Object Detection and Instance Segmentation using Mask R-CNN Algorithm

Rahul K. Kher[1], Priyanka Raninga[2]
[1]*Senior Member IEEE, EC Department, G H Patel College of Engineering & Technology, Vallabh Vidyanagar, Gujarat, India*
[2]*EC Department, G H Patel College of Engineering & Technology, Vallabh Vidyanagar, Gujarat, India*

***Abstract*- In the ever-advancing field of computer vision, image processing plays a prominent role. We can extend the applications of Image processing into solving real-world problems like substantially decreasing Human interaction over the art of driving. In the process of achieving this task, we face several challenges like Segmentation and Detection of objects. Mask RCNN is the superior model over the existing CNN models and yields accurate detection of objects more efficiently. In this paper, a mask R-CNN algorithm has been implemented using Python and the object detection results are shown.**

***Index terms*- Mask R-CNN, Segmentation, Object detection, Python**

1. INTRODUCTION

In modern technology, image segmentation contributes a major role in computer vision. Image segmentation is described as, segmenting into set of pixels or multiple significant regions as per specific application. The major intention of segmentation is for easy analysis by reducing information complexity and it is additionally useful in compressing the images. Image segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same lab share certain characteristics. Each of the as identifying objects. It may be challenging for beginners to distinguish between different related computer vision tasks. Image classification involves activities such as predicting the class of one object in an image.

Region-based Convolutional Neural Networks, or R-CNNs, is a family of techniques for addressing object localization and recognition tasks, designed for model performance. R-CNN generated region proposals based on selective search and then processed each proposed region, one at time, using Convolutional Networks to output an object label and its bounding box. The aim of object detection is to detect all instances of objects from a known class, such as people, cars or faces in an image. Generally, only a small number of instances of the object are present in the image, but there is a very large number of possible locations and scales at which they can occur. Mask RCNN is highly capable of achieving state-of-the-art results on a range of object detection tasks with high accuracy.

2. MASK R-CNN ALGORITHM: AN OVERVIEW

Segmentation Algorithms have been developed to segment the images; they are based on the two basic properties, discontinuity and similarity. This process detects outlines of an object and boundaries between objects and the background in the image. In recent years, there has been a continuous research going on CNN. Mostly, the R-CNN, and the extensions of it namely Fast R-CNN and Faster R-CNN and overcomes the issues of the previous method. Mask RCNN is an improvement over Faster RCNN by including a mask predicting branch parallel to the class label and bounding box prediction branch as shown in the image below. It adds only a small overhead to the Faster R-CNN network and hence can still run at 5 fps on a GPU. Object detection is an important task, yet challenging vision task. It is a critical part of many applications such as image search, image auto-annotation and scene understanding, object tracking. Moving object tracking of video image sequences was one of the most important subjects in computer vision.

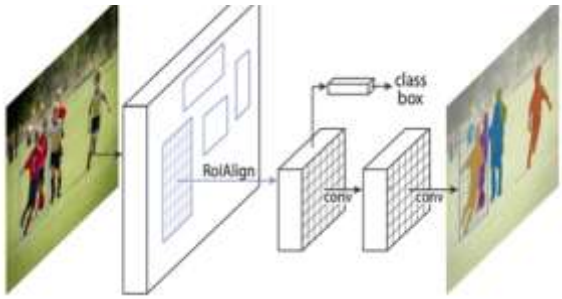


Figure 1. Mask R-CNN framework [4]

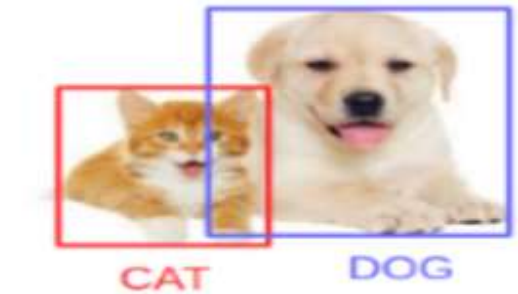
During training, each of these proposals (ROIs) go through the second part which is the object detection and mask prediction network, as shown above. Compared to other object detectors like YOLOv3, the network of Mask-RCNN runs on larger images. Mask R-CNN is a sophisticated model to implement, especially as compared to a simple or even state-of-the-art deep convolutional neural network model.

3. IMPLEMENTATION OF MASK R-CNN ALGORITHM

Mask R-CNN is basically an extension of Faster R-CNN. Faster R-CNN is widely used for object detection tasks. For a given image, it returns the class label and bounding box coordinates for each object in the image. So, let's say you pass the following image:

(a)

The Fast R-CNN model will return something like this:



(b)

Figure 2. (a) Original image and (b) object identified image by R-CNN

The original R-CNN algorithm is a four-step process:

- Step 1: Input an image to the network.
- Step 2: Extract region proposals (i.e., regions of an image that potentially contain objects) using an algorithm.
- Step 3: Use transfer learning, specifically feature extraction, to compute features for each proposal (which is effectively an ROI) using the pre-trained CNN.
- Step 4: Classify each proposal using the certain python algorithms.

Each selected ROI goes through R-CNN layer where there is label prediction of objects, bounding box prediction and Mask prediction around the object. The model is trained using a COCO dataset in which certain trained ROI dataset is present.

The implementation procedure is followed by different steps as follows: The first step is to install the library. The library can be installed directly via pip. It is always a good idea to confirm that the library was installed correctly. You can confirm that the library was installed correctly by querying it via the pip command. First, download the weights for the pre-trained model, specifically a Mask R-CNN trained on the MS Coco dataset. First, the model must be defined via an instance Mask RCNN class. This class requires a configuration object as a parameter. The configuration object defines how the model might be used during training or inference. In this case, the configuration will only specify the number of images per batch, which will be one, and the number of classes to predict. The next step will be to load the model and to make the prediction using the COCO dataset.

First, the model must be defined via an instance MaskRCNN class. The class requires a Will extract information from the image and store it in the vector form which helps to identify the object. We will now define the model and load into our algorithm. Now we will create algorithm which carries out the segmentation of the object detected in the image. With the help of COCO dataset and certain algorithm the object will be detected which are listed in the class of objects. Now using algorithm of instance segmentation, the object which is detected will be

segmented. Following steps summarize the R-CNN in Python.

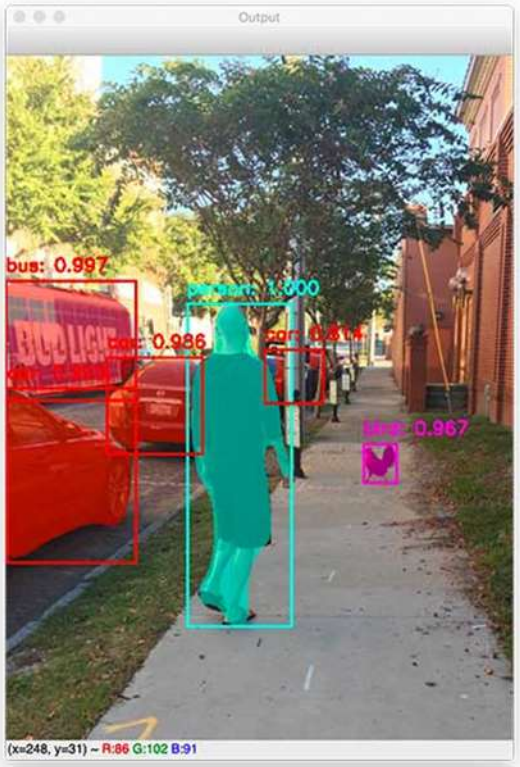
- Step 1: Install certain libraries such as: Numpy, imultis, open cv2, scikit-image, scipy, os, matplotlib, math
- Step 2: Download the pretrained weights of COCO
- Step 3: These weights are obtained from a model that was trained on the MS COCO dataset. Once you have downloaded the weights, paste this file in the samples folder of the Mask_RCNN.
- Step 4: Predicting for our image

Finally, we will use the Mask R-CNN architecture and the pretrained weights to generate predictions for our own images.

4. OUTPUT RESULTS

This is a sample image we feed to the algorithm and expect our algorithm to detect and identify objects in the image and label them according to the class assigned to it. As expected, our algorithm identifies the objects by its classes and assigns each object by its tag and has dimensions on detected image. Figure 3 (a) and (b) show the result of object identification using the mask R-CNN algorithm.

(a)



(b)

Figure 3. Objects identified: (a) car and (b) woman

5. CONCLUSION

In this paper, we described how to use the Mask R-CNN for Instance segmentation. The accuracy of detection was quite high as compared to other detection techniques. Moreover, the processing time required to identify was quite less. The model successfully detected and segmented the object in the image. The instance segmentation and object detection can be used in various fields and technologies like Real time application of self-driving cars, Counting the persons in real time or counting the objects from an image , Identifying features from an image like identifying cancer cells from an image , can be used in traffic footage to identify a required vehicle etc,

REFERENCES

[1] A. Ess, T. Muller, H. Grabner, and L. J. Van Gool, “Segmentation- based urban traffic scene

understanding", in BMVC, vol. 1, p. 2, Citeseer (2009).

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in 2012 IEEE Conference on Computer Vision and Pattern Recognition, June 2012, pp. 3354– 3361.

[3] M. Hameed, M. Sharif, M. Raza, S. W. Haider, and M. Iqbal, "Framework for the comparison of classifiers for medical image segmentation with transform and moment-based features," Research Journal of Recent Sciences, vol. 2, no. 6, pp. 1-10, June 2013

[4] Ravalisri.Vasam and Padmalaya Nayak, "Instance Segmentation on Real time Object Detection using Mask R-CNN", International Journal of Engineering and Advanced Technology (IJEAT), vol. 9, issue-1, October 2019.

[5] Ross Girshick, Jeff Donahue, Trevor Darrell and Jitendra Malik, "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence, 38: 1, 142-158.

[6] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks", Proceedings of the 2014 Int. Conf. on Computer Vision and Pattern Recognition (CVPR- 2014), Columbus, USA, June 24-27, 2014.

[7] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. "CAFFE: Convolutional architecture for fast feature embedding", arXiv:1408.5093, 2014.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", Proceedings of the 2014 Int. Conf. on Computer Vision and Pattern Recognition (CVPR-2014), Columbus, USA, June 24-27, 2014.

[9] Y. Zhu, R. Urtasun, R. Salakhutdinov, and S. Fidler, "segDeepM: Exploiting segmentation and context in deep neural networks for object detection", Proceedings of the 2015 Int. Conf. on Computer Vision and Pattern Recognition (CVPR-2015), Boston, USA, June 8-10, 2015.

[10] Sharif Razavian, A., Azizpour, H., Sullivan, J., & Carlsson, S., "CNN features off-the-shelf: an astounding baseline for recognition", Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 806-813).

[11] Redmon, J., & Farhadi, A. (2016). YOLO9000: better, faster, stronger. arXiv preprint arXiv:1612.08242.

[12] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation", Proceedings of the 2017 Int. Conf. on Computer Vision and Pattern Recognition (CVPR-2017), Honolulu, USA, 21-26 July, 2017.

[13] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2-D pose estimation using part affinity fields", Proceedings of the 2017 Int. Conf. on Computer Vision and Pattern Recognition (CVPR-2017), Honolulu, USA, 21-26 July, 2017.

[14] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," IEEE transactions on pattern analysis and machine intelligence, vol. 35, no. 8, pp. 1915–1929, 2013.

[15] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, "Speed/accuracy trade-offs for modern convolutional object detectors", Proceedings of the 2017 Int. Conf. on Computer Vision and Pattern Recognition (CVPR-2017), Honolulu, USA, 21-26 July, 2017.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks", Proceedings of 2015 Int. Conf. on Neural Information Processing System (NIPS-2015), Montreal, Canada, Dec 7-10, 2015.