# Advancements in Object Detection Algorithms: Evaluating, Enhancing and Envisioning.

*[1]Abhinav Sharma*
*SCOPE*
*Vellore Institute of Technology*
*Vellore, India*

*[1]Rishabh Kumar*
*SCOPE*
*Vellore Institute of Technology*
*Vellore, India*

*[2]Geraldine Bessie Amali D*
*SCOPE*
*Vellore Institute of Technology*
*Vellore, India*

**ABSTRACT:** The advancement in computational power and multicore processing along with better learning capabilities like the attention module has enabled several applications in the fields of Robotics, Natural Language Processing and Image Processing. Object detection is one key branch that finds its implementation in several real-world activities such as object tracking, surveillance systems, facial recognition systems, driverless cars, and many more. In this paper, we present a comprehensive study of all relevant object detection algorithms, starting from R-CNN (Region-based Convolutional Neural Networks) and its successors, YOLO (You Only Look Once) along with its different versions starting from YOLOv1 to YOLOv4. In this page, we proposed and experimented with an architecture that is union of YOLO as well as R-CNN model with the use of U-net model. And ultimately providing a framework for future research.

***Keywords*** *- Object Detection, YOLO, RCNN, Deep Learning, Computer Vision, PyTorch*

## [1] INTRODUCTION:

The human sight is one of the primary senses that enable us to perform various tasks, among which is the gift of recognizing and identifying objects. This task may seem simple to us but implementing it for machines is a significant challenge. Various models have been developed to enable machines to do the same.

For the scope of this paper, we are primarily focusing on CNN[11] (Convolutional Neural Networks) and YOLO [5](You Only Look Once) models. Both of these models have their merits as well as demerits.

The CNN family is used to solve problems related to classification and has a high level of accuracy (Average Precision, AP). However, this high AP comes at the expense of overall speed. On the other hand, the YOLO family is used to solve problems related to object identification and, therefore, requires higher processing speed. To achieve this, it sacrifices AP.

Most real-world computer vision problems require high overall speed with a decent AP value. Hence, we propose a model that combines the strengths of the two best models: the high accuracy of the CNN model and the high speed of the YOLO model. In doing so, we propose a framework for a model with decent speed and a high AP value.

## [2] LITERATURE SURVEY:

[1] **A.Bochkovskiy.et al.in 2020** paper titled "YOLOv4: Optimal Speed and Accuracy of Object Detection" presented an architecture which can be described as backbone which utilizes CNN to extract vital features from the images at various scales and also enhances as well as refines these extracted features and a head which makes object detection prediction. Additionally, they also introduce the idea of bag-of-freebies and bag-of-specials.

[2] **J.Redmon.et al.in 2018** paper titled "YOLOv3: An Incremental Improvement" proposed an improved version of YOLO, here the architecture is larger and can better detect small objects as compared to its previous version because of its support for multi scale prediction.

[3] **K.He.et al.in 2018** paper titled "Mask R-CNN" introduced an instance segmentation model, which combes the Faster R-CNN object detection framework with precise pixel-level object masks. This model significantly improved then state-of-the-art computer vision model and enabled accurate object recognition and segmentation in a single model.

[4] **J.Redmon.et al.in 2016** paper titled "YOLO9000: Better, Faster, Stronger" introduced a method for training joint classification and detection of the object. It uses labelled data from COCO dataset to learn bounding box coordinates and classification data from ImageNet to increase the number of detection categories. The result is this YOLO model which can detect more than 9000 categories.

[5] **J.Redmon.et al.in 2016** paper titled "You Only Look Once: Unified, Real-Time Object Detection" introduce the very first YOLO architecture which uses unified object detection steps by detecting all the bounded boxes simultaneously. It treated detection task as a single regression problem using 24 convolutional layers followed by 2 fully connected layers it extracts features and identifies objects, and Thus, achieving state of the art speed at the time.

[6] **S. Ren.et al.in 2016** paper titled "Faster R-CNN" introduced a model for object detection with more emphasis speed as compared to previous R-CNN models. Faster R-CNN architecture introduced a two staged approach, utilizing region proposal network (RPN) to improve object localization and overall object detection accuracy.

[7] **R. Girshick 2015** paper titled "Fast R-CNN" achieved a fast R-CNN model by using a single-staged approach it integrates object classification and bounding box regression thus simplifying training process.

[8] **O.Ronneberger.et al.in 2015** paper titled "U-Net: Convolutional Networks for Biomedical Image Segmentation" proposed a semantic segmentation model, named U-Net which was designed for applications in medical image analysis. It features a contracting and expansive path, enabling accurate pixel-wise segmentation.

[9] **R. Girshick.et al.in 2014** paper titled "Rich Feature hierarchies for accurate object detection and semantic segmentation" is the introducing paper of R-CNN. The authors proposed use of region-based convolutional neural networks for object localization and classification using two staged approach for combining region proposal generation with CNNs, leading to improved object detection performance.

## [3] PROBLEM IDENTIFICATION:

Lack of an algorithm that can cater to all the below mentioned problems-

1.An algorithm with a significant better AP value for detecting small objects as compared to large objects.

2.Slow inference generation and a high data as well as computational power is required for training due to their large model size.

3.An algorithm with a proper balance between speed and accuracy.

## [4] METHODOLOGY:

### [4.1]Proposed System:

In this paper, we propose a modified version of the U-Net model [10] which is used for image segmentation. We have modified it for our use case with some slight modification. Every task we are training for, we introduce some removable layers at the end of the conventional U-Net layer and after which training the entire network end to end. It will be further discussed on the later part of the paper.

We choose U-Net model because of its efficiency in domain of image segmentation. While working with region proposed model it is not guaranteed to generate the desired result in the output, to tackle this we trained the model in a way to generate important layers in the feature map towards the end. In order to avoid conflicting nomenclature, we call them Pseudo Region Proposals (PSR). We intend to generate Pseudo region proposals like the region proposals of Faster RCNN[6] and Mask RCNN[3] due to its object detection capability. Secondly, the U-Net model also has connections similar to the skip connections in ResNet[9] which allow the knowledge of the initial layers to be passed to the deeper layers directly. Here we are replicating this to specific information at a particular location that might not be the immediate input to the layer in consideration. Thirdly, the U-Net model only has convolution layers thus we need to train few parameters compared to a model like YOLO1 which trained on full connected layers. Lastly, U-Net is sequential thus one forward pass is enough for generating the region proposals and object detection.

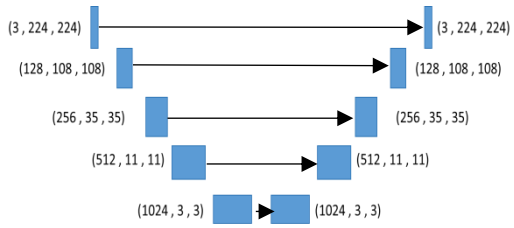


***Figure 1.*** *Proposed Architecture representing different convolutional layers.*

### [4.2]Expected outcomes

By generating Pseudo region proposals, we expect final added layers(that are task dependent) of the model will perform better if interesting regions of the image (PSR) are given directly to them. The skip connections will enable detecting smaller objects efficiently compared to YOLO model. The information of the initial layer is directly sent to the deeper layers, the information which could be lost due to continuous convolution and maxpool operations are regained in the deep layers directly. In case of smaller objects their presence in the feature maps in the deeper layer become smaller and smaller and eventually disappear. We also expect the model to be very fast during inference, as our model, even though generating PSR, is fully sequential unlike the RCNN family. This means we will be able to get the inference results for an image or a batch of images in a single forward pass of the model. Along with that having only convolution layers also means that we have fewer parameters to learn. This again helps in faster inference but also means that lesser data to train the model and convergence on the datasets will be faster.

### [4.3]Training paradigm:

We have devised a different training method for our modified U-Net model. We call it the hierarchical training. Hierarchy of tasks are based on the relative difficulty. We train the model first on more complex tasks like image segmentation and then on simpler task like image classification. With a wider use case, for the scope of this paper we are restricting it to two tasks. We train the trained model to do object which will be final task. The idea is that the first two tasks will act like unsupervised pre-training which is very famous right now in the deep learning domain. However, the tasks here are just to give better insights to the model about domain of the images and then we assume that this generalization will help the model to perform easier task like object detection easily.

### [4.4]Training procedure:

While training on a task is different with each task having a dedicated set of final layers, we pretty much follow the same procedure for training for all the tasks. Input: A 224*224 RGB image which is logically divided into S*S sections. We assume the value of S to be 7, but it can be changed. Each of the S*S cell will be responsible to predict one object and that cell will be responsible to predict the object which has the centre of the object in the image. The image is then passed to the model and PSR maps are generated at the end of the U-Net model. The PSR maps are of the same size as the image i.e. 224*224 pixels. The PSR maps are then fed into the task specific final layers and the output shape is changed according to the task. For the segmentation task, we don't use any final layers and the PSRs themselves help find the region proposals. In the classification task, the new set of final layers end to predicting single feature map of size (n,1) which will be passed through the SoftMax activation function to generate the final output of the model. In the object detection part, we generate a S*S*5*5 feature map for each image. The 5 * 5 feature map will have 20 values for the 20 classes of the PASCAL VOC dataset. The next five values will be probability of the object and the four bounding box coordinates. The first 20 values will be passed through the SoftMax function and the loss will be cross entropy loss. The probability and bounding box values will be used to compare against the ground truth values and the loss for them will be Root Mean Squared Error(RMSE). Finally, the two loss values will be added, and the model will be trained using backpropagation.

## [5] EXPERIMENT AND RESULT:

We experimented with the implementation of the architecture in PyTorch. As stated earlier, we used the PASCAL VOC 2007 dataset for out experiments. The dataset consists of 20 classes of common objects. We train the model on the dataset for 60 epochs. The loss function that we used was a combination of three losses, i.e. classification loss, object confidence loss and the localization loss. The loss function that we used in the paper is same as the YOLO v1. We chose this loss function

as it gives stable gradients which is helpful during the gradient descent. We weigh all the losses (classification loss, object confidence loss, localization loss) equally. Classification loss uses Cross Entropy Loss and localization loss, confidence loss uses Root Mean Square Error.

$$\frac{\partial L}{\partial x_i} = \sum_r \sum_j [i = i * (r, j)] \frac{\partial L}{\partial y_{rj}}$$

**Equation 1: Loss Function for Classification Loss: Cross Entropy Loss.**

For each mini-batch RoI r and also for each pooling output unit of $y_{rj}$ , the partial derivative $\delta L = \delta y_{rj}$ is accumulated if and only if $i^{th}$ is the argmax selected for an $y_{rj}$ by max pooling.

$$\sum_{i=0}^{S^2} \sum_{j=0}^{B} [(x_i - \tilde{x_i})^2 + (y_i - \tilde{y_i})^2] +$$
$$\sum_{i=0}^{S^2} \sum_{j=0}^{B} [(\sqrt{w_i} - \sqrt{\tilde{w_i}})^2 + (\sqrt{h_i} - \sqrt{\tilde{h_i}})] +$$
$$\sum_{i=0}^{S^2} \sum_{j=0}^{B} (C_i - \tilde{C_i})^2 + \sum_{i=0}^{S^2} (p_i(c) - \tilde{p_i}(c))^2$$

***Equation 2: Loss Function of YOLOv1: Root Mean Square Error.***

where $1_i^{object}$ is being denoted if object appears in cell $i^{th}$ and $1_{ij}^{object}$ denotes that the $j^{th}$ bounding box predictor in cell $i^{th}$ is responsible for that prediction.

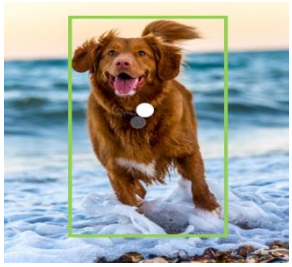

***Figure 2:*** *Bounding box Detected*

Our model trains with loss decreasing after each epoch. The loss first decreases slowly and then decreases rather quickly. The same can be verified from the diagram of the loss per epoch (Figure 1).

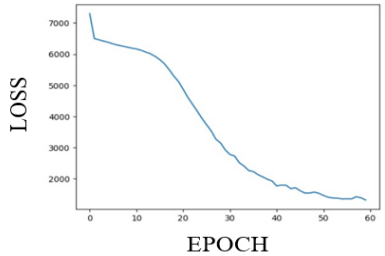


***Figure 3.*** *Loss vs Epoch graph.*

The graph above shows the loss plotted against the training epochs. The loss first decreases during the initial phase as the model was initialised randomly. Then we see the loss decreasing quite consistently. We conjecture that the loss will decrease further giving us better generalization on the dataset. We also would like to state that using another dataset which is significantly bigger and recent will also give us similar training results in terms of the loss values.

## [6] CONCLUSION:

In this paper we have summarised various image recognition and classification models ranging from YOLO family to R-CNN family. And have proposed an architecture for a unified model which has features of YOLO as well as R-CNN to achieve a better balance between the speed vs accuracy parameters.

## [7] REFERENCES:


[1] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," in Proceedings of the European Conference on Computer Vision (ECCV), 2020, pp. 34-51.

[2] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 8793-8802.

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980-2988.

[4] J. Redmon and S. Divvala, "YOLO9000: Better, Faster, Stronger," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 7263-7271.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788.

[6] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2015, pp. 91-99.

[7] R. Girshick, "Fast R-CNN," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440-1448.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," in Proceedings of the European Conference on Computer Vision (ECCV), 2014, pp. 346-361.

[9] 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 (CVPR), 2016, pp. 770-778.

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2015, pp. 234-241.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580-587.

[12] N. Darapaneni, A. Tiwari, A. Balaraman, M. Ravikumar, S. Das, and G. Pratap, "Computer Vision Application in Automobile Error Detection," in 2022 Interdisciplinary Research in Technology and Management (IRTM), 2022, pp. 1–7.