Optimized Yolo with Dropout

Muhammad Affan Abbas¹, Mubashir Ali²

¹Department of Computer Science, Lahore Garrison University, Lahore, Pakistan.

²Department of Software Engineering, Lahore Garrison University, Lahore, Pakistan.

Abstract:

The goal is to recognize different objects by applying the YOLO (You Only Look Once) technique. This technique has a few benefits in contrast to any other technique for object detection and tracking. In some codes as Fast Convolutional Neural Network (FCNN) and Convolutional Neural Network (CNN) the code will not focus at the picture entirely but for the case of YOLO, the code focuses the entire image by concluding the detection boxes utilizing a convolutional neural framework and the probability of classes for the bounding boxes and finds the image immediately in contrast to some different codes. The dropout layer is also programmed at the end to avoid overfitting issues. It is seen that using dropout the results have improved much.

Keywords: Bounding Boxes, Fast-Convolutional, YOLO, Neural Network, Convolutional Neural Network (CNN) and Dropout.

1. INTRODUCTION

Detecting objects is a computer application technique linked to computer vision and Image Processing that identifies case of a certain type of objects (such as people, homes, or cars) in videos and digital images. Closely examined object detection areas include pedestrian detection, face detection/recognition, and identifying individuals. Object recognition has implementations in many fields of machine vision and image analysis, including video monitoring, self-driving cars, image retrieval of and object tracking. We have different object detection methods that can be classified into two groups; the first is the classification based technique. RNN and CNN fall within that group. From this, we have to choose the required areas of the image and have to identify them using CNN. This approach is rather slow because we must carry out a forecast for each area of interest. In type 2 come the Regression-based algorithms. A single convolution network in this group predicts class probabilities and multiple bounding boxes at the same time. YOLO focuses on the class probabilities of complete photos for

certain boxes. This merged representation has varying advantages over conventional object detection techniques. At first, YOLO is quite rapid. We do not need a complicated workflow because we frame detection as a regression problem. To understand objects, we just run the neural network for test time on another image of interest. Without using batch processing on a GPU, our neural network runs at 45 frames per second, and a quick version looks for more than 150 frames per second.

2. LITERATURE SURVEY

The very first detector of objects was published in 2001 and was named the Viola Jones Object Detector [7]. While it has been theoretically graded as an image predictor, it has been the main application for facial detection. It proposed an alternative in real-time and was embraced at the time by many image processing libraries. With the emergence of Deep Learning, the area was significantly expanded. The first Deep Learning target tracker framework was named the Over feat Network [13], which used Convolutional Neural Networks; together with a

sliding window method (CNNs). This categorized every aspect of the frame as an object/non-object and then merged the details to produce the final session of assumptions. This approach for using CNNs to tackle recognition led to the creation of new networks which further advanced the latest technology even more.

The prior work is to use a regression algorithm to detect objects. In this paper, we have proposed YOLO code to have higher performance and better guessing [1]. Comprehension of object detection By Juan Du, depending on CNN types and YOLO. In that article, they generally elaborated on object detection techniques such as CNN and Region-based Convolutional Neural Networks (R_CNN) and related their performance and proposed the YOLO algorithm to boost the performance [2]. This article relates to the position and identification of objects using the bounding box approach to locate targets to address the disadvantages of the sliding window system [3].

3. OPERATION OF YOLO NEURAL NETWORKS

At the start, the clip was taken and then YOLO is implemented. The picture in our example is translated as 3x3 matrix grids. Based on the complexity of the image it can be subdivided into any combination of grids. Once completed, each grid is subjected to object classification and localization. Each grid's confidence score is found. If the grid does not have a proper object, then the grid's object and the value of bounding box should be null, or if an object is found in the grid, the object value is considered to be 1 and the bounding box score is relevant to the bounding box result for the object detected. The estimation of boundary boxes is described below. Anchor boxes were also used to increase the performance of detecting objects which are discussed in particular below.

Fig. 1. Functioning of YOLO

3.1. Predictions for Boundary Box:

YOLO code is necessary to predict precision image bounding boxes. Estimating the bounding boxes for class probabilities and in each grid, the object's image is broken down into grids. For each image grid, both the localization of objects and the image classification techniques are applied, and a mark is assigned to each grid. The program then tests each grid separately and labels an object's label and marks its bounding boxes as well. All labels are marked as zero which does not have an object. Considering the case below, an image is considered and is subdivided in a 3 x 3 matrix form. Each framework is numbered, and each framework is subjected to the techniques of locating objects and classifying images. The mark is called Y. Y is created by 8 values. Pc- Represents whether or not there is an entity in the grid. When pc=1 exists 0. Bh, bw, Bx and by are object boundary boxes (if exists). Classes stand for C1, C2, and C3. If the car is the object so C3 and C1 are 0 and value for c2 considered to be 1. The first grid, in this case, is a picture does not contain a proper entity. So they're described as since the grid doesn't have a proper pc value for the entity is 0. And the remaining ideals don't matter, because no object exists there.

Figure 4: Bounding box and Class values of grid 1.

So, they're listed as '??'. Start gridding with an object's life. The image's 5th and 6th grid each comprises an object. Let's look at the 6th line, as it is represented.

Figure 5: Bounding box and class value of grid 6.

'1' Is the existence of an entity in this list. And 'bh', 'bw', 'bx' and 'by', are the object's boundary boxes in grid 6. Car is a grid object so that the classes are (0,1,0). It is written in matrix form as $Y=3 \times 3 \times 8$. Depending on the position of the objects in the respective row, the matrix will be similarly identical for the 5th column and bounding boxes. When two or more grids have identical object then the object's middle point is placed and the grid containing that point is taken. For this, we can use two methods to get the precise object detection. They are intersection over suppression of Non-Max and Union. For IoU it selects the actual and expected bounding box value and by considering the formulas it calculates the IoU of two boxes. $IoU = \text{Intersection zone} / \text{Union zone}$. The IoU value is greater than or equal to our target level (0.5), this is a reasonable guess. The predicted value is known to be an expected value. The higher threshold value can also be used to better predict the object and increase the precision. A further strategy is Non-max suppression, involving taking boxes with high likelihood and suppressing items with higher IoU. This cycle is performed until an object is chosen and the bounding box for that element is called.

3.2. Accuracy Advancement Anchor Box:

Utilizing the detecting Box to detect objects, a grid may recognize only one object. So, we go for an anchor box to detect more than one object.

Figure 6: An example image for anchor box.

The figure above represents the image's anchor box which we considered. The anchor box in the horizontal is a car and the human being has the vertical anchor box. Considering such a scenario, in that the halfway point for both the car and the human comes under the same cell of the grid. We use an anchor box approach for such a situation. For such entities, the two anchor frames are the red grid cells. Any set of

anchor boxes can be used to detect multiple objects for one single image. In our case, we took two boxes for anchor.

The diagram mentioned shows the image's anchor box that we considered. For the individual, the vertical detection area is a human and the horizontal detection area is a vehicle(car).

Figure 8: Anchor box prediction values

p_c in the anchor box both shows the object's existence. b_x, b_y, b_h, b_w reflects the corresponding bounding box values in both anchor boxes.

The value in the detection area 1 is (1, 0,0) since the object targeting is a person.

Car is the tracked object, so the value of the class is (0, 1, and 0) in the case of anchor box 2,

The matrix shape of Y , in this case, is $Y=3 \times 3 \times 2 \times 8$ or $3 \times 3 \times 16$. It's 2×8 , owing to two anchor boxes.

4. RESULTS & DISCUSSIONS

YOLO's idea is to create a Convolutional Neural Network for predicting a tensor (7, 7, 30).

This has used a Convolutionary neural network, with 1024 output channels, to ramp up the spatial parameter to 7×7 for each position perform a linear regression by projecting the bounding box $7 \times 7 \times 2$ using two fully connected layers. Eventually, it makes a forecast by considering a box's high confidence rate. Intersection over union rate is $\text{iou_threshold} = 0.5$ several convolutional and pooling layers were added and in the end came fully connected layer to prevent overfitting. After 30 such layers dropout layer was added which improved the performance of the YOLO algorithm.

Fig 9 Design of CNN Network

4.1. The YOLO algorithm's Loss factor:

The algorithm predicts several bounding boxes for a single grid cell. We only use one detection area to measure the loss function for object liability. We use the high IoU value to pick one among the bounding boxes. The target will be charged to the box with high IoU. Varying function's loss is:

Localization function loss
Confidence function loss
Classification loss function.

The loss among assumed boundary box and ground truth value is the Localization loss. The box's abjectness is the loss of confidence itself.

4.2. Dropout and Regularization:

In a neural network, dropout is applied to each layer. It can be applied to most layer types, such as the dense layer of convolution layers, fully connected layers, and recurring layers. Dropout may be introduced on any or all of the network's hidden layers, as well as the visible or input layer. It is impossible to use on the output layer. Simple speaking regularization refers to a set of different techniques during training that reduce the complexity of a neural network model, thus preventing overfitting. There are three very popular and efficient regularization techniques called L1, L2, and dropout.

5. CONCLUSION

We have introduced the YOLO method in such a document to use a simple neural network to detect objects using a simple neural network. The dropout layer is added to the neural network to reduce overfitting and increasing performance. This technique has good generalized and once it generalizes the technique is easy to create and can be directly trained on an entire image of natural images to different domains it outperforms different strategies. The model is simple to set up and can be directly trained on an entire object. Regional Proposal Techniques restrict the algorithm to a particular area. In guessing boundaries YOLO accesses the entire frame. And also in field regions, it predicts fewer false positives. The methodology is far more successful and quickest to use in real-time compared with other classifier algorithms.

References

- [1] Joseph Redmon, Santosh Divvala, Ross Girshick, "You Only Look Once: Unified, Real-Time Object Detection", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788.
- [2] YOLO Juan Du1," Understanding of Object Detection Based on CNN Family", New Research, and Development Center of Hisense, Qingdao 266071, China.
- [3] Matthew B. Blaschko Christoph H. Lampert, "Learning to Localize Objects with Structured Output Regression", Published in Computer Vision—ECCV 2008 pp 2-15.
- [4] Wei Liu, Dragomir Anguelov, Dumitru Erhan, "SSD: Single Shot MultiBox Detector", Published in Computer Vision – ECCV 2016 pp 21-37.
- [5] Lichao Huang, Yi Yang, Yafeng Deng, Yinan Yu DenseBox, "Unifying Landmark Localization with End to End Object Detection", Published in Computer Vision and Pattern Recognition (cs.CV)
- [6] Dumitru Erhan, Christian Szegedy, Alexander Toshev, "Scalable Object Detection using Deep Neural Networks", The IEEE Conference on Computer Vision and Pattern

Recognition (CVPR), 2014, pp. 2147-2154.

[7] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", Published in Advances in Neural Information Processing Systems 28 (NIPS 2018).

[8] Joseph Redmon, Ali Farhadi, "YOLO9000: Better, Faster, Stronger", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 7263-7271.

[9] Jifeng Dai, Yi Li, Kaiming He, Jian Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks", published in: Advances in Neural Information Processing Systems 29 (NIPS 2018).

[10] Karen Simonyan, Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", published in Computer Vision and Pattern Recognition (cs.CV).

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In European conference on computer vision,

[12] I. Newton. *Philosophiae naturalis principia mathematica*. William Dawson & Sons Ltd., London, 1687. 1

[13] J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. Rubenstein. Animal population censusing at scale with citizen science and photographic identification. 2017.

[14] J. Redmon. Darknet: Open source neural networks in c. <http://pjreddie.com/darknet/>, 2013–2016. 3

[15] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pages 6517–6525. IEEE, 2017. 1, 2, 3

[16] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. arXiv, 2018. 4

[17] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497, 2015. 2

[18] O. Russakovsky, L.-J. Li, and L. Fei-Fei. Best of both worlds: human-machine collaboration for object annotation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2121–2131, 2016.

[19] M. Scott. Smart camera gimbal bot scanlime: 027, Dec 2017.

[20] A. Shrivastava, R. Sukthankar, J. Malik, and A. Gupta. Beyond skip connections: Top-down modulation for object detection. arXiv preprint arXiv:1612.06851, 2016. 3

[21] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. 2017. 3