Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/6/20230690

Research of image object detection using deep learning

Xuanang Feng

College of mechanical and electronic engineering, Northwest A&F University No.22, Xinong Road, Yangling District, Xianyang, China

fxa-icdesigner@nwafu.edu.cn

Abstract. One of the major issues in the realm of computer vision is the work of object recognition, which is to identify all interesting items in the image and establish their location and size. The most difficult issue in the field of machine vision has always been target recognition because of the varied differences in appearance, shape, and position of diverse objects as well as the interference of light, occlusion, and other elements in imaging. This article evaluates numerous target detection publications based on the advancement of target detection technology. The different target detection networks are analysed and discussed. In addition, test datasets and evaluation indicators are summarized. In the end, the paper summarizes and looks forward to the full text.

Keywords: object detection, computer vision, deep learning.

1. Introduction

One of the main issues in the realm of computer vision is the work of object identification, which is to identify all interesting items (objects) in the image and establish their location and size. The main objective is to discover all the qualified targets in the input image based on the target information that users are interested in, and then assess the target's category and position. The deep learning algorithm for object detection has very broad application value and development prospects. At this stage, it can be applied in many scenes of daily life and industrial development, such as OCR technology, digital recognition, fingerprint recognition, face recognition, license plate recognition, agricultural product pest recognition, defect detection, pathological detection and other fields.

2. Target detection network model

Since the emergence of deep learning, object identification research has mostly focused on two-stage algorithms such as the R-CNN series or one-stage algorithms such as YOLO and SSD. The two phases differ largely in that the two-stage technique must first present a suggestion (a pre-selection box that may include the item to be identified) before doing fine-grained object identification. For the purpose of predicting item categorization and location, the one-stage approach will immediately extract features from the network.

2.1. Two stage algorithms

2.1.1. R-CNN. R-CNN operates on a basic principle: it first uses selective search to extract a set of object suggestions (object candidate boxes) [1]. Each proposal is then reduced to a fixed-size image and put into an ImageNet-trained CNN model (like AlexNet) to extract features. To establish whether or not there are items in each location and to identify the object type, a linear SVM classifier is utilized. R-CNN adheres to the classical concept of target detection. It also detects targets using extraction boxes in four steps: feature extraction for each box, picture classification, and non-maximum suppression. Only during the feature extraction process are standard features (such as SIFT, HOG, and so on) substituted with depth convolution network features. RCNN has significantly improved the performance of VOC07, with the average accuracy (mAP) greatly raised to 58.5% from 33.7% (DPM v5).

Even though RCNN has advanced significantly, its disadvantage is self-evident: superfluous feature calculation for a substantial number of overlapping recommendations (more than 2000 boxes in an image) resulting in extremely sluggish detection speed (14 seconds per image when using GPU).

The R-CNN system is separated into three stages: first, candidate areas with independent production categories, which contain R-final CNN's positioning results; second, R-final CNN's positioning results; and third, R-final CNN's positioning results. Second, for each candidate region, the neural network obtains a fixed length feature vector. Finally, a sequence of SVM classifiers should be configured.

2.1.2. Fast RCNN. One of the main disadvantages of R-CNN is that it requires multi-stage separate training. Fast RCNN solves this problem by developing a unified end-to-end trainable system [2]. The network sends a picture via a number of convolution layers, and the target's recommendations are also mapped to the gathered feature maps. Girshick uses the ROI Pooling layer to replace the pyramid structure Pooling in SPP net, followed by two full connection layers, and then divides it into N+1 softmax layer and a border regression layer that also has a full connection. The model additionally shifts the border regression loss function from L2 to smooth L1 to boost performance and incorporates multi task loss to train the network.

Fast R-CNN is mainly used to solve the problems of R-CNN: slow test training speed, mainly slow feature extraction of candidate regions: R-CNN first extracts 2000 candidate regions from the test map, and then inputs these 2000 candidate regions into the pre trained CNN to extract features. Because candidate regions have a large number of overlaps, this method of feature extraction will repeatedly calculate the features of overlapping regions.

In Fast RCNN, input the whole graph into CNN to extract features, and then map to each candidate region during adjacency. In this way, only a few layers at the end need to process each candidate box separately. Training requires extra space to save the extracted feature information: R-CNN needs to save the extracted features for training individual SVM classifiers and border regressors for each class. In Fast R-CNN, the category judgment and border regression are implemented using CNN in a unified way, without additional storage features.

2.1.3. Faster R-CNN. Algorithms based on candidate boxes, such as Selective Search, are all based on CPU and cannot be accelerated by taking advantage of the high parallelism of GPU. However, neural network-based methods can take full advantage of GPU parallel computing and greatly improve the algorithm speed. If candidate boxes can be extracted by neural network, the execution speed of the whole algorithm can be further improved. Faster R-CNN is based on this idea. It uses a full convolutional network as the Region Proposal Network (RPN) to extract the waiting frames of various scales and aspect ratios.

In order to further improve the efficiency, the subsequent target detection algorithm and RPN network share convolution features, making the whole detection process smoother and the overall speed significantly improved. The quicker R-CNN algorithm has two primary parts. For extracting candidate frames, the first module uses a complete convolutional network (RPN), while the second uses a Fast R-CNN target detector. The whole detection process is completed through a network. Faster R-CNN uses the so-called attention mechanism. The RPN module tells Faster R-CNN where to focus. Faster RCNN foregoes the standard sliding window and SS approaches in favor of using RPN directly to produce detection frames [3]. This is another important advantage of Faster R-CNN, which might greatly increase the production speed of the detection frame.

2.2. One stage algorithm

2.2.1. YOLOv2. The author offers in this study an enhanced YOLOv2 based on YOLOv1, followed by a combined training technique for detection and classification. The YOLO9000 model is developed using this hybrid training strategy on the ImageNet classification dataset and the COCO detection dataset, which can detect over 9000 different types of objects. In this network, the author puts forward a bold attempt. In YOLO v2, The Batch Normalization (BN) layer is placed after every convolution layer, and the drop out layer is removed. This improvement is also used by many models today.

The issue of gradient disappearance and explosion during back propagation can be resolved by batch normalization, which can also lessen sensitivity to particular super factors (for example, the pace of learning, the size range of the network parameters, and the choice of activation function), increase the model's rate of convergence and use a specified regularization impact to lessen overfitting. After using Batch Normalization. YOLOv2's mAP increased by 2.4%. Referring to the practice of Faster R-CNN, YOLOv2 does away with the whole connection layer of YOLOv1 and instead predicts the boundary box using convolution kernel anchor boxes [4]. By presetting a set of borders with different sizes and aspect ratios in each cell, the whole image can be covered at different locations and at different scales. At the same time, a pooling layer in the network is removed in order to improve the resolution of the detection practical feature map. And the network input image size becomes 416 × 416 instead of the original 448 × 448.The convolution layer of YOLO uses the value of 32 to down sample the image, so the network input is 416 × 416, output 13 × 13. The use of Anchor Box will reduce the accuracy slightly, but it can make YOLO predict 13 × thirteen × 9=1521 boxes. Therefore, compared with the recall rate of 81% of YOLOv1, the recall rate of YOLOv2 has increased significantly to 88%.

The parameters (length and breadth) of an a priori box is set manually in SSD and Faster R-CNN, which is subjective. The model is simpler to learn and produces better predictions if the a priori box dimension chosen is the right one. As a result, YOLOv2 clusters the bounding boxes in the training set using the k-means clustering approach to determine the bounding box size that best fits the samples [5].

2.2.2. YOLOv4. YOLOv4 optimizes YOLOv3 in IoU threshold (positive sample matching). In YOLOv3, only one anchor is assigned to each GT, but in YOLOv4, a GT can be assigned to multiple anchors at the same time. They directly use the anchor template to roughly match GT Boxes, and then locate the corresponding anchor of the corresponding cell.

Yolov4 improved the input terminal during training. Mosaic introduced Mosaic data augmentation mechanism by referring to the CutMix data improvement approach proposed for the end of 2019. It spliced four pictures into one picture during data preprocessing to increase the diversity of learning samples. Especially, random scaling increased many small targets, making the network more robust.
CBN will consider the first k time statistics when calculating the current time statistics, so as to expand the batch size operation. At the same time, the author points out that CBN operation will not introduce large memory overhead, and the training speed will not affect much, but the training will be slower. Therefore, the author uses Cross Mini Batch Normalization (CMBN) [6]. CmBN is an improved version of CBN, which takes the four mini batches inside the big batch as a whole and isolates them from the outside. At time t, CBN will also consider the statistics of the first three times for merging, while the CmBN operation will not, and will no longer slide cross. It will only perform merging operation within the mini batch, and keep the BN updated with trainable parameters once per batch.

In addition, the author develops self-confrontation training (SAT), a novel data augmentation strategy. The neural network alters the original picture rather than the network weight in the initial step. In this manner, the neural network launches an adversarial assault against itself, altering the original picture and creating the illusion that there is no target on the image. The trained neural network detects normal targets on the changed picture in the second step. Self-Adaptive Training is a data improvement system that can withstand some assaults. CNN computes Loss, then modifies the image information via back propagation to create the illusion that there is no target on the image, before doing regular target detection on the changed image. In the process of back propagation of SAT, the network weight does not need to be changed. The decision boundary of learning's weak links can be strengthened by using confrontation generation, and the model's resilience may be increased [7]. Consequently, this data improvement technique is being utilized by an increasing number of object identification frameworks.

2.3. Vision transformer

Transformer was first suggested for the NLP discipline and has had considerable success. This paper is also inspired by it, trying to apply Transformer to the CV field. Through the experiment in this article, the accuracy of the best model given can reach 88.55% on ImageNet1K (pre training was conducted on Google's own JFT dataset first), which shows that Transformer is indeed effective in the CV field. The four primary components of the ViT core process are the MLP classification processing, the Transformer encoder, the image block processing (create patches), and the image block embedding and position coding [8].

One may consider the first step to be an image preparation phase. In CNN, the image can be convolved directly without special pre-processing flow [9]. Nevertheless, the Transformer structure is unable to analyze the picture directly. Before that, it needs to be partitioned. Consider the following image: x H W C. Now that it has been divided into P, P, and C patches, the dimensions of all the patches may be stated as N, P, P, and C. The equivalent data dimension is thus N (P2 C), after flattening each patch. Here, N may be thought of as the length of the sequence fed into the Transformer, C as the quantity of channels in the input picture, and P as the size of the image patch.

2.3.1. Swin transformer. Swin Transformer is a new type of transformer, which can be used as a unified model for visual and language processing. Swin Transformer has suggested a shifted window [10], which is related to the higher layer of Windows and increases the model's performance. Query patches in the same window use the same key set, which increases memory access performance. On the contrary, the self-attachment access to memory in the previous slide window method is inefficient, because different query pixels have different key sets.

For efficient modeling, Swin Transformer suggests calculating self-attention in local windows [11]. The windows are arranged to evenly segment the image in a non-overlapping manner. Assume that each window contains M × M patches, global MSA module and window based MSA module in h × The computational complexity on the w patches images is:

$$\Omega(MSA) = 4hwC^2 + 2(hw)^2C \tag{1}$$

$$\Omega(W - MSA) = 4hwC^2 + 2M^2hwC \tag{2}$$

The former is the number of patches h × The second power of w, which is linear when M is fixed (default setting is 7). For greater h W (high resolution picture), global self-attention computation is typically unpleasant, but window based self-attention is scalable.

3. Test data set and evaluation indicators

The target detector uses a variety of indicators to evaluate the performance of the detector, such as FPS, precision, recall, and the most commonly used mAP. Precision is generated from the intersection and merger ratio between prediction frame and GT, which is denoted as IoU (Intersection over Union). Then, set an IoU threshold to assess the accuracy of the detection result: If IoU exceeds the threshold, the outcome is categorized as True [12]. Positive (TP). If it is less than the threshold value, it is characterized as a False Positive. (FP). If the network fails to identify targets in the GT, these targets are classified as False Negative (FN).

3.1. mAP

MAP stands for Mean Average Precision [13], which stands for mean average precision. It is used as an index to determine target detection accuracy. The calculating formula is as follows:

mAP=Sum of average precision of all categories divided by all categories. ndicator to measure the detection accuracy in object decision.

3.2. MS COCO dataset

The COCO data set is a Microsoft team-released data collection that may be used for picture recognition+segmentation+capping [14]. This data set captures a large number of daily scene photographs with common items and offers pixel level instance annotation to more properly evaluate the effect of detection and segmentation algorithms, with the goal of advancing scene comprehension research advancement. The competition, which currently encompasses the core tasks of machine vision such as detection, segmentation, key point identification, annotation, and so on, is held once a year using this data set. Since the ImageNet Challenge, it has been one of the most significant academic contests [15].

In comparison to ImageNet, COCO prioritizes non-audio images in which the target and its scene appear together. Such images can reflect visual semantics and fulfill the image understanding task criteria. The relative iconic pictures, on the other hand, are better suited for tasks like shallow semantic image categorization. COCO's detection tasks include 80 categories in total. The data size released in 2014 was divided into train/val/test of 80k/40k/40k, separately [16]. In the academic environment, using the train and 35k val subset as the training set (trainval35k), the remaining val set as the test set (minival), and submitting the findings (test dev) to the official evaluation server is more typical. Additionally, COCO also sets aside a portion of the test data for use as the competition's assessment set.

3.3. Google open image

Nine point two million photos make up Google's Open Images collection. It has segmentation masks, object bounding boxes, and image level labels [17]. It debuted in 2017 and has been updated six times since then. Open Photos provides the biggest target location collection, with sixteen million bounding boxes and six hundred categories on one point nine million images for target identification. Its designers chose intriguing, complicated, and diversified photos, each containing 8.3 item categories. The AP introduced in Pascal VOC has undergone some adjustments, including as disregarding uncommented classes and identifying classes and subclasses.

4. Comparison of network effects

Table 1: Performance evaluation of several object detectors using datasets from PASCAL VOC 2012 and MS COCO with comparable input picture sizes.

Table 1. Performance evaluation of several object detectors.

Model	year	*Backbone*	size	AP [0.5:0.95]	AP0.5	FPS
R-CNN*	2014	AlexNet	224	-	58.5%	~0.02
Fast-R-CNN*	2015	VGG-16	Variable	-	65.7%	~0.43
Faster R-CNN	2016	VGG-16	600	-	67%	5
YOLOv2	2016	DartNet-19	352	21.6%	44%	81
YOLOv4	2020	CSP-DartNet-53	512	43.0%	64.9%	31
Swin-L	2021	HTC++	-	57.7%	-	-

The comparison of models denoted by an asterisk (*) is done using PASCAL VOC 2012, whereas comparisons of other models are done using MS COCO. Real-time detectors (>30 FPS) are indicated by gray-colored rows.

5. Conclusion

Although great progress has been made in target detection in the past decade, the performance of the best detectors is not yet reached saturation. Along with people's demand for the convenience of life progress and the technical progress of microprocessor NPU and GPU dedicated to neural network technology to provide hardware computing support, the deployment of target detection in life scenes and the demand for lightweight network models will also grow exponentially.

In this article, the author summarizes the two stage and one stage network, as well as common evaluation indicators and open-source datasets with high recognition in this field. Through list comparison, we can observe for ourselves that the model's target detection accuracy is increasing, and of course the model will be applied to more and more fields. The appearance of VIT technology has applied the transformer originally applicable to the NLP field to the CV field, which has also greatly improved the accuracy. At present, the most precise detector is the Swin Transformer. With the development of GPU and NPU hardware technology, the efficiency of training neural networks will also be greatly enhanced, and the accuracy of target detection technology will also be enhanced. It is also an inevitable trend for large-scale deployment in many life scenes, such as automatic driving, commodity recognition, machine translation, etc.

References

[1] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.

[2] Tu, SQ , Yuan, WJ ; Liang, Y , 2016 Automatic Detection and Segmentation for Group-Housed Pigs Based on PigMS R-CNN, *Sensors* **4**.

[3] Li, JN , Liang, XD ; Shen, SM ; Xu, TF, 2018 Scale-Aware Fast R-CNN for Pedestrian Detection, *IEEE Trans. Multi-media* **10**

[4] Nguyen, H , 2019 Improving Faster R-CNN Framework for Fast Vehicle election, *Math. Prob. Eng.***32.**

[5] Ahmad, T , Ma, YL, Yahya, M , 2021 Object Detection through Modified YOLO Neural Network, *Sci. Rog.*452-461.

[6] Chen, ZC, Li, ZM , Hu, WJ , 2018 An Efficient Pedestrian Detection Method Based on YOLOv2, *Math. Prob. Eng.***29.**

[7] Yu, ZW , Shen, YG , Shen, CK , 2021 A real-time detection approach for bridge cracks based on YOLOv4-FPM, *Auto. Constr.* ***122****: 103514*

[8] Wang, BL , Li, SA , Gao, XZ , Xie, T , 2021 UAV Swarm Confrontation Using Hierarchical Multiagent Reinforcement Learning, *Inter. J. Aer. Eng.***23.**

[9] Han, Kai; Wang, Y.; Chen, H. 2022 A Survey on Vision Transformer., *IEEE Trans. PA. Anal. Mach. Intel.***726.**

[10] Zhao, JF ; Mao, X; Chen, LJ , 2018 Learning deep features to recognize speech emotion using merged deep CNN, *Iet. Signal Pro.***81(2).**

[11] Xu, XK ; Feng, ZJ Cao, CQ, Li, MY; Wu, J, 2021 An Improved Swin Transformer-Based Model for Remote Sensing Object Detection and Instance Segmentation, *Rem. Sen.* 431-442.

[12] Xu, J , Ma, YX ; He, SH, Zhu, JH, 2019 3D-GIoU: 3D Generalized Intersection over Union for Object Detection in Point Cloud, *Sensors* 239-242.

[13] Saleem, MH ; Khanchi, S; Potgieter, J ; Arif, KM , 2020 Image-Based Plant Disease Identification by Deep Learning Meta-Architectures, *Plants Basel* 731-737.

[14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll´ar, and C. L. Zitnick, 2014 Microsoft coco: Common objects in context, *Comput. Vis.* **39**

[15] Srivastava, S ; Divekar, AV ; Anilkumar, C ; Naik, I ; Kulkarni, V ; Pattabiraman, V , 2014 Comparative analysis of deep learning image detection algorithms, *J. Big Data* **33**

[16] Xu, HY ; Lv, XT ; Wang, XY; Ren, Z ; Bodla, N ; Chellappa, R , 2019, Deep Regionlets: Blended Representation and Deep Learning for Generic Object Detection, *IEEE Trans. PA. Anal. Mach. Intel.***76**

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma,Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, andL. Fei-Fei, 2015 *ImageNet large scale visual recognition challenge,* **115(3)** 211–252.

[18] Mullissa, A ; Vollrath, A ; Odongo-Braun, C ; Slagter, B; Balling, J ; Gou, YQ; Gorelick, N ; Reiche, J , 2021 Sentinel-1 SAR Backscatter Analysis Ready Data Preparation in Google Earth Engine, *Remote Sensing* **47(10).**