International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 06 | Jun 2022 www.irjet.net p-ISSN: 2395-0072

Object Single Frame Using YOLO Model

Mrs. NVN. SOWJANYA[1], NENAVATH PARAMESH[2], ANUMULA PRANAY KUMAR[3], MAAKAM ROSHINI[4]

[1]Assistant Professor in Department of CSE, Teegala Krishna Reddy Engineering College, JNTUH University, Telangana, India,
[2,3,4] UG Scholar in Department of CSE, Teegala Krishna Reddy Engineering College, JNTUH University, Telangana, India.

Abstract: *Our project on this area has been making great progress in many directions. The main goal of the project is to detect multiple object in a single frame. In this we have increased the classification accuracy of detecting objects and an overview of past research on object detection. You Only Look Once (YOLO) algorithm is the fastest, most efficient algorithm and uniquely detects.. In this comparative analysis, using the Microsoft COCO(Common Object in Context) dataset, the performance of the algorithm is evaluated and the strengths and limitations are analyzed based on parameters such as accuracy and precision.*

Keywords: Object Detection, YOLO Algorithm, Prediction

1. Introduction

A computer views all kinds of visual media as an array of numerical values. As a consequence of this approach, they require image processing algorithms to inspect contents of images. Object detection is a key ability required by most computer and robot vision systems. Our project on this area has been making great progress in many directions. In our project, we have increased the classification accuracy of detecting objects and we give an overview of past research on object detection, outline the current main research directions, and discuss open problems and possible future directions. You Only Look Once (YOLO) algorithm correlates activities and uniquely detects. The fastest and most efficient of algorithm. In this comparative analysis, using the Microsoft COCO (Common Object in Context) dataset, the performance of the algorithm is evaluated and the strengths and limitations are analyzed based on parameters such as accuracy and precision. The comparison between Single Shot Detection (SSD), Faster Region based Convolutional Neural Networks (Faster R-CNN) and You Only Look Once (YOLO), From the results of the analysis, YOLO processes images at 30 FPS and has a mAP of 57.9% on COCO test-dev. In an identical testing environment, YOLO outperforms SSD and Faster R-CNN, making it the best of these algorithms. Finally, we propose a method to jointly train on object detection and classification. Using this method, we train YOLO simultaneously on the COCO detection dataset and the ImageNet classification dataset.

2. Literature Survey

In the recent few years, diverse research work happened to develop a practical approach to accelerate the development of deep learning methods. Numerous developments accomplished excellent results and followed by continuous reformations in deep learning procedures. Object localization is the identification of all the visuals in a photograph, incorporating the precise location of those visuals. By using deep learning techniques for object identification and localization, computer vision has reached a new zenith. Due to significant inconstancies in viewpoints, postures, dimensions, and lighting positions, it is challenging to succeed in the identification of objects perfectly. Accordingly, considerable concern has been given by researchers to this area in the past few years. There are two types of object detection algorithms. Object detection algorithms using region proposal includes RCNN , Fast RCNN, and Faster RCNN, etc. These techniques create region proposal networks (RPN), and then the region proposals are divided into categories afterward. On the other side, object detection algorithms using regression includes SSD and YOLO, etc. These methods also generate region proposal networks (RPN) but divide these region proposals into categories at the moment of generation. All of the procedures mentioned above have significant accomplishments in object localization and recognition. YOLO consolidates labels in diverse datasets to form a tree-like arrangement, but the merged labels are not reciprocally exclusive. YOLO9000 enhances YOLO to recognize targets above 9000 categories employing hierarchical arrangement. Whereas YOLOv3 uses multilabel classification, it replaces the approach of estimating the cost function and further exhibits meaningful improvement in distinguishing small targets. The arrangement of this paper is as follows. Below in section 2, background information of object detection methods is covered. It includes two stage detectors with their methodologies and drawbacks. Section 3 elaborates one stage detectors and the improved version YOLO v3-Tiny. Section 4 describes implementation results and comparison of object detection methods based on speed and accuracy. Finally, section 5 summarizes the conclusion.

2.1 Related work:

In this section, we present background information. It elaborates the most representative and pioneering two-stage object detection methods with their significant contributions in object detection. First, we examine their methodologies and then explain their drawbacks.

2.1.1. HOG

HOG is a feature descriptor that is extensively applied in various domains to distinguish objects by identifying their shapes and structures. Local object structure, pattern, aspect, and representation can usually be characterized by the arrangement of gradients of local intensity or the ways of edges. In the HOG detection method, the first step is to break the source image into blocks and then distribute each block in small regions. These regions are called cells. Commonly, the blocks of image overlap each other, due to this corresponding cell may be a part of many blocks. For each pixel inside the cell, it calculates the gradients vertically and horizontally.

2.1.2 RCNN

Region based convolutional neural networks (RCNN) algorithm uses a group of boxes for the picture and then analyses in each box if either of the boxes holds a target. It employs the method of selective search to pick those sections from the picture. In an object, the four regions are used. These are varying scales, colours, textures, and enclosure.

Drawbacks of RCNN method- Based on a selective search, 2,000 sections are excerpted per image. For every region or part of the image, we have to select features using CNN. For this, if we have 'i' number of images, then selected regions will become i×2,000. The whole method of target identification through RCNN utilizes the following three models: Linear SVM classifier for the identification of objects, CNN is employed for characteristic extraction, and a regression model is required to tighten the bounding boxes. All these three processes combine to take a considerable amount of time. It increases the running time of RCNN method. Therefore, RCNN needs almost 40 to 50 seconds to predict the result for several new images .

2.1. FAST RCNN

In place of using three different models of RCNN, Fast RCNN employs one model to excerpt characteristics from the different regions. Then it distributes the regions into several categories based on excerpted features, and the boundary boxes of recognized divisions return together.

Fast RCNN uses the method of spatial pyramid pooling to calculate only one CNN representation for the whole image.

It passes one region for each picture to a particular convolutional network model by replacing three distinct models for excerption of characteristics, distributing into divisions, and producing bounding boxes.

Drawbacks of Fast RCNN method- Fast RCNN also employ a selective search method to detect concerned regions. This method is prolonged and demands a lot of time. Usually, for the detection of objects, this complete procedure needs almost two seconds for each picture. Therefore its speed is quite good in contrast to RCNN. However, if we contemplate extensive real-life datasets, then the execution of fast RCNN approach is still lacked in speed.

2.2. Faster RCNN

Faster RCNN is a transformed variant of fast RCNN. The significant difference between both is that faster RCNN implements region proposal network (RPN), but fast RCNN utilizes a selective search technique for producing concerned regions. In input, RPN accepts feature maps of picture and produces a collection of object recommendations and an objectness score per recommendation in output. Usually, this approach takes ten times less time in contrast to fast RCNN approach because of RPN.

Drawbacks of faster RCNN method- To excer2.pt all the targets in a given picture, this procedure needs multiple passes for that particular picture. Different systems are working in a sequence therefore, the performance of the upcoming operation is based on the performance of preceding operations. This approach uses region proposal networks to localize and identify the objects in a picture.

2.3. Table

Type	Characteristics	time	Drawbacks
RCNN	To generate regions, it uses selective search. From each picture, it extracts around 2000 regions.	40-50 sec	Time taken for prediction is large because several regions pass through CNN definitely, and it employs three distinct models for the detection of targets
Fast RCNN	To excerpt the features, each picture passes one time through CNN. All distinct models applied in RCNN are combined collectively to form a single model.	2 sec	The method used is prolonged and time consuming Therefore, computation time is still high.
Faster RCNN	The previous approach is replaced with the region proposal networks. Therefore, this procedure works much faster compared to previous methods.	0.2 sec	Object region proposal is timeconsuming. Different types of systems are operating in sequence. Thus, the performance of entire procedure is based on the working of the preceding operations.

3. Existing System

Object detection consists of various approaches such as fast R-CNN, Retina-Net, and Single-Shot Multi Box Detector (SSD). Although these approaches have solved the challenges of data limitation and modeling in object detection, they are not able to detect objects in a single algorithm run. We have different object detection algorithms, they are applied based on their accuracy, precision.

4. Proposed System

It will detect the objects, by using the application of Linear Support Vector Machine. We apply a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.

YOLO is extremely fast and accurate. In mAP measured at 0.5, IOU YOLO is on par with Focal Loss but about 4x faster. Moreover, we can easily tradeoff between speed and accuracy simply by changing the size of the model, no retraining required.

5. Architecture

Architecture describes how the application is going to function. Project Architecture of Object Detection describes how the user's request is taken as input and how the output is delivered. The detailed architecture is shown in

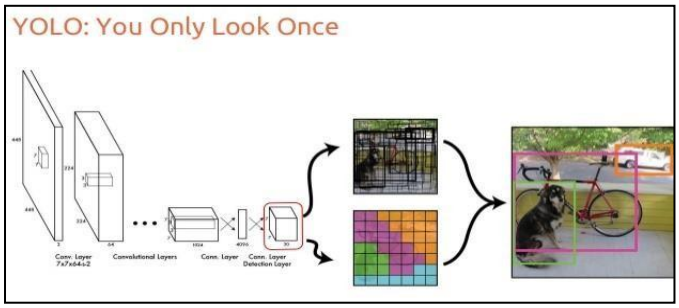

Fig 1. Project Architecture of Object Detection

You Only Look Once: Unified, Real-Time Object Detection. In this it is introduced a new approach to object detection. The feature extraction and object localization were unified into a single monolithic block. Further more the localization and classification heads were also united. Their single-stage architecture, named YOLO (You Only Look Once) results in a very fast inference time. The frame rate for 448x448 pixel images was 45 fps (0.022 s per image) on a Titan X GPU while achieving state-of-the-art mAP (mean average precision). Smaller and slightly less accurate versions of the network reached 150 fps. This new approach, together with other detectors built on light-weight Google's MobileNet backbone, brought the vision (pun intended) of detection networks and other CV tasks on edge devices ever closer to reality.

6. Implementation

For realistic execution we are using Operating System Multi Platform (Windows 7 & above, Linux GCC),and Backend as Python 3.6 & above. Dataset as COCO (Common Objects In Context)and Machine Learning Model YOLO V3 (You Only Look Once). Using this application, we can detect the objects and specify them. In order to use the application the user has to run the application and can upload a video file or image file to the program by giving the file path. Itis designed to detect many objects with the specification. It can easily find out common objects such as chairs, remotes, bus etc. The application gives a glimpse, of where the object is located with the accuracy. The core features of this project are it provides feedback in the form of video file or image file, it detects most of the common objects in context.The other features include detection of each image is reported with some form of pose information. For example, for face detection in a face detector system compute the locations of the eyes, nose and mouth, in addition to the bounding box of the face.

6.1. Python 3

If you are on Ubuntu, it's most likely that Python 3 is already installed. Run python3 in terminal to check whether its installed. If its not installed use

sudo apt-get install python3

For macOS please refer my earlier post on deep learning setup for macOS.

I highly recommend using Python virtual environment. Have a look at my earlier post if you need a starting point.

6.2. Libraries

6.2.1. Numpy

pip install numpy

This should install numpy. Make sure pip is linked to Python 3.x (pip -V will show this info)

If needed use pip3. Use sudo apt-get install python3-pip to get pip3 if not already installed.

6.2.2 OpenCV Python

OpenCV-Python

You need to compile OpenCV from source from the master branch on github to get the Python bindings. (recommended)

Adrian Rosebrock has written a good blog post on PyImageSearch on this.

(Download the source from master branch instead of from archive)

If you are overwhelmed by the instructions to get OpenCV Python bindings from source, you can get the unofficial Python package using

pip install opencv-python

This is not maintained officially by OpenCV.org. It's a community maintained one. Thanks to the efforts of Olli-Pekka Heinisuo.

6.2.3. OpenCV dnn module

DNN (Deep Neural Network) module was initially part of opencv_contrib repo. It has been moved to the master branch of opencv repo last year, giving users the ability to run inference on pre-trained deep learning models within OpenCV itself.

(One thing to note here is, dnn module is not meant be used for training. It's just for running inference on images/videos.)

Initially only Caffe and Torch models were supported. Over the period support for different frameworks/libraries like TensorFlow is being added.

Support for YOLO/DarkNet has been added recently. We are going to use the OpenCV dnn module with a pre-trained YOLO model for detecting common objects.

7. Application of YOLO

YOLO algorithm can be applied in the following fields:

7.1. Autonomous driving:

YOLO algorithm can be used in autonomous cars to detect objects around cars such as vehicles, people, and parking signals. Object detection in autonomous cars is done to avoid collision since no human driver is controlling the car.

7.2. Wildlife:

This algorithm is used to detect various types of animals in forests. This type of detection is used by wildlife rangers and journalists to identify animals in videos (both recorded and real-time) and images. Some of the animals that can be detected include giraffes, elephants, and bears.

7.3. Security:

YOLO can also be used in security systems to enforce security in an area. Let's assume that people have been restricted from passing through a certain area for security reasons. If someone passes through the restricted area, the YOLO algorithm will detect him/her, which will require the security personnel to take further action.

8. Output Screens

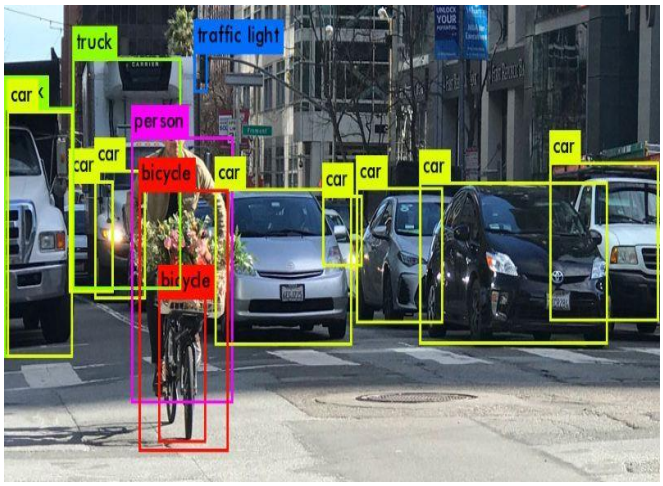

Fig 2. Object detection

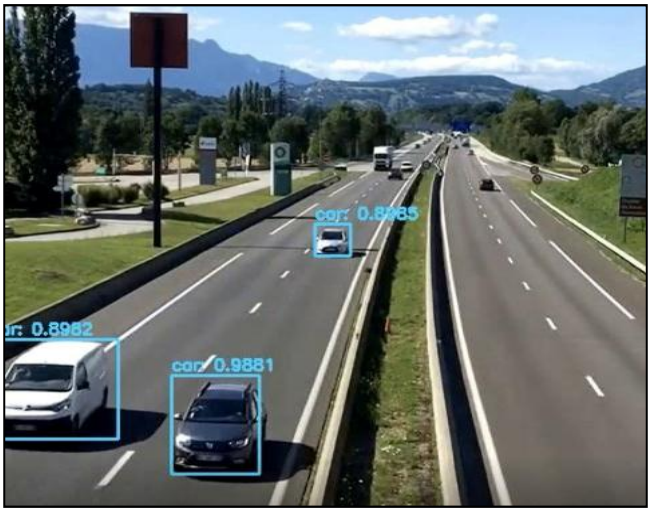

Fig 3. Motion Object Detection

9. Conclusion

Here is our project that address problems with existing system and solves them effectively. In the end, we have achieved a fully functional RCNN model that efficiently extracts ships from high resolution satellites images.

Approach helps in increasing the accuracy and speedand achieves the desired results. By using method, weare able to detect object more precisely and identify the objects individually with exact location of an object in the picture in x, y axis. implementations of the YOLO algorithm on the web using Darknet is one open-source neural network frame work.Dark net was written in the C Language, which makes it really fast and provides for making computations on a GPU,essential for real-time predictions.

10. Future Enhancement

In future we can add a faster model that runs on the GPU and use a camera that provides a 360 field of view and allows analysis completely around the person. We can also include a Global Positioning System and allow the person to detect the objects instantly without any delay in frames and seconds.

11. References

[1] G. Dahl, D. Yu, L. Deng, and A. Acero, "Contextdependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition," IEEE Transactions on Audio Speech & Language Processing, vol. 20, Jan. 2012, pp. 30-42, doi: 10.1109/TASL.2011.2134090.

[2] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition," CoRR, vol. 22, Nov. 2010, pp. 3207-3220, doi: 10.1162/NECO_a_00052.

[3] R. Collobert and J.Weston, "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning," International Conference on Machine learning (ICML 08), ACM press, Jul. 2008, pp. 160-167, doi: 10.1145/1390156.1390177. 12, 05012 (2017) DOI: 10.1051/ 7120 ITA 2017 ITM Web of Conferences itmconf/201 5012 5

[4] R. Raina, A. Madhavan, and A. Y. Ng, "Large-scale Deep Unsupervised Learning Using Graphics Processors," International Conference on Machine Learning (ICML 09), ACM press, Jun. 2009, pp. 873- 880, doi: 10.1145/1553374.1553486.

[5] Z. Y. Han and J. S. Chong, "A Review of Ship Detection Algorithms in Polarimetric SAR Images," International Conference on Signal Processing (ICSP 04), IEEE press, vol. 3, Sept. 2004, pp. 2155-2158, doi: 10.1109/ICOSP.2004.1442203.

[6] R. Raina, A. Madhavan, and A. Y. Ng, "Large-scale Deep Unsupervised Learning Using Graphics Processors," International Conference on Machine Learning (ICML 09), ACM press, Jun. 2009, pp. 873- 880, doi: 10.1145/1553374.1553486.

[7] Z. Y. Han and J. S. Chong, "A Review of Ship Detection Algorithms in Polarimetric SAR Images," International Conference on Signal Processing (ICSP 04), IEEE press, vol. 3, Sept. 2004, pp. 2155-2158, doi: 10.1109/ICOSP.2004.1442203.

[8] K. Eldhuset, "An Automatic Ship and Ship Wake Detection System for Spaceborne SAR Images in Coastal Regions," IEEE Transaction on Geoscience and Remote Sensing, vol. 34, Jul. 1996, pp. 1010-1019, doi: 10.1109/36.508418.

[9] H. Greidanus, P. Clayton, N. Suzuki, and P. Vachon, "Benchmarking Operational SAR Ship Detection," International Geoscience and Remote Sensing Symposium (IGARSS 04), IEEE press, vol. 6, Dec. 2004, pp. 4215-4218, doi: 10.1109/IGARSS.2004.1370065.

[10] C. C. Wackerman, K. S. Friedman, and X. Li, "Automatic Detection of Ships in RADARSAT-1 SAR Imagery," Canadian Journal of Remote Sensing, vol. 27, Jul. 2014, pp. 568-577, doi: 10.1080/07038992.2001.10854896.

[11] D. J. Crisp, "The State of the Art in Ship Detection in Synthetic Aperture Radar Imagery," Organic Letters, vol. 35, May 2004, pp. 2165-2168.

[112] C. Zhu, H. Zhou, R. Wang and J. Guo, "A Novel Hierarchical Method of Ship Detection from Spaceborne Optical Image Based on Shape and Texture Features," IEEE Transactions on Geoscience and Remote Sensing, vol. 48,

Sept. 2010, pp. 3446-3456, doi: 10.1109/ TGRS.2010.2046330.

Author Profile

<Authors >

Author 1 NVN. SOWJANYA

Author 2 NENAVATH PARAMESH

Author 3 ANUMULA PRANAY KUMAR

Author 4 MAAKAM ROSHINI