Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 01, March 2021.

A NOVEL INSTANCE SEGMENTATION ALGORITHM BASED ON IMPROVED DEEP LEARNING ALGORITHM FOR MULTI-OBJECT IMAGES

Suhaila Farhan Ahmad Abuowaida[1], Huah Yong Chan[1], Nawaf Farhan Funkur Alshdaifat[1] and Laith Abualigah[2]

(Received: 26-Oct.-2020, Revised: 15-Dec.-2020 and 17-Jan.-2021, Accepted: 2-Feb.-2021)

ABSTRACT

A Deep Learning (DL) algorithm is highly common and attractive in recent years because of its encouraging achievements in many areas. DL lies in image-based detection and instance segmentation of an entity, which is a critical issue that needs further investigation. This paper aims to study the fundamental challenges in using object instance segmentation of images. This paper proposes a novel algorithm for multi-object image instance segmentation algorithm in three stages. A novel backbone approach improves the image recognition algorithm by extracting low and high characteristic levels from the given images in the first stage. The ResNet is the fundamental building block and connects with the Squeeze-and-Excitation Network (SENet) for each ResNet block. The Region Proposal Network (RPN) is used to determine the object item's placement, followed by the third stage, which suggests an average position RoI layer to choose the optimal boundaries of the instance segmentation. The experiments are conducted and validated using a standard benchmark image dataset, called COCO. The proposed algorithm's performance is validated using standard evaluation criteria and compared against the recent image segmentation algorithms that use object instances. The results show that the proposed algorithm gets better results than other well-known instance segmentation algorithms in terms of average accuracy over IoU (AP) threshold measures using various thresholds.

KEYWORDS

Deep learning, Multi-object detection, Recognition, Instance segmentation, Average position RoI layer.

1. INTRODUCTION

DL as a branch of machine learning is given this term, since it uses a Deep Neural Network (DNN) [1]. DNN has attracted significant interest and attention over the years due to its ability to handle complex data by its very nature and having a high-level of dimensions [2], such as computer vision [3]-[4], speech recognition [5]-[6], Rate Control (RC) [7], depth estimation from singe image [8] and neural language processing [9]-[10]. One of the essential characteristics of the DNN is dealing with a broad set of data in its various forms in the training phase through optimization algorithms. Within a short period of time, the vision community has been improved rapidly for object recognition [11]-[12], object detection [13]-[14], semantic segmentation [15] and instance segmentation [16]-[17] based on DL. The object detection algorithms, which are used to get object information, have two main problems. First, the traditional algorithms cannot solve the object detection [18]-[19] and recognition problems [20]-[21] effectively. This problem mainly focuses on distinguishing the object from the background and addressing labels of the object class. Second, addressing the bounding boxes of each object is a critical issue to solve the object localization. The development process has been motivated by a powerful baseline algorithm. A Region-Convolution Neural Network (R-CNN) [22] is used to solve the problem of multiple-object detection by generating a particular region search, which can draw bounding boxes over all of the objects. Afterward, the algorithm applies the VGG backbone with a modified Fully Connected (FC) layer using Support Vector Machine (SVM), which extracts a feature map for each region and image detection. R-CNN's main

1. S. F. A. Abuowaida, H. Y. Chan and N. F. F. Alshdaifat are with School of Computer Sciences, Universiti Sains Malaysia, 11800, Pulau Pinang, Malaysia. Emails: suhilaowida@student@usm.my, hychan@usm.my and nawaf@student@usm.my

2. L. Abualigah is with Faculty of Computer Science and Informatics, Amman Arab University, Amman 11953, Jordan. Email: Aligah.2020@gmail.com

drawbacks are that its detection process is slow, requires multiple stages and is computationally costly. Nevertheless, the Fast R-CNN enhances the R-CNN to solve the low accuracy and slow detection problems by sharing computation of the convolution layers of various proposals and running the CNN [23]. This is the primary motivation to generate proposals for the region using selective search. Henceforth, the proposal region is sent to Region of Interest (RoI), which is a proposed region from the input image and RoI pooling that converts the feature inside each region to make small feature maps using max-pooling. The main issue of the algorithm is its slow detection, since using selective search produces bottlenecks. Notably, Faster R-CNN [24] is the enhanced version of Fast R-CNN by combining the RPN and Fast R-CNN. Furthermore, transforming the Faster R-CNN image into a convolution network requires output, which is a set of feature maps on the last convolution layer used. Hence, a sliding window is implemented for each feature map. The Faster R-CNN uses 3×3 as the sliding window size and a set of nine anchors, which computes how much these anchors have overlapped with the ground-truth bounding boxes. Finally, each feature map extracted from the convolution layer has fed a smaller network with two tasks: classification and regression.

The instance segmentation is an essential task of the object recognition system in recognising each object based on the image. The instance segmentation task is challenging due to several challenges, including diversity, difference between the colours, sizes of the object items and overlapping between the objects. Li et al. in [25] have proposed a new algorithm for image segmentation, called Full Convolution Instance Segmentation (FCIS). However, in perdition edges, FCIS suffers from overlapping instances and errors. Another research [26] has proposed a Multi-task Network Cascade (MNC) algorithm for instance segmentation. The MNC comprises of three stages; each stage has a particular task to predict the instance level for each object. The first stage proposes the bounding box for each object in the image, the second stage presents a mask for each bounding box and the third stage distinguishes between instances. However, the MNC, which has numerous predicting instance segmentation gaps, is inflexible and takes much time when predicting instance segmentation. Also, the main problem in MNC is that the three stages do not work in a parallel way and require many parameters for each stage, leading to prolonged time to predict instance segmentation. Mask R-CNN is used to predict instance-level segmentation [27]. The proposed algorithm utilizes the Faster R-CNN to predict each object's mask by adding a branch for each bounding box after the Faster R-CNN. The Mask R-CNN works in parallel to decrease training and testing time. Moreover, the main contribution of Mask R-CNN is that it uses RoI align and provides highly accurate results. However, it has taken a long time in the training stage and lost some features at the instance level. A real-time algorithm is proposed in [28], called You Only Look at CoefficienTs (YOLACT), where it uses the parallel concept in its main procedure. However, this algorithm does not receive satisfying feedback because of instance segmentation's accuracy values. Regarding the architecture of YOLACT [28], the researcher in [29] has constructed a Cascade R-CNN that minimizes the overfitting using a sequential threshold. However, the Cascade R-CNN increases the threshold training and testing time. The architecture of Cascade R-CNN consists of many stages, where each stage extends Faster R-CNN for the localization of the object for each input image and extends Mask R-CNN to instance segmentation. The main advantage of Cascade R-CNN is decreasing overfitting through a sequence of detectors trained with increasing IoU thresholds and is sequentially more selective against close false positives.

The main goal of this paper is to build a new algorithm able to perform the instance segmentation process effectively. Therefore, it combines elements from the classical computer vision object detection tasks. The objective is to classify individual objects, locate each using a bounding box and instance segmentation. The objective is to classify each pixel into a fixed set of categories without differentiating object instances. Normally, a complex algorithm is required to achieve good results. We demonstrate a surprisingly simple, versatile and fast algorithm that can overcome the results of well-known instance segmentation algorithms. This paper has proposed a new algorithm for instance segmentation of objects to solve the problems mentioned above. Hence, a novel image instance segmentation algorithm is proposed; namely, multi-object instance segmentation is divided into three phases. In the first phase, a novel backbone architecture is proposed, aiming to extract the object's feature maps with high precision value and less time. In the second phase, the RPN is adapted to identify multiple objects. In the third phase, the Fully Convolution Network

(FCN) [30] is used to generate instance segmentation to prevent the overlapping problem between objects. This prevention manages the various sizes of RPN's feature maps using the average position RoI layer. So, the instance segmentation of multiple-object instances is the main aim of this research. The proposed multi-object instance segmentation algorithm is evaluated using two measures; the AP with different thresholds and time. The obtained results of the proposed algorithm are compared against other well-known instance segmentation algorithms published in the literature, such as MNC, FCIS, Mask R-CNN, YOLACT and Cascade R-CNN. The experimental results have shown that the proposed image instance segmentation algorithm has obtained better results compared to other well-known image instance segmentation algorithms. The rest of this paper is organized as follows. Section 2 presents the full details of the proposed image instance segmentation algorithm and its main procedures. Section 3 shows the experiments and discussion. Finally, the conclusions and future work directions are given in Section 4.

2. THE PROPOSED ALGORITHM

This section presents the proposed instance segmentation algorithm based on the improved DL algorithm for multi-object detection. We have focused on the multi-object instance segmentation by identifying multi-object tasks.

Figure 1 shows the proposed algorithm which consists of several improved algorithms, including the backbone, detection and instance segmentation algorithms in order to obtain the most significant results by accurately presenting the multi-object image. The mathematical presentation of the given problem is explained as follows.

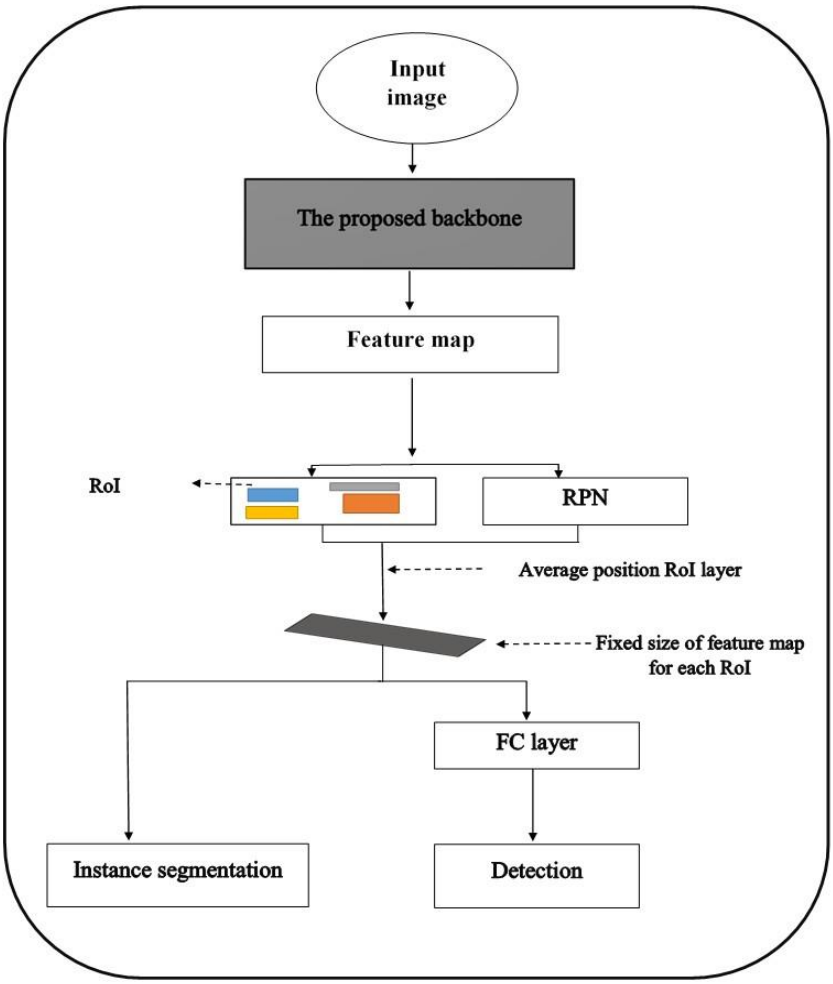

Figure 1. The proposed algorithm.

For each object, the overall loss function of the multi-object instance segmentation is calculated using Equation (1).

$$L = L_{detection} + L_{instance\ segmentation} \tag{1}$$

where $L_{detection}$ loss function is given according to Equation (2).

$$L(p_i, t_i) = \frac{1}{N}\sum_{i=1} |p_i^* - p_i|^2 + \lambda \frac{1}{N}\sum_{i} p^* L_{loc}(t_i, t_i^*) \tag{2}$$

where *pi* is the predicted probability of anchor *i*, p_i^* is the ground truth of anchor *i*, t_i stands for the coordinates predicted, t_i^* is the coordinates ground truth, *N* is the normalization term and λ = *a* is the balancing parameter.

The $L_{instance\ segmentation}$ loss function is identified using the per-pixel sigmoid and average binary cross-entropy to generate boundaries for each class, as shown in Equation (3).

$$L_{instance\ segmentation} = -\frac{1}{s^2}\sum_{1\le i,j\le s} [y_{ij} log y_{ij}^k + (1 - y_{ij} \log(1 - y_{ij}^k)] \tag{3}$$

where y_{ij} is the ground truth of boundaries of size region (s^2), y_{ij}^k is the predicted value of boundaries and *k* is the ground truth class.

The pseudocode of the proposed algorithm is as shown in Algorithm 1.

Algorithm 1: Pseudocode of the proposed algorithm

```
block 1, block 2, block 3, block 4, block 5 = build proposed backbone()
anchors = generate-anchors()
rpn = build-rpn()
rois = Proposal-Layer(rpn, anchors)
if mode == 'training': then
    ground-truth-values = values from the training dataset bbox,
    classes = classifier(rois)
    target-detection = Detection-Target-Layer(ground-truth-values) instance-
    segmention = instance-segmention (rois from target detection)
    loss = loss-functions(target-detection, bbox, classes, instance-segmention ) algorithm
    = [bbox, classes, instance-segmention, loss]
else
    bbox, classes = classifier(rois)
    target-detection = Detection-Layer(bbox, classes)
    instance-segmention = instance-segmention (rois)
    algorithm = [bbox, classes, instance-segmention]
end
return algorithm
```

2.1 The Proposed Backbone

The ResNet has attracted significant interest and attention due to its ability to handle complex data and high accuracy compared to other backbones [31]; it consists of a series of blocks to overcome the problem of the vanishing of gradient [31]. Therefore, there are problems with ResNet backbone, such as: 1) determination of ResNet block that has failed to receive sufficient training, 2) determination of ResNet block that has received more than sufficient training and 3) adopting a large filter size in the first convolution layer. In this paper, ResNet is improved in the proposed backbone as the fundamental building block based on the proposed Equation 4. The ResNet block is fed forward directly and linked to all other layers, which consist of a series of blocks to overcome the vanishing of the gradient.

$$H(y) = \sum_{I=1}^{n} F(y, \{y_i, I\}) + y \tag{4}$$

where *y* is the building block's input, *H(y)* is the block's output, *F(y, W)* is the remaining mapping that you acquired during the training stage and *I* represents the number of iterations to every ResNet block. In the case of a layer of insufficient training, *I* should be raised, while additional training must be decreased. The chosen filter size in the first convolution layer has been smaller than ResNet due to the feature map's extraction. In the proposed backbone, the first convolution layer uses a 5x5 filter size accompanied by max-pooling of the 2x2 matrix to obtain more features, as shown in Figure 2. In order to further enhance information flow across layers, the performance of the convolution layer is used for the input to the ResNet block of the proposed ResNet backbone.

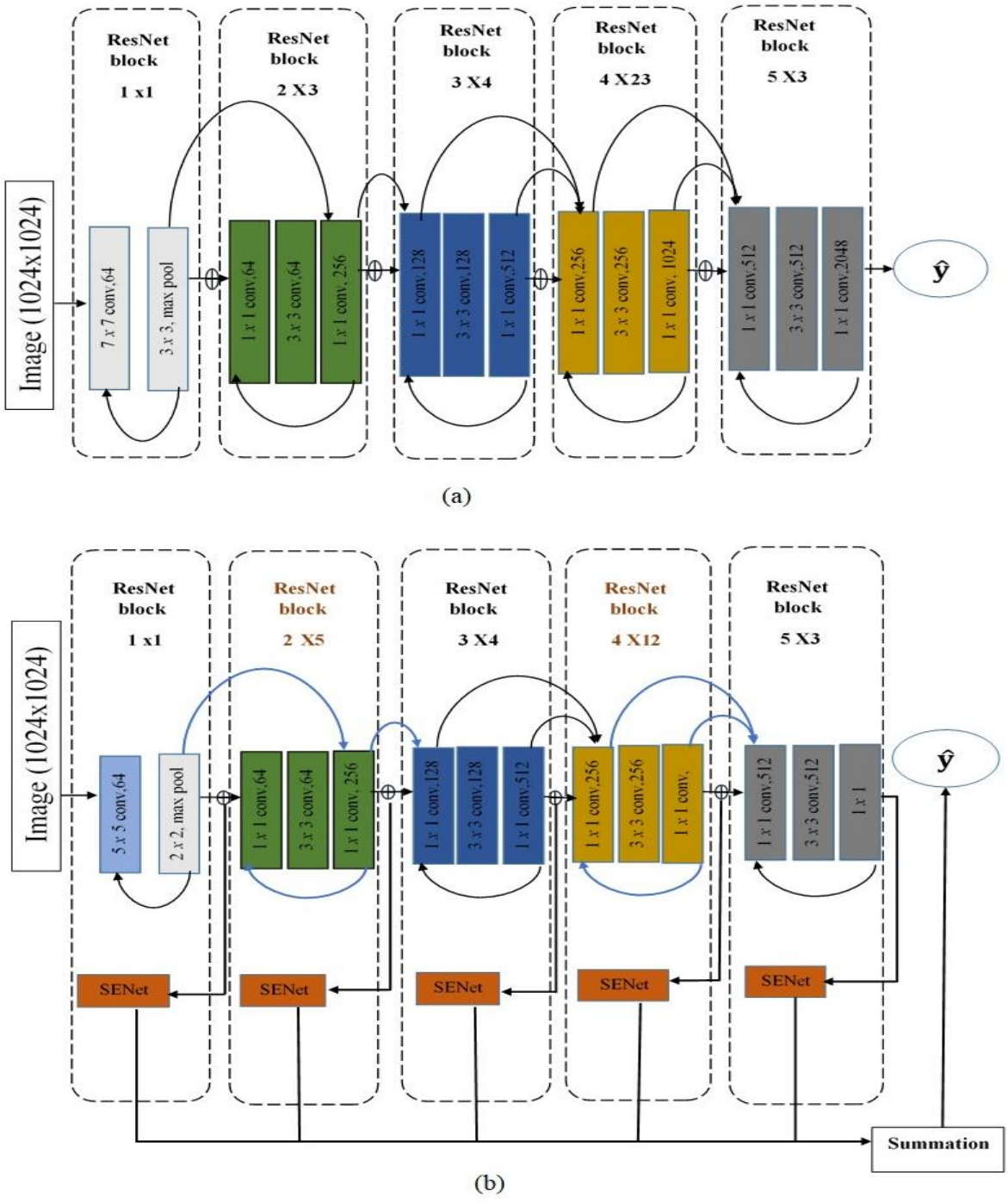

Figure 2. (a) The existing ResNet-101 backbone [31] (b) The improvement ResNet backbone.

For incorporating a content-aware mechanism to weigh each channel adaptively, the production of each proposed ResNet block is transmitted *via* the SENet network to provide a linear scalar of how relevant each proposed ResNet block is, which can be written as:

$$F_{feature\ map} = \sum_{i=1}^{W}\sum_{j=1}^{H} y_q(i,j) \tag{5}$$

where y_q is the element of the feature map with spatial dimension $H \times W$, where H is the height and W is the width.

Sequentially, we have obtained five samples of the building blocks consisting of ResNet blocks and a network SENet. Outputs are integrated from each of the SENet networks to combine all features from various depth levels by summation of the characteristics of feature maps derived from the five couples of the ResNet blocks, as shown in Figure 2. In the following sub-section, the architecture of SENet will be addressed.

2.1.1 The architecture of SENet

The feature-generating maps from the ResNet improvement have been fed to the SENet network [32] to obtain further channel information and enhance the sharing of information, as shown in Figure 3. It selectively uses global information to illustrate and eliminate less valuable features by using weights on each feature map's layer. It contains five operations, including a global average pooling, an FC layer, an ReLU function, an FC layer and the sigmoid function. The role of the sigmoid activation for channel weights is suited to the input. The SENet architecture is illustrated in Figure 3. As represented in Figure 3, the SENet architecture mainly consists of two processes, which are:

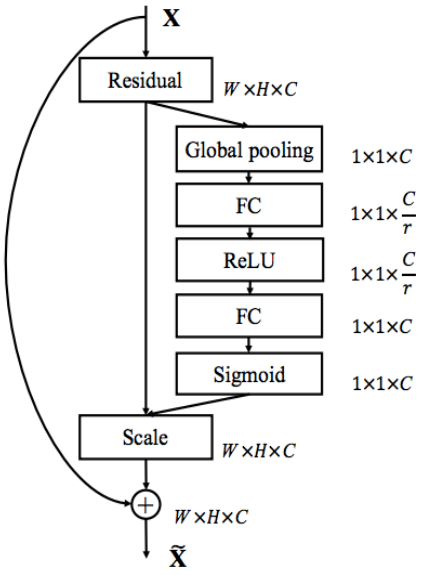

Figure 3. The SENet architecture [32].

- The squeezing process: It produces channel-wise statistics (Se $\in R^D$) through global average pooling, which can be written as:

$$Se = F_{SENet}(y_q) = \frac{1}{H \times W}\sum_{i=1}^{W}\sum_{j=1}^{H} y_q(i,j) \tag{6}$$

where F_{SENet} (.) is the function of squeezing. y_q is the element of the feature map with spatial dimensions $H \times W$, where y_q is the q^{th} element of Se and q = 1, 2, ..., D.

- Excitation process: It provides and identifies channel-wise dependencies and significantly minimizes the number of parameters through FC layers, sigmoid and ReLU functions, as shown in the following formula.

$$T = F_{excitation}(Se, W) = \sigma\big(G(Se, W)\big) = \sigma\big(W_2\,\delta(W_1 Se)\big) \quad (7)$$

where $T = t_1, t_2, \ldots, t_D$, $F_{\text{excitation}}$ is the function of excitation and $t_q \in R^{H\times W}$. $\delta(x) = MAX(x, 0)$ is reference to ReLU function, $G(.,.)$ is the reference to global function and $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigma mode function.

$$\hat{P} = F_{scale}(Se_q, y_q) = Se_q . y_q, \quad (8)$$

where $y_q \in \mathrm{R}^{H\times W}$ and F_{scale} is the reference to channel-wise multiplication between the scalar Se_q and the feature map y_q.

2.2 Localization

The RPN is implemented in the multi-object form to decide the location of multiple objects in the input image [24]. Besides, the RPN approved any sizes of the feature map that would act as the output. In the meantime, the proposed CNN functions as the input to produce multiple proposals for rectangular objects. The object is illustrated in the current technique for rectangular objects, while the sliding window is seen in all the feature maps collected by the proposed CNN's last convolution layer. The sliding window in RPN comprises nine anchors, which are the center points of the sliding window. In particular, the location for each anchor is calculated based on the input image, which gives the sliding window different Aspect Ratio (AR) and Scale (S) values, as seen in Figure 4.

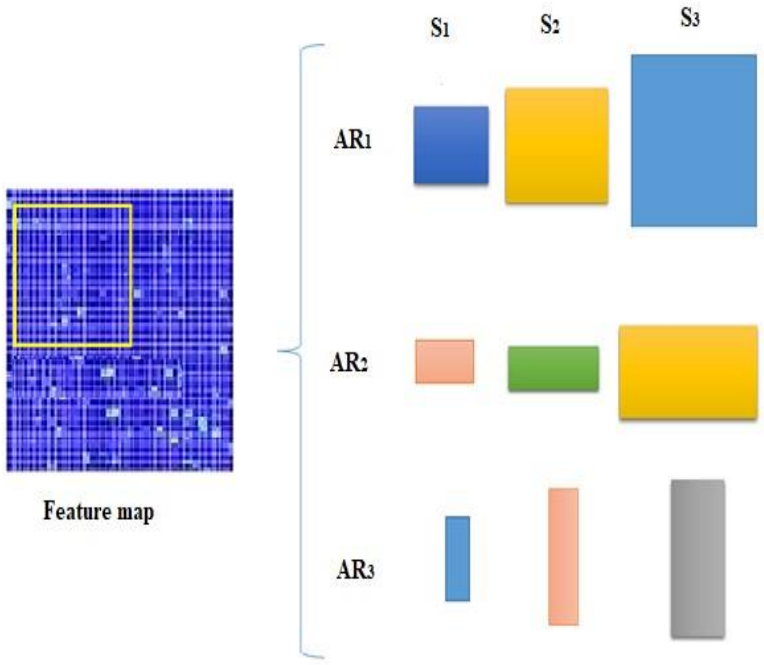

Figure 4. The sliding window with different aspects in ratio and scale.

As a consequence, the value of p^* to every anchor is determined on the two parameters that are as follows:

1) The anchors with the largest intersection-over-union overlap and a ground truth box.
2) For each anchor, the overlap intersection-over-union (IoU) is higher than 0.7.

The IoU is defined by the following formula:

$$IoU = \frac{Anchor \cap ground\ truth\ box}{Anchor \cup ground\ truth\ box} \quad (9)$$

2.3 Average Position Region of Interest Pooling Layer (Average Position RoI Layer)

Several propositions of rectangular objects are created on feature maps from RPN, as represented in Figure 1. Consequently, various map size features are designed, which affected the instance segmentation accuracy. This paper, therefore, has suggested a novel layer for handling the feature map's different sizes. The function map has been reduced over the following two steps to a fixed scale, known as the average position RoI layer. Suppose that the feature map's size is 5x5, where the rectangular object proposals are encoded in red colour as represented in Figure 5.

0.2	0.05	0.03	0.27	0.53
0.2	0.23	0.32	0.34	0.19
0.65	0.76	0. 26	0.26	0.25
0.55	0.58	0.25	0.18	0.15
0.39	0.29	0.2	0.38	0.55

Figure 5. The rectangular object proposals in red colour on the feature map.

The first move is to preserve the position of feature maps by stopping implemented quantification to each RoI boundary *via* the RoI Pool [25], as represented in Figures 6 and 7. Nevertheless, because of the strong quantization levels for every pixel and success in order to achieve optimal performance in segmentation, low performance is found in the RoI Pool segmentation [25]. This paper has solved the question of misalignment by annulling quantization. For each bin, the second step has utilized average pooling to reduce computational complexity and extract low-level features from the neighbourhood, as represented in Figure 8.

0.2	0.05	0.03	0.27	0.53
0.2	0.23	0.32	0.34	0.19
0.65	0.76	0. 26	0.26	0.25
0.55	0.58	0.25	0.18	0.15
0.39	0.29	0.2	0.38	0.55

(a)

0.2	0.05	0.03	0.27	0.53
0.2	0.23	0.32	0.34	0.19
0.65	0.76	0. 26	0.26	0.25
0.55	0.58	0.25	0.18	0.15
0.39	0.29	0.2	0.38	0.55

(b)

Figure 6. (a) The RoI Pool after implementing of quantization (b) The prevention of quantization by average position RoI.

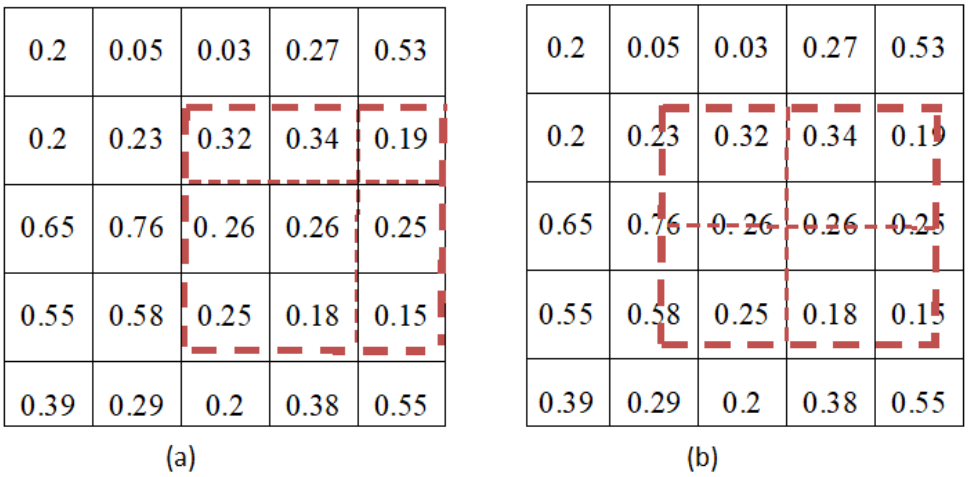

Figure 7. (a) The RoI Pool after the second implementation of quantization (b) The second prevention of quantization by average position RoI.

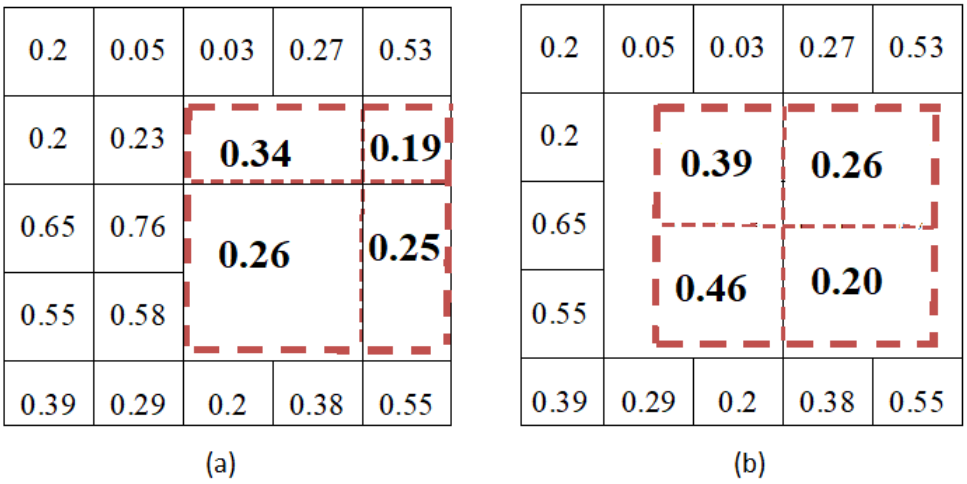

Figure 8. (a) The RoI Pool result (b) The average position RoI result.

After that, the result of the average position RoI layer is passed to the FC layer. This process consists of detection for each object element, as shown in Figure 1.

2.4 Instance Segmentation

The FCN [30] is done to differentiate between class levels, while the segmentation of the instance has returned the boundaries of each object element. This process consists of three stages, which are as follows:

1) The first stage consists of translating the average position RoI layer performance for the object item into a sequence of 3 x 3 convolution stages, multiple times after applying ReLU for the creation of limits for each area from the average position RoI layer.
2) The second stage requires a 1 x 1 convolution layer to every feature map acquired from the last convolution.
3) The third stage converts the segmentation dimension based on the input image by bi-linear interpolation.

3. Results and Discussion

The experimental findings and the proposed algorithm evaluation from the preceding sections are summarized in this section. The multi-object instance segmentation algorithm assessment is carried out

using the following measurements: AP with specific thresholds is used to evaluate the multi-object instance segmentation algorithms, including improved CNN, RPN and instance segmentation. To ensure a judgment on the enhanced segmentation of multi-object instances, the outcome is compared with those of other well-known algorithms that have been used for the multi-object instance.

3.1 Benchmark Datasets

The experiments are conducted on MS-COCO dataset [33], which includes 1118k images for training, 5k for validation (Val) and 20k for annotated testing (test-dev). The calculation of COCO AP was done from 0.5 to 0.95, with an interval of 0.05. All models have been trained on the COCO training set and tested on the Val set. For a fair comparison, the final results are compared with the state-of-the-art instance segmentation algorithm on the test-dev package.

3.2 Experimental Specifications

A new algorithm for the instance segmentation of multi-object instances is introduced using TensorFlow [34]. The algorithms are tested with the GPU-Us-Tesla V100 16 GB and the VCPUs-8 cores 61 GB on Amazon Web Services (AWS) and Amazon Machine Image (AMI). In the training stage, the RoI is determined positive if it has IoU with a ground-truth box of at least 0.5 and negative otherwise and the L segmentation loss function is defined on positive RoIs. The instance segmentation is the intersection between an RoI and its associated ground-truth of instances segmentation. We train on GPU-Us-Tesla V100 16 GB and the VCPUs-8 cores 61 GB on Amazon Web Services (AWS) and Amazon Machine Image (AMI). The weight decay for 50 epochs was 0.0001 with a learning momentum of 0.9 and a learning rate of 0.001. Every one of the epochs is an iteration of 1000. Besides, the optimization algorithm used in the context of this analysis is Stochastic Gradient Descent (SGD) [27].

3.3 Comparison with the State-of-the-Art Layers

In this sub-section, a comparison of the proposed algorithm with the state-of-the-art layers is conducted to validate the average position RoI layer ability. Table 1 compares the performance of the average position RoI layer with state-of-the-art layers.

Table 1. Evaluation results of the proposed backbone with various RoI layers.

RoI layers	Instance segmentation			Detection		
	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
RoI pooling [25]	31.0	53.6	30.12	35.6	60.1	34.3
RoI wrap [26]	28.3	50.2	28.1	31.8	53.7	31.6
Average position RoI layer	**45.8**	**66.7**	**47.1**	**47.6**	**68.6**	**47.1**

Based on the results, the average position RoI layer ability has obtained high accuracy with different AP values. The feature maps are reduced to a fixed size while maintaining the map's location obtained from the previous algorithms by avoiding quantization, while RoI Wrap and RoI pooling are still considered quantization in the RoI boundary, which leads to losing alignment with the input image. As a result, significant results in the detection and instance segmentation are obtained due to the impact on the detection and instance segmentation process for each pixel value in the feature maps.

3.4 Comparison with the State-of-the-Art Detection Algorithms

We compare our proposed algorithm with the state-of-the-art algorithms on COCO detection to validate the proposed algorithm's ability.

Table 2. Multi-object detection algorithm performance with different thresholds (0.5, 0.75, Small (S), Medium (M), Large (L)).

Algorithms	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Fast R-CNN	VGG	19.7	35.9	-	-	-	-
Faster R-CNN	VGG	21.9	42.7	-	-	-	-
Faster R-CNN	ResNet-101	34.9	55.7	37.4	15.6	38.7	50.9
Mask R-CNN	ResNet-101	38.2	60.3	41.7	20.1	41.1	50.2
Cascade R-CNN	ResNet-101	42.8	62.1	46.3	23.7	45.5	55.2
Proposed algorithm	**ResNet-50**	**40.5**	**62.7**	**43.3**	**23.5**	**42.8**	**51.5**
Proposed algorithm	**ResNet-101**	**44.8**	**66.0**	**46.6**	**30.2**	**48.1**	**56.8**
Proposed algorithm	**Proposed backbone without SENet**	**45.5**	**66.4**	**46.8**	**30.4**	**48.9**	**57**
Proposed algorithm	**Proposed backbone with SENet**	**47.6**	**68.6**	**47.1**	**32.8**	**50.8**	**58.9**

The efficiency of the proposed algorithm indicates better results with different AP values. The proposed algorithm AP accuracy has amounted to 47.6, 68.6, 47.1, 32.8, 50.8 and 58.9. Particularly, these values are considerably higher than comparable algorithm values due to the recommendation of a new backbone in this paper, which addresses insufficient training and determines the best possible filter size. Additionally, inserting an SENet network for each block helps increase the efficiency of the method's productivity by capturing several local and precise features from the input image.

3.5 Comparison with the State-of-the-Art Instance Segmentation Algorithms

In this sub-section, a comparison of the proposed algorithm with the state-of-the-art algorithms is conducted to validate the proposed algorithm's ability.

It is clear from Table 3 that the proposed algorithm is compared with the state-of-the-art algorithms, including MNS [26], FCIS [25], Mask R-CNN [27], YOLACT [28] and CASCADE R-CNN [29].

Table 3. Multi-object instance segmentation algorithm performance with different thresholds (0.5, 0.75, Small (S), Medium (M), Large (L)).

Algorithms	**Backbone**	**AP**	**AP_{50}**	**AP_{75}**	**AP_S**	**AP_M**	**AP_L**
MNC	ResNet-101	24.6	44.3	24.8	4.7	25.9	43.6
FCIS	ResNet-101	29.2	49.5	-	7.1	31.3	50.0
Mask R-CNN	ResNet-101	35.7	58.0	37.8	15.5	38.1	52.4
YOLACT	ResNet-101	31.2	50.6	32.8	12.1	33.3	47.1
CASCADE R-CNN	ResNet-101	42.8	62.1	46.3	23.7	45.5	55.2
Proposed algorithm	ResNet-50	33.8	55.1	36.4	19.4	42.2	51.1
Proposed algorithm	ResNet-101	36.3	57.4	38.7	18.8	40.5	53.8
Proposed algorithm	**Proposed backbone without SENet**	**43.9**	**64.8**	**45.3**	**22.4**	**44.4**	**54**
Proposed algorithm	**Proposed backbone with SENet**	**45.8**	**66.7**	**47.1**	**24.3**	**46.2**	**55.9**

The performance of the proposed algorithm has been found to exhibit better results with different AP values. Our algorithm's AP accuracy has amounted to 45.8, 66.7, 47.1, 24.3, 46.2 and 55.9. Notably, these values are significantly higher than the comparative algorithms' values due to the suggestion of a new backbone in this paper, which is essential to achieve better results.

The proposed algorithm has solved gradient vanishing *via* an identity shortcut based on a new gradient equation while taking into account the efficiency training for each convolution block.

This function is considered to remove a certain amount of feature from the input image and transfer it to other layers by means of a similar duplicate increase or decrease in training and reduction in the filter size. Also, the SENet network has increased the performance of the feature selection process. The proposed

backbone incorporates deeper and shallow feature maps. Because there are several local and accurate feature maps on the shallow layers, there are also rich feature maps on the deep network layers. Consequently, with the proposed backbone, we should effectively collect feature maps in different improved ResNet blocks and transfer them to the SENet for additional spatial feature maps.

Compared to other algorithms, the results obtained from the produced function are positive results. In addition to the structure, this paper has also proposed a novel layer for managing different sizes of the feature map created from the RPN, known as the average position RoI. Furthermore, the average position RoI layer has reduced the feature maps to a fixed size while preserving the map's position obtained from previous algorithms by avoiding quantization.

As a result, significant results in the instance segmentation are obtained due to the impact on the instance segmentation process of every pixel value in the feature maps. Given the significant success of the proposed algorithm in the multi-object segmentation setting, it is expected that the proposed algorithm obtains the potential for substantial results. The visual experimental results are presented in Figure 9.

Figure 9. The visual experimental results from the proposed algorithm for instance segmentation.

3.6 Time Measure

In this sub-section, the time measure is used to evaluate the proposed algorithm's training time and frame per second in the testing process. The training and frame per second are significant factors in evaluating algorithm efficiency. The comparative algorithms are evaluated in terms of the training time in second per image and frame per second, as shown in Table 4.

Table 4. Evaluation of training time in second per image and frame per second of multi-object segmentation using different state-of-the-art algorithms.

Algorithms	Training time in second	Frame per second
MNC	0.21	5.45
FCIS	0.14	5.75
Mask R-CNN	0.11	6.07
YOLACT	0.27	29.5
CASCADE R-CNN	0.41	8.03
Proposed algorithm	0.10	8.71

Based on the evaluation frame per second in the testing process of multi-object instance segmentation algorithm with different state-of-the-art algorithms, the proposed algorithm has produced 8.71 frames per second, making it the second algorithm following YOLACT algorithm despite of the YOLACT algorithm's high-speed characterization in this process. At the same time, the precision is omitted in AP50, AP75, AP90, APS, APM and APL due to its reliance on the proposed backbone, which has resulted in the avoidance of the issue in ResNet. This trend has happened with the reduction of the filter size and the emphasis on the layer, allowing further training and reduction in the amount of excessive layer testing to obtain good results in the shortest possible time. In contrast, the MNS, FCIS, Mask R-CNN and CASCADE R-CNN algorithms have produced less frames per second due to ResNet101 implementation as the backbone network. ResNet has faced many problems, including using a large filter size affecting the increased consumption time parameters, as stated earlier. In addition, several layers have not undergone preparation, which has resulted in a substantial time investment in the training cycle, due to the three-fold repetition of the bounding box by the algorithm.

4. CONCLUSION

Multi-object image is improved by extracting features low and large *via* creating a novel backbone by several connected copies of the ResNet blocks of enhancement ResNet network connected with SENet to obtain additional channel features, improve the use of a more significant feature in images and provide a linear scalar of how relevant each proposed ResNet block is. The second phase is to adopt RPN to locate the object item and create a new layer called the average position RoI layer, which manages to map various features to obtain the best boundaries for the multi-object image. In the third phase, the FCN is used to generate instance segmentation to prevent the overlapping problem between objects. This prevention manages the various sizes of RPN's feature map using the average position RoI layer. So, the segmentation of multiple-object instances is the main aim of this research. The proposed multi-object instance segmentation algorithm is evaluated using two measures; AP with different thresholds and training time and frames per second. The proposed algorithm's obtained results are compared against other well-known segmentation algorithms published in the literature, such as MNC, FCIS, Mask R-CNN, YOLACT and Cascade R-CNN. Better performance regarding the accuracy AP with different thresholds; namely, AP50, AP75, AP90, APS, APM and APL, is observed from the proposed algorithm. Such thresholds are higher than those of the state-of-the-art instance segmentation algorithms. Notably, the proposed system has rapidly and reliably defined, located and segmented the multi-object precision, preparation, training and frames per second. The proposed algorithm can be enhanced in future work by adding an edge detection algorithm to capture multiple objects' fine details. Also, we will implement variations of CNN architectures to investigate instance segmentation.

REFERENCES

[1] Q. Zhang, L. T. Yang, Z. Chen and P. Li, "A Survey on Deep Learning for Big Data," Information Fusion, vol. 42, pp. 146–157, 2018.

[2] L. Liu, W. Ouyang et al., "Deep Learning for Generic Object Detection: A Survey," International Journal of Computer Vision, vol. 128, pp. 261–318, 2020.

[3] N. F. F. Alshdaifat, A. Z. Talib and M. A. Osman, "Improved Deep Learning Framework for Fish Segmentation in Underwater Videos," Ecological Informatics, vol. 59, p. 101121, DOI: 10.1016/j.ecoinf.2020.101121 2020.

[4] Z.-C. He, L.-Y. An et al., "Comment on “Deep Learning Computer Vision Algorithm for Detecting Kidney Stone Composition”," World Journal of Urology, DOI: 10.1007/s00345-020-03181-4, April 2020.

[5] A. B. Nassif, I. Shahin et al., "Speech Recognition Using Deep Neural Networks: A Systematic Review," IEEE Access, vol. 7, pp. 19143–19165, 2019.

[6] J. Jiang and H. H. Wang, "Application Intelligent Search and Recommendation System Based on Speech Recognition Technology," International Journal of Speech Technology, pp. 1–8, DOI: 10.1007/s10772-020-09703-0, April 2020.

[7] M. Zhou, X. Wei, S. Kwong et al., "Rate Control Method Based on Deep Reinforcement Learning for Dynamic Video Sequences in HEVC," IEEE Transactions on Multimedia, pp. 1-1, DOI: 10.1109/TMM.2020.2992968, May 2020.

[8] S. F. A. Abuowaida and H. Y. Chan, "Improved Deep Learning Architecture for Depth Estimation from Single Image," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 6, no. 4, pp. 434-445, 2020.

[9] R. S. T. Lee, "Natural Language Processing," in: Artificial Intelligence in Daily Life Book, pp. 157–192, ISBN 978-981-15-7695-9, Springer, 2020.

[10] K. Shuang, Z. Zhang et al., "Convolution–deconvolution Word Embedding: An End-to-end Multi-prototype Fusion Embedding Method for Natural Language Processing," Information Fusion, vol. 53, pp. 112–122, DOI: 10.1016/j.inffus.2019.06.009, 2020.

[11] Spyridon Thermos et al., "Deep Sensorimotor Learning for RGB-D Object Recognition," Computer Vision and Image Understanding, vol. 190, p. 102844, DOI: 10.1016/j.cviu.2019.102844, 2020.

[12] N. Wang, Y. Wang and M. J. Er, "Review on Deep Learning Techniques for Marine Object Recognition: Architectures and Algorithms," Control Engineering Practice, p. 104458, DOI: 10.1016/j.conengprac.2020.104458, 2020.

[13] Qiaoyong Zhong et al., "Cascade Region Proposal and Global Context for Deep Object Detection," Neurocomputing, vol. 395, pp. 170–177, 2020.

[14] Francisco Pérez-Hernández et al., "Object Detection Binary Classifiers Methodology Based on Deep Learning to Identify Small Objects Handled Similarly: Application in Video Surveillance," Knowledge-based Systems, vol. 194, p. 105590, DOI: 10.1016/j.knosys.2020.105590, 2020.

[15] M. Rezaei, H. Yang and C. Meinel, "Recurrent Generative Adversarial Network for Learning Imbalanced Medical Image Semantic Segmentation," Multimedia Tools and Applications, vol. 79, pp. 15329–15348, DOI: 10.1007/s11042-019-7305-1, 2020.

[16] B. Xu, W. Wang, G. Valzon et al., "Automated Cattle Counting Using Mask R-CNN in Quadcopter Vision System," Computers and Electronics in Agriculture, vol. 171, p. 105300, 2020.

[17] M. Bellver, A. Salvador, J. Torres et al., "Mask-guided Sample Selection for Semi-supervised Instance Segmentation," Multimedia Tools and Applications, vol. 79, pp. 25551–25569, DOI: 10.1007/s11042-020-09235-4, 2020.

[18] D. Larlus, J. Verbeek and F. Jurie, "Category Level Object Segmentation by Combining Bag-of-words Models with Dirichlet Processes and Random Fields," International Journal of Computer Vision, vol. 88, pp. 238–253, DOI: 10.1007/s11263-009-0245-x, 2010.

[19] X. Zhao, Y. Satoh et al., "Object Detection Based on a Robust and Accurate Statistical Multipoint-pair Model," Pattern Recognition, vol. 44, no. 6, pp. 1296–1311, 2011.

[20] J. Walsh, N. O'Mahony et al., "Deep Learning *vs.* Traditional Computer Vision," Proc. of the Science and Information Conference (CVC), pp. 128–144, DOI: 10.1007/978-3-030-17795-9_10, Springer, Las Vegas, USA, 2019.

[21] Z. Xue, D. Ming et al., "Infrared Gait Recognition Based on Wavelet Transform and Support Vector Machine," Pattern Recognition, vol. 43, no. 8, pp. 2904–2910, DOI: 10.1016/j.patcog.2010.03.011, 2010.

[22] R. Girshick, J. Donahue et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587, DOI: 10.1109/CVPR.2014.81, Columbus, USA, 2014.

[23] R. Girshick, "Fast R-CNN," Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448, DOI: 10.1109/ICCV.2015.169, Santiago, Chile, 2015.

[24] S. Ren, K. He et al., "Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks," Advances in Neural Information Processing Systems, pp. 91–99, [Online], Available: https://arxiv.org/pdf/1506.01497.pdf, 2015.

[25] Yi Li et al., "Fully Convolutional Instance-aware Semantic Segmentation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2359–2367, DOI: 10.1109/CVPR.2017.472, Honolulu, USA, 2017.

[26] J. Dai, K. He and J. Sun, "Instance-aware Semantic Segmentation *via* Multi-task Network Cascades," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3150–3158, DOI: 10.1109/CVPR.2016.343, Las Vegas, 2016.

[27] K. He et al., "Mask R-CNN," Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2961–2969, DOI: 10.1109/ICCV.2017.322, Venice, Italy, 2017.

[28] D. Bolya, C. Zhou et al., "YOLACT: Real-time Instance Segmentation," Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 9157–9166, DOI: 10.1109/ICCV.2019.00925, Seoul, Korea (South), 2019.

[29] Zhaowei Cai and Nuno Vasconcelos. "Cascade R-CNN: High Quality Object Detection and Instance Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019.

[30] J. Long, E. Shelhamer and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3431–3440, DOI: 10.1109/CVPR.2015.7298965, Boston, USA, 2015.

[31] K. He, et al., "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, DOI: 10.1109/CVPR.2016.90, Las Vegas, USA, 2016.

[32] J. Hu, L. Shen and G. Sun, "Squeeze-and-excitation Networks," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7132–7141, DOI: 10.1109/CVPR.2018.00745, Salt Lake City, USA, 2018.

[33] T.-Y. Lin, M. Maire et al., "Microsoft COCO: Common Objects in Context," Proc. of the European Conference on Computer Vision (ECCV), pp. 740–755, DOI: 10.1007/978-3-319-10602-1_48, Part of the Lecture Notes in Computer Science Book Series (LNCS, vol. 8693), Springer, 2014.

[34] M. Abadi, A. Agarwal, P. Barham et al., "TensorFlow: Large-scale Machine Learning on Heterogeneous Distributed Systems," Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation (OSDI'16), pp. 265-283, arXiv preprint arXiv: 1603.04467, 2016.

" A Novel Instance Segmentation Algorithm Based on Improved Deep Learning Algorithm For Multi-object Images", S. F. A. Abuowaida, H. Y. Chan, N. F. F. Alshdaifat and L. Abualigah.

ملخص البحث:

تعدّ خوارزميات التعلُّم العميق واسعة الانتشار وجذّابة في السنوات الأخيرة، وذلك بالنظر الى ما تحقّق بفضلها من إنجازاتٍ في العديد من المجالات. ويكمن التعلُّم العميق في الكشف المستند الى الصُّور وتجزئة المراحل الخاصة بكيانٍ او شيء ما، وتلك مسألة حرجة في حاجةٍ الى المزيد من البحث والاستقصاء.

تهدف هذه الورقة الى اقتراح خوارزمية مبتكرة من أجل تجزئة المراحل في الصور متعددة المواضع، من ثلاث مراحل؛ بهدف البحث في التحديات التي تواجه استخدام خوارزميات تجزئة المراحل في الصور. في المرحلة الأولى، التي تشكل العمود الفقري أو الجزء الأساسي، يجري تحسين خوارزمية تمييز الصور عبر استخلاص المستويات الدنيا والعليا المميزة من الصور موضوع المعالجة. ووحدة البناء الأساسية هي تلك الشبكة المسماة (ResNet)، وهي تتصل مع شبكة الضغط والإثارة التي تعرف باسم (SENet) لكل وحدةٍ من وحدات شبكة (ResNet). وتستخدم شبكة اقتراح المنطقة (RPN) للعمل على تحديد موضع العُنصر الهدف، وتتبعها المرحلة الثالثة التي تقترح طبقة الموضع (RoI) من أجل اختيار الحدود المثالية للتجزئة.

تم إجراء التجارب وتقييمها باستخدام قاعدة بياناتٍ مرجعية للصور تُعرف باسم (COCO). وجرى تقييم الخوارزمية المقترحة من حيث الأداء باستخدام معايير تقييم مرجعية، ومقارنتها بعددٍ من خوارزميات تجزئة المراحل الحديثة الخاصة بالصور. وبينت النتائج أن الخوارزمية المقترحة في هذه الورقة كانت أفضل من خوارزميات تجزئةٍ أخرى معروفة جيداً تستخدم المراحل، وذلك من حيث متوسط الدقة باستخدام قياسات العتبة (AP)، وذلك لعتباتٍ مختلفة.