoU threshold of 0.5, and the average accuracy with an IoU threshold of 0.75.

3.2 Experimental Process

Tibetan Mastiff and Samoyed data

This dataset corresponds to data with significant differences in target features. For the ResNet-50-FPN network, the batch size during training is set to 2 images per GPU, and the batch size during testing is also set to 2 images per GPU. After annotating the patterns, each pattern has a height of 600 pixels and a width of 800 pixels. Therefore, in terms of data augmentation, the author set the minimum edge length of the training image input to [8001200] and the maximum edge length to 1333. In this way, the short edges will be randomly scaled at [8001200], while the long edges will be scaled according to a certain proportion, but the maximum will not exceed 1333. The same applies to the setup of the test set. The number of iterations is 20k, and the accuracy is output every 5k iterations. The initial learning rate is 0.00025.

The results showed that training with ResNet-50-FPN network for 20k took 1.25 hours, and the bounding box performance on AP was 68.34, AP50 was 99.72, and AP75 was 79.45. SEGM shows a performance of 79.76 on AP, 97.80 on AP50, and 93.14 on AP75. As shown in Table 1.
For the ResNet-50-C4 network, training 20k with the same parameters took 1.80h, but the effect was not ideal. The learning rate was readjusted to 0.001, and training 20k times took 1.5h. The bounding box performance on AP was 55.00, AP50 was 89.02, and AP75 was 66.32. SEGM shows a performance of 63.30 on AP, 89.80 on AP50, and 76.25 on AP75. As shown in Table 2.

Table 1: Training Results of ResNet-50-FPN Network on TM&S Dataset

ResNet-50-FPN	AP	AP50	AP75
boundingbox	68.34	99.72	79.45
segm	79.76	97.80	93.14

Table 2: Training Results of ResNet-50-C4 Network on TM&S Dataset

ResNet-50-C4	AP	AP50	AP75
boundingbox	55.00	89.02	66.32
segm	63.30	89.80	76.25

Border Collie and Malamute data

This dataset corresponds to data with similar feature differences. Because the proportion of image sizes in this dataset is similar to the former, in order to provide a comparative effect, the same parameters were used for training in the selection of parameters.

The results showed that training with ResNet-50-FPN network for 20k took 1.36h, and the bounding box performance on AP was 62.11, AP50 was 95.38, and AP75 was 72.58. SEGM shows 71.96 on AP, 95.18 on AP50, and 86.94 on AP75. As shown in Table 3.

Table 3: Training Results of ResNet-50-FPN Network on BC&M Dataset

ResNet-50-FPN	AP	AP50	AP75
boundingbox	62.11	95.38	72.58
segm	71.96	95.18	86.94

3.3 Increase the Number of Iterations

Increasing the number of iterations often leads to significant performance improvements. Increase the number of iterations for both datasets to 30k, and at 20k iterations, the learning rate decays to 10% (0.000025). By increasing the number of iterations, the AP performance of the bounding box on the corgi and Samoyed data is 73.24, AP50 is 100.00, and AP75 is 92.72. SEGM shows 86.43 on AP, 100.00 on AP50, and 98.02 on AP75. As shown in Table 4.

On the Border Collie and Malamute data, the AP performance of the bounding box is 70.00, AP50 is 99.88, and AP75 is 83.27. SEGM performs 82.57 on AP, 99.88 on AP50, and 97.97 on AP75. As shown in Table 5.

Table 4: Increase the number of training iterations to 30k on the TM&S dataset

TM&S	AP	AP50	AP75
boundingbox	73.24	100.00	92.72

segm	86.43	100.00	98.02

Table 5: Increase the number of training iterations to 30k on the BC&M dataset

BC&M	AP	AP50	AP75
boundingbox	70.00	99.88	83.27
segm	82.57	99.88	97.97

3.4 Data Comparison and Analysis

By comparing the AP values of the training test results, it was found that the ResNet-50-FPN network using FPN on TM&S data compared to the general ResNet-50-C4 network trained 20k times. On the bounding box, the AP difference was 13.34 percentage points, and on SEGM, the AP difference was 16.46 percentage points, with a time difference of 0.25 hours. Therefore, it can be seen that using FPN has significant advantages in both speed and accuracy. Therefore, ResNet-50-C4 network was not used for training on the BC&M dataset. In the results obtained using the ResNet-50-FPN network, the coco standard indicators obtained from both datasets were much larger than those obtained from training and testing on the original coco dataset, which may be due to the small total amount of data and the limited number of classifications.

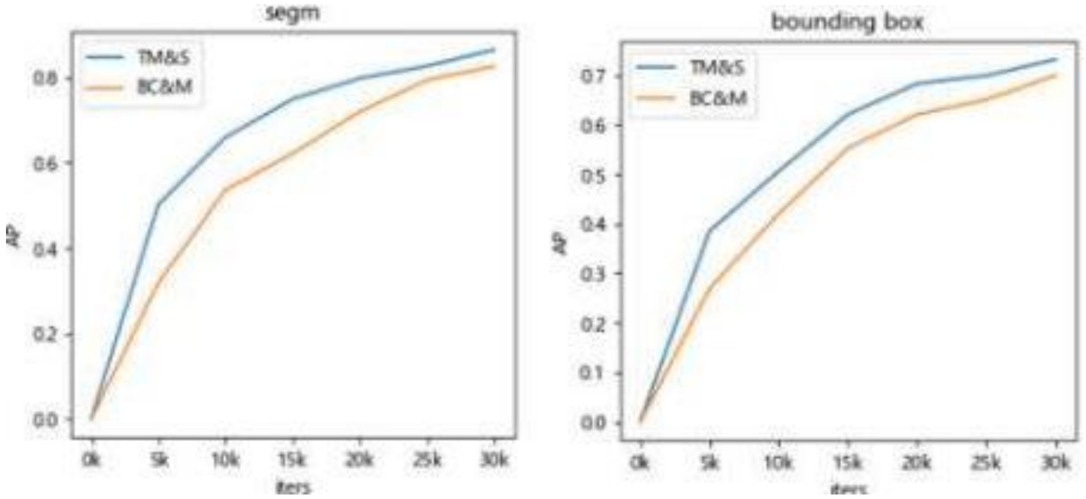


Figure 2: Average accuracy variation curves of two datasets, SEGM and BBOX

The variation curves of the average accuracy of the two datasets bbox and segm for 30k iterations are shown in Figure 2. Comparing the training data results of the two datasets, the TM&S dataset showed a difference of 6.23 percentage points in AP on the bounding box and 7.80 percentage points on the segm when the training times reached 20k compared to the BC&M dataset. However, when the training times reached 30k, the difference in AP on the bounding box and segm was 13.21 percentage points and 3.86 percentage points, respectively. From the difference in average accuracy AP between classification and segmentation, it can be seen that Mask R-CNN improves the classification and segmentation performance faster for targets with larger feature differences, while improving the classification and segmentation performance slower for targets with smaller feature differences. This result is also reasonable. For targets with significant differences in features, Mask R-CNN can converge quickly, so it can achieve high accuracy with fewer training iterations. However, when the feature differences are small, the convergence speed will decrease to a certain extent, and more iterations are required to achieve the same accuracy.

So overall, as the number of iterations increases, the difference in accuracy gradually decreases. It can be seen that Mask R-CNN also has strong robustness in classification and segmentation performance when dealing with similar feature targets.

4. SUMMARY

Mask R-CNN has strong robustness on different datasets, even when the target feature attributes are similar. Mask R-CNN has achieved great success in instance segmentation due to its unique ideas and structure. As for the Mask branch, its instance segmentation performance has already exceeded the performance of all models in instance segmentation at that time. Mask R-CNN achieved a leading position in instance segmentation tasks at that time, and the original text also indicated that the code was open-source, hoping that Mask R-CNN could become a framework for more advanced algorithms in related tasks in the future. And the fact shows that the proposal of Mask R-CNN has indeed made significant contributions to subsequent tasks such as instance segmentation and even instance level human segmentation.

REFERENCES

[1] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao and Li Fei-Fei. Novel dataset for Fine-Grained Image Categori- zation. First Workshop on Fine-Grained Visual Categorization (FGVC), IEEE Conference on Computer Vision and Pattern Rec- ognition (CVPR), 2011.

[2] T.-Y. Lin, P. Doll,ar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In CVPR, 2017.

[3] K. He, G. Gkioxari, P. Doll,ar, and R. Girshick. Mask r-cnn. In ICCV, 2017.

[4] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015.

[5] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional net- works for semantic segmentation. InCVPR,2015.

[6] Yang L, Song Q, Wang Z, et al. Parsing R-CNN for In- stance-Level Human Analysis[J]. 2018.

[7] Ji, H., Xu, X., Su, G., Wang, J., & Wang, Y. (2024). Utilizing Machine Learning for Precise Audience Targeting in Data Science and Targeted Advertising. Academic Journal of Science and Technology, 9(2), 215-220.

[8] Ma, Y., Shen, Z., & Shen, J. (2024). Cloud Computing and Hyperscale Data Centers: A Comparative Study of Usage Patterns. Journal of Theory and Practice of Engineering Science, 4(06), 11-19.

[9] Ren, Z. (2024). VGCN: An Enhanced Graph Convolutional Network Model for Text Classification. Journal of Industrial Engineering and Applied Science, 2(4), 110-115.

[10] Ren, Z. (2024). Enhanced YOLOv8 Infrared Image Object Detection Method with SPD Module. Journal of Theory and Practice in Engineering and Technology, 1(2), 1–7. Retrieved from https://woodyinternational.com/index.php/jtpet/article/view/42

[11] Yuan, B., & Song, T. (2023, November). Structural Resilience and Connectivity of the IPv6 Internet: An AS-level Topology Examination. In Proceedings of the 4th International Conference on Artificial Intelligence and Computer Engineering (pp. 853-856).

[12] Yuan, B., Song, T., & Yao, J. (2024, January). Identification of important nodes in the information propagation network based on the artificial intelligence method. In 2024 4th International Conference on Consumer Electronics and Computer Engineering (ICCECE) (pp. 11-14). IEEE.

[13] Wang, Z. (2024, August). CausalBench: A Comprehensive Benchmark for Evaluating Causal Reasoning Capabilities of Large Language Models. In Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10) (pp. 143-151).

[14] Wu, Z. (2024). Deep Learning with Improved Metaheuristic Optimization for Traffic Flow Prediction. Journal of Computer Science and Technology Studies, 6(4), 47-53.

[15] Lyu, H., Wang, Z., & Babakhani, A. (2020). A UHF/UWB hybrid RFID tag with a 51-m energy-harvesting sensitivity for remote vital-sign monitoring. IEEE transactions on microwave theory and techniques, 68(11), 4886-4895.

[16] Lu, Q., Guo, X., Yang, H., Wu, Z., & Mao, C. (2024). Research on Adaptive Algorithm Recommendation System Based on Parallel Data Mining Platform. Advances in Computer, Signals and Systems, 8(5), 23-33.

[17] Wu, X., Wu, Y., Li, X., Ye, Z., Gu, X., Wu, Z., & Yang, Y. (2024). Application of adaptive machine learning systems in heterogeneous data environments. Global Academic Frontiers, 2(3), 37-50.