# ENHANCEMENT OF YOLOV3

**B.KRISHNA, G.SWETHA**
*CVR College of engineering Hyderabad TS, India*
*Department of computer science and engineering*
*krishna900026@gmail.com, mailid:shreyasnaani@gmail.com*

**Abstract - Object detection is one of the most important research directions for computer vision. Object detection is a technique that detects the semantic objects of a particular class in digital images and videos. Modifications have been done to YOLO!. It have been trained this new network that's pretty's well. A small design changes have done to make better than last time but more accurate. At 416X416 YOLOv3 runs in 22 ms at 28.2 mAP, as accurate as MSD but three times faster. When we look at the 0 .5 IOU mAP detection metric.YOLOv3 is quite good. It achieves 61 AP50 in 43 ms on a Titan X, compared to 57:9 AP50 in 51 ms by RetinaNet, similar performance but 4.3x faster. we have overcome to detect the small objects that appear in group. It also learn to predict bounding boxes from the data.**

**Key Words: YOLO, image processing, object detection, Deep Learning, Neural Network.**

## I. INTRODUCTION

Object detection is a vision activity for computers that both recognize and classifies one or more objects in an image. This is a computer-specific task that involves effective object location to locate and build a bounding box in the image around of object and classification of objects to predict the proper type of object. I have managed to boost YOLO. But frankly, it's just a few minor improvements that make it better, nothing like crazy interesting. I have contributed a bit to the study of other people.

One of its real-time applications consists of self-driving cars or even a visually impaired application that senses and notifies the disabled person that certain objects are facing them. Object detection algorithms can be divided into conventional approaches using the sliding window technique where the particular size window shifts the entire picture and the profound learning methods like YOLO algorithm. Our objective here is to detect several objects in an image. Bus, bottle and tablet are the most common items to detect. We use object location concepts to find more than one object in real-time systems to find the objects in the image.
In comparison to other classification algorithms, the YOLO algorithm is faster. YOLO algorithm makes position errors, but in the context it predicts fewer false positives.

These algorithms do not have a test with degraded pictures, i.e. they have been trained with academic data sets such as the Net, COCO and VOC, etc. In the real scene, the key problems of the photos are:

1) The captured images can be blurred due to the camera's instability.
2) The pictures cannot also be adequately clear because they can block the object.
3) Pictures can be of low quality, overexposure or low resolution due to bad weather.

## II. LITERATURESURVEY

Object detection is a typical problem of detection. Different approaches for enhancing efficiency have been implemented. Deep learning approaches nowadays show additional possibilities for solving this problem.
As this study aims to investigate how CNNs are operating on human detection, previous research on object detection as well as various mechanisms of CNNs will be discussed. In particular, given that detection networks are somewhat a transfer training from CNNs for classification, many of the structures are built on the basis of classification architectures (this concept of transfer learning will be introduced). The related work in YOLO will also include widely used classification structures.

Object Identification Comprehension Focused on Juan Du's CNN Family and YOLO. In this article they generally described the detection families of artifacts like CNN, R-CNN and compared their efficiency to the YOLOv3 algorithm. Structured performance regression, object position, learning to locate objects. We used the Bounding box approach to locate the objects in order to address the limitations of the sliding window approach.

The mechanism can be seen as an extension of oriented gradient histograms (HOG). The objects predicted are scored in a higher resolution according both to a rough global image template and six sections of the subject. All inputs are HOG-described. This means that the HOG's multi-model will tackle the issue of change of perspective. The use of latent vector support machine (Latent SVM) during the training reduces the detection problem to area classification. Sections are called latent variable positions. Because of its robustness, the approach was extremely significant.

## III. WORKING OF YOLOALGORITHM

YOLOv3 is one of the most common algorithms because it is extremely accurate and can work in real time. The algorithm works on images, webcam, videos. I.e. only one forward propagation is required to predict. It gives the name of the object recognized along with the anchor boxes around the object after not being removed.

One object can be identified by grid using bounding boxes for object detection. So we go to the Anchor Box to detect more than one thing. The input image is divided into a S * S grid by YOLO. Only one entity is forecast in each grid cell. The yellow grid cell below, for example, attempts to predict the "personal" entity whose centre falls into the grid cell (the blue dot).

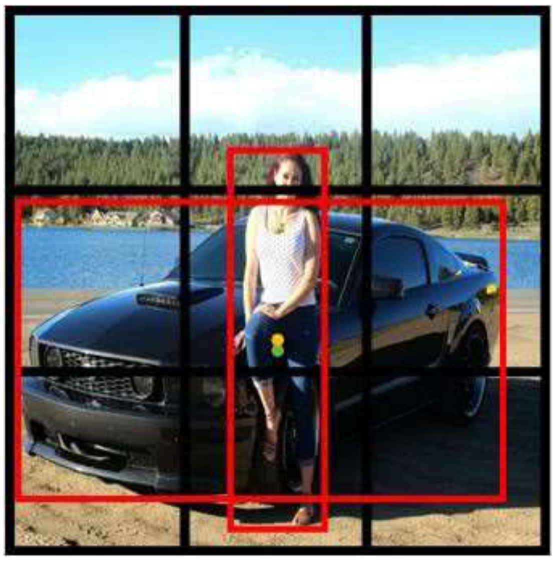


*Fig 1- Detection of object using anchor box*

Take the image above, since both the person and the midpoint are under the same grid cell. The anchor box approach is used in this case. Two anchor boxes of these items are labeled with purple grid cells. For a single image, multiple objects can be found in any number of anchor boxes.

### *Prediction Class*

Each box predicts the classes which the bounding box may identify with a classification of several labels. Instead of using independent logistics classifiers, we don't use a Softmax because we find it redundant for good results. We use binary cross-entropy loss for class predictions throughout the course of training.
This formulation allows you to move to more complicated areas such as open pictures. There are multiple overlapping labels (i.e. woman and person) in this dataset. With a softmax, each box has exactly one class that sometimes is not the case. It must be assumed that a multi-label approach models the data more efficiently.

### *Prediction of Bounding Box*

After YOLO9000 our device forecasts bordering boxes using anchor boxes with dimensional clusters. For each bounding box, TX, ty, tw, Th, the network predicts 4 coordinates. If the cell is offset by (cx; cy) and the previous bounding box is pw in width and high, ph, from the upper left corner of the image, then the predictions are the right ones.

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

We use a squared error loss total during preparation. If the basis truth is ^t * our gradient is the value of ground truth (computed from the bottom truthbox) minus our prediction ^t * t *. The above equations can easily measure this fundamental truth value. YOLOv3 forecasts an object score in logistic regression for each bounding box. That should be 1 if the bounding box before is more than every other bounding box before the ground truth object is overlapped. If the previous bounding box isn't the strongest, but overlaps a real object with a threshold that's more than some threshold, then we disregard this prediction. Unlike our method, for any ground truth object only one bounding box is allocated. If a binding box before is not allocated to a real entity, there is no loss in the predictions for coordinates or classes.

On the COCO dataset the 9 clusters were:

(10×13),(16×30),(33×23),(30×61),(62×45),(59×119), (116 × 90), (156 × 198), (373 × 326).
Multi-scale Detector We have three features vectors, we can now feed them into the detector. Multiple 1x1 and 3x3 Conv layers are used before a final 1x1 Conv layer to form the final output. For medium and small scale, it also concatenates features from the previous scale.small scale detection can also benefit from the result of large scale detection. the input image is (416, 416, 3), the final output of the detectors will be in shape of *[(52, 52, 3, (4 + 1 + num_classes)), (26, 26, 3, (4 + 1 + num_classes)), (13, 13, 3, (4 + 1 + num_classes))]*. The three items in the list represent detections for three scales.

## IV. TRAINING

YOLOv3 is one of the most common computer vision real-time object sensors. The first big task is to create the dataset. COCO is an outstanding dataset for object detection with 80 classes, 80,000 images of preparation, and 40,000 images of validation. Models for object recognition, from images to health enhancement, are extremely effective, training computers that recognize the pixels that make up objects, have nearly unlimited potential.

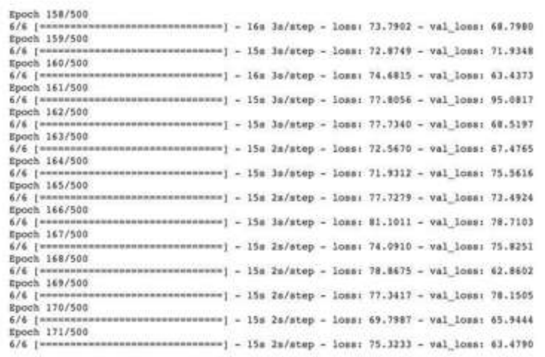
After data set is prepared, i.e. label files should be generated for each image, all images and their label files should be maintained together. The yolo.cfg file was used to train three yolo-layer configurations.

Each object should be trained for at least 2000 iterations as a conventional approach. Thus, the dataset was trained as 3(total classes) in 6000 iterations * 2000 = 6000. For optimal training speed, the batch and subdivisions values were set at 64 and 8, respectively. At a maximum speed and improved detection precision, the width and height values were set to 416. The number of filters used in the condensation layer was set to 24 since the number of filters = (classes + 5)*3 depend on the total class number in this case.

The weights thus produced were used to detect and analyses the output after 6000 iterations. It displays the load number, the current lot loss, the average loss to the current lot, the current level of learning, the batch time and photos used up to the current lot. The number of photos you can see below is up to 64 times per batch. We set the size of the batch to 64.

**Training**

This particular model is a one-shoot learner, so that each image passes only once across the network, to predict the design to be very effective and to predict video feeds with up to sixty frames per second. Basically, YOLOv3 divides an images into subcomponents before combining them to create a prediction, and performs revolutions on each of the subcomponents.



```
Epoch 158/500
6/6 [==============================] - 16s 3s/step - loss: 73.7902 - val_loss: 68.7980
Epoch 159/500
6/6 [==============================] - 15s 3s/step - loss: 72.8749 - val_loss: 71.9348
Epoch 160/500
6/6 [==============================] - 16s 3s/step - loss: 74.6815 - val_loss: 63.4373
Epoch 161/500
6/6 [==============================] - 15s 3s/step - loss: 77.8056 - val_loss: 95.0817
Epoch 162/500
6/6 [==============================] - 15s 3s/step - loss: 77.7340 - val_loss: 68.5197
Epoch 163/500
6/6 [==============================] - 15s 2s/step - loss: 72.5670 - val_loss: 67.4765
Epoch 164/500
6/6 [==============================] - 15s 3s/step - loss: 71.9312 - val_loss: 75.5616
Epoch 165/500
6/6 [==============================] - 15s 2s/step - loss: 77.7279 - val_loss: 73.4924
Epoch 166/500
6/6 [==============================] - 15s 3s/step - loss: 81.1011 - val_loss: 78.7103
Epoch 167/500
6/6 [==============================] - 15s 2s/step - loss: 74.0910 - val_loss: 75.8251
Epoch 168/500
6/6 [==============================] - 15s 2s/step - loss: 78.8675 - val_loss: 62.8602
Epoch 169/500
6/6 [==============================] - 15s 2s/step - loss: 77.3417 - val_loss: 78.1505
Epoch 170/500
6/6 [==============================] - 15s 2s/step - loss: 69.7987 - val_loss: 65.9444
Epoch 171/500
6/6 [==============================] - 15s 2s/step - loss: 75.3233 - val_loss: 63.4790
```

We primarily do six things in our notebook:

- Choose our environment, design of the model and weights for prefitting.
- Loading our information through the above shared Roboflow code snippet
- Find out our configuration of templates, such as how much time to practice, the size of the training batch, the size of our training vs.
- Training startup.
- Use our well-trained inference model.
- Save our weight to our newly qualified Google Drive, so that we can foresee the future without waiting for the training to be completed.

## V. PERFORMANCE OFALGORITHMS

MAP, IoU, and f1 score are the parameters used to monitor for model completeness. Mean average precise (mAP) is the mean value of the average precision and the average intersection over union (IoU) is dependent on the precision and reminder of the intersected over the intersection between objects and detection and is calculable based on the confusion matrix.

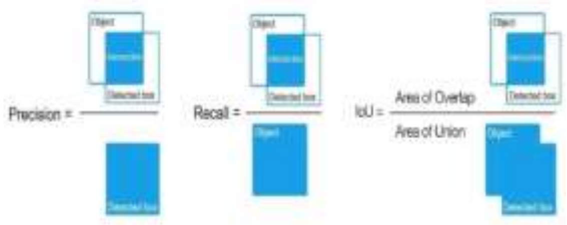
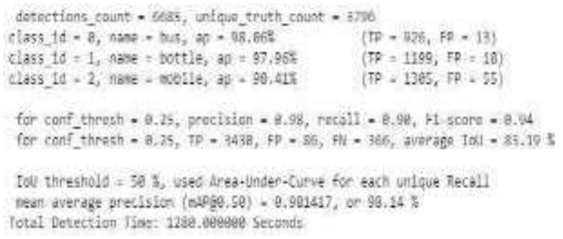


*Fig 2- Performance metrics graphical representation*

The following values are the performance metrics obtained on the training data

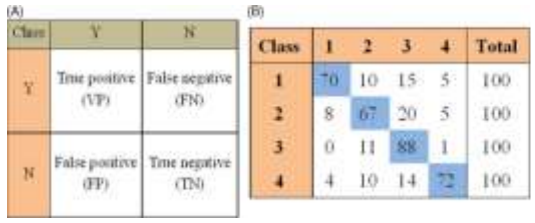


```
 detections_count = 6685, unique_truth_count = 3796
class_id = 0, name = bus, ap = 98.86%           (TP = 926, FP = 13)
class_id = 1, name = bottle, ap = 97.96%        (TP = 1199, FP = 18)
class_id = 2, name = mobile, ap = 98.41%        (TP = 1305, FP = 55)

 for conf_thresh = 0.25, precision = 0.98, recall = 0.90, F1-score = 0.94
 for conf_thresh = 0.25, TP = 3430, FP = 86, FN = 386, average IoU = 85.19 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.981417, or 98.14 %
Total Detection Time: 1280.000000 Seconds
```

**Confusion Matrix**

An uncertainty array indicates that the classification model has the number of correct and incorrect predictions in relation to the actual results (target value) of the data. NxN is the matrix, where N is the number of destinations (classes). These models are typically assessed with the data in the matrix. A 2x2 uncertainty matrix for two groups (Positive and Negative) appears in the following table.

For Bus (ClassId = 0), TP = 926 and FP = 13
For Bottle (ClassId = 1), TP = 1199 and FP =18
For Mobile (ClassId = 2), TP = 1305 and FP = 55


(A)

| Class | Y | N |
|---|---|---|
| Y | True positive (VP) | False negative (FN) |
| N | False positive (FP) | True negative (TN) |

(B)

| Class | 1 | 2 | 3 | 4 | Total |
|---|---|---|---|---|---|
| 1 | 70 | 10 | 15 | 5 | 100 |
| 2 | 8 | 67 | 20 | 5 | 100 |
| 3 | 0 | 11 | 88 | 1 | 100 |
| 4 | 4 | 10 | 14 | 72 | 100 |

**Table:Confusion Matrix**

**IoU:**

For confidence threshold 0.25, the average IoU is 83.19%

**MAP:**

For IoU threshold of 0.5 i.e. 50%, the mAP is 98.14%

**F1-score:**

The f1 mark is 0.94 for a trust 0.25 threshold

Since the code does not use the system's GPU capabilities for image processing, it is essential to process CPU frames. A frame and the bounding box are shown over the detected objects in around eight seconds. By using the GPU in the respective system, the efficiency can be greatly improved.

As pictures in the training dataset were centred on the objects and thus had a larger object body to image size, detection of the objects kept away from the camera view failed. In the best light conditions the model performs better.

## VI. RESULTS

Some web-cam and android camera detections are supported in real time:

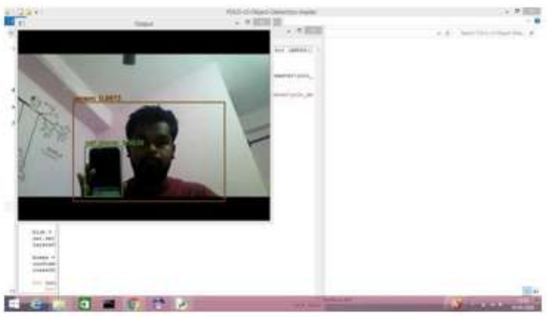


*Fig 3-Identifying objects from video*

## VII. CONCLUSION

A strong detector's YOLOv3. Its fast, it's precise. The average AP of COCO between 0.5-0.95 IOU is less good. It is not as good. But the old IOU detection metric is very fine. Never the less.

This semantic sentence is only present in the original COCO paper: "The assessment calculation measures are fully addressed until the assessment server is complete." Russakovsky et al. claim that people find it difficult to discern an IOU from 1.3 from.5! "Es remarkably difficult to train people in visually examining an IOU 0.3 bounding box and separate it from an IOU 0.5 one."

I have much confidence that most people who use a computer vision are content, good things, like counting the number of zebras in a national park, or monitoring their cat while walking around. But computer vision is already being used, and as scientists, it is our duty at least to consider the damage that our work can do and to consider ways of mitigating it. We owe so much to the universe.

## REFERENCES

*[1] Analogy. Wikipedia, Mar 2018.1*

*[2] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2):303–338, 2010.6*

*[3] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg.Dssd: Deconvolutional single shot detector. arXiv preprintarXiv:1701.06659, 2017.3*

*[4] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox,and A. Farhadi. Iqa: Visual question answering in interactive environments. arXiv*

*[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.3*

*[6]J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional acti- vation feature for generic visual recognition. arXiv preprint arXiv:1310.1531, 2013.4*

*[7]J. Dong, Q. Chen, S. Yan, and A. Yuille. Towards unified object detection and semantic segmentation. In Computer Vision–ECCV 2014, pages 299–314. Springer, 2014.7*

*[8]D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Confer-ence on, pages 2155–2162. IEEE, 2014.5,6*

*[9]M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual ob- ject classes challenge: A retrospective. International Journalof Computer Vision, 111(1):98–136, Jan. 2015.2*

*[10]P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ra- manan. Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis andMachine Intelligence, 32(9):1627–1645, 2010.1,4*

*[11]S. Gidaris and N. Komodakis. Object detection via a multi- region & semantic segmentation-aware CNN model. CoRR, abs/1505.01749, 2015.7*

*[12]S. Ginosar, D. Haas, T. Brown, and J. Malik. Detecting peo- pleincubistart.InComputerVision-ECCV2014Workshops,pages 101–116. Springer,2014.7*

*[13]R. Girshick, J. Donahue, T. Darrell, and J. Malik.*

*Rich fea- ture hierarchies for accurate object detection and semantic segmentation. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 580–587. IEEE, 2014.1,4,7*
*[14]R. B. Girshick. Fast R-CNN. CoRR, abs/1504.08083, 2015.*
*2,5,6,7*

*[15]S. Gould, T. Gao, and D. Koller. Region-based segmenta- tion and object detection. In Advances in neuralinformation processing systems, pages 655–663,2009.4*

*[16]B.Hariharan,P.Arbeláez,R.Girshick,andJ.Malik.Simul - taneous detection and segmentation. In Computer Vision–*
*ECCV 2014, pages 297–312. Springer, 2014.7*

*[17]K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. arXiv preprint arXiv:1406.4729, 2014.5*
*[18]G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and*
*R. R. Salakhutdinov. Improving neural networks by pre- venting co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580, 2012.4*

*[19]D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In Computer Vision–ECCV 2012, pages 340–353. Springer,2012.6*