Scientific Research Publishing

Journal of Computer and Communications, 2022, 10, 115-132
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Overview of Object Detection Algorithms Using Convolutional Neural Networks

Junsong Ren, Yi Wang*

School of Computer Science and Engineering, Sichuan University of Science and Engineering, Yibing, China
Email: *38871010@qq.com

How to cite this paper: Ren, J.S. and Wang, Y. (2022) Overview of Object Detection Algorithms Using Convolutional Neural Networks. *Journal of Computer and Communications*, **10**, 115-132.
https://doi.org/10.4236/jcc.2022.101006

Received: December 6, 2021
Accepted: January 26, 2022
Published: January 29, 2022

Abstract

In today's world, computer vision technology has become a very important direction in the field of Internet applications. As one of the basic problems of computer vision, object detection has become the basis of many vision tasks. Whether we need to realize the interaction between images and text or recognize fine categories, it provides reliable information. This article reviews the development of object detection networks. Starting from RCNN, we introduce object detection based on candidate regions, including Fast R-CNN, Faster R-CNN, etc.; and then start to introduce single-shot networks including YOLO, SSD, and Retina Net, etc. Detectors are the most excellent methods at present. By reviewing the current research status of object detection networks, it provides suggestions for the further development trend and research of object detection.

Keywords

Deep Learning, Convolutional Neural Network, Object Detection, Computer Vision

1. Introduction

Convolutional Neural Network [1] (CNN) has made great progress in recent years and is a very bright pearl in the booming deep neural network treasure house. And computer vision technology allows artificial intelligence to have the ability of visual perception and understanding. In recent years, thanks to the improvement of computer hardware performance and the creation of large-scale image annotation data sets, computer vision algorithms based on deep learning have achieved great success in classic computer vision tasks such as image classification, object detection, and image segmentation.

At present, object detection has not only received a lot of research in academia, but also has been widely used in real life, such as video fire detection [2], unmanned driving [3], security monitoring [4], and UAV scene analysis [5]. At present, object detection algorithms are mainly divided into two types, traditional object detection algorithms based on image processing and object detection algorithms based on convolutional neural networks. In 2014, Girshick *et al.* proposed R-CNN [6] on this basis. For the first time, convolutional neural networks were applied to object detection, and the detection accuracy was improved by nearly 30% compared with traditional detection algorithms, which caused a great response. From the current academic research and practical application, the object detection algorithm based on the convolutional neural network has higher accuracy and shorter test time than the traditional method, and it has almost completely replaced the traditional algorithm.

2. Convolutional Neural Network

A common CNN example is displayed in **Figure 1**.

Convolutional neural networks are mainly composed of these types of layers: input layer, convolutional layer, ReLU layer, pooling layer, and fully connected layer (the fully connected layer is the same as the conventional neural network). By superimposing these layers, a complete convolutional neural network can be constructed. In practical applications, the convolutional layer and the ReLU layer are often collectively referred to as the convolutional layer, so the convolutional layer also passes through the activation function after the convolution operation. Specifically, when the convolutional layer and the fully connected layer perform transformation operations on the input, not only the activation function will be used, but also many parameters, namely the weight w and the deviation b of the neuron; and The ReLU layer and the pooling layer perform a fixed function operation. The parameters in the convolutional layer and the fully connected layer will be trained as the gradient drops so that the classification score calculated by the convolutional neural network can match the label of each image in the training set.

Convolutional neural networks have the concepts of local receptive fields [7], sparse weights, and parameter sharing. These three concepts make convolutional neural networks have a certain translation and scale invariance compared with other neural networks [8], and are more suitable for image data learning.

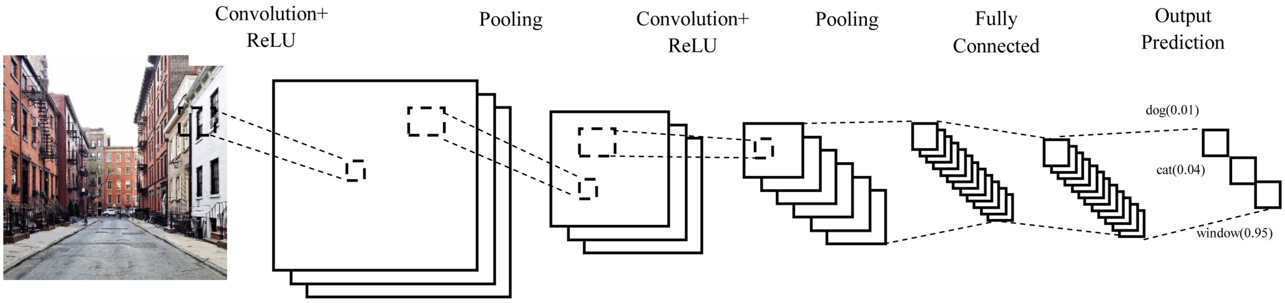

Figure 1. Architecture of CNN framework.

2.1. Convolutional Layer

The convolutional layer is the core layer to construct a convolutional neural network, which generates most of the calculations in the network. Note that the amount of calculation is not the number of parameters. The convolution operation can effectively reduce the training complexity of the network model and reduce the network connection and parameter weights, which makes it easier to train than a fully connected network of the same scale. Common convolution operations are as follows: ordinary convolution [9], transposed convolution [10], hole convolution [11], and depth separable convolution [12].

Ordinary convolution is a process in which the convolution kernel is used to slide on the image, and after a series of matrix operations, the process of calculating the gray value of all image pixels is finally completed. Transposed Convolution is the convolution method from low-dimensional feature mapping to high-dimensional feature mapping, which is the opposite of ordinary convolution. It is widely used in semantic analysis [13], image recognition [14], and other fields. Fractionally-Strided Convolution realizes the sampling operation of input features by reducing the step size of transposed convolution and improving the feature dimension. Transposed Convolution, also known as dilated convolution, is a convolution method that does not increase the number of parameters while increasing the receptive field [7] of the unit. Depthwise Separable Convolution is used in the lightweight network model MobileNets [12], where a single filter is applied to each input channel by depthwise convolution, and then pointwise convolution is used to combine the outputs of different depth convolutions. Depth separable convolution realizes the separation of channels and regions in ordinary convolution operations. This decomposition process can greatly reduce the amount of calculation and the size of the model.

2.2. Activation Layer

Activation Function is a function added to artificial neural networks to help the network learn complex patterns in data. Similar to the neuron-based model in the human brain, the activation function ultimately determines the content to be emitted to the next neuron.

Common activation functions include Rectified Linear Unit (ReLU) [15], Randomized LeakyReLU (RReLU) [16], Exponential Linear Units (ELU) [17] and so on. The linear rectification function ReLU is one of the most significant unsaturated activation functions. As shown in **Figure 2**, its mathematical expression is as follows:

$$f(x) = \max(0, x) \tag{1}$$

2.3. Pooling Layer

The pooling layer was first seen in the LeNet [18] article, called Subsample, and named after the publication of the AlexNet [15] paper. It is one of the commonly used components in current convolutional neural networks. The pooling layer is

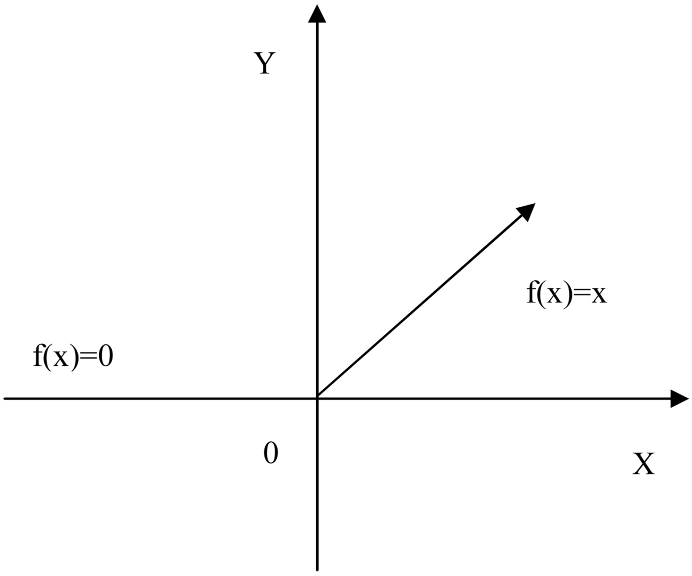

Figure 2. ReLU function image.

sandwiched between successive convolutional layers to compress the amount of data and parameters to reduce overfitting. If the input is an image, then the main function of the pooling layer is to compress the image.

The pooling layer can effectively reduce the size of the matrix, that is, it can perform collective statistical operations on the special diagnosis at different positions in the local area of the image, thereby alleviating the excessive sensitivity of the convolutional layer to the image position, reducing the parameters in the final fully connected layer, and speeding up calculation speed. The common operations of the pooling layer include the following: max-pooling [19], average pooling [20], Spatial Pyramid Pooling [18], etc.

3. Classification of Object Detection Algorithms

In recent years, with the development of deep learning, object detection algorithms have made great breakthroughs. The current popular object detection methods can be divided into two categories. One is the R-CNN algorithm based on Region Proposal, such as R-CNN, Fast R-CNN [21], Faster R-CNN [22], etc. They are two-stage and require the First use of heuristic methods for example Selective search [23], or CNN network to generate Region Proposal [22] and then perform classification and regression on Region Proposal. The other is one-stage algorithms such as Yolo [24] and SSD [25], which only use a CNN network to directly predict the categories and positions of different targets. The two-stage object detection algorithm needs to perform region extraction operations, first use the CNN backbone network to extract image features, then find possible candidate regions from the feature map, and finally perform sliding window operations on the candidate regions to further determine the target category and location information.

The one-stage object detection algorithm does not extract candidate regions through the intermediate layer, but performs feature extraction, target classification, and position regression in the entire convolutional network, and then obtains the target position and category. The recognition accuracy is slightly weaker than that of the two-stage object detection algorithm. Under the premise, the speed has been greatly improved. The development process of one-stage and two-stage algorithms is shown in Figure 3 and Figure 4, respectively.

4. Common Object Detection Network Model

4.1. R-CNN

In 2014, Ross Girshick proposed R-CNN [6], which uses a selective search algorithm to replace the sliding window, which solves the problem of window redundancy and reduces the time complexity of the algorithm. At the same time, the traditional hand-made feature extraction part is replaced with a convolutional neural network, which can more effectively extract the features of the image and improve the network's anti-interference ability.

RCNN first selects possible object frames from a set of object candidate frames through Selective Search algorithm, and then resizes the images in these selected object frames to a fixed size image, and feeds them to CNN The model (a CNN model trained on the ImageNet data set, such as AlexNet) extracts features, and finally sends the extracted features to the classifier to predict whether the image in the object frame has the target to be detected, and uses the regression to further predict Which category does the detection target belong to.

The performance of the R-CNN model has been greatly improved compared to traditional object detection algorithms, but there are also many problems. For example, R-CNN generates about 2000 candidate regions, and feature extraction takes too much time; convolutional neural networks require fixed-size input, and image cropping or stretching will cause loss of image information; training speed is slow, not only training image classification The network also needs to train the SVM [26] classifier and regressor. The structure of R-CNN network is shown in Figure 5.

4.2. SPPNet

In 2015, SPPNet [18] was published on IEEE. In R-CNN, to generate a vector of equal dimensions for all candidate regions, the candidate regions are forcibly scaled, which will destroy the proportional relationship of the image, which is

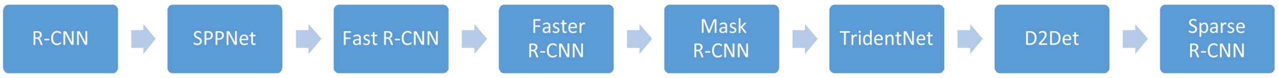

Figure 3. The development history of the two-stage object detection network framework.

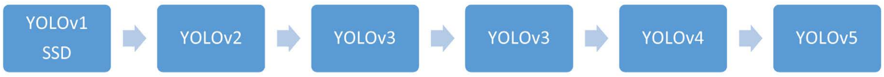

Figure 4. The development history of the one-stage object detection network framework.

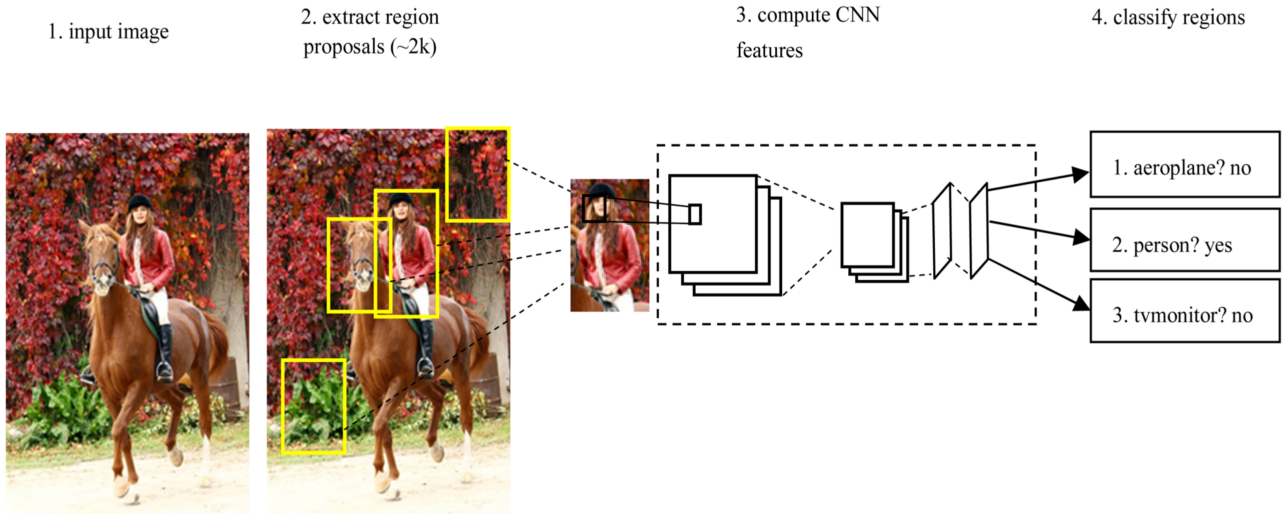

Figure 5. The architecture of the R-CNN framework.

not good for feature extraction, and this extraction process is quite time-consuming, so SPPNet is optimized here, using spatial pyramid pooling.

The spatial pyramid pooling layer in the figure below is the core of SPPNet, and its main purpose is to generate a fixed size output for any size input. The idea is to first divide a feature map of any size into 16, 4, or 1 blocks, and then pool the maximum on each block. The pooled features are spliced to obtain a fixed-dimensional output to meet the needs of the fully connected layer. Obviously, for images of different sizes, we get vectors of the same size. This is the advantage of spatial pyramid pooling. The structure of SPPNet network is shown in Figure 6.

4.3. Fast R-CNN

Fast R-CNN published in 2015, Comparing Fast R-CNN and R-CNN frameworks, it can be found that there are two main differences: one is that an ROI pooling layer is added after the last convolutional layer, and the other is that the loss function uses a multi-task loss function (multi-task loss), The Bounding Box Regression is directly added to the CNN network for training.

Fast R-CNN uses the CNN network to first extract the features of the entire image instead of extracting multiple times for each image block. Then, we can apply the method of creating candidate regions directly to the extracted feature maps. For example, Fast R-CNN chose the conv5 layer in VGG16 to generate the mapped feature block of the ROI region on the corresponding feature map and used it in the object detection task. We use ROI pooling to convert feature tiles to a fixed size and send them to the fully connected layer for classification and positioning. Because Fast-RCNN does not repeatedly extract features, it can significantly reduce processing time. Fast R-CNN uses Softmax Loss and Smooth L1 Loss to jointly train classification probability and Bounding box regression in the training process. The structure of Fast R-CNN network is shown in Figure 7.

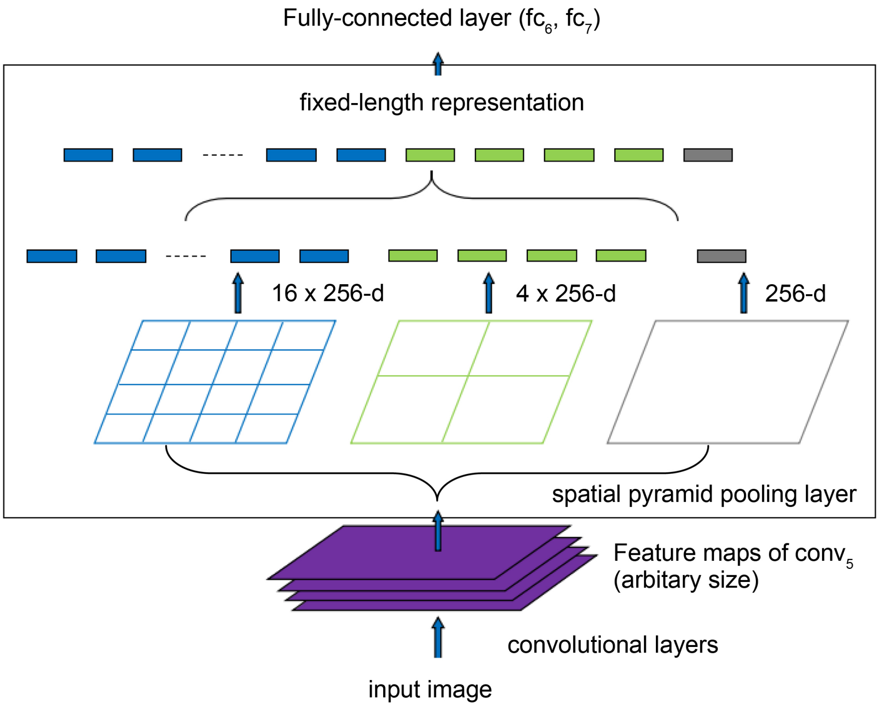
Fully-connected layer (fc_6, fc_7)
fixed-length representation
16 x 256-d
4 x 256-d
256-d
spatial pyramid pooling layer
Feature maps of $conv_5$
(arbitary size)
convolutional layers
input image

Figure 6. The architecture of the SPPNet framework.

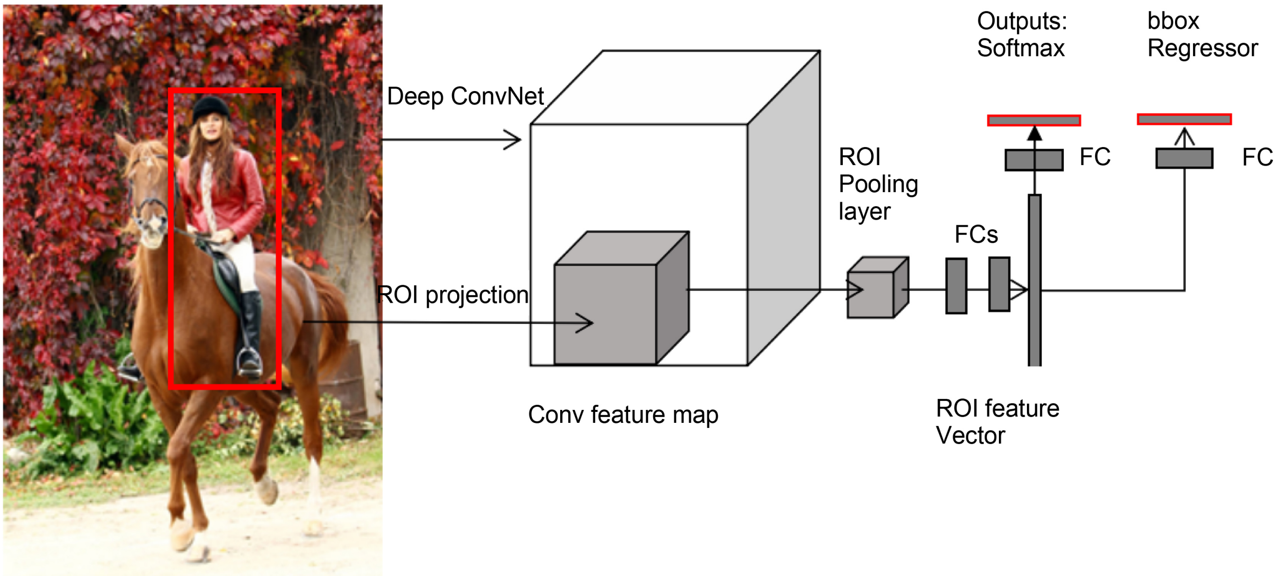

Figure 7. The architecture of Fast R-CNN framework.

4.4. Faster R-CNN

The problem with Fast R-CNN: There is a bottleneck: selective search to find all candidate boxes, which is also very time-consuming. To obtain these candidate frames more efficiently, Faster R-CNN has added a neural network region proposal network RPN (region proposal network) that extracts edges. After the convolutional neural network is added RPN, the work of finding candidate frames can be completed, the region proposal network As part of the Faster

R-CNN model, it is trained together with the entire model. Realizing the integration of candidate frame extraction into the deep network, RPN can learn how to generate high-quality proposed regions, reduce the number of proposed regions learned from the data, and still maintain the accuracy of object detection.

Faster R-CNN [22] is another masterpiece of the author Ross Girshick after Fast R-CNN. It also uses VGG-16 as the backbone of the network. The inference speed reaches 5fps on the GPU (including the generation of candidate regions), that is, it can detect every second of Five pictures, the accuracy rate has also been further improved, and won first place in multiple projects in the 2015 ILSVRC and COCO competitions. The structure of Faster R-CNN network is shown in Figure 8.

4.5. Mask R-CNN

HeKaiming launched Mask R-CNN [27] on ICCV in 2017. Mask R-CNN is an extension of the original Faster-RCNN, adding a branch to use existing detection to predict the target in parallel. At the same time, this network structure is relatively easy to implement and train, and the speed is relatively fast, and it can be easily applied to other fields, such as object detection, segmentation, and key point detection of people.

The image is first extracted through ResNet-FPN for feature extraction; then through the RPN network to predict Proposal; then RoI Align is used for feature extraction, then the classification and detection heads, and finally the Mask detection head, that is, each category predicts a Mask.

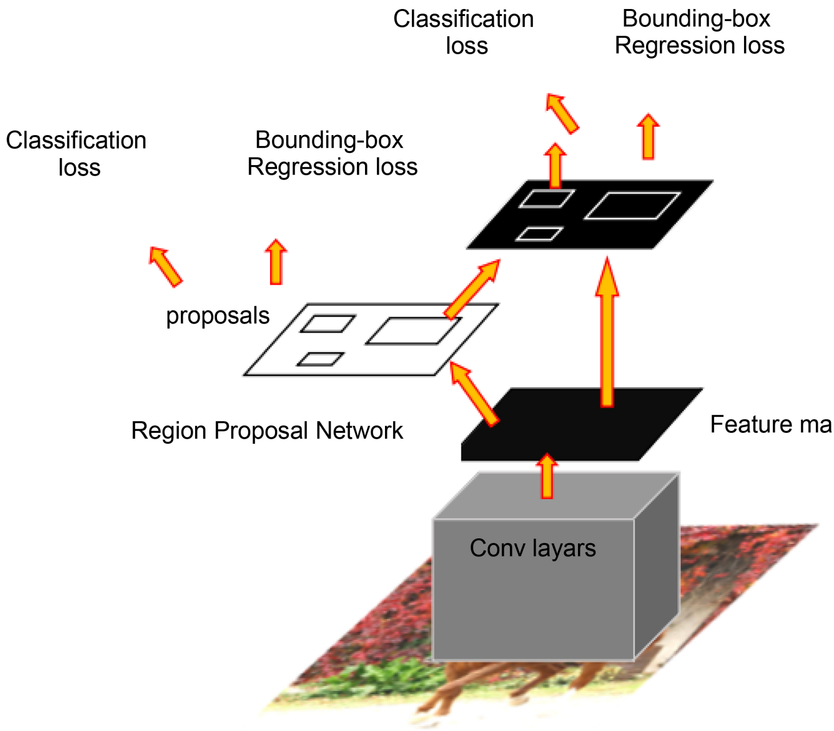

Figure 8. The architecture of faster R-CNN framework.

RoI Pooling extracts fixed-scale features by performing pooling operations on the feature map by round, but two rounding operations will cause large errors. The first rounding is when the proposal of RPN is mapped to the feature map for rounding. The second rounding is the deviation of the coordinates due to the image downsampling when the feature map is mapped to the original image.

So RoI Align solves this problem in two ways, The first way is to map the proposal to the fractional part of the feature map position. The second method is to use bilinear interpolation to process the feature map where the proposal is located to avoid inaccurate features caused by the boundary. The structure of Mask R-CNN network is shown in Figure 9.

4.6. Trident Net

In 2019, Li *et al.* proposed TridentNet [28], which was the first to propose the influence of receptive fields on objects of different scales in object detection tasks, and carried out relevant experimental verifications. Using the parameter sharing method, three branches are proposed during training, and only one of the branches is used during testing, to ensure that there will be no additional parameters and an increase in calculation during inference.

The TridentNet module mainly includes 3 of the same branches, the only difference is the expansion rate of the expanded convolution. From top to bottom, the expansion rates are 1, 2, and 3 respectively, which can detect small, medium, and large targets respectively, which can better realize multi-scale object detection. The three branches share weights.

The whole model successfully solves the problem of constant scale in object detection, but it does not improve the detection speed. The structure of TridentNet network is shown in Figure 10.

4.7. D2Det

In 2020, based on the two-stage method, Cao *et al.* improved the classification

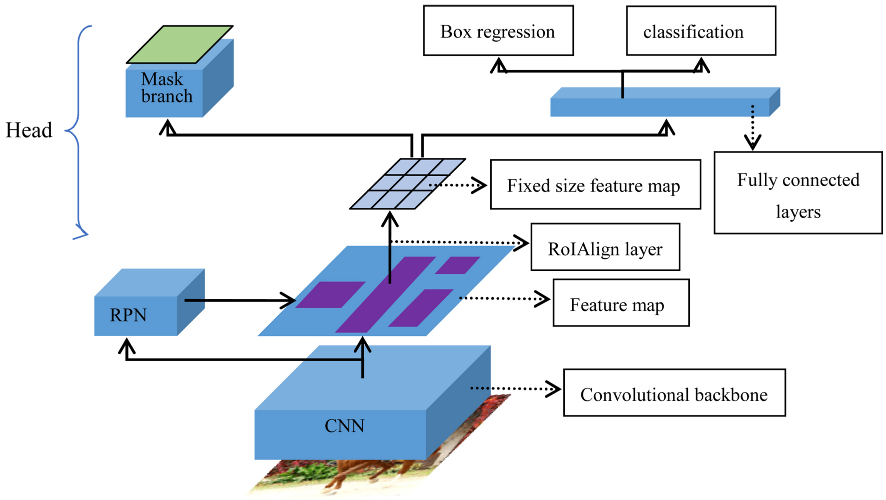

Figure 9. The architecture of Mask R-CNN framework.

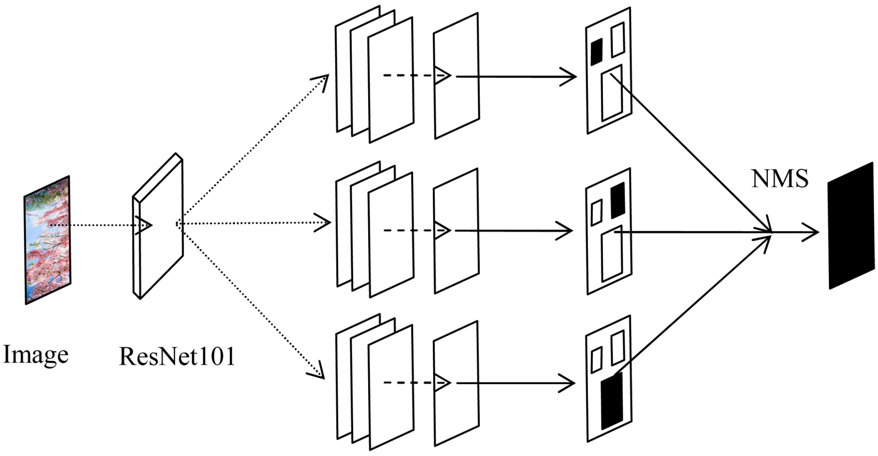

Figure 10. The architecture of Trident Net framework.

and regression branches to further improve the accuracy of object detection and instance segmentation. They proposed D2Det [29], a method that can both accurately locate and accurately classify.

For precise positioning, this paper introduces a dense local regression method, which is used to predict multiple dense box offsets for each target candidate box.

For accurate classification, this paper introduces a discriminative RoI pooling scheme. For a candidate area, it can sample from different sub-regions, and then assign adaptive weights during the calculation to obtain discriminative features. The structure of D2Det network is shown in Figure 11.

4.8. Sparse R-CNN

Most of the previous target detectors are dense detectors, which are based on dense recommendations (sliding-windows, anchor-boxes, reference-points), which are present in the image grid or feature map network in advance On the grid, the scores and offsets of these suggestions are predicted, judged by IOU (Intersection Ratio), and then filtered by NMS (Non-Maximum Suppression) [30].

The small part is the dense-sparse detector (Dense-Sparse), which first extracts relatively few foreground boxes from the dense suggested regions, that is, regional candidate boxes, and then classifies and regresses the position of each regional candidate box, from thousands of candidates, were eliminated to a few prospects.

Sparse R-CNN [31] avoids the manual setting of a large number of hyperparameters for candidate boxes and many-to-one positive and negative sample allocation. More importantly, the final prediction result can be directly output without NMS (Non-Maximum Suppression).

4.9. YOLOv1

YOLO [24] was proposed in 2016 and published in CVPR, the computer vision conference.

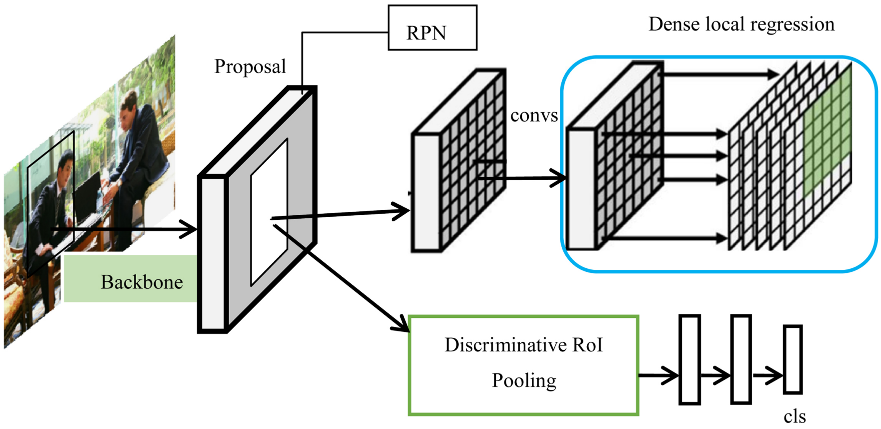

Figure 11. The architecture of the D2Det framework.

Unlike the R-CNN series that needs to find the candidate area first, and then identify the objects in the candidate area, YOLO's prediction is based on the entire picture, and it will output all detected target information at one time, including category and location.

The first step of YOLO is to divide the picture. It divides the picture into grids, and the size of each grid is equal. The core idea of YOLO is to turn object detection into a regression problem, using the entire image as the input of the network, and only going through a neural network to get the location of the bounding box and its category. Its detection speed is extremely fast, the generalization ability is strong, the speed is provided, and the accuracy is reduced. The disadvantage is that for small objects, overlapping objects cannot be detected.

4.10. YOLOv2

In 2017, Joseph Redmon and Ali Farhadi made a lot of improvements based on YOLOv1, and proposed YOLOv2 [32], focusing on solving the shortcomings of YOLOv1's recall rate and positioning accuracy.

Compared with YOLOv1, which uses the fully connected layer to directly predict the coordinates of the Bounding Box, YOLOv2 draws on the idea of Faster R-CNN and introduces the Anchor mechanism. The K-means clustering method is used to cluster and calculate a better Anchor template in the training set, which greatly improves the recall rate of the algorithm. At the same time, combining the fine-grained features of the image, the shallow features are connected with the deep features, which is helpful for the detection of small-scale targets.

The article proposes a new training method—a joint training algorithm. This algorithm can mix these two data sets. Use a hierarchical view to classify objects, and use a huge amount of classification data set data to expand the detection data set, thereby mixing two different data sets.

4.11. YOLOv3

In 2018, Redmon made some improvements based on YOLOv2. The feature extraction part uses the darknet-53 network structure to replace the original darknet-19 and uses the feature pyramid network structure to achieve multi-scale detection. The classification method uses logistic regression instead of softmax to ensure the accuracy of object detection while taking into account real-time performance.

YOLOv3's prior detection system reuses the classifier or locator to perform detection tasks. They apply the model to multiple locations and scales of the image. Those areas with higher scores can be regarded as the test results. In addition, compared to other object detection methods, they use a completely different method. They apply a single neural network to the entire image. The network divides the image into different regions and predicts the bounding box and probability of each region. These bounding boxes are weighted by the predicted probability. The model has some advantages over classifier-based systems. It looks at the entire image during the test, so its prediction uses the global information in the image. Unlike R-CNN, which requires thousands of single target images, it makes predictions through a single network evaluation. This makes YOLOv3 [33] very fast, generally, it is 1000 times faster than R-CNN and 100 times faster than Fast R-CNN.

4.12. YOLOv4

In 2020, Bochkovskiy and others launched YOLOv4 [33]. YOLOv4 conducted a lot of tests on some commonly used Tricks in deep learning and finally selected these useful Tricks: WRC, CSP, CmBN, SAT, Mish activation, Mosaic data augmentation, CmBN, DropBlock regularization, and CIoU loss. YOLOv4 adds these practical skills based on traditional YOLO to achieve the best trade-off between detection speed and accuracy.

4.13. SSD

The full name of the SSD [25] algorithm is Single Shot MultiBox Detector. Single-shot indicates that the SSD algorithm is a one-stage method, and MultiBox indicates that the SSD is a multi-frame prediction.

Compared with Yolo, SSD uses CNN to directly perform detection instead of performing detection after the fully connected layer as Yolo does. In fact, the direct detection of convolution is only one of the differences between SSD and Yolo. There are also two important changes. One is that the SSD extracts feature maps of different scales for detection. Feature maps) can be used to detect small objects, while small-scale feature maps (the later feature maps) can be used to detect large objects; the second is that SSD uses prior boxes with different scales and aspect ratios (Prior boxes, Default boxes), Called Anchors in Faster R-CNN). The disadvantage of the Yolo algorithm is that it is difficult to detect small targets and the positioning is not accurate, but these important improvements enable SSD to overcome these shortcomings to a certain extent.

SSD uses VGG16 as the basic model, and then adds a new convolutional layer based on VGG16 to obtain more feature maps for detection.

4.14. RetinaNet

In 2017, Lin *et al.* proposed RetinaNet [34]. They believe that the one-stage method is fast but not as accurate as the two-stage because the positive and negative samples are not balanced.

The one-stage detector designed a new loss, focal loss for the obstacle problem of category imbalance during the training process, and the cross-entropy error of the regression task was changed to focal loss.

Focal loss is a cross entropy loss that can be dynamically zoomed. When the confidence of the correct category increases, the zoom factor attenuates to 0. The zoom factor can automatically reduce the weight of the loss contributed by easy examples during training so that the model pays attention to hard examples.

FPN serves as the Backbone. It adds a top-down path and a lateral path to the ResNet [35] network and builds a rich, multi-scale feature pyramid from the single resolution of the picture. The features of each layer of the pyramid are used to detect targets of different sizes. The structure of RetinaNet network is shown in **Figure 12**.

4.15. CornerNet

In 2018, Hei *et al.* published CornerNet [36] on ECCV2018. They proposed to solve the object detection problem as a key point detection problem, that is, to obtain the prediction frame by detecting the two key points of the upper left corner and the lower right corner of the target frame. Therefore, there is no concept of anchor in the CornerNet algorithm. This approach is used in object detection. The field is relatively innovative and can achieve good results. The training of the entire detection network is started from scratch and is not based on a pre-trained classification model. This allows users to freely design a feature extraction network without being restricted by the pre-training model. CornerNet also proposed a new pooling method: corner pooling. The structure of CornerNet network is shown in **Figure 13**.

4.16. CenterNet

CenterNet [37], it can be seen from the name of the algorithm that this algorithm

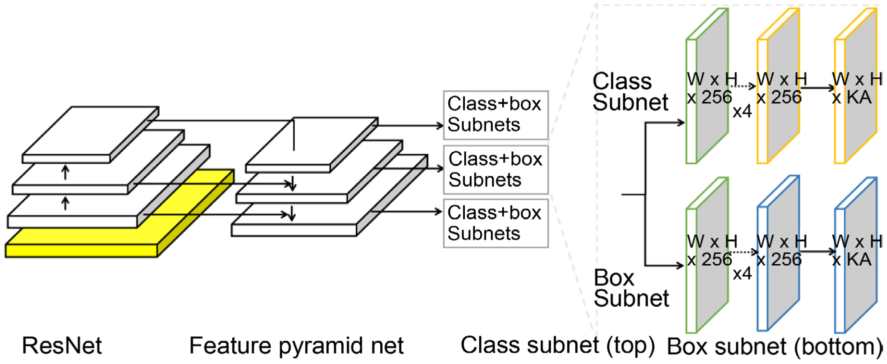

Figure 12. The architecture of RetinaNet framework.

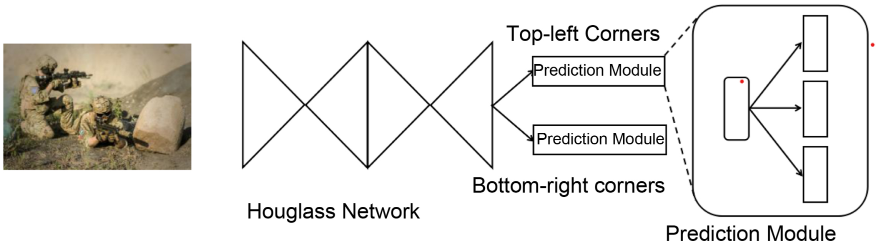

Figure 13. The architecture of CornerNet framework.

is to predict the center point of the target, instead of the two corner points in CornerNet; CenterNet uses a heat map to achieve this, introducing the Gaussian distribution area of the predicted points Calculate the true predicted value. At the same time, the heat map output by the network will first be normalized to 0 to 1 through the sigmoid function and then transferred to the loss function.

CenterNet does not include operations such as corner pooling, because the probability of the center point of the target frame falling on the target is relatively large, so the conventional pooling operation can extract effective features.

CenterNet also uses the same offset prediction as CornerNet, which represents the coordinate error caused by the rounding operation when the annotation information is mapped from the input image to the output feature map, but the calculation in CornerNet is 2. The offset of the corner point, and CenterNet calculates the offset of the center point.

4.17. EffcientNet

An *et al.* proposed the EffcientDet [38] algorithm on CVPR 2020. They believe that the current object detection, either pursues more accurate detection results, but costs a lot, or is more efficient, but at the expense of accuracy. Therefore, the paper designs a set of object detection frameworks to adapt to different constraints, while satisfying high precision and high efficiency. They mainly proposed BiFPN and compound scale methods.

BiFPN is an improvement based on FPN. The original FPN module adds edges to add contextual information and multiplies each edge by corresponding Weights. It allows simple and fast multi-scale feature fusion, secondly, the compound scale method can uniformly scale the resolution, depth and width, feature network, and box/class prediction network of all backbones.

4.18. CentripetalNet

CentripetalNet [39] published in CVPR 2020 uses centripetal displacement to pair corner points in the same instance. CentripetalNet predicts the position and centripetal shift of corner points and matches the aligned corner points as a result of the shift.

Combining location information, CentripetalNet matches corner points more accurately than traditional embedding methods. corner pooling extracts the information in the bounding box to the boundary. To make the information at the corner points more sensitive, CentripetalNet designed a corner-star deformable convolution network for feature adaptation. In addition, CentripetalNet also explores instance segmentation on anchorless detectors by installing a mask prediction module on the centripetal network.

5. Conclusions

In recent years, with the rapid development of deep learning technology, universal object detection technology has developed rapidly and made breakthroughs. However, there is still a huge gap between the efficiency and speed of the detection model and the humanized performance. Existing research methods show that: based on depth The problems to be solved and future research trends of the learned general object detection technology mainly include:

1) Unsupervised object detection: Automated labeling technology is exciting and promising. Unsupervised object detection can eliminate manual labeling.

2) To study a detection method that can have the advantages of both Tow Stage and One Stage models at the same time.

3) Design an efficient feature extraction network.

4) GAN object detector: We know that deep learning object detectors usually require a lot of data for training. In contrast, the GAN target detector is an important structure for producing false images. The combination of real scenes and GAN simulation data helps the detector to be more robust and general.

5) Multi-domain object detection: a general object detector is mainly developed, which can detect multi-domain objects without prior knowledge.

Funding

Supported by the scientific research project of the Sichuan University of Science and Engineering (ZHZJ19-01).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2017) ImageNet classification with Deep Convolutional Neural Networks. *Communications of the ACM*, **60**, 84-90. https://doi.org/10.1145/3065386

[2] Kim, B. and Lee, J. (2019) A Video-Based Fire Detection Using Deep Learning Models. *Applied Sciences*, **9**, Article No. 2862. https://doi.org/10.3390/app9142862

[3] Li, P., Chen, X. and Shen, S. (2019) Stereo R-CNN Based 3D Object Detection for Autonomous Driving. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, 15-20 June 2019, 7636-7644. https://doi.org/10.1109/CVPR.2019.00783

[4] Zhang, X., Yi, W.-J. and Saniie, J. (2019) Home Surveillance System Using Computer Vision and Convolutional Neural Network. 2019 *IEEE International Confe-*

rence on Electro Information Technology (*EIT*), Brookings, 20-22 May 2019, 266-270. https://doi.org/10.1109/EIT.2019.8833773

[5] Zhang, R., Shao, Z., Huang, X., Wang, J. and Li, D. (2020) Object Detection in UAV Images via Global Density Fused Convolutional Network. *Remote Sensing*, **12**, Article No. 3140. https://doi.org/10.3390/rs12193140

[6] Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2013) Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. 2014 *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 23-28 June 2014, 580-587. https://doi.org/10.1109/CVPR.2014.81

[7] Fukushima, K. and Miyake, S. (1982) Neocognitron: A New Algorithm for Pattern Recognition Tolerant of Deformations and Shifts in Position. *Pattern recognition*, **15**, 455-469. https://doi.org/10.1016/0031-3203(82)90024-3

[8] Singh, B. and Davis, L.S. (2018) An Analysis of Scale Invariance in Object Detection Snip. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 3578-3587. https://doi.org/10.1109/CVPR.2018.00377

[9] Ngiam, J., Chen, Z., Chia, D., Koh, P., Le, Q. and Ng, A. (2010) Tiled Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, **23**.

[10] Zeiler, M.D. and Fergus, R. (2014) Visualizing and Understanding Convolutional Networks. *European Conference on Computer Vision*, Zurich, 6-12 September, 818-833. https://doi.org/10.1007/978-3-319-10590-1_53

[11] Yu, F. and Koltun, V. (2015) Multi-Scale Context Aggregation by Dilated Convolutions. https://arxiv.org/abs/1511.07122

[12] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., *et al.* (2017) Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications. https://arxiv.org/abs/1704.04861

[13] Noh, H., Hong, S. and Han, B. (2015) Learning Deconvolution Network for Semantic Segmentation. *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, 7-13 December 2015, 1520-1528. https://doi.org/10.1109/ICCV.2015.178

[14] Chen, R., Wang, M. and Lai, Y. (2020) Analysis of the Role and Robustness of Artificial Intelligence in Commodity Image Recognition under Deep Learning Neural Network. *PLoS ONE*, **15**, Article ID: e0235783. https://doi.org/10.1371/journal.pone.0235783

[15] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, **25**, 1097-1105.

[16] Xu, B., Wang, N., Chen, T. and Li, M. (2015) Empirical Evaluation of Rectified Activations in Convolutional Network. https://arxiv.org/abs/1505.00853

[17] Clevert, D.-A., Unterthiner, T. and Hochreiter, S. (2015) Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). https://arxiv.org/abs/1511.07289

[18] He, K., Zhang, X., Ren, S. and Sun, J. (2015) Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **37**, 1904-1916. https://doi.org/10.1109/TPAMI.2015.2389824

[19] Graham, B. (2014) Fractional Max-Pooling. https://arxiv.org/abs/1412.6071

[20] Zhai, S., Wu, H., Kumar, A., Cheng, Y., Lu, Y., Zhang, Z., *et al.* (2017) S3pool:

Pooling with Stochastic Spatial Sampling. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 21-26 July 2017, 4003-4011. https://doi.org/10.1109/CVPR.2017.426

[21] Girshick, R. (2015) Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, 7-13 December 2015, 1440-1448. https://doi.org/10.1109/ICCV.2015.169

[22] Ren, S., He, K., Girshick, R. and Sun, J. (2017) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **39**, 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031

[23] Uijlings, J.R., van de Sande, K.E.A., Gevers, T. and Smeulders, A.W.M. (2013) Selective Search for Object Recognition. *International Journal of Computer Vision*, **104**, 154-171. https://doi.org/10.1007/s11263-013-0620-5

[24] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2015) You Only Look Once: Unified, Real-Time Object Detection. 2016 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, 27-30 June 2016, 779-788. https://doi.org/10.1109/CVPR.2016.91

[25] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. and Berg, A.C. (2016) SSD: Single Shot Multibox Detector. *European Conference on Computer Vision*, Amsterdam, 8-16 October 2016, 21-37. https://doi.org/10.1007/978-3-319-46448-0_2

[26] Platt, J. (1998) Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines.

[27] He, K., Gkioxari, G., Dollár, P. and Girshick, R. (2017) Mask R-CNN. 2017 *IEEE International Conference on Computer Vision* (*ICCV*), Venice, 22-29 October 2017, 2980-298. https://doi.org/10.1109/ICCV.2017.322

[28] Li, Y., Chen, Y., Wang, N. and Zhang, Z.-X. (2019) Scale-Aware Trident Networks for Object Detection. *Proceedings of the IEEE*/*CVF International Conference on Computer Vision*, Seoul, 27 October-2 November 2019, 6053-6062. https://doi.org/10.1109/ICCV.2019.00615

[29] Cao, J., Cholakkal, H., Anwer, R.M., Khan, F.S., Pang, Y. and Shao, L. (2020) D2det: Towards High Quality Object Detection and Instance Segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, 13-19 June 2020, 11482-11491. https://doi.org/10.1109/CVPR42600.2020.01150

[30] Neubeck, A. and Van Gool, L. (2006) Efficient Non-Maximum Suppression. 18*th International Conference on Pattern Recognition* (*ICPR*'06), Hong Kong, 20-24 August 2006, 850-855. https://doi.org/10.1109/ICPR.2006.479

[31] Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., *et al*. (2021) Sparse R-CNN: End-to-End Object Detection with Learnable Proposals. *Proceedings of the IEEE*/*CVF Conference on Computer Vision and Pattern Recognition*, Nashville, 20-25 June 2021, 14449-14458. https://doi.org/10.1109/CVPR46437.2021.01422

[32] Redmon, J. and Farhadi, A. (2017) YOLO9000: Better, Faster, Stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 21-26 July 2017, 6517-6525. https://doi.org/10.1109/CVPR.2017.690

[33] Bochkovskiy, A., Wang, C.-Y. and Liao, H.-Y.M. (2020) YOLOv4: Optimal Speed and Accuracy of Object Detection. https://arxiv.org/abs/2004.10934

[34] Lin, T.-Y., Goyal, P., Girshick, R., He, K. and Dollár, P. (2017) Focal Loss for Dense Object Detection. *Proceedings of the IEEE International Conference on Computer Vision*, Venice, 22-29 October 2017, 2999-3007.

https://doi.org/10.1109/ICCV.2017.324

[35] He, K., Zhang, X., Ren, S. and Sun, J. (2016) Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 770-778. https://doi.org/10.1109/CVPR.2016.90

[36] Law, H. and Deng, J. (2018) Cornernet: Detecting Objects as Paired Keypoints. *Proceedings of the European Conference on Computer Vision* (*ECCV*), Munich, 8-14 September 2018, 765-781. https://doi.org/10.1007/978-3-030-01264-9_45

[37] Zhou, X., Wang, D. and Krähenbühl, P. (2019) Objects as Points. https://arxiv.org/abs/1904.07850

[38] Tan, M. and Le, Q. (2019) EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *International Conference on Machine Learning*, Long Beach, 10-15 June 2019, 6105-6114.

[39] Dong, Z., Li, G., Liao, Y., Wang, F., Ren, P. and Qian, C. (2020) Centripetalnet: Pursuing High-Quality Keypoint Pairs for Object Detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, 13-19 June 2020, 10516-10525. https://doi.org/10.1109/CVPR42600.2020.01053