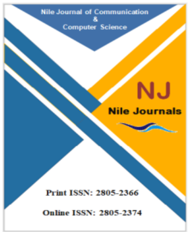
Nile Journals

Nile Journal of Communication & Computer Science

Volume 3 , Number 1, May 2022

Journal Webpage: https://njccs.journals.ekb.eg



Nile Journal of Communication & Computer Science

NJ Nile Journals

Print ISSN: 2805-2366

Online ISSN: 2805-2374

Review: Mask R-CNN Models

***Esraa Hassan*[1], *Nora El-Rashidy*[1] *Fatma M. Talaa*[1*]**

fatma.nada@ai.kfs.edu.eg

Faculty of Artificial Intelligence, Kafrelsheikh University, Kafrelsheikh, Egypt

**: corresponding author; ORCID: 0000-0001-6116-2191*

Abstract

Instance segmentation is a challenging computer vision task that requires the prediction of object instances and their per-pixel segmentation mask. This makes it a hybrid of semantic segmentation and object detection. It detects and delineates each distinct object of interest appearing in an image. Mask R-CNN model is common for instance segmentation that has several versions for improving this task. We proposed a simple comparison between Fifteenth different version frameworks from Mask-RCNN for object instance segmentation. Our survey representing the difference between the popular versions of Mask R-CNN. The Mask R-CNN method extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition. The results in most versions were implemented on of the COCO dataset that created for instance segmentation tasks.

Keywords: *Open CV, CNN, Mask R-CNN models, COCO dataset, instance segmentation tasks.*

1. Introduction

Computer vision tasks include methods for acquiring, processing, analyzing and understanding digital images, and extraction of high-dimensional data from the real world[1]. Real-time computer vision can be performed using the open-source computer vision (OpenCV) programming library. OpenCV has vast application areas such as facial recognition systems, human-computer interaction, object identification, mobile robotics, motion tracking, and augmented reality. R-CNNs (Region Based Convolutional Neural Networks) are a kind of machine learning model used in computer vision, specifically object detection. As a result, Mask R-CNN is a natural and intuitive concept. However, the additional mask output differs from the class and box outputs, necessitating the extraction of a much finer spatial layout of an object. For each candidate item, Mask R-CNN produces two outputs: a class label and a bounding-box offset; to this, we add a third branch that produces the object mask[2]. As a result, Mask R-CNN is a natural and intuitive concept. However, the additional mask output differs from the class and box outputs, necessitating the extraction of a much finer spatial arrangement of an item. This is accomplished by adding a branch for anticipating an object mask alongside the existing branch for bounding box recognition as shown in Figure 1.

Mask R-CNN is an intuitive extension of Faster R-CNN in theory, but properly designing the mask branch is important for good results. Most crucially, faster R-CNN was not designed to align network inputs and outputs pixel by pixel [1]. Mask R-CNN is straightforward to train and adds only a minor overhead to Faster R-CNN, which runs at 5 frames per second. Furthermore, Mask R-CNN is easily generalizable to various tasks, such as estimating human poses within the same framework. We achieved top results in all three COCO challenge tracks, including instance segmentation, bounding box object detection, and person key point detection. Mask R-CNN surpasses all previous, single-model entrants on every job, including the COCO 2016 challenge winners, even without any bells and whistles [3]. Over a short period of time, the vision community has quickly improved object detection and semantic segmentation outcomes. These advancements have been fueled in large part by powerful baseline systems, such as the Fast/Faster RCNN and Fully Convolutional Network (FCN) frameworks for object detection and semantic segmentation, respectively[4].

These approaches are theoretically simple and provide flexibility and resilience, as well as quick training and inference times. It is based on the R-CNN series, FPN, FCIS, etc. The idea of MRCNN is very simple: Faster R-CNN has two outputs for each candidate area: category label and box bias[5]. MRCNN adds another branch based on the Faster R-CNN and adds an output, the object mask. We instantiate Mask R-CNN with several architectures to showcase the generality of our technique [4]. To be clear, we distinguish between the convolutional backbone architecture used for feature extraction over the whole image and (ii) the network head for bounding-box identification (classification and regression) and mask prediction that is applied to each RoI separately. Mask R-CNN is an enhancement of the Faster RCNN method that adds a segmentation mask along with the bounding boxes to each RoI. This additional segment facilitates a wide range of use cases. Mask R-CNN requires an inference time of 350-200 ms. Mask rcnn is a new convolutional network proposed based on the previous fast rcnn architecture. The object instance segmentation is completed in one fell swoop[6]. This method accomplishes high-quality semantic segmentation while effectively targeting. The main idea of the article is to extend the original Faster-RCNN, add a branch, and use the existing detection to predict the target in parallel. At the same time, this network structure is relatively simple to implement and train, with a speed of 5fps that can be easily applied to other areas such as target detection, segmentation, and character key point detection.is better than the existing algorithm, and it is shown in the later experimental results [3]. Fast/Faster R-CNN, Good speed, Good accuracy, Intuitive and simple to use. Fully Convolutional Net (FCN), fast and accurate, intuitive and simple to use. Instance Segmentation Mask R-CNN's goals are to be fast, accurate, intuitive, and simple to use[7].

Mask R-CNN is simple to train and adds only a small overhead to Faster R-CNN, running at 5 fps. Moreover, Mask R-CNN is easy to generalize to other tasks, e.g., allowing us to estimate human poses in the same framework as shown in figure (2). We show top results in all three tracks of the COCO suite of challenges, including instance segmentation, bounding box object detection, and person key point detection. Without bells and whistles, Mask R-CNN outperforms all existing, single-model entries on every task, including the COCO 2016 challenge winners. The vision community has rapidly improved object detection and semantic segmentation results over a short period of time. In large part, these advances have been driven by powerful baseline systems, such as the Fast/Faster RCNN and Fully Convolutional Network (FCN) frameworks for object detection and semantic segmentation, respectively [5]. These methods are conceptually intuitive and offer flexibility and robustness, together with fast training and inference time. Our goal in this work is to develop a comparably enabling framework for instance segmentation. First, let's review the Faster R-CNN. The Faster R-CNN consists of two phases: (i) The Region Proposal Network (RPN) and the basic Fast R-CNN model. (ii) RPN is employed in the generation of candidate regions.

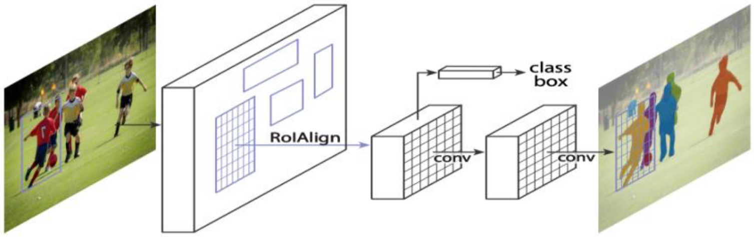


Figure 1:The general mechanism of R-CNN mask model

This survey is organized as follows: Section 2 describes the Mask RCNN model Versions, and finally the paper concludes in section 3.

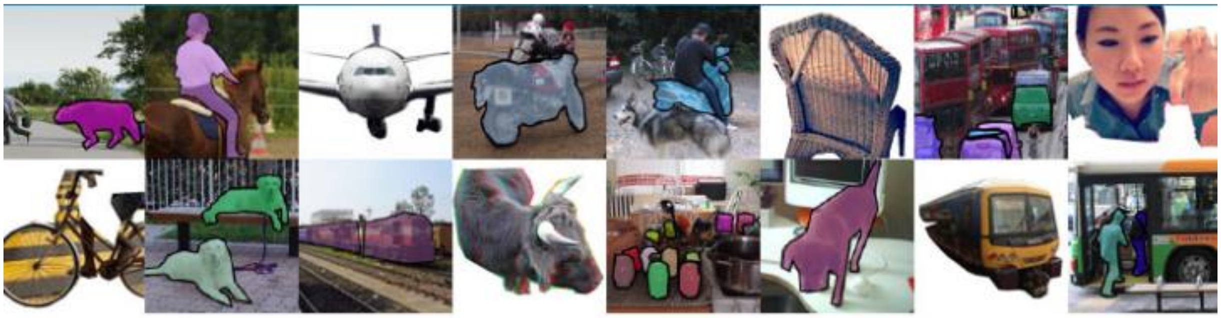


Figure 2: common tasks for RCNN mask model

2. Mask RCNN model Versions

In this section, we describe different Mask R-CNN models.

2.1. Mask R-CNN (R101-C4, 3x)

Mask R-CNN (R101-C4, 3x) is a common version of the Mask R-CNN framework for instance segmentation. It is distinguished with several metadata descriptions where it used the COCO dataset in the training data process. The COCO dataset is a large-scale object detection, segmentation, and captioning dataset. The implementation with COCO is a fantastic idea with many features. Object segmentation, recognition in context, super pixel stuff segmentation, etc. R101 depends on SGD with Momentum training techniques and 8 NVIDIA V100 GPUs as training resources that make it a robust framework with speed and flexibility. The R101 has a Floating Point Operations (FLOP) Input Number of 100 Per Second., it depends on the convolution layer, RoIAlign Softmax layer, RPN layer, Dense Connections layer, ResNet layers and it The maximum number of iterations was 270000, with 55 million parameters, a training time of 0.652 seconds per iteration, and an inference time of 0.145 seconds per iteration. It has a huge amount of data that requires training memory of 6.3 (GB). Table 1 represents extra and main details about this version.

2.2. Mask R-CNN (R101-DC5, 3x)

Mask R-CNN (R101-DC5, 3x) is a common version of the Mask R-CNN framework for instance segmentation. It is distinguished with several metadata descriptions where it used the COCO dataset in the training data process. R101 depends on 8 NVIDIA V100 GPUs as training resources, which makes it a robust framework with speed and flexibility to use. The R101 has a Floating Point Operations (FLOP) Input Number of 100 per Second. It depends on the convolution layer, RoIAlign layer, Softmax layer, RPN layer, Dense Connections layer, and ResNet layer. Its Max Iteration reached 270000 iterations with 191 Million Parameters, training time of 0.545 (s/iter) and inference time of 0.092 (s/im). It has a huge amount of data that requires training memory of 7.6 (GB). Table 1 represents extra and main details about this version.

2.3. Mask R-CNN (R101-FPN, 1x, LVIS)

Mask R-CNN (R101-FPN, 1x, LVIS) is a version of the Mask R-CNN framework for instance segmentation. It is distinguished with several metadata descriptions where it used the COCO dataset in the training data process. R101 depends on 8 NVIDIA V100 GPUs as training resources, which makes it a robust framework with speed and flexibility to use. The R101 has a Floating Point Operations (FLOP) Input Number of 100 Per Second., it depends on the convolution layer, RoIAlign layer, Softmax layer, RPN layer, Dense Connections layer, and ResNet layer and it The maximum number of iterations was 90000, with 70 million parameters, a training time of 0.371 seconds per iteration, and an inference time of 0.114 seconds per iteration. It has a huge amount of data that requires training memory of 7.8 (GB). Table 1 represents extra and main details about this version.

2.4. Mask R-CNN (R101-FPN, 3x)

Mask R-CNN (R101-FPN, 1x, LVIS) is a version of the Mask R-CNN framework for instance segmentation. It is distinguished with several metadata descriptions where it used the COCO dataset in the training data process. R101 depends on 8 NVIDIA V100 GPUs as training resources, which makes it a robust framework with speed and flexibility to use. The R101 has a Floating Point Operations (FLOP) Input Number of 100 Per Second., it depends on the convolution layer, RoIAlign layer, Softmax layer, RPN layer, Dense Connections layer, and ResNet layer and it The maximum number of iterations was 270000, with 63 million parameters, a training time of 0.34 seconds per iteration, and an inference time of 0.056 seconds per iteration. It has a huge amount of data that requires training memory of 4.6 (GB). Table 1 represents extra and main details about this version.

2.5. Mask R-CNN (R50-C4, 1x)

Mask R-CNN (R50-C4, 1x) is a version of the Mask R-CNN framework for instance segmentation. It is distinguished with several metadata descriptions where it used the COCO dataset in the training data process. R101 depends on 8 NVIDIA V100 GPUs as training resources, which makes it a robust framework with speed and flexibility to use. The R101 has a Floating Point Operations (FLOP) Input Number of 100 per second. it depends on the convolution layer, RoIAlign layer, Softmax layer, RPN layer, Dense Connections layer, and ResNet layer. Its maximum iteration reaches 90000 iterations with 36 million parameters, training time of 0.584 (s/iter), and inference time of 0.11 (s/im). It has a huge amount of data that requires training memory of 5.2 (GB). Table 1 represents extra and main details about this version.

2.6. Mask R-CNN (R50-C4, 3x)

Mask R-CNN (R101-C4, 3x) is a common version of the Mask R-CNN framework for instance segmentation. It is distinguished with several metadata descriptions where it used the COCO dataset in the training data process. The COCO dataset is a large-scale object detection, segmentation, and captioning dataset. The implementation with COCO is a fantastic idea with many features. Object segmentation, recognition in context, super pixel stuff segmentation, etc. R101 depends on SGD with Momentum training techniques and 8 NVIDIA V100 GPUs as training resources that make it a robust framework with speed and flexibility[8].

2.7. Mask R-CNN (R50-C4, VOC)

Mask R-CNN (R50-C4, VOC) is a version of the Mask R-CNN framework for instance segmentation. It is distinguished with several metadata descriptions where it used the PASCAL VOC 2007 dataset in the training data process. R101 depends on 8 NVIDIA V100 GPUs as training resources, which makes it a robust framework with speed and flexibility to use. The R101 has a Floating Point Operations (FLOP) Input Number of 100 per Second. It depends on the convolution layer, RoIAlign layer, Softmax layer, RPN layer, Dense Connections layer, and ResNet layer. Its Max Iteration reached 18000 iterations with 33 Million Parameters, training time of 0.537 (s/iter) and inference time of 0.081 (s/im). It has a huge amount of data that requires training memory of 4.8 GB. Table 1 represents extra and main details about this version[9].

2.8. Mask R-CNN (R50-DC5, 1x)

Mask R-CNN (R50-DC5, 1x) is a version of the Mask R-CNN framework for instance segmentation. It is distinguished with several metadata descriptions where it used the COCO dataset in the training data process. R101 depends on 8 NVIDIA V100 GPUs as training resources, which makes it a robust framework with speed and flexibility to use. The R101 has a Floating Point Operations (FLOP) Input Number of 100 per Second. It depends on the convolution layer, RoIAlign layer, Softmax layer, RPN layer, Dense Connections layer, and ResNet layer. Its max iteration reached 90000 iterations with 172 Million Parameters, training time of 0.471 (s/iter) and inference time of 0.076 (s/im). It has a huge amount of data that requires training memory of 6.5 (GB). Table 1 represents extra and main details about this version.

2.9. Mask R-CNN (R50-FPN, 1x)

Mask R-CNN (R50-FPN, 1x) is a version of the Mask R-CNN framework for instance segmentation. It is distinguished with several metadata descriptions where it used the COCO dataset in the training data process. R101 depends on 8 NVIDIA V100 GPUs as training resources, which makes it a robust framework with speed and flexibility to use. The R101 has a Floating Point Operations (FLOP) Input Number of 100 per Second. It depends on the convolution layer, RoIAlign layer, Softmax layer, RPN layer, Dense Connections layer, and ResNet layer. Its max iteration reached 270000 iterations with 36 million parameters, training time of 0.47 (s/iter) and inference time of 0.076 (s/im). It has a huge amount of data that requires training memory of 5.2 (GB). Table 1 represents extra and main details about this version.

2.10. Mask R-CNN (R50-FPN, 1x, LVIS)

Mask R-CNN (R50-FPN, 1x, LVIS) is a version of the Mask R-CNN framework for instance segmentation. It is distinguished with several metadata descriptions where it used the COCO dataset in the training data process. R101 depends on 8 NVIDIA V100 GPUs as training resources, which makes it a robust framework with speed and flexibility to use. The R101 has a Floating Point Operations (FLOP) Input Number of 100 Per Second., it depends on the convolution layer, RoIAlign layer, Softmax layer, RPN layer, Dense Connections layer, and ResNet layer and it The maximum number of iterations was 90000, with 44 million parameters, a training time of 0.261 seconds per iteration, and an inference time of 0.043 seconds per iteration. It has a huge amount of data that requires training memory of 3.4 (GB). Table 1 represents extra and main details about this version.

2.11. Mask R-CNN (R50-FPN, 3x)

Mask R-CNN (R50-FPN, 3x) is a version of the Mask R-CNN framework for instance segmentation. It is distinguished with several metadata descriptions where it used the COCO dataset in the training data process. R101 depends on 8 NVIDIA V100 GPUs as training resources, which makes it a robust framework with speed and flexibility to use. The R101 has a Floating Point Operations (FLOP) Input Number of 100 per Second. It depends on the convolution layer, RoIAlign layer, Softmax layer, RPN layer, Dense Connections layer, and ResNet layer. Its maximum iteration reaches 90000 iterations with 50 million parameters, training time of 0.292 (s/iter), and inference time of 0.0107 (s/im). It has a huge amount of data that requires training memory 7.1 (GB). Table 1 represents extra and main details about this version.

2.12. Mask R-CNN (X101-FPN, 1x, LVIS)

Mask R-CNN (X101-FPN, 1x, LVIS) is a version of the Mask R-CNN framework for instance segmentation. It is distinguished with several metadata descriptions where it used the COCO dataset in the training data process. R101 depends on 8 NVIDIA V100 GPUs as training resources, which makes it a robust framework with speed and flexibility to use. The R101 has a

Floating Point Operations (FLOP) Input Number of 100 Per Second., it depends on the convolution layer, RoIAlign layer, Softmax layer, RPN layer, Dense Connections layer, and ResNet layer and it The maximum number of iterations was 27000, with 44 million parameters, a training time of 3.4 seconds per iteration, and an inference time of 0.043 seconds per iteration. It has a huge amount of data that requires training memory of 0.261 (GB). Table 1 represents extra and main details about this version.

2.13. Mask R-CNN (R50-FPN, Cityscapes)

Mask R-CNN (R50-FPN, Cityscapes) is a version of the Mask R-CNN framework for instance segmentation. It differentiates with several metadata descriptions where it used the Cityscapes dataset in the training data process. R101 depends on 8 NVIDIA V100 GPUs as training resources, which makes it a robust framework with speed and flexibility to use. The R101 has a Floating Point Operations (FLOP) Input Number of 100 per Second. It depends on the convolution layer, RoIAlign layer, Softmax layer, RPN layer, Dense Connections layer, and ResNet layer. Its max iteration reached 90000 iterations with 36 million parameters, training time of 0.24 (s/iter) and inference time of 0.078 (s/im). It has a huge amount of data that requires training memory of 4.4 (GB). Table 1 represents extra and main details about this version.

2.14. Mask R-CNN (X101-FPN, 1x, LVIS)

Mask R-CNN (X101-FPN, 1x, LVIS) is a version of the Mask R-CNN framework for instance segmentation. It is distinguished with several metadata descriptions where it used the COCO dataset in the training data process. R101 depends on 8 NVIDIA V100 GPUs as training resources, which makes it a robust framework with speed and flexibility to use. The R101 has a Floating Point Operations (FLOP) Input Number of 100 per Second. It depends on the convolution layer, RoIAlign layer, Softmax layer, RPN layer, Dense Connections layer, and ResNet layer and the maximum number of iterations was 90000, with 114 million parameters, training time of 0.712 seconds per iteration, and inference time of 0.151 seconds per iteration. It has a huge amount of data that requires training memory of 10.2 GB. Table 1 represents extra and main details about this version[10].

2.15. Mask R-CNN (X101-FPN, 3x)

Mask R-CNN (X101-FPN, 3x) is a version of the Mask R-CNN framework for instance segmentation. It is distinguished with several metadata descriptions where it used the COCO dataset in the training data process. R101 depends on 8 NVIDIA V100 GPUs as training resources, which makes it a robust framework with speed and flexibility to use. The R101 has a Floating Point Operations (FLOP) Input Number of 100 per Second. It depends on the convolution layer, RoIAlign layer, Softmax layer, RPN layer, Dense Connections layer, and ResNet layer. Its max iteration reached 27000 iterations with 107 Million Parameters, training time of 0.69 (s/iter) and inference time of 0.103 (s/im). It has a huge amount of data that requires training memory 7.2 (GB). Table 1 represents extra and main details about this version.

Table. 1. General Comparison between the all types of Mask RCNN framework

		Parameters	FLOPs	File Size	Training Data	Training Resources	Training Time	Architecture	ID	Max Iter	lr sched	FLOPs Input No	train time (s/iter)	Training Memory (GB)	inference time (s/im)
(1) (2) (3) (4) (5) (6)	R101-C4, 3x	55 Million	937 Billion	210.10 MB	COCO	8 NVIDIA V100 GPUs	2.04 days	, Convolution ,RoIAlign Softmax, RPN, Dense Connections, ResNet	138363239	270000	3x	100	0.652	6.3	0.145
	R101-DC5, 3x	191 Million	0.092	730.60 MB	COCO	8 NVIDIA V100 GPUs	1.7 days	, RoIAlign, Softmax, RPN, Dense Convolution Connections, ResNet	138363294	270000	3x	No	0.545	7.6	0.092
	R101-FPN, 1x, LVIS	70 Million	527 Billion	265.90 MB	COCO	8 NVIDIA V100 GPUs		, RoIAlign, Softmax, RPN, Dense Convolution Connections, ResNet	144219035	90000	1x	100	0.371	7.8	0.114
	R101-FPN, 3x	63 Million	290 Billion	242.29 MB	COCO	8 NVIDIA V100 GPUs	1.06 days	, RoIAlign, Softmax, RPN, Dense Convolution Connections, ResNet	138205316	270000	3x	100	0.34	4.6	0.056
	R50-C4, 1x	36 Million	890 Billion	137.42 MB	COCO	8 NVIDIA V100 GPUs	15 hours	, RoIAlign, Softmax, RPN, Dense Convolution Connections, ResNet	137259246	90000	1x	100	0.584	5.2	0.11
	R50-C4, 3x	36 Million	890 Billion	137.42 MB		8 NVIDIA V100	1.8 days	Convolution , RoIAlign, Softmax, RPN, Dense Connections, ResNet	137849525	270000	3x	100	0.575	5.2	0.111
(7)	R50-C4, VOC	33 Million	0.081	127.00 MB	PASCAL VOC 2007	8 NVIDIA V100 GPUs	3 hours	, RoIAlign, Softmax, RPN, Dense Convolution Connections, ResNet	142202221	18000		100	0.537	4.8	0.081
(8)	R50-DC5, 1x	172 Million	0.076	657.92 MB	COCO	8 NVIDIA V100 GPUs	12 hours	, RoIAlign, Softmax, RPN, Dense Convolution Connections, ResNet	137260150	90000	1x	100	0.471	6.5	0.076
(9)	R50-DC5, 3x	172 Million	0.076	657.92 MB	COCO	8 NVIDIA V100 GPUs	1.47 days	, RoIAlign, Softmax, RPN, Dense Convolution Connections, ResNet	137849551	270000	3x	100	0.47	6.5	0.076
(10)	R50-FPN	44 Million	0.043	169.60 MB	COCO	8 NVIDIA V100	7 hours	, RoIAlign, Softmax, RPN, Dense Convolution Connections, ResNet	137260431	90000	1x	100	0.261	3.4	0.043

	, 1x					GPUs								
(11)	R50-FPN, 1x, LVIS	50 Million	460 Billion	193.21 MB	COCO	8 NVIDIA V100 GPUs	7 hours	, RoIAlign, Softmax, RPN, Dense Convolution Connections, ResNet	144219072	90000	1x	100	0.292	7.1	0.107
(12)	R50-FPN, 3x	44 Million	0.043	169.60 MB	COCO	8 NVIDIA V100 GPUs	20 hours	, RoIAlign, Softmax, RPN, Dense Convolution Connections, ResNet	137849600	270000	3x	100	0.261	3.4	0.043
(13)	R50-FPN, Cityscapes	44 Million	464 Billion	168.13 MB	Cityscapes	8 NVIDIA V100 GPUs	2 hours	, RoIAlign, Softmax, RPN, Dense Convolution Connections, ResNet	142423278	0.01		100	0.24	4.4	0.078
(14)	X101-FPN, 1x, LVIS	114 Million	686 Billion	435.04 MB	COCO	8 NVIDIA V100 GPUs	18 hours	, RoIAlign, Softmax, RPN, Dense Convolution Connections, ResNeXt			1x	100	0.712	10.2	0.151
(15)	X101-FPN, 3x	107 Million	0.103	411.43 MB	COCO	8 NVIDIA V100 GPUs	2.16 days	, RoIAlign, Softmax, RPN, Dense Convolution Connections, ResNeXt	139653917	270000	3x		0.69	7.2	

3. Conclusions

In this survey, we proposed common types of RCNN mask models that are used in instance segmentation task. Versions of RCNN mask have different properties which mask it helpful for more many situations and cases. The Mask R-CNN modes and Faster R-CNN are helping for expecting an object mask in parallel with the branch for bounding box recognition. The results in most versions were implemented on of the COCO dataset that generated for instance segmentation tasks.

References

[1] E. Hassan, M. Shams, N. A. Hikal, and S. Elmougy, “Plant Seedlings Classification using Transfer,” no. July, pp. 3–4, 2021.

[2] Q. Li, L. Shen, S. Guo, and Z. Lai, “Wavelet Integrated CNNs for Noise-Robust Image Classification.”

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, 2020, doi: 10.1109/TPAMI.2018.2844175.

[4] S. Elmuogy, N. A. Hikal, and E. Hassan, “An efficient technique for CT scan images classification of COVID-19,” vol. 40, pp. 5225–5238, 2021, doi: 10.3233/JIFS-201985.

[5] H. tao Zhang *et al.*, “Automated detection and quantification of COVID-19 pneumonia: CT

imaging analysis by a deep learning-based software," *Eur. J. Nucl. Med. Mol. Imaging*, vol. 47, no. 11, pp. 2525–2532, 2020, doi: 10.1007/s00259-020-04953-1.

[6] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 4438–4446, 2017, doi: 10.1109/CVPR.2017.472.

[7] V. Birodkar, Z. Lu, S. Li, V. Rathod, and J. Huang, "The surprising impact of mask-head architecture on novel class segmentation."

[8] N. K. Chowdhury, M. A. Kabir, M. M. Rahman, and N. Rezoana, "ECOVNet: An Ensemble of Deep Convolutional Neural Networks Based on EfficientNet to Detect COVID-19 From Chest X-rays," 2020, doi: 10.7717/peerj-cs.551.

[9] F. O. Giuste and J. C. Vizcarra, "CIFAR-10 Image Classification Using Feature Ensembles," pp. 1–5, 2020, [Online]. Available: http://arxiv.org/abs/2002.03846.

[10] J. C. Spall, *Stochastic Optimization*. 2012.