YOLO (You Only look Once) Technology and Its' Impact in Field of Object Detection

Aditya Kumar Sharma
Cyber Security Analyst
TATA Consultancy Services
Mumbai, India
a141997@gmail.com

Akshita Singh
Department of Computer Science
and Engineering
Amity University, Mumbai.

Chirag Shetty
Department of Computer Science
and Engineering
Amity University, Mumbai

Prof. Sushila Ratre
Department of Computer Scinece
and Engineering
Amity University, Mumbai

***Abstract*--This paper will discuss in length about YOLO (You Only Look Once), a path breaking addition concerning object detection, YOLO technology in detail, the advantages and risks it carries and how this technology will massively shape up the future generations. Two review papers are considered which explain YOLO technology and the latest version of YOLO, namely YOLOv3. Also discussed are the characteristics of YOLO technology along with its' advantages, drawbacks and improvements. The Network design, architecture and the training provided to the neural network YOLOv3, a new updated version over YOLO technology and in comparison to its' predecessor how does it fare is mentioned. How YOLO technology will shape up in future, what will be its impact. A clear insight about YOLO and its' functionality is provided here.**

Keyword: YOLO, R-CNN, DPM, Detection System

I. INTRODUCTION

It is really wonderful to see how a Human mind works. A single look at an image and in an instant the mind knows, in that image what all objects are present, where exactly they are, and how they interact. The eyes of a human are so quick and perfect, thereby being able performing complicated tasks like driving with scant awareness with relative easiness. Object detection using fast, accurate type of algorithm, would make computers drive cars without any kind of special sensors. It would also facilitate assistive devices to communicate actual-time scenario to the user, thus bringing the general purpose responsive robotic system within the realms of possibility

The Current group of detection systems remodel classifiers to perform Detection problems. In order to detect a particular object, the systems will consider a classifier of that object and estimates it at various scales and location in the test image. Consider Systems like, Deformable Parts Models (DPM). In all these systems the concept of sliding window approach is used. In this approach, the classifier begins to scan each and every evenly spaced location over the entire image [1]

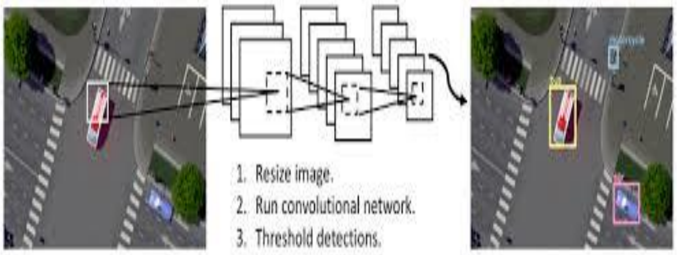


Figure 1: Demonstration on how YOLO algorithm works. YOLO has made processing of the image easy and direct. Through structure (1) input image is bumped to 448 x 448, (2) the structure makes use of a single convolutional network on the image, and (3) threshold set by the system for the generated detections by the confidence of the model.[21]

New Technology like R-CNN (Region-Convolutional Neural Network) uses region proposal methods mainly for two things: To generate numerous potential bounding boxes in an image and then make the classifier run on these proposed boxes.

Once classification is done, there is a use of post-processing. The process is carried out to filter the bounding boxes, eliminate any kind duplicate detections, and rescoring of the boxes depending on other objects in that particular scene. The complex pipelines which are used are sluggish and it is difficult to enhance them as every single part has to be skilled independently. [2]

The entire component of object detection is considered as a single regression problem, taking into consideration are components ranging from each and every image pixel to bounding box coordinates along with class probabilities. Using YOLO, user can now just look once at an image to foresee the objects which are available and what location they are available, hence YOLO (You Only Look Once)

YOLO is absolutely simple: It makes use of a single convolutional network which predicts multiple bounding boxes along with class probabilities for the boxes simultaneously. The training is carried out over the entire image and straightway enhances the performance of detection. Such kind of unified model has various advantages over other classical ways of detecting an object.

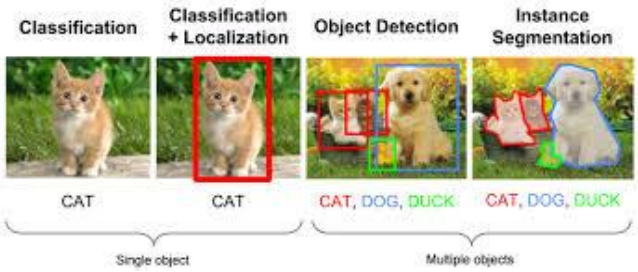


Figure 2: Demonstrating how image detection occurs [20]

II. BACKGROUND

A. CHARACTERISTIC OF YOLO TECHNOLO GY

a. Lightning fast and quick

Since regression problem is considered as frame detection, This YOLO technology looks really fast. Due to this, complex pipeline is not needed at all. The only thing that has to be done is run the neural network over the given new image during test time to predict detections. The base network is clocked at *45 frames per second* considering absence of batch processing on a Titan X GPU and a fast version clocking at more than *150 frames per second*. This indicates that streaming video can be processed by it in real-time with latency of less than 25 milliseconds. More, YOLO looks to achieve two times or more, the mean average precision over other real-time systems.

b. Global Image Specification

Considering predictions, YOLO covers each and every part of the image while making the predictions. What YOLO does differently from techniques like sliding window, region proposal-based techniques is that YOLO sees an entire image during training and test time. Due to this information depending on the classes, their appearance all are encoded implicitly. [4]

Fast R-CNN, a top detection method, commits a blunder by considering background patches available in an image for objects because it is not able picture the larger situation. Almost 50 % less of background errors are done by YOLO in comparison to Fast R-CNN. [14]

c. Easy to genelarize and reperesent

YOLO generally represents the object, when the training provided is on natural images and artworks. Better operation is observed on YOLO in comparison other top detection methods like Deformable Parts Model (DPM) and R-CNN by a huge margin. Since it is considered to be hugely theorized, YOLO has less chance of breaking down, while being in application of new domains or entirely unanticipated inputs.

B. UNIFIED OBJECT DETECTION

All separate fundamentals involved with object detection are combined within a particular single neural network. Every single feature present in the entire image is made to use by the network to predict each and every bounding box. Along with this, all bounding boxes across all classes for an image are predicted in a simultaneous manner. It implies the network reasons globally about the full image and all the objects present in the image.

The structure that YOLO is built in permits full scope training and real-timespeeds whilst high average precision is maintained. The system segregates the input image into a S*S grid. If in grid cell center there is a presence of an object, then the responsibility of detecting that object falls on that particular grid cell. Each grid cell is responsible for prediction of B bounding boxes and confidence scores for each of those boxes. The confidence scores generated are clear indication on how self-assured the model is, when it comes the box containing an object and along with it, the it is sure about how much correct the predictions are about accurate box. The confidence is given by $\Pr(Object) * IOU_{Pred}^{Truth}$. If there is an absence of other object in that cell, the confidence scores is set to zero. If that is not it, the *Intersection over Union* (IOU) between the predicted box & ground truth needs to be as equal as confidence score.

There are total of 5 predictions contained in each Bounding Boxes namely **x, y, w, h**, and **confidence.** The **(x; y)** coordinates represents the center of the box which is relative to the bounds of the grid cell. The width **(w)** and height **(h)** are predicted relative in respect to the entire image. At last **confidence** prediction will represent the IOU between the predicted box and any ground truth box.

Along with all this, Each grid cell predicts the required C conditional class probabilities represented by $\Pr(Class_i|Object)$. All the probabilities are conditioned on the grid cell which contains an object. There is exactly one set of class probabilities per grid cell which has to be predicted, disregarding the number of boxes B. While at time of testing, multiply the conditional class probabilities and the individual box confidence predictions. This is how final equation is generated.

$$\Pr(Class_i|Object) * \Pr(Object) * IOU_{Pred}^{Truth} = \Pr(Class_t) * IOU_{Pred}^{Truth}$$

[1]

This demonstrates class-specific confidence scores pertaining to each box. The generated scores clearly cover the class probability that appears in the box along with the prediction that the box fits the object perfectly.

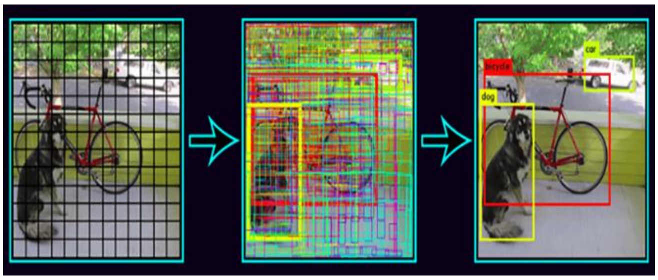


FIGURE 3: The YOLO model used for testing [22]

Considering **Figure 3**, This uses system models considers using of detection as a problem of regression. The image is divided into an S X S grid where each grid cell predicts B bounding boxes, confidence for those boxes along with C class probabilities. These predictions are encoded in this formula given below

S * S * (B * 5 + C) worth of tensor.

Figuring out YOLO on PASCAL VOC, the values considered is **S = 7, B = 2**
PASCAL VOC contains **20** labelled classes so
C = 20.
The prediction that finally generates will be a **7 * 7 * 30** worth of tensor.

C. NETWORK DESIGN

The model is implemented as a Convolutional Neural Network (CNN). Evaluation is done over a PASCAL VOC detection dataset. [3] The opening convolutional layers of the network derives component from the image whereas the layers that are connected wholly are responsible for prediction of the output probabilities and coordinates

The network consists of convolutional layers which totals up to 24 along with fully connected layers that totals up to maximum of 2. There layers used are 1 x 1 reduction layers followed with 3 x 3 Layers of Convolution. The entire network is demonstrated at **Figure 4** below

A quick version of YOLO is trained. The design is such so as to push the thresholds of fast object detection. A neural network is made to use by YOLO with lesser convolutional layers i.e. 9 layers in place of 24 along with less filters in all the layers. Along with the network size, all training and parameters pertaining to testing are similar among YOLO and Fast YOLO.

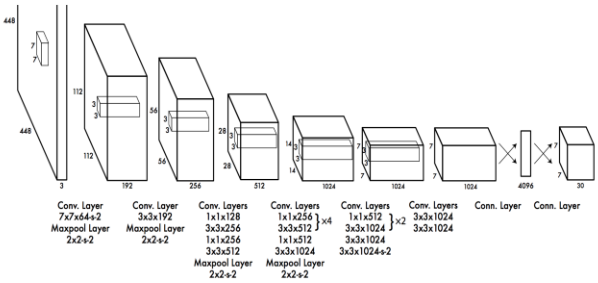


FIGURE 4: Network Design of YOLO[19]

The output of the network that is finally accomplished will **be 7 *7 *30** tensor worth of predictions.

III. TRAINING

Considering the above network, it was pre-trained to convolutional layers for pre-training; the initial 20 convolutional layers from Figure 2 were used succeeded by an average-pooling and wholly connected layer. To be able to acquire 88% of single crop top-5 accuracy, the network is made to train for at least a week. Then the model is transformed into performing the task of detection. Combination of both connected and convolutional layers is done to pre-trained networks. The result that is achieved is improvement in the overall performance. [19]

Predictions for class probabilities and bounding box is done by the coordinates. Image width and height normalizes the bounding box. This is done to make the value available somewhere in the range of 0 to 1. The bounding box specifies x and y coordinates to be offsets to a precise grid cell location by bounding them in the range of 0 to 1.

This is optimized to generate a sum-squared error in the output of the

$$Ø(x) = \begin{cases} x, & if\ x > 0 \\ 0.1x, & otherwise \end{cases} \quad \textbf{(2)}$$

model. The usage of sum-squared error is done due to its' easy optimization, but here is the catch; There is no perfect alignment towards the intention of magnifying average precision because localization error and classification error are equally weighted which is never an optimal scenario. It is also worth noting that in all the images many grid cells lack presence of the object. Thereby "confidence" scores of the cells is thrusted to zero, which in turn overpowers the gradient of the cells that have the presence of an object. This would result in model instability,

This would result in diverging of the training at the earliest stages. To tackle this, Increment is done in bounding box coordinate predictions, and decrement in loss from confidence predictions happens for boxes having no presence of an objects. The criterions λ_{coord} **and** λ_{noobj} are used to achieve this. By setting λ_{coord} **= 5 and** λ_{noobj} **= 0.5.**

Due to this in large and small boxes, Sum squared error is equally weighted. The one inference that error metric shows is worth noting. Small deviations matter less in Larger boxes compared to ones' in the small boxes. To focus on this issue partially, square root associated with the width and height of the bounding box is predicted rather than directly predicting width and height.

Multiple bounding boxes per grid cell are predicted via YOLO. During time of training, just one bounding box

predictor is needed which will carry out responsibility of every object. Single predictor is assigned to be "responsible" in order to predict an object based on which prediction has the highest current IOU with respect to ground truth. Which in turn results in specialization between the bounding box predictors. Each predictor improves and improves at predicting aspect ratios, certain sizes or classes of object. This improves the total recall. Whilst training, given multi-part loss-function is optimized

$$\lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} 1_{t_i}^{obj} [(x_t-\hat{x}_t)^2 - (y_t-\hat{y}_t)^2]$$

$$+ \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} 1_{t_i}^{obj} [(\sqrt{w_t-\hat{w}_t})^2 - \sqrt{(y_t-\hat{y}_t)^2}$$

$$+\sum_{i=0}^{s^2} \sum_{j=0}^{B} 1_{t_i}^{obj} [c_t - \hat{c}_t] + \lambda_{noobj}$$

$$\sum_{i=0}^{s^2} \sum_{j=0}^{B} 1_{t_i}^{noobj} [c_t - \hat{c}_t] +$$

$$\sum_{i=0}^{s^2} 1_{t_i}^{noobj} \sum_{ceclasses} [p(c) - \hat{p}(c)]^2 \quad \textbf{(3)}$$

Where 1_i^{obj} indicating if presence of object in cell i $1_{i,j}^{obj}$ indicating that the jth bounding box predictor in cell i is "responsible" for the particular prediction.

One thing to remember is the loss function only punishes the classification error only when there is a presence of an object in the particualr grid cell (That's why the conditional class probability). Along with that bounding box coordinate error is also punished if for ground truth box, the predictor is "responsible" i.e. containing IOU higher of any other predictor in that grid cell). The network was given training for epochs equal to 135

Validation data sets from PASCAL VOC 2007 and 2012 was considered. During the testing was done 2012, VOC 2007 test data was also included in the training. During the course of the training, 64 batch size, 0:9 momentums and 0:0005 decay was taken into use

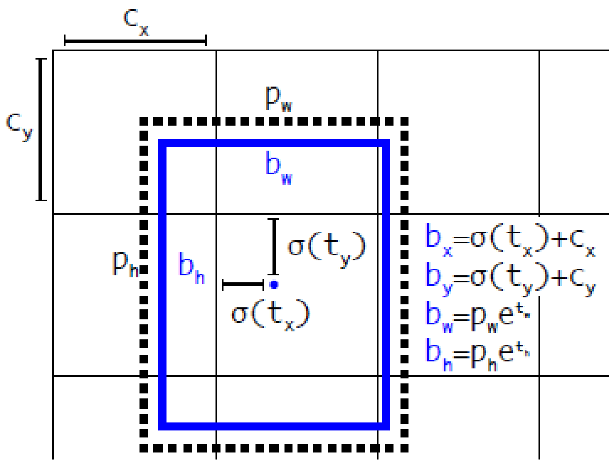
This is how rate of learning schedule presents itself : In the first epoch, the learning rate is raised gently from 10^{-3} to 10^{-2} . Commencing from a comparatively high learning rate, the model is seen deviating, mainly because of presence of highly unstable gradients. The process is continued with 10^{-2} for 75 epochs, then 10^{-3} for 30 epochs, and ending with 10^{-4} for 30 epochs. To avert the issue of overfitting, dropout along with extensive data augmentation is used. Dropout layer with rate = 0.5 succeeding intial connected layer will prevent co-adaptation amongst the layers. [5] Translation and random scaling till 20% of the Images' original size is used by Data Augmentation. Random adjust of the exposure is made along with saturation of the image with a factor of 1:5 in terms of HSV color space.

A. BOUNDING BOX PREDICTION

Considering the YOLOv3 technology, Predictions of the bounding boxes are carried out by the system with use of dimension clusters acting as anchor boxes. Total of 4 coordinates is predicted by the network for each bounding box namely, t_x, t_y, t_w, t_h. [16]. So, if (c_x; c_y) are offsets belonging to the top left corner and the p_w, p_h, are bounding box that consist of width and height (4) Accordingly, predictions that will be generated correspond to following equations:

- $b_x = \sigma(t_x) + (c_x)$
- $b_y = \sigma(t_y) + (c_y)$
- $b_w = p_w e^{tw}$ (4)
- $b_h = p_h e^{th}$

At the training time, the totality of squared error loss is considered. Ground truth for prediction of a random coordinate is $\hat{t}*$, after that the gradient will be considered as the ground truth value(This is figured out from the difference between the ground truth box and prediction) given by: $\hat{t}* - t*$. If the above equations are inverted, ground truth value can easily be calculated.



[1 8]

Figure 5: This demonstrates Bounding boxes including components like dimension priors and location prediction. As offsets, the width and height are considered which is generated from cluster centroids. In the box, coordinates present at center is predicted relative to the location of filter application with the help of a sigmoid [16] [18] function.

With the help of logistic regression, an objectness score is predicted by YOLO for each bounding box. When the bounding box prior protrude the ground truth object further than any other bounding box prior, then score sets to 1. Prediction is neglected, if the bounding box prioris seldom in best way but protrudes a ground truth object with a significant amount, more than threshold. The threshold that is considered is of :5. Different from others, the system delegates just a single bounding box prior for every ground truth object. If there are no assignment of bounding box prior to a ground truth object, zero loss is incurred by coordinate or class predictions, excluding objectness.

B. CLASS PREDICTION

Every box generates prediction for classes the bounding box might contain with the help of a multi label classification. Using softmax is deemed as unnecessary since it doesn't bring required performance, in place of that, the independent logistic classifiers are used. The class predictions when they are trained, a binary cross-entropy loss is used. The formulation will be of great help when striding towards more convulated domains such as the Open Images Dataset. The dataset consists of numerous overlapping labels (i.e. People, Man, Animals). Softmax usage generates the presupposition that each box consists of no more than one class. This is not the scenario at all. Data Modelling is achieved better with the use of Multilabel Approach. [17] [18]

C. INFERENCE GENERATED

Only single network evaluation is recommended to predict detections on test images, just like during a training. Considering PASCAL VOC, the network is able predict around 98 bounding boxes per image along class probabilities for each box.

During test time, YOLO seems to be remarkably fast. This is because unlike other classifier-based methods, only single network is needed for evaluating,

During bounding box predictions, spatial diversity is invoked by the grid design. A transparent idea regarding which grid cell an object falls in to and due to this the network predicts only one box for each particular object. Yet, some large objects or object present at the vicinity of the multiple cells borders can be well localized by multiple cells. To solve these multiple detections, Non-maximal suppression is made use of. Suppressions that are Non-maximal tallies up 2-3% in mAP, compared to R-CNN or DPM [19]

IV. LIMITATIONS PRESENTED BY YOLO TECHNOLOGY

Strong spatial constraints are imposed by YOLO on bounding box predictions because only two boxes are predicted by each grid cell and they are able to possess a maximum of one class. The close-by object count that model is able to predict is limited due to this spatial constraint. The model has shown its issues and conflicts regarding small object which are present in groups.

As model understand predicting the bounding boxes from data, object generalization in unique or uncommon aspect configurations, ratio or sizes, the model has had its struggles. Considerably raw features for predicting bounding boxes are used because considering the input image, the architecture has multiple downsampling layers. Finally, when the training is carried out over loss function which calculates and assumes detection performance, the loss function considers the errors in small bounding boxes versus large bounding box as same. A minor failure in a small box would have a superlative effect on IOU in comparison to error that are minor in a sizable box which is considered to be benignant. Incorrect localizations are hub of the grave errors.

V. HOW DOES YOLO FARE IN COMPARISON TO OTHER REAL TIME SYSTEMS

Numerous research efforts concerning are targeted in making object detections' standard detection pipelines quick. [6] [7] [8] [9] [10]

[11] For the time being only Sadeghi et al. has developed a detection system that has been able to run in real-time i.e., 30 frames per second or more. [12]. By comparing YOLO to its' DPMs' GPU implementation, which operates somewhere in between 30Hz or 100Hz. While in comparison to other system using its' mAP and Speed, the desired results have not been achieved.

Using PASCAL, Fast YOLO happens to be the fastest object detection method; it is the quickest extant object detector. It has two time more the accuracy as prior work done on real-time detectors, by having mAP of 52.7%. mAP boundary is pushed by YOLO to 63:4% whilst sustaining real-time performance. When YOLO is trained with VGG-16, The overall accuracy rises but also becomes considerably slower in comparison to YOLO. It is slower compared to different type of models taken into consideration here.

Fastest DPM productively increases the speed of DPM with mAP kept intact but with some compromise regarding real-time performance. It is affected by a factor of 2 [7]. DPM's relatively low accuracy affects real time performance which is evident during detection phase whilst comparing to neural network approaches.

Table A gives a detailed information on how effective YOLO is.

Real Time Detector	*Train*	*mAP*	*FPS*
100Hz DPM [8]	2007	16.0	100
30Hz DPM [8]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less than Real-Time			
Fastest DPM [7]	2007	30.4	15
R-CNN minus R [13]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16 [11]	2007+2012	73.2	7
Faster R-CNN ZF [11]	2007+2012	62.1	18

YOLO VGG-16	2007+2012	66.4	21

Table A demonstrates all systems that belong to real time. Running on PASCAL VOC, Fast YOLO looks to be fastest detector amongst all with accuracy two times more than any other real-time detector. YOLO consists of mAP more in comparison with fast version whilst being considerably above real-time in speed.[9]

Selective Search is replaced by R-CNN minus R along with static bounding box proposals [13]. Although it is quicker than R-CNN, real-time still prevails over it and a significant accuracy hit is faced by it due to absence of good proposals. The classification stage is significantly speeded up by R-CNN but there is still heavy reliance on selective search. A Total of 2 seconds per image is needed to generate bounding box proposals. This results in high mAP but is still far from real-time at 0:5 fps

Selective search now is replaced with Faster R-CNN which consists of a neural network which proposes bounding boxes, bearing similarities to Szegedy et al. [15] During testing, the most accurate model is able to achieve 7 fps while 18 fps is generated by a smaller, less accurate one. The VGG-16 version of Faster R-CNN higher mAP by 10 but is slacks by a factor of 6 than YOLO. The Zeiler-Fergus Faster R-CNN is sluggish than YOLO by 2.5 times as well as it has less accuracy.

VI. FUTURE OF YOLO AND YOLOv3 TECHNOLOGY

YOLOv3 is very stable. Considering COCOs average mean AP metric, even though being on level among the other SSD (Single Shot Detector) variants, it is three times faster than them. Yet it still lags behind a bit than other models like RetinaNet considering the above metric.

Considering old detection metric, YOLOv3 emerges victorious. It is on level with RetinaNet and atop many other SSD variants, thereby indicating YOLOv3 is a very powerful detector outshining others at generating decent boxes for objects but here is a catch; with increase in the IOU threshold, drop in the performance is pretty significant which indicates struggles of YOLOv3 with respect to getting the boxes in perfect alignment to the object. [18]

Formerly there were struggles for YOLO concerning the smaller objects. Now a turnabout can be observed Thanks to new multi-scale predictions, the YOLOv3 is considered to have a relatively high APs performance but worse performance is observed considering medium and larger size object.

Plotting accuracy vs speed using the AP$_{50}$ metric, one thing evident is YOLOv3 having significantly huge benefit over other detection systems. It is just better and faster!

Powerful performance gains are shown by YOLO while running at real-time performance. This is an integral middle ground in the age which is dominated by resource craving deep learning algorithms. Proceeding towards forthcoming life revolving towards automation and technology, YOLO and SSD500 type of system will prosper grandly in many folds thereby concluding the eternal AI imagination. New updated version like YOLOv3 offer a great deal of flexibility

VII. CONCLUSION

With the benefit over old classification techniques, YOLO has offered to be clearer and sharper with respect to detection of object in the image and classifying it. We did encounter an issue with YOLO regarding smaller object and issues detecting it. YOLOv3 has proved to be more improved and faster version compared to its' predecessor. Although there are issues yet to be solved with YOLOv3 regarding larger objects, its' struggles with getting objects aligned perfectly.

These problems are surely can be rectified and worked upon in future times. YOLO has given a different dynamic in terms of image processing. With the help of this fast evolving technology we are able to witness drastic change in the field of neural network for years to come.

REFERENCES

1. P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan.Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9):1627–1645, 2010.

2. R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich featurehierarchies for accurate object detection and semanticsegmentation. In Computer Vision and PatternRecognition(CVPR), 2014 IEEE Conference on, pages 580–587. IEEE,2014.

3. M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective International Journal of Computer Vision, 111(1):98–136, Jan. 2015

4. R. B. Girshick. Fast R-CNN. CoRR, abs/1504.08083, 2015

5. G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, andR. R. Salakhutdinov. Improving G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580, 2012.

6. T. Dean, M. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan,J. Yagnik, et al. Fast, accurate detection of 100,000 object classes on a single machine In ComputerVision and Patter Recognition (CVPR) 2013 IEEE Conference on, pages 1814–1821. IEEE, 2013.

7. J. Yan, Z. Lei, L. Wen, and S. Z. Li. The fastest deformable part model for object detection. In Computer Vision and PatternRecognition (CVPR), 2014 IEEE Conference on, pages 2497–2504. IEEE, 2014

8. M. A. Sadeghi and D. Forsyth. 30hz object detection with dpm v5. In Computer Vision–ECCV 2014, pages 65–79. Springer, 2014

9. R. B. Girshick. Fast R-CNN. CoRR, abs/1504.08083, 2015.

10. K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. arXiv preprint arXiv:1406.4729, 2014.

11. 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

12. M. A. Sadeghi and D. Forsyth. 30hz object detection with dpm v5. In Computer Vision–ECCV 2014, pages 65–79. Springer, 2014.

13. K. Lenc and A. Vedaldi. R-cnn minus r. arXiv preprint arXiv:1506.06981, 2015.

14. R. B. Girshick. Fast R-CNN. CoRR, abs/1504.08083, 2015

15. D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalableobject detection using deep neural networks. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 2155–2162. IEEE, 2014

16. 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.

17. I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, A. Veit, S. Belongie, V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan, and K. Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. Dataset available from https://github.com/openimages,

18. Joseph Redmon, Ali Farhadi. YOLOv3: An Incremental Improvement Pages 1-6. IEEE 2016

19. Joseph Redmon_, Santosh Divvala_y, Ross Girshick{, Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection Pages 1-10. 2016

20. https://www.kdnuggets.com/2018/09/object-detection-image-classification-yolo.html

21. https://www.researchgate.net/figure/The-YOLO-Detection-System-Redmon-et-al-2016-It-1-resizes-the-input-image-to-448_fig3_330194561

22. https://blog.statsbot.co/real-time-object-detection-yolo-cd348527b9b7