ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 8, August 2019

Real Time Object Detection in Images using YOLO

Ulli Tirupataiah[1], Kunjam Nageswara Rao[2], Sita Ratnam Gokuruboyna[3], Seepana Sridhara Rao[4]

M.Tech Student, Dept. of CSSE, Andhra University College of Engineering, Visakhapatnam, Andhra Pradesh, India[1]

Professor, Dept. of CSSE, Andhra University College of Engineering, Visakhapatnam, Andhra Pradesh, India[2]

Associate Professor, Dept. of CSSE, Lendi Institute of Engineering and Technology, Visakhapatnam, India[3]

Research Scholar, Dept. of CSSE, Andhra University College of Engineering, Visakhapatnam, Andhra Pradesh, India[4]

ABSTRACT: OpenCV is a library of programming functions mainly aimed at real-time computer visionsuch as object detection**.** Object detection framed as a regression problem to spatially separated bounding boxes and associated class probabilities.YOLO, an approach to object detection, works on object detection repurposes classifiers to perform detection.. Single neural network predicts the bounding boxes and its class probabilities directly from full images. It optimized end-to-end directly on object detection performance. YOLO architecture is extremely fast, processes images in real-time at 45 fps.FastR-CNN use region proposal methods to first generate potential bounding boxes in an image and then run a classifier on these proposed boxes. Post-processing is used to refine the bounding boxes, eliminate duplicate detections, and rescore the boxes based on other objects in the scene.YOLO makes fewer background mistakes than Fast R-CNN. In this paper YOLO and fast R-CNN is combined to produce better results.

KEYWORDS: object detection, combining YOLOand Fast R-CNN.

I. INTRODUCTION

Humans glance at an image and instantly know what objects are in the image, where they are, and how they interact. The human visual system is fast and accurate, allowing us to perform complex tasks like driving with little conscious thought. Fast, accurate algorithms for object detection would allow computers to drive cars without specialized sensors, enable assistive devices to convey real-time scene information to human users, and unlock the potentialfor general purpose, responsive robotic systems. Current detection systems repurpose classifiers to perform detection. To detect an object, these systems take a classifier for that object and evaluate it at various locations and scales in a test image. Systems like deformable parts models (DPM) use a sliding window approach where the classifier is run at evenly spaced locations over the entire image.More recent approaches like R-CNN use region proposalmethods to first generate potential bounding boxes in an image and then run a classifier on these proposed boxes. After classification, post-processing is used to refine the bounding boxes, eliminate duplicate detections, and rescore the boxes based on other objects in the scene.Reframeobject detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. Using system like, you only look once (YOLO) at an image to predict what objects are present and where they are. YOLO is refreshingly simple. A single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes.YOLO trains on full images and directly optimizes detection performance. This unified model has several benefits over traditional methods of object detection. First, YOLO is extremely fast.Simplyrunneural network on a new image at test time to predict detections.Base network runs at 45 frames per second with no batch processing on a Titan X GPU and a fast version runs at more than 150 fps. To get more accuracy we combine YOLO and fast R-CNN.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 8, August 2019

II. RELATED WORK

Object detection (OD) system finds objects in the real world by making use of the object models which is known a priori. This task is comparatively difficult to perform for the machines as compared to Humans who perform OD very effortlessly and instantaneously. This paperwill give a review of the various techniques and approaches that are used to detect objects in images. Basically an OD system can be described easily by seeing, which shows the basic stages that are involved in the process of OD. The basic input to the OD system can be an image or a scene in case of videos.
The basic aim of this system is to detect objects that are present in the image or scene or simply in other words the system needs to categorise the various objects into respective object classes [2].

Fast R-CNN first fine-tunes a CommNet on object proposals using log loss. Then, it fits SVMs to ConvNet features. These SVMs act as object detectors, replacing the softmax classifier learnt by fine-tuning. In the third training stage, bounding-box repressors are learned.R-CNN is slow because it performs a ConvNet forward pass for each bject proposal, without sharing computation. Spatial pyramid pooling networks (SPPnets) were proposed to speed up R-CNN by sharing computation. The SPPnet method computes a convolutional feature mAP for the entire input image and then classifies each object proposal using a feature vector extracted from the shared feature mAP. Features are extracted for a proposal by maxpooling the portion of the feature mAP inside the proposal into a fixed-size output (e.g., 6 × 6)multiple output sizes are pooled and then concatenated as in spatial pyramid pooling. SPPnet accelerates R-CNN by 10 to 100× at test time. Training time is also reduced by 3× due to faster proposal feature extraction [5].

III. METHODOLOGY

OpenCV-Python:
OpenCV 4.0 supports the deep learning frameworks like Fast R-CNN, YOLO, and easy integration with an OpenCV application, OpenCV's CPU implementation of the DNN module is astonishingly fast. For example, Darknet when used with OpenCV takes about 2 seconds on a CPU for inference on a single image. In contrast, OpenCV's implementation runs in a mere 0.22 Compared to other programming languages python is suitable for my project, OpenCV supports Python 3.7, and it is a package based language. It is very easy and simple.

1.Real-time detection:
Unify the separate components of object detection into a single neural network.Network uses features from the entire image to predict each bounding box. It also predicts all bounding boxes across all classes for an image simultaneously. This means network reasons globally about the full image and all the objects in the image. The YOLO design enables end-to-end training and real time speeds while maintaining high average precision. System divides the input image into an S × S grid. If the centre of an object falls into a grid cell, that grid cell is responsible for detecting that object. Each grid cell predicts B bounding boxes and confidence scores for those boxes. These confidence scores reflect how confident the model is that the box contains an object and also how accurate it thinks the box is that it predicts.If no object exists in that cell, the confidence scores should be zero. Otherwise want the confidence score to equal the intersection over union (IOU) between the predicted box and the ground truth. Each bounding box consists of 5 predictions: x, y, w, h, and confidence. The (x, y) coordinates represent the centre of the box relative to the bounds of the grid cell. The width and height are predicted relative to the whole image. Finally the confidence prediction represents the IOU between the predicted box and any ground truth box.Each grid cell also predicts C conditional class probabilities. These probabilities are conditioned on the grid cell containing an object, that only predict,one set of class probabilities per grid cell, regardless of the number of boxes B. At test time multiply the conditional class probabilities and the individual box confidence predictions,which gives us class-specific confidence scores for each box. These scores encode both the probability of that classappearing in the box and how well the predicted box fits the object.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 8, August 2019

2.Network design

Implementing this model as a convolutional neural network and evaluate it on the PASCAL VOC detection dataset. The initial convolutional layers of the network extract features from the image while the fully connected layers predict the output probabilities and coordinates. Networkarchitecture is inspired by the GoogLeNet model for image classification, network has 24 convolutional layers followed by 2 fully connected layers. Instead of the inception modules used by GoogLeNet, simply use 1 × 1 reduction layers followed by 3 × 3 convolutional layers, similar to Lin et al.

The full network is shown in Figure 1. Also train a fast version of YOLO designed to push the boundaries of fast object detection. Fast YOLO uses a neural network with fewer convolutional layers (9 insteadof 24) and fewer filters in those layers. Other than the size of the network, all training and testing parameters are thesame between YOLO and Fast YOLO.

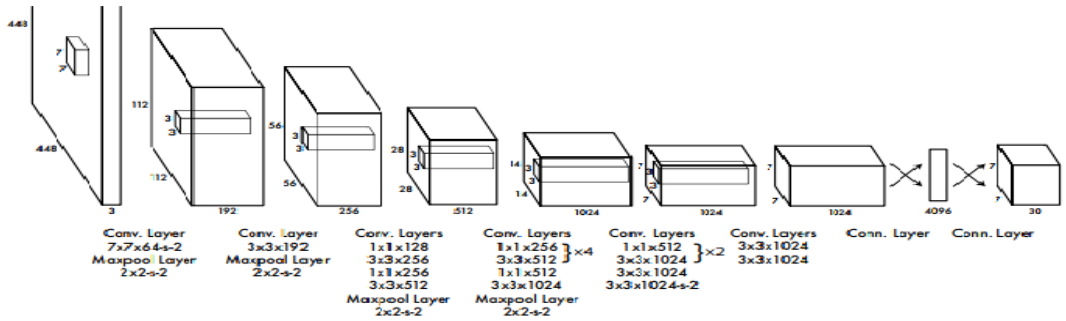

Figure 1: Shows Architecture detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1 × 1 convolutional layers reduce the features space from preceding layers. Network pre train the convolutional layers on the PASCAL VOC 2012 classification task at half the resolution (224 × 224input-image) and then double the resolution for detection.

3. Training

Apretrainedconvolutional layers on the PASCAL VOCdataset. For pretrainingusethe first 20 convolutional layers from Figure 1 followed by a average-pooling layer and a fully connected layer. For train this network approximately a week and achieve a single crop top-5 accuracy of 88% on the PASCAL VOC 2012 validation dataset, comparable to the GoogLeNet models in Caffe's Model Zoo. Final layer predicts both class probabilities and bounding box coordinates. To normalize the bounding box width and height by the image width and height so that they fall between 0 and 1. Parameterize the bounding box x and y coordinates to be offsets of a particular grid cell location so they are also bounded between 0 and 1. Figure 2 shows the outcome of all the three steps individually.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 8, August 2019

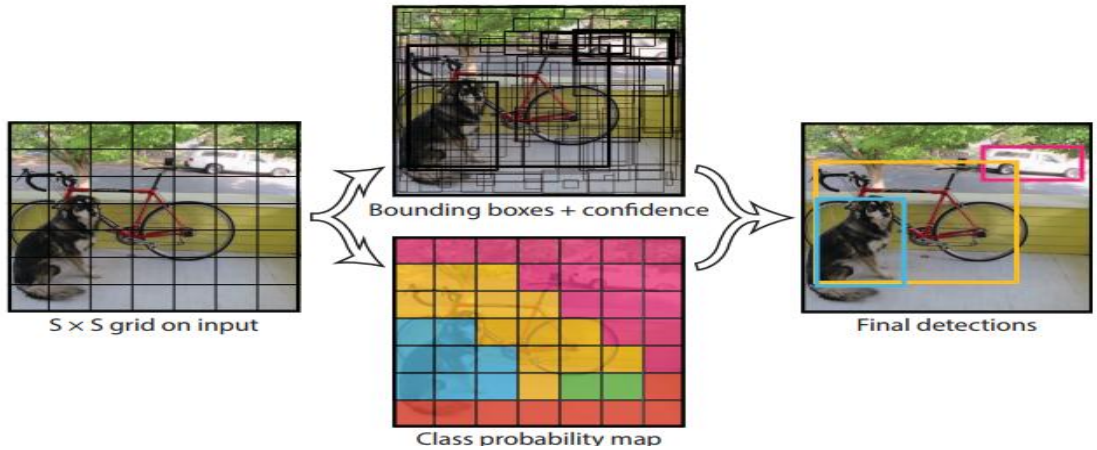

Figure 2: Input (a), system model detection as a regression problem. It divides the image into an S × S grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. Thesepredictions are encoded as an S × S × (B ∗ 5 + C) tensor. For evaluating YOLO on PASCAL VOC, use S = 7, B = 2. PASCAL VOC has 20 labelled classes so C = 20.final prediction is a 7 × 7 × 30 tensor as output (b).

Limitations to YOLO

YOLO imposes strong spatial constraints on bounding box predictions since each grid cell only predicts two boxes and can only have one class. This spatial constraint limits the number of nearby objects that model can predict. This model struggles with small objects that appear in groups, such as flocks of birds. Since model learns to predict bounding boxes from data, it struggles to generalize to objects in new or unusual aspect ratios or configurations. This model also uses relatively coarse features for predicting bounding boxes since architecture has multiple down sampling layers from the input image.Finally, while train on a loss function that approximates detection performance, loss function treats errorsthe same in small bounding boxes versus large bounding boxes. A small error in a large box is generally benign but a small error in a small box has a much greater effect on IOU, main source of error is incorrect localizations.

IV. EXPERIMENTAL RESULTS

First of all compare YOLO with other real-time detection systems on PASCAL VOC 2007. To understand the differences between YOLO and R-CNN variants explore the errors on VOC 2007 made by YOLO and Fast R-CNN, one of the highest performing versions of R-CNN . Based on the different error profiles show that YOLO can be used to rescore Fast R-CNN detections and reduce the errors from background false positives, giving a significant performance boost. Also present VOC 2012 results and compare mAP to current state-of-the-art methods. Finally, show that YOLO generalizes to new domains better than other detectors on two artwork datasets.To further examine the differences between YOLO and state-of-the-art detectors, look at a detailed breakdownof results on VOC 2007. Compare YOLO to Fast RCNN since Fast R-CNN is one of the highest performing detectors on PASCAL VOC and its detections are publicly available. Using the methodology and tools of Hoiem et al. For each category at test time look at the top N predictions for that category. Each prediction is either correct orit is classified based on the type of error.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 8, August 2019

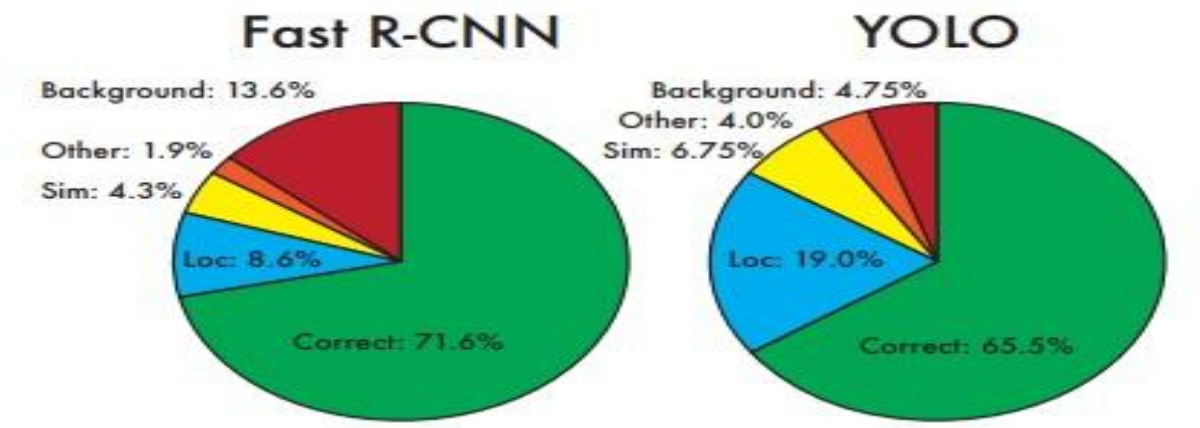

Figure 3: Error Analysis: Fast R-CNN vs. YOLO these charts show the percentage of localization and background errors in the top N detections for various categories (N = # objects in that category).

Combing Fast R-CNN and YOLO

YOLO makes itself fewer background errors compared to Fast R-CNN. Background detections in YOLO to eliminate from Fast R-CNN got a significant boost in performance and every bounding box that R-CNNpredicts. To check to see if YOLO predicts a similar box. If it does, that prediction a boost based on the probability predicted by YOLO and the overlap between the two boxes. The R-CNN model achieves a mAP of 66.9% on the PASCAL VOC test set. When both combinedYOLO and Fast R-CNN, mAP (mean average precision) increased by 6.8% to 75.7.0%, also tried combining the top Fast R-CNN models with several its versions. Finallyit ensembles produced small increases in mAP between 4 and 8%, see the table 1.Fnallally based on Gian in mAP by combining YOLO + Fast R-CNN is useful to get more accuracy than any other.

Dataset name	PASCAL VOC	VGG-M	CaffeNet
Fast R-CNN(mAP)	66.9	59.2	57.1
Fast R-CNN+YOLO(mAP)	75.7	64.2	62.2
Gain(mAP)	6.8	5.0	5.1

Table 1: Examining the Gain on combining Fast R-CNN with YOLO on various data sets.

On the VOC 2012 test set, YOLO scores 57.9% mAP. This is lower than the current state of the art, closer to the original R-CNN using VGG-16, see Table 2.YOLO System struggles with small objects compared to its closest competitors. On categories like bottle, sheep, and tv YOLO scores 8-10% lower than R CNN or Feature Edit. However, on other categories like cat and train YOLO achieves higher performance.However combined Fast R-CNN + YOLO model is one of the highest performing detection methods.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 8, August 2019

Algorithm with dataset	mAP	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	train	tv
MR-CNN	73.9	82.9	76.6	57.8	62.7	79.4	77.2	86.6	55.0	79.1	62.2	87.0	83.4	84.7	78.9	65.8	80.3	74.0	45.3
Hyper VGG	71.4	78.5	73.6	55.6	53.7	78.7	79.8	87.7	49.6	74.9	52.1	86.0	81.7	83.3	81.8	48.6	73.5	59.4	48.3
HyperNet-SP	71.3	78.3	73.3	55.5	53.6	78.6	79.6	87.5	49.5	74.9	52.1	85.6	83.2	81.6	48.4	73.2	79.7	65.6	65.6
Fast R CNN+YOLO	**70.7**	**78.5**	**73.5**	**73.5**	**55.8**	**43.4**	**79.1**	**73.1**	**89.4**	**49.4**	**57.0**	**87.5**	**80.9**	**81.0**	**74.1**	**41.8**	**71.5**	**82.1**	67.2
MR-CNN –S.CNN	70.7	79.6	71.5	55.3	57.7	76.0	73.9	84.6	50.5	73.4	61.7	85.5	79.9	81.5	76.4	41.0	69.0	77.1	72.2
Fast R-CNN	**68.4**	**79.8**	**70.8**	**52.3**	**49.8**	**77.5**	**75.9**	**88.5**	**45.6**	**77.1**	**55.3**	**86.9**	**81.7**	**80.9**	**79.6**	**40.1**	**69.6**	**81.0**	61.5
DEEP-ENS COCO	70.1	79.4	79.0	51.9	51.1	74.1	72.1	86.6	48.3	73.4	57.8	86.1	80.0	80.7	70.4	46.6	69.6	75.9	71.4
NOC	68.8	79.0	79.4	52.3	53.7	74.1	69.0	84.9	46.9	74.3	53.1	85.0	81.3	79.5	72.2	38.9	59.5	76.7	687
UMICH-FGS STRUCT	66.4	76.1	64.1	44.6	49.4	70.3	71.2	84.6	42.7	68.6	58.5	82.7	77.1	79.9	68.7	41.4	69.0	72.0	66.2
NUS-NIN-C200	63.8	73.8	61.9	43.7	43.0	70.3	67.6	80.7	41.9	69.7	51.7	78.2	75.2	76.9	65.1	38.6	68.3	68.7	63.3
BabyLearning	63.2	74.2	61.3	45.7	42.7	68.2	66.8	80.2	40.6	70.0	49.8	79.0	74.5	77.9	64.0	35.3	67.9	68.7	62.6
NUS-NIN	62.4	73.1	62.6	39.5	43.3	69.1	66.4	78.9	39.1	68.1	50.0	77.2	71.3	76.1	64.7	38.4	66.9	66.9	62.7
R-CNN VGG BB	62.4	72.7	61.9	41.2	41.9	65.9	66.4	84.6	38.5	67.2	46.7	82.0	74.8	76.0	65.2	35.6	65.4	65.4	60.3
R-CNN VGG	59.2	70.9	56.6	37.5	36.9	62.9	63.6	81.1	35.7	64.3	43.9	80.4	71.6	74.0	60.0	30.8	63.4	63.5	58.7
YOLO	**57.9**	**67.2**	**57.7**	**38.3**	**22.7**	**68.3**	**55.9**	**81.4**	**36.2**	**60.8**	**48.5**	**77.2**	**72.3**	**71.3**	**63.5**	**28.9**	**52.2**	**73.9**	**50.8**
Feature Edit	56.3	69.1	54.4	39.1	33.1	65.2	62.7	69.7	30.8	56.0	44.6	70.0	64.4	71.1	60.2	33.3	61.3	61.7	57.8
R-CNN BB	53.3	65.8	52.0	34.1	32.6	59.6	60.0	69.8	27.6	52.0	41.7	69.6	61.3	68.3	57.8	29.6	57.8	59.3	54.1
SDS	50.7	58.4	48.5	28.3	28.8	61.3	57.5	70.8	24.1	50.7	35.9	64.9	59.1	65.8	57.1	26.0	58.8	58.9	50.7
R-CNN	49.6	63.8	46.1	29.4	27.9	56.6	57.0	65.9	26.5	48.7	39.5	66.2	57.3	65.4	53.2	26.2	54.5	50.6	51.6

Table 2: On PASCAL VOC 2012 mean average precision and per-class average precision are shown for a variety of detection methods. YOLO is the only real-time detector. Fast R-CNN + YOLO is the fourth highest scoring method, with a 2.3% boost over Fast R-CNN.

V. CONCLUSION

Object detection is a key ability for most computer and robot vision systems. Although great progress has been observed in the last years, and some existing techniques are now part of many consumer electronics (e.g., face detection for auto-focus in smartphones) or have been integrated in assistant driving technologies, People still far from achieving human-level performance, in particular in terms of open-world learning. Being user of YOLO, it is a unified model for object detection and easy to build and can be trained directly on images. Unlike classifier-based methods, YOLO is well trained on a loss function that directly corresponds to detection performance and whole model is trained jointly. FastYOLO is the latest and fastest general-purpose object detector in the literature. YOLO has pushing the state-of-the-art in real-time object detection. YOLO canGeneralizes well to new domains making it ideal for application.In spite of rapid development and achieved progress of object detection, there are still many open issues for future work.Multi-task joint optimization and multi-modal information fusion. Due to the correlations between different tasks within and outside object detection, multi-task joint optimization has already been studied by many researchers. Scale adaption. Objects usually exist in different scales, which is more apparent in face detection and pedestrian detection. To increase the robustness to scale changes, it is demanded to train scale-invariant, multi-scale or scale adaptive detectors. Spatial correlations and contextual modelling. Spatial distribution plays an important role in object detection. So region proposal generation and grid regression are taken to obtain probable object locations. However, the correlations between

multiple proposals and object categories are ignored. Besides, the global structure information is abandoned by the position-sensitive score mAPs in R-FCN. Unsupervised and weakly supervised learning. It's very time consuming to manually draw large quantities of bounding boxes.

REFERENCES

[1] Sharma, K.U. and Thakur, N.V. (2017) 'A review and an approach for object detection in images', Int. J. Computational Vision and Robotics, Vol. 7, Nos. 1/2, pp.196–237.
[2] Sandeep Kumar, Aman Balyan and Manvi Chawla." Object Detection and Recognition in Images ", Volume 5, Issue 4 | ISSN: 2321-9939, 2011.
[3] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. CoRR, abs/1312.6229, 2013.
[4] R. B. Girshick. Fast R-CNN. CoRR, abs/1504.08083, 2015.
[5] J. Redmon and A. Angelova. Real-time grasp detection using convolutional neural networks. CoRR, abs/1412.3128, 2014.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 8, August 2019

[6] B.Hariharan, P. Arbelaez, R. Girshick, and J. Malik. Simultaneous detection and segmentation in computer Vision,ECCV 2014, Springer, 2014.
[7] Object Detection with Deep Learning: A Review, Zhong-Qiu Zhao, Member, IEEE, PengZheng, Shou-tao Xu, and Xindong Wu, Fellow, IEEE, arXiv, 1807.05511v2, Apr 2019.
[8] You Only Look Once: Unified, Real-Time Object Detection, Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, University of Washington, Allen Institute for AI, Facebook AI Research, DOI:10.1109/CVPR.2016.91
[9] Understanding of Object Detection Based on CNN Family and YOLO, Juan Du1, New Research and Development Centre of Hisense, Qingdao 266071, China, IOP Conf. Series: Journal of Physics: Conf. Series (2018) 012029 doi: 10.1088/1742-6596/1004/1/012029.
[10] Liming Wang1, Jianbo Shi2, Gang Song2, and I-fan Shen1," Object Detection Combining Recognition and Segmentation Fudan University, Shanghai, PRC,200433,2010.