Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 199 (2022) 1066–1073

Procedia Computer Science

www.elsevier.com/locate/procedia

The 8th International Conference on Information Technology and Quantitative Management (ITQM 2020 & 2021)

A Review of Yolo Algorithm Developments

Peiyuan Jiang, Daji Ergu*, Fangyao Liu, Ying Cai, Bo Ma

Key Laboratory of Electronic and Information Engineering (Southwest Minzu University), Chengdu, 610041, China. * Corresponding author. ergudaji@163.com

Abstract

Object detection techniques are the foundation for the artificial intelligence field. This research paper gives a brief overview of the You Only Look Once (YOLO) algorithm and its subsequent advanced versions. Through the analysis, we reach many remarks and insightful results. The results show the differences and similarities among the YOLO versions and between YOLO and Convolutional Neural Networks (CNNs). The central insight is the YOLO algorithm improvement is still ongoing.This article briefly describes the development process of the YOLO algorithm, summarizes the methods of target recognition and feature selection, and provides literature support for the targeted picture news and feature extraction in the financial and other fields. Besides, this paper contributes a lot to YOLO and other object detection literature.

Keywords: Review; Yolo; Object Detection; Public Data Analysis

1. Introduction

You Only Look Once (YOLO) is a viral and widely used algorithm [1]. YOLO is famous for its object detection characteristic. In 2015, Redmon et al. gave the introduction of the first YOLO version [2]. In the past years, scholars have published several YOLO subsequent versions described as YOLO V2, YOLO V3, YOLO V4, and YOLO V5 [3-10]. There are a few revised-limited versions, such as YOLO-LITE [11-12]. This research paper only focused on the five main YOLO versions.

This paper will compare the main differences among the five YOLO versions from both conceptual designs and implementations. The YOLO versions are improving, and it is essential to understand the main motivations, features development, limitations, and even relationships for the versions. This reviewing paper will be meaningful and insightful for object detection researchers, especially for beginners.

The following first section will give a version comparing from the technique perspective along with the version similarities. The second section describes them through public data. The insightful results are displayed using both figures and tabular. The two primary analyses are focused on the YOLO trends and YOLO-related queries.

10.1016/j.procs.2022.01.135

2. Yolo Algorithm Developments

2.1 Main Differences (Features)

The core of the YOLO target detection algorithm lies in the model's small size and fast calculation speed. The structure of YOLO is straightforward. It can directly output the position and category of the bounding box through the neural network. The speed of YOLO is fast because YOLO only needs to put the picture into the network to get the final detection result, so YOLO can also realize the time detection of video. YOLO directly uses the global image for detection, which can encode the global information and reduce the error of detecting the background as the object. YOLO has a strong generalization ability because YOLO can learn highly generalized features to be transferred to other fields. It converts the problem of target detection into a regression problem, but detection accuracy needs to be improved. YOLO's test results are poor for objects that are very close to each other and in groups. This poor performance is because only two boxes in the grid are predicted and only belong to a new class of objects of the same category, so an abnormal aspect ratio appears, and other conditions, such as weak generalization ability.

Due to the loss function, the positioning error is the main reason for improving the detection efficiency. Especially the handling of large and small objects needs to be strengthened. In the implementation, the most important thing is how to design the loss function so that these three aspects can be well balanced. YOLO uses multiple lower sampling layers, and the target features learned from the network are not exhaustive so that the detection effect will be improved.

The original YOLO architecture consists of 24 convolution layers, followed by two fully connected layers. YOLO predict multiple bounding boxes per grid cell but those bounding boxes having highest Intersection Over Union (IOU) with the ground truth is selected, which is known as non-maxima suppression [13].

YOLO has two defects: one is inaccurate positioning, and the other is the lower recall rate compared with the method based on area recommendations. Therefore, YOLO V2 mainly improves in these two aspects. Besides, YOLO V2 does not deepen or broaden the network but simplifies the network.

Two improvements of YOLO V2: Better and Faster.

2.2 Better

2.2.1 Batch normalization

It is equivalent to standardizing the input of each layer, speeding up the convergence rate, deleting the loss, and increasing mAP by 2%.

2.2.2 High-resolution classifier

The original YOLO network uses 224×224 pixels before training and then uses 448×448 pixels in detection. When switching from the classification model to the detection model, the model adapts to image classification. YOLO V2 divides pre-training into two steps: from the beginning, train the network with 224×224 (160 epochs) pixels, then adjust the pixels to 448×448 and train for ten epochs.

2.2.3 Fine features

The most important thing is to add a layer: through the layer. The function of this layer is to connect the 26×26 feature map of the previous layer with the 13×13 feature map of this layer because of the 13×13 feature. It is sufficient to predict large objects, but predicting small objects is not necessarily effective and easy to understand. Smaller objects may eventually disappear after multiple layers of convolution and merging. Therefore, the larger

functions of the previous layer should be merged.

2.2.4 Multi-scale training

This network training method enables the same network to detect images of different resolutions. Although the training speed is slower when the input size is large, the training speed is faster when the input size is small, multi-scale. Training can improve accuracy, so there is a good balance between accuracy and speed.

2.3 Faster

2.3.1 Darknet-19

In YOLO, the training network used is based on GooleNet. Here, the author makes a simple comparison between GooleNet and VGG16.In terms of computational complexity (8.25 billion operations and 30.69 billion operations), GooleNet is superior to VGG16. The former in ImageNet is slightly lower than the latter (88% vs. 90%). In YOLO V2, the author uses a new classification model, Darknet-19, as the primary network.

Table 6 is the final network structure: Darknet-19 only needs 5.58 billion operations. The network contains 19 convolutional layers and five maximum pooling layers, and in YOLO. The GooleNet used in V1 has 24 convolutional layers and two complete connection layers. Therefore, the convolution and convolution operations in Darknet-19 are less than those used in GoogleNet. Overall, YOLO is the key to reducing the amount of calculation. Finally, the average pooling layer is used to replace the entire connection layer for prediction.

2.3.2 Training for Classification

The classification training here is all pre-training on ImageNet, which mainly includes two steps. The data set is ImageNet, 160 Epochs are trained, the input image size is 224×224, and the initial learning rate is 0.1. Standard data increment methods are used during the training process, such as random cropping, rotation, and chroma and brightness adjustments.

Then fine-tune the network: At this time, using a 448×448 input, all parameters remain unchanged except for the epoch and the learning rate. Here, the learning rate is changed to 0.001, and training is performed ten times.

The results show that the accuracy of top-1 and top-5 after fine-tuning are 76.5% and 93.3%, respectively. According to the original training method, the accuracy of Darknet-19's top-1 and top-5 is 72.9% and 91.2%.

2.3.3 Training for Detection

First, delete the last convolution layer, and then add three convolution layers 33. Each convolution layer has 1024 filters, and each convolution layer is connected to 11 convolution layers. The category probabilities of the two boxes corresponding to this cell are the same, but in YOLO V2, the category probabilities belong to this box, and each box corresponds to a category probability instead of a grid.

Compared with YOLO V2, YOLO V3 has two points: using multi-scale features for object detection and adjusting the basic network structure.

On the one hand, YOLO V3 adopts feature graphs of three scales (when the input is (416×416), (13×13), (26×26) and (52×52)). YOLO V3 uses three prior boxes for each position, so K-means is used to get nine prior boxes and divide them into three scale feature maps. Feature maps with larger-scale use smaller prior boxes.

On the other hand, YOLO V3 feature extraction network used the residual model. Compared with Darknet-19 used by YOLO V2, it contained 53 convolution layers, so it was called Darknet-53.

YOLO V4 style has a significant change, more focus on comparing data, and has a substantial improvement.

The integrator characterizes it and finally achieves very high performance.

We can summarize it like this: YOLO V4=CSP Darknet53+SPP+Pan+YOLO V3

The main contributions are as follows:

- An efficient and powerful target detection model is proposed. It allows everyone to train super-fast and accurate target detectors;
- The influence of SOTA's bag-of-freebies and bag-of-specials methods was verified during detector training;
- Improved SOTA methods to make them more efficient and suitable for single GPU training, including CBN, PAN, SAM, etc.

It splits the current mainstream target detector frameworks: Input, Backbone, Neck, and Head. In the previous YOLO V3, one anchor point was responsible for one ground truth, while in YOLO V4, several anchor points were responsible for one ground truth. This means that the number of anchor frames remains unchanged, but the selection ratio of positive samples is increased, thereby alleviating the problem of imbalance between positive and negative samples. The advantage is that due to the range of the sigmoid function, the grid sensitivity is also eliminated as the open time interval, and the actual position of the boundary is not available. The CIOU (Complete Intersection over Union) loss function is adopted, which converges quickly and eliminates the problem of the bounding box containing ground truth. YOLO V4 provides a faster and more accurate advanced detector compared to all available alternatives. The original concept of a detector based on a single-stage anchor has proven its feasibility. We have verified many features and chose to use these features to improve the accuracy of classifiers and detectors. YOLO V4 can be used as a best practice for future research and development.

Multiple network architectures of YOLO V5 are more flexible to use, have a very lightweight model size, and are on par with the YOLO V4 benchmark in terms of accuracy. However, people still have reservations about YOLO V5 because it is less innovative than YOLO V4, but it has some performance improvements, with the following significant advantages:

- The PyTorch framework is user-friendly and easy to train your data set, making it easier to put into production than the Darknet framework used in YOLO V4;
- Easy to read code, integration of a large number of computer vision technology, is conducive to learning and reference;
- Easy to configure the environment, model training is very fast, and batch reasoning produces real-time results.

YOLO V5 provides each batch of training data through the data loader and enhances the training data simultaneously. The data loader performs three types of data enhancement: scaling, color space adjustment, and mosaic enhancement.

The data proves that Mosaic enhancement can indeed effectively solve the most troublesome small object problem in the model training. That is, the small object detected is not as accurate as of the large object. But it must also be admitted that the naming of YOLO V5 is controversial, and its implementation is not static and has not been fully completed. Now it leaves us more room.

Main improvement measures of YOLO network from V1 to V5:

- YOLO: The grid division is responsible for detection, confidence loss;
- YOLO V2: Anchor with K-means added, two-stage training, full convolutional network;
- YOLO V3: Multi-scale detection by using FPN;
- YOLO V4: SPP, MISH activation function, data enhancement Mosaic/Mixup, GIOU(Generalized Intersection over Union) loss function;
- YOLO V5: Flexible control of model size, application of Hardswish activation function, and data enhancement.

2.4 Relationship

Because YOLO and YOLO V2 are not effective in detecting small targets, multi-scale detection is added to YOLO V3. YOLO V3 is a well-received master of the previous generations. YOLO V4 sorted out and tried all possible optimizations in the entire process and found the best effect in each permutation and combination. YOLOv4 runs twice faster than EfficientDet with comparable performance. Improves YOLOv3's AP and FPS by 10% and 12%, respectively [15]. YOLO V5 can flexibly control models from 10+M to 200+M, and its small model is very impressive. The overall network diagrams of YOLO V3 to YOLO V5 are similar, but they also focus on detecting objects of different sizes from three different scales.

3. Public Data Insights

This section gives a brief overview of the YOLO versions through the public data. In 2015, the YOLO algorithm was published, which is used for object detection. YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection [14]. YOLO's performance was competitive, but it still has room for improvement. The first subsection presents the algorithm trends. The second one gives more details about people's insights. Both subsections use numerical data and text show the YOLO is still ongoing and keeps updating. All the data is collected from GOOGLE (www.google.com) open dataset (we have deleted the YOLO V1 results in this section due to the noise).

3.1 Trends

This subsection has collected the publication data for displaying the trends. Table 1 gives us the academic research paper numbers of each version. The breakdown illustrates the research paper number has increasing a lot in the year 2019 and year 2020. Besides, YOLO V3 and V2 versions have attracted most of the researcher's eyes, although the time fact can be another element. V4 and V5 versions' number is less because they are very new.

Table 1. YOLO versions breakdown by years

	YOLO V2	YOLO V3	YOLO V4	YOLO V5	Total
2016	0	0	0	0	0
2017	5	0	0	0	5
2018	47	19	0	0	66
2019	48	210	0	0	258
2020	36	496	81	13	626
Total	136	725	81	13	955

Fig. 1 presents the interests over time. The data is based on the web search performance, including news search, image search, and YouTube search. The scale is a relative measurement. The highest point is 100, and the lowest is zero. For example, a value of 50 means that the term is half as popular. Statistically, the value of zero means there data maybe not enough or people are not interested in this topic.

The figure shows that the V2 and V3 versions are more prevalent most of the time. However, after April 2020, V4 and V5 are getting more popular. This result matches the numerical results from previous Table 1.

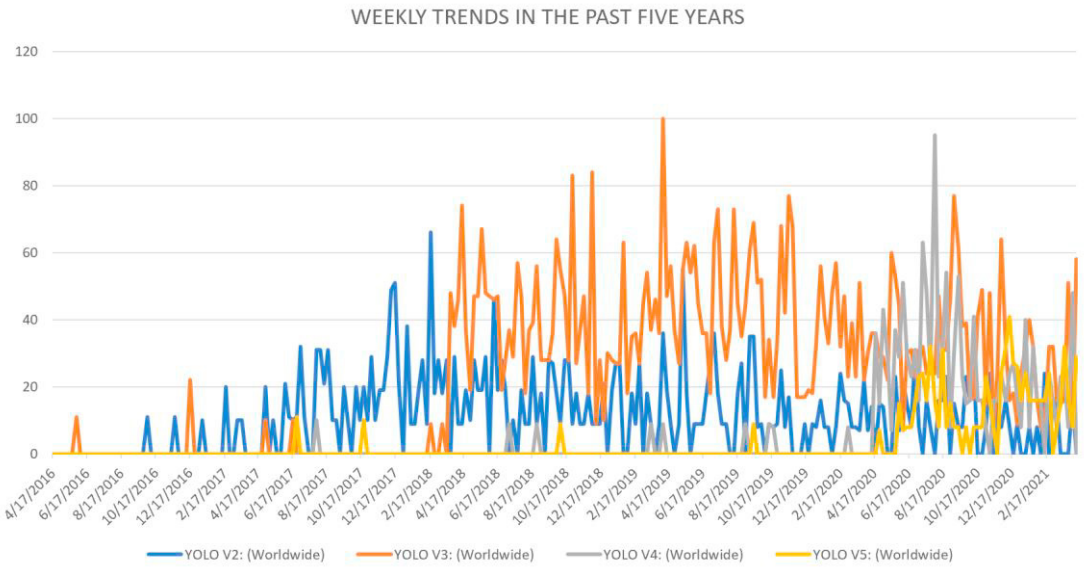

Fig. 1. Weekly trends for YOLO versions in the past five years

3.2 Public queries

This subsection gives us more details about the public's interest. We use Google and YouTube as the baseline platforms to compare the four YOLO versions. According to the search keyword analysis. Table 2 and Table 3 summarized the top ten queries for each of them.

Table 2. Top ten queries for each YOLO version (V2 and V3)

YOLO V2		**YOLO V3**	
GOOGLE	**YOUTUBE**	**GOOGLE**	**YOUTUBE**
yolo v2	yolo v2	yolo v3	yolo v3
yolo v2 paper	yolo v2 matlab	yolo v3 paper	yolo v3 vs v4
yolo v2 architecture	yolo v2 object detection	yolo v3 github	yolo v3 training
yolo v2 github	yolo v2 pytorch	yolo v3 pytorch	yolo v3 demo
yolo v2 pytorch	yolo v2 loss function	yolo v3 vs v4	yolo v3 object detection
yolo v2 matlab	yolo v2 vs v3	yolo v3 architecture	yolo v3 tutorial
yolo v2 vs v3	yolo v2 explained	yolo v3 vs v4 vs v5	yolo v3 colab
yolo v2 loss function	yolo v2 algorithm	yolo v3 weights	yolo v3 in hindi
yolo v2 explained	yolo v2 tiny	yolo v3 tensorflow	yolo v3 custom dataset

yolo v2 algorithm	yolo v2 tensorflow	yolo v3 object detection	yolo v3 code

Table 3. Top ten queries for each YOLO version (V4 and V5)

YOLO V4		YOLO V5	
GOOGLE	**YOUTUBE**	**GOOGLE**	**YOUTUBE**
yolo v4	yolo v4	yolo v5	yolo v5
yolo v4 alexeyab	yolo v4 tutorial	yolo v5 github	yolo v5 vs v4
yolo v4 github	yolo v4 demo	yolo v5 paper	yolo v5 tutorial
yolo v4 pytorch	yolo v4 video	yolo v5 tutorial	yolo v5 object detection
yolo v4 tiny	yolo v4 colab	yolo v5 vs v4	yolo v5 colab
yolo v4 vs v5	yolo v4 tiny	yolo v5 architecture	yolo v5 demo
yolo v4 tensorflow	yolo v4 google colab	yolo v5 darknet	yolo v5 video
yolo v4 tutorial	yolo v4 object detection	yolo v5 tensorflow	yolo v5 pytorch
yolo v4 python	yolo v4 training	yolo v5 tensorflow github	yolo v5 paper
yolo v4 training	yolo v4 tensorflow	yolo v5 tensorrt	yolo v5 architecture

The top ten queries in Table 3 give us more details about what the researchers care. One of the insights is the versions comparing. Many researchers care about the difference among the several YOLO versions. Also, many people want to know what is the code or tutorial, which is implementation-oriented.

4. Conclusion

This paper gives us a review of the YOLO versions. Here we draw the following remarks. First, the YOLO version has a lot of differences. However, they still have some features in common. Hence, they are still similar. Second. The YOLO versions are still very new, have a lot of room for future research. Especially for scenario implementations.

There is still room for future improvement. This paper can focus more on the implementations comparing, such as scenario analysis. Further, the research for YOLO V1 is very limited in this paper. For example, in the trend subsection, both the figure and tabular have ignored YOLO V1. Future research can do better on this point.

Funding statement

This research has been partially supported by grants from the National Natural Science Foundation of China (Nos. 71774134, U1811462). This research is also supported by the Fundamental Research Funds for the Central Universities,Southwest Minzu University(Grant Number 2020NGD04,and 2018NZD02).

References

[1] Sultana, F., Sufian, A., & Dutta, P. (2020). A review of object detection models based on convolutional neural network. Intelligent Computing: Image Processing Based Applications, 1-16.

[2] Zhiqiang, W., & Jun, L. (2017, July). A review of object detection based on convolutional neural network. In 2017 36th Chinese Control Conference (CCC) (pp. 11104-11109). IEEE.

[3] Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 30(11), 3212-3232.

[4] Zou, X. (2019, August). A Review of Object Detection Techniques. In 2019 International Conference on Smart Grid and Electrical Automation (ICSGEA) (pp. 251-254). IEEE.

[5] Laroca, R., Severo, E., Zanlorensi, L. A., Oliveira, L. S., Gonçalves, G. R., Schwartz, W. R., & Menotti, D. (2018, July). A robust real-time automatic license plate recognition based on the YOLO detector. In 2018 International Joint Conference on Neural Networks (IJCNN) (pp. 1-10). IEEE.

[6] Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., & Liang, Z. (2019). Apple detection during different growth stages in orchards using the improved YOLO-V3 model. Computers and electronics in agriculture, 157, 417-426.

[7] Jamtsho, Y., Riyamongkol, P., & Waranusast, R. (2021). Real-time license plate detection for non-helmeted motorcyclist using YOLO. ICT Express, 7(1), 104-109.

[8] Han, J., Liao, Y., Zhang, J., Wang, S., & Li, S. (2018). Target fusion detection of LiDAR and camera based on the improved YOLO algorithm. Mathematics, 6(10), 213.

[9] Lin, J. P., & Sun, M. T. (2018, November). A YOLO-based traffic counting system. In 2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI) (pp. 82-85). IEEE.

[10] Lu, J., Ma, C., Li, L., Xing, X., Zhang, Y., Wang, Z., & Xu, J. (2018). A vehicle detection method for aerial image based on YOLO. Journal of Computer and Communications, 6(11), 98-107.

[11] Huang, R., Pedoeem, J., & Chen, C. (2018, December). YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 2503-2510). IEEE.

[12] Gong, B., Ergu, D., Cai, Y., & Ma, B. (2020). A Method for Wheat Head Detection Based on YOLO V4.

[13] Jamtsho, Y. , Riyamongkol, P. , & Waranusast, R. . (2019). Real-time bhutanese license plate localization using yolo. ICT Express, 6(2).

[14] Redmon, J. , Divvala, S. , Girshick, R. , & Farhadi, A. . (2016). You only look once: unified, real-time object detection.

[15] Bochkovskiy, A. , Wang, C. Y. , & Liao, H. . (2020). Yolov4: optimal speed and accuracy of object detection.