# Real-Time Object Detection using Deep Learning and Open CV

**P.Devaki, S.Shivavarsha, G.Bala Kowsalya, M.Manjupavithraa, E.A. Vima**

***Abstract— The object detection is used in almost every real-world application such as autonomous traversal, visual system, face detection and even more. This paper aims at applying object detection technique to assist visually impaired people. It helps visually impaired people to know about the objects around them to enable them to walk free. A prototype has been implemented on a Raspberry PI 3 using OpenCV libraries, and satisfactory performance is achieved. In this paper, detailed review has been carried out on object detection using region – conventionaal neural network (RCNN) based learning systems for a real-world application. This paper explores the various process of detecting objects using various object detections methods and walks through detection including a deep neural network for SSD implemented using Caffe model.***

***Keywords— Object Detection, RCNN, SSD, Caffe model, Open CV libraries, Neural Networks.***

## I. INTRODUCTION

Advanced concepts like neural networks and deep learning are gaining its ground in the area of computer vision. The solution provided using these techniques can be highly adaptive and reliable in real time. Traditionally, visually challenged people use a white cane for their navigation in outdoors which provides them limited utility. A smart system is required to ensure safety and to make the individual highly aware of his/her surrounding to improve assistance.

Before implementing object detection and classifying the object based on its category, we need to understand the difference between object detection and image classification. Image classification is the process of classifying the image to a category based on the recognized features and patterns, whereas object detection is the process of obtaining the bounding box of coordinates exactly where a particular object is present in the image. We can detect more than one object of a different class in an image. In short, object detection can not only tell us **what** is in an image but also **where** the object is as well. There are several ways to detect objects in an image.

**Revised Manuscript Received on September 14, 2019.**

**Dr.P.Devaki,** Professor, Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore, Tamil Nadu, India.

**S.Shivavarsha,** UG Final Year, Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore, Tamil Nadu, India.

**G.Bala Kowsalya,** UG Final Year, Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore, Tamil Nadu, India.

**M.Manjupavithraa,** UG Final Year, Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore, Tamil Nadu, India.

**E.A. Vima,** Associate Professor, Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore , Tamil Nadu, India.

## II. APPROACHES TO OBJECT DETECTION

Usually, we categorize object detection as a classification problem involving the classification of an object based on categories. There are several approaches to detect the objects and it can be broadly classified as a classification problem and regression problem.

## III. CLASSIFICATION-BASED APPROACH

*a) Object detection using HOG features*

In this approach, we input hog features calculated from each window obtained by running a sliding window to SVM (Support Vector Machine) algorithm to create classifiers. HOG (Histogram of Oriented Gradients) is a feature descriptor which decomposes an image into small-sized square cells and computes a histogram of oriented gradients in each cell, normalizes the result using a block-wise pattern, and return a descriptor for each cell[1]. Though this technique was introduced decades ago, still it is one such algorithm used for object detection because of its computational inexpensiveness.

*Problem with HOG features*

Since we are working on the classification based problem, the accuracy of the detection is the foremost requirement. It has disadvantages such as,

1. Very sensitive to image rotation.
2. Final feature descriptor grows larger and it takes too much time to extract objects.

**Fig 1. Image is decomposed into small-sized square cells and computes a histogram of oriented gradients in each cell**

*b) Regional based Convolutional Neural Networks*

Though HOG based detections are inexpensiveness and simpler, the accuracy obtained by them is not that effective. Region Conventional Neural Network (CNN) gives rise to a more accurate result than HOG based implementation. R-CNN technique is slower and very expensive, using the sliding window concept. Because the sliding window method can make the algorithm run slower. It is an exhaustive search for objects in an image/each frame of the video stream. Sliding window method needs to capture all the possible locations for objects in an image and also at different scales too, this is really complex.

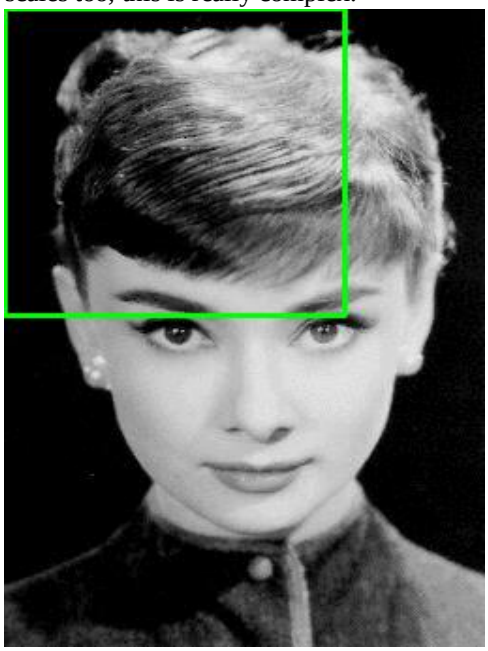

**Fig 2. Sliding window captures all the possible locations in an image**

Therefore, we use an alternative method which is Selective Search algorithm. It uses properties such as color similarity, texture similarity, size similarity, and shape compatibility, to group similar pixels to a segment.

*Selective Search-Region Proposal Algorithm*

Instead of looking for every possible patch in the image as in sliding window, this method groups the pixels of similar properties into segments. This can highly reduce the overhead of recognizing false positives of an image. An important advantage of the Region Proposal Algorithm is that it has **high recall.** Selective search is the region based algorithm, it uses a bottom-up approach to detect an object in the image. It uses a concept of segmenting similar pixels of the image at first and in each iteration, the segments will be grouped together based on the similarity. This results in a larger segment from the smaller segments with which the algorithm starts [3].

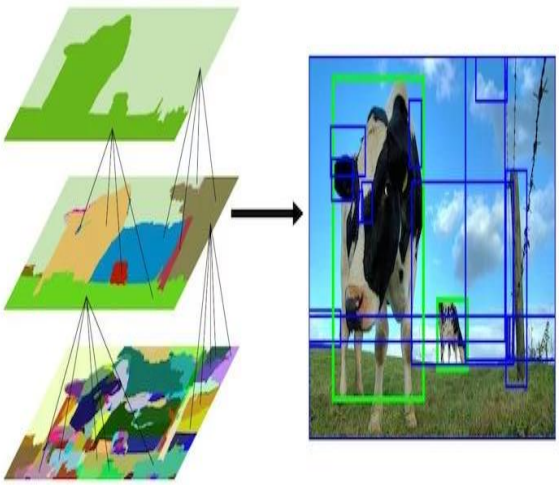

**Fig 3. Selective Search method iteratively groups pixels of similar properties into segments and detects the object once the segment is large enough**

*Problems with RCNN*

Though RCNN has several improvements in its way of detecting objects, this approach still has several limitations such as it is very slow in recognizing objects. It has to extract more than 2,000 regions even while using selective search approach and it has complex manipulation processes such as feature extraction, identifying objects using SVM model and a regression model for bounding boxes creation[4]. These highly complex underlying computations make the algorithm to take 30-50 seconds to detect objects in an image. This can not be the better choice for detecting objects in real time.

*Fast R-CNN*

Fast RCNN overcomes the limitations of RCNN. Instead of scanning the image for 2,000 times, Fast RCNN runs the neural network only once and creates the convolutional feature map using that[5]. In addition to that, it uses a single model to extract the features, classify the object and return boundary box instead of using three different models in each process as in RCNN. This can improve the performance of the algorithm greatly.

Steps involved in Fast RCNN:

1. An image is given as an input to the convolutional neural network and it generates the feature map which gives us region of interest[6].
2. All the obtained RoI's will be given as an input to the RoI pooling layer to reshape it into a fixed sized so that it can be fed further into fully- connected layer.
3. A softmax layer is used to classify the proposed regions to a predicted class obtained from the RoI feature vectors

*Problem with Fast RCNN*

When we see the underlying computations of Fast RCNN, it also uses Selective search method for extracting the region of interest. It takes around 2 seconds to detect objects in an image. The system will run into the bottleneck situation while using extremely large dataset. This can reduce the

effectiveness of the algorithm and make it not the best choice for real-time object detection too.

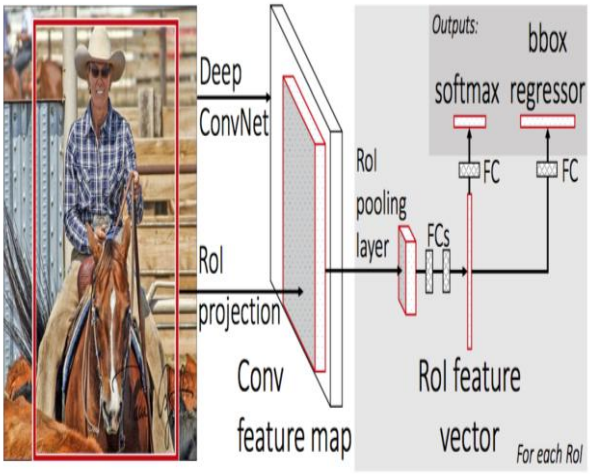


**Fig 4. Fast RCNN scans the entire image only once and creates a feature map and generates the regions of interest**

*Faster R-CNN*

The major advancement from Fast CNN to its modified version is that it uses Region Proposal Network(RPN) instead of using Selective Search method. Faster-RCNN is 10 times faster than Fast-RCNN[8].

Steps involved in Faster RCNN:

1. An image is fed as an input to the convolutional neural network and returns a feature map.
2. Objectness score of the proposals is obtained by applying region proposal network.
3. These proposals are reshaped into fixed size using RoI pooling layer and passed to a fully-connected layer, which has softmax layer and linear regression layer to classify the objects.

*Problem with Faster RCNN*

Faster RCNN runs at 7 frames per second.

*Why Faster RCNN still a better option?*

1. Faster RCNN solves the bottleneck of running Selective Search method on each image. It replaces it with a very small network Region Proposal Network (RPN). 2. It generates anchor boxes to handle the variations in aspect ratio and scale of objects.

At each location in an image, it uses 3 kinds of anchor boxes with various scales such as 128 x 128, 256 x 256, 512 x 512 and 3 the other 3 with different aspect ratios 1:1, 2:1, and 1:2, totally 9 anchor boxes of each combination.

3. It takes only ~0.2 seconds to detect objects in an image.

These merits make it a better option for object detection.

## IV. REGRESSION-BASED APPROACH

*Single Shot Detectors for object detection*

Single Shot Detectors are equivalent to the CNN in Faster RCNN but it runs a convolutional neural network only once and provides better speed and accuracy. Similar to Faster RCNN, SSD also uses anchor boxes to handle the variations in aspect ratio and scale of objects. It computes both the location and classification scores using the classification filters. After extracting the feature maps, SSD applies 3 × 3 convolution filters for each cell to make predictions. Each filter outputs 25 channels: 21 scores for each class plus one boundary box.

SSD uses non-maximum suppression to remove the duplicate predictions around the object. It sorts the objectness score the retains only the top 200 predictions per image and ignores the rest.

*Problems with SSD*

1. The performance of SSD is poorer for the images with low resolution. The smaller objects can only be detected using higher resolution layers.

## V. NETWORK ARCHITECTURE & RESULTS

While building an object detection network, we tend to use the already existing network architectures such as VGG , ResNet. But the disadvantage with those networks is that it is larger in size almost in the order of 200-500 MB. This makes it not suitable for implementing it in the computationally less powered systems.

*MobileNet*

MobileNet is a network architecture developed by Google suitable for mobile and less power embedded vision applications.

Salient features:

1. Its peculiarity lies in using depthwise separable convolution.
2. It uses only a less number of parameters it reduces overfitting.
3. It computes fewer multiplications and additions
4. It introduces two parameters **Width Multiplier α** and **Resolution Multiplier ρ** to tune easily.

*Proposed Solution*

From the above discussion on various approaches to detect an object in real-time and framework used for object detection, we can conclude that SSD and MobileNet combination can provide a better solution for detecting objects in real-time. This solution is faster and accurate as needed.

For the prototyped version, we have used this approach to detect objects and run in a Raspberry Pi 3 to make it as a standalone system to assist visually impaired people. The output of the object detection will be communicated to them as auditory information and haptic feedback.

## VI. CONCLUSION

In this paper, we discussed various approaches for object detection with its merits and demerits. So many research and improvements are taking place in the field of object detection and recognition too. This can be used in real time applications such as driverless cars.

## REFERENCES


1. Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, Avidan, S., "Fast Human Detection Using a Cascade of Histograms of Oriented Gradients," Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference, pp.1491-1498, 2006
2. Yoshua Bengio. Learning deep architectures for ai. Foundations and Trends R in Machine Learning, 2(1):1–127, 2009
3. J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. IJCV, 2013
4. R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. TPAMI, 2015
5. R. Girshick, "Fast R-CNN," in IEEE International Conference on Computer Vision (ICCV), 2015
6. S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun, "Object detection networks on convolutional feature maps," arXiv:1504.06066, 2015
7. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In Proc. of the ACM International Conf. on Multimedia, 2014
8. Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", arXiv:1506.01497, 2016
9. [9] Andrew G. Howard Menglong Zhu Bo Chen Dmitry Kalenichenko Weijun Wang Tobias Weyand Marco Andreetto Hartwig Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", arXiv:1704.04861v1, 2017
10. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. FeiFei. ImageNet: A large-scale hierarchical image database. In CVPR, 2009
11. A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in Neural Information Processing Systems (NIPS), 2012
12. M. Wang, B. Liu, and H. Foroosh. Factorized convolutional neural networks. arXiv preprint arXiv:1608.04337, 2016.