International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 10 | Oct 2023 www.irjet.net p-ISSN: 2395-0072

OBJECT IDENTIFICATION

Prof. Heena Patil[1], Anurag Shekhar[2], Renuka Dingore[3] , Archana sahu[4]

[1]*Head of Department, Dept. of AIML Diploma, ARMIET ,Maharashtra, India*
[2]*Lecturer, Dept. of AIML Diploma, ARMIET ,Maharashtra, India*
[3]*Lecturer, Dept. of AIML Diploma, ARMIET ,Maharashtra, India*
[4]*Lecturer, Dept. of AIML Diploma, ARMIET ,Maharashtra, India*

Abstract - *Artificial neural networks are the best and most popular method for classifying images and identifying objects in images. The paper examines them as a technique that greatly enhances the aforementioned, extremely challenging computer calculations later section of the publication includes a picture of the chosen object detector we used for our introduction experiment as well as a brief overview of its development. Also presented is a fresh way for automatically producing brand-new domain-specific datasets, which are essential during the training stage of neural networks. This proposal for future study will be based on the experiment that was completed.*

Key Words: Object detection, Convolutional neural networks, YOLO, Deep learning, Computer vision

1. INTRODUCTION

Object detection is crucial in computer systems industrial automation, automated vehicles, and vision. Real-time object detection is a difficult task. Deep Object detection training is superior to classical target detection. Region suggestion is one deep learning technique. object detection methods that produce regions of interest network proposals, and then categorize them. SPPnet, region-based convolutional neural networks, fast CNN, faster-RCNN, etc. are a few examples. Object regression detection SSD and YOLO algorithms produce region proposals networks while simultaneously classifying them. This paper lists the many real-time object-detecting techniques and methods based on "YOLO" (You Only Look Once)[1].

2. NEURAL NETWORK

Many general approaches solve problems in distinctive ways while taking the least amount of time possible. In the modern era, neural networks have emerged as one of these approaches that have gained commercial traction as a result of the significant daily advancements being made in both hardware and software. They are now widely employed in a variety of computer science fields, from authentication to Arduino microcontroller interfaces to our study on image categorization and object recognition.Neurons are interconnected groupings of nodes that make up neural networks. These neurons receive multivariable linear combinations of variables from input functions from the data, where the values are multiplied by each function variable (i.e. weights)[2]. Later nonlinearity is given to this linear combination, giving the neural networks the ability to model intricate nonlinear relationships. More layers are possible in neural networks, where the input for one layer serves as the output for the next. Additionally, learned datasets are used by neural networks during the learning and detection operations.There are several algorithms today that use different kinds of neural networks. Their historical development is discussed in section 2.1 after that.

2.1 History of neural network

Since 1958 [3], when Frank Rosenblatt began researching how information from the physical environment is stored in biological systems to be used for detection or behavioral effects in the future, there have been easy methods for building one of the first neural networks.Later models with numerous sequentially non-linear layers of neurons were created; these models date to the 1960s [4] and 1970s [5]. The gradient descent method, often known as backpropagation [6], was initially applied to a neural network in 1981 for supervised learning in discrete, differentiable networks of any depth.Because neural networks had so many different layers at the time, it was difficult to develop them, and their advancement stalled until the introduction of unsupervised learning [7] techniques at the beginning of the 1990s .

There were notable advancements in this type of field during the 1990s and 2000s of the previous century. The agent looks into a foreign world and uses the trial-and-error technique [8] to learn about its surroundings, getting better with each new activity it tries. This newly created reinforcement learning method [9] is used by the agent.The use of neural networks in numerous fields drew a lot of researchers in the third millennium [10], leading to some of the greatest algorithms. Since 2009, neural networks have excelled in several contests, particularly those involving pattern recognition.When convolutional neural networks were created for the image classification task on the ImageNet challenge by Alex Krizhevsky et al. in 2012, the pattern recognition was significantly improved [11]. He and his team triumphed in the contest and produced a cutting-edge image classification technique that is still in use today.

2.2 Datasets

There are many different datasets available today for machine learning, but we'll focus on picture datasets because they're crucial for tasks like object detection and image categorization.Since their meaning is only understood when they contain a vast amount of data, creating image datasets takes a fair bit of effort. Objects are labeled and precisely located using bounding boxes to produce the picture datasets required for object detection and image classification. These fully automated solutions for tagging and locating things are no longer available.The community is interested in developing a mechanism that automates the production of these datasets since we want to focus our research on domain-specific environments.In our two trials, we experimentally tested the object detector for images within the YOLO architecture and the convolutional neural network, both of which we plan to be used in its construction (section 3.2). We want to compile photographs from the internet of objects belonging to the same classes in a range of shapes and hues against a transparent or solid-colored background[12]. To collect more images for the training phase, we might later extract certain objects from the images and programmatically change their brightness, light settings, shadows, etc. As seen in the following graphic, our goal is to place those objects onto backgrounds that were generated at random, with random placements and overlapping (Fig.1).

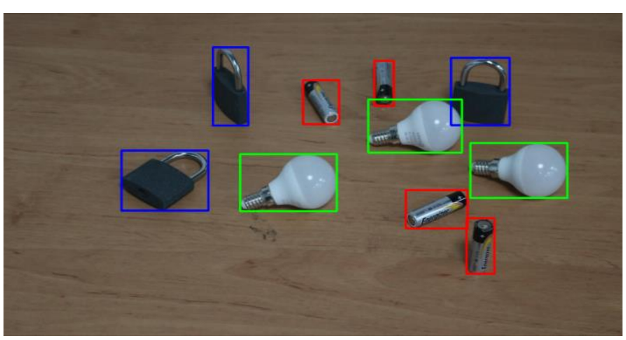

Fig -1: Background created at random, using elements that are randomly placed and overlapped.

3. THE EXPERIMENTS AND DETECTOR YOLO

The object detector known as YOLO was developed by Redmon, J., et al. [13]. We employed the YOLO authors' state-of-the-art Darknet neural network in our research since they claim [14] that it is the fastest and most accurate image object detector currently available.

3.1 Detector

Every cell in the grid predicts B bounding boxes and their confidence, and each image is partitioned by YOLO into a grid of size S x S. The precision and dependability of the bounding box used to locate and categorize an object depends on its confidence level. The confidence of an object is described as follows:

$$\mathrm{P}(Object) * IOUpred\ truth \qquad (1)$$

3.2 Experiments

On a dataset that had already been trained using COCO, we ran two tests using the detector YOLO [15]. In the first, we demonstrated how the detector operates using the image below (Fig. 2), and in the second, we put the detector to the test on a set of 500 photos to experimentally verify its performance.

various resolutions for image detectors In this study, we compared object detection and image categorization using the same image processed on the Intel Core i7-7700K processor (Table 1) and GeForce GTX 1070 graphics card (Table 2).Fig.2. Used Image for this experiment .

Comparing the two tables reveals that the processor processes data classify images, and detects objects substantially more slowly than the graphic card. Additionally, when the resolution increases, more things are detected, which is a result of improved image clarity.

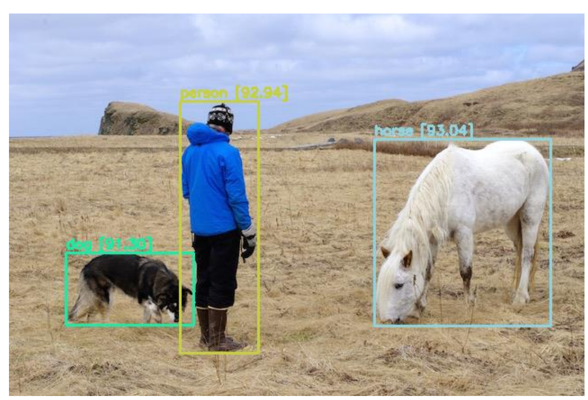

Fig -2: Used Image for this experiment.

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 10 | Oct 2023 www.irjet.net p-ISSN: 2395-0072

Table-1: Processor Testing for Object Detection.

Resolution	Objects detected	Time in ms
368x274	9	1649.214
766x577	10	1653.339
1009x757	12	1820.924
2026x1522	14	1699.288
4042x3034	15	1530.113

Table-2: Testing for Objects Detection on the Graphic Card

Resolution	Objects detected	Time in ms
368x274	9	184.574
766x577	10	222.665
1009x757	12	262.817
2026x1522	14	178.224
4042x3034	15	164.799

Applying the detector to the 500-image sequence. We expanded our initial experiment to include the detection of items on the 500-image series. Additionally, based on the findings of our last study, Different resolutions were no longer used because they had no effect[16]. We also reported the average time and quantity of detected items for the entire set of photos. Similar to the prior experiment, we employed processing from the processor and graphics card. The following tables display the findings (Table 3 and Table 4)

Table-3: Testing of Object Detection on 500 Images on the Processor.

Property	Objects detected	FLOPS	Time in ms
quickest detection	18	64.867	2288.822
The last rapid detection	18	64.867	1301.273
average period	14.23	64.867	1363.503
the most things	42	64.867	1605.41
the smallest items	3	64.867	1872.51

Table-4: Testing Object Detection in 500 Images on the Graphic Card

Property	Objects detected	FLOPS	Time in ms
quickest detection	18	64.867	210.742
The last rapid detection	18	64.867	157.003
average period	14.23	64.867	122.299
the most things	42	64.867	169.685
the smallest items	3	64.867	188.625

The processing time for the series of 500 photos ranges from 1301.273 to 2288.822 milliseconds, with an average detection time of 1686.303 milliseconds for the processor and an average detection time of 169.670 milliseconds for the graphic card[17].

According to the results of this experiment, the quantity of items discovered has no bearing on the speed of detection (both the quickly and slowly processed photos include the same quantity of objects), and both the quickly and slowly processed images take nearly the same length of time.

4. FUTURE WORK

In the future, we plan to process a sizable number of photos using the YOLO detector to train an automated dataset-generating system. We will direct the processes toward a graphic card based on our findings. The card we were using managed 5 FPS.

Additionally, fully new methodologies for the automated production of domain-specific datasets will be designed as part of future research. We anticipate that the method will play a significant role in speeding up the process of building new datasets, particularly during the labeling stage where each object on an image needs to be accurately placed within its bounding box. This method should fully eliminate human involvement in the labeling process, which is now done by hand. The technique would also be used for real-time detections and a variety of other jobs, for example teaching pupils how to study certain objects in the same way that youngsters do from their very first days of existence or identifying specific species of a particular kind.The last point I want to make is how challenging it is to build these datasets because the majority of object labeling and positioning in the images is done manually. Our approach would enable researchers in many different domains to acquire notably improved findings because their datasets are typically fairly tiny and could have an impact on the result's accuracy, as is indicated in these studies [18, 19].We might train neural networks in specific contexts using our approach for automatically generating domain-specific datasets, which would substantially help in identifying not just the class of an object but also its types and subclasses. A detected flower, for example, would be more properly classified as a forget-me-not, and a detected tree, as a baobab[20].

5. CONCLUSION

In this study, we suggest employing the YOLO network model to detect objects. We put the image that has deteriorated during the degenerative model training process. Investigations reveal that by Using degraded photos as training data, a network can learn more features and become more adaptable to complicated scene types.

The outcomes demonstrate that the model enhances the object detection's average precision. Better generalization and more resilience are characteristics of the model that was developed using the degraded training sets.

REFERENCES

[1] Diwan, Tausif, G. Anirudh, and Jitendra V. Tembhurne. "Object detection using YOLO: challenges, architectural successors, datasets and applications." Multimedia Tools and Applications (2022): 1-33.

[2] Ahmad, Tanvir, et al. "Object detection through modified YOLO neural network." Scientific Programming 2020 (2020).

[3] Lee, Yong-Hwan, and Youngseop Kim. "Comparison of CNN and YOLO for Object Detection." Journal of the semiconductor & display technology 19.1 (2020): 85-92.

[4] Liu, Chengji, et al. "Object detection based on YOLO network." 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC). IEEE, 2018.

[5] Long, Xiang, et al. "PP-YOLO: An effective and efficient implementation of object detector." arXiv preprint arXiv:2007.12099 (2020).

[6] Huang, Rachel, Jonathan Pedoeem, and Cuixian Chen. "YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers." 2018 IEEE International Conference on Big Data (Big Data). IEEE, 2018.

[7] Liu, Wenyu, et al. "Image-adaptive YOLO for object detection in adverse weather conditions." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 36. No. 2. 2022.

[8] Huang, Zhanchao, et al. "DC-SPP-YOLO: Dense connection and spatial pyramid pooling based YOLO for object detection." Information Sciences 522 (2020): 241-258.

[9] Nguyen, Duy Thanh, et al. "A high-throughput and power-efficient FPGA implementation of YOLO CNN for object detection." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 27.8 (2019): 1861-1873.

[10] Krišto, Mate, Marina Ivasic-Kos, and Miran Pobar. "Thermal object detection in difficult weather conditions using YOLO." IEEE access 8 (2020): 125459-125476.

[11] Wong, Alexander, et al. "Yolo nano: a highly compact you only look once convolutional neural network for object detection." 2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMC2-NIPS). IEEE, 2019.

[12] Lee, Yong-Hwan, and Youngseop Kim. "Comparison of CNN and YOLO for Object Detection." Journal of the semiconductor & display technology 19.1 (2020): 85-92.

[13] Yin, Yunhua, Huifang Li, and Wei Fu. "Faster-YOLO: An accurate and faster object detection method." Digital Signal Processing 102 (2020): 102756.

[14] Lee, Jeonghun, and Kwang-il Hwang. "YOLO with adaptive frame control for real-time object detection applications." Multimedia Tools and Applications 81.25 (2022): 36375-36396.

[15] Lu, Yonghui, Langwen Zhang, and Wei Xie. "YOLO-compact: an efficient YOLO network for single category real-time object detection." 2020 Chinese control and decision conference (CCDC). IEEE, 2020.

[16] Liu, Chengji, et al. “Object detection based on YOLO network.” 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC). IEEE, 2018.

[17] Liu, Chengji, et al. “Object detection based on YOLO network.” 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC). IEEE, 2018.

[18] Du, Juan. “Understanding of object detection based on CNN family and YOLO.” Journal of Physics: Conference Series. Vol. 1004. No. 1. IOP Publishing, 2018.

[19] Zhang, Shuo, et al. “Tiny YOLO optimization oriented bus passenger object detection.” Chinese Journal of Electronics 29.1 (2020): 132-138.

[20] Bathija, Akansha, and Grishma Sharma. “Visual object detection and tracking using Yolo and sort.” International Journal of Engineering Research Technology 8.11 (2019).