

# Object Detection in Railway Track using Deep Learning Techniques



R. S. Rampriya, R. Suganya, Sabarinathan, Aarthi Ganesan, P. Prathiksha, and B. Rakini

**Abstract** Nowadays, deep neural networks are one of the ongoing trends which are having their uses in various kinds of fields. One of the most important applications of neural networks is the object detection framework. Object detection in railways field is the novel one, which includes the detection of obstacles on the railway tracks. But some researchers deployed this model using you only look once (YOLO) and single shot detector (SSD). We found these models to produce a lesser accuracy when compared to that of faster R-CNN. The previously mentioned models consume some time for training. To overcome the existing drawbacks, the upgraded model is being deployed by using faster R-CNN, which comprises two modules, namely regional proposal network (RPN) and fast R-CNN. It helps by detecting the obstacles such as branches, boulders, iron rods, animals, vehicles, and people. So, with the help of both the models (faster R-CNN and YOLO), we can detect the obstacles present on the track. Finally, with the quantitative and qualitative comparisons made on these two models, we chose the best fit model for this purpose. Hence, this provides the novel idea to prevent railway accidents as much as possible.

**Keywords** Object detection · Deep neural network · Faster R-CNN · YOLO

## 1 Introduction

Deep learning is the successor of machine learning, which is an AI function that mimics the workings of the human brain in processing data that is used in various fields like object detection, speech recognition, language translation, and decision. The main idea behind this technology is the network of structures. It builds more

---

R. S. Rampriya (✉) · A. Ganesan · P. Prathiksha · B. Rakini  
Anna University (MIT Campus), Chennai, India

R. Suganya  
Thiagarajar College of Engineering, Madurai, India  
e-mail: [rsuganya@tce.edu](mailto:rsuganya@tce.edu)

Sabarinathan  
Cougher Inc, Tokyo, Japan

networks to train the model with unstructured or unlabeled data. They also called these deep neural networks. This kind of framework requires an enormous amount of data to fetch greater accuracy. Hence, inputs were given as huge datasets to the model. One of the most important features of the deep neural networks is to process large numbers of features so that it becomes powerful when dealing with unstructured data. Some of the frequently used algorithms in deep learning are convolutional neural network, long short-term memory networks, stacked auto-encoders, etc. One such application of this framework is the object detection technique.

Object detection plays a significant role in many real-life instances. Its role in railways is quite important for overcoming the challenges that cause many railway accidents. Accidents because of the obstacles on the tracks have become more common nowadays, particularly in rural areas. It also has a tremendous impact on wildlife as most of the accidents happen because of animals crossing the track. All these should be monitored and necessary actions should be taken. For this, we implement object detection on railways that detects the obstacles on tracks such as branches, animals, and people.

The primary purpose of this paper is to find the best fit object detection model to detect the obstacles present on the railway tracks. Here, we consider the two models YOLO and faster R-CNN. With YOLO, it uses a single neural network to process the full image and divides the image into regions and predicts bounding boxes and probabilities of each region whereas faster R-CNN uses region proposal networks to predict the region in which the object is present. This helps to minimize the number of railway accidents or collisions caused by the trains because of the lack of signals.

## 2 Related Works

They gave a brief introduction to deep learning and CNN in [1]. They have discussed various kinds of object detection, namely generic object detection, salient object detection, face detection, pedestrian detection. But it mainly focuses on typical generic object detection architectures. CNN architectures comprise feature maps and transformations—filtering and pooling. By comparing all these models on various datasets, the efficiency of the models was studied and found the best model for the object detection purpose. This paper has explained the best model with the pedestrian detection application. To do this, the complete process of pedestrian detection starting from dataset creation to computing the evaluation metrics for the results obtained.

In the work described in [2], they compared models SSD and faster R-CNN for object detection. A region proposal network that shares full image convolutional features with the detection network has been introduced in [3]. It focuses on RPN, which is the most important technique of faster R-CNN. This RPN tells the faster R-CNN, where to look at the image to detect the correct one. They did experiments on the MS CoCo and Pascal VOC dataset. Here, they have used 80,000 samples

for training and 40,000 for validation. In [4], three major improvements on faster R-CNN algorithms are made, namely feature pyramid structure, region of interest align, usage of soft NMS algorithm (non-maximum suppression)—it sorts all detection boxes based on their detection score, and the one with maximum score is selected while the others are suppressed.

Different neural networks have been shown [5] to achieve classification using the faster R-CNN. For object detection, it exploded the speed of detection as it integrates the process of feature extraction, proposal extraction, and rectification. Experimental results show that its effectiveness comes from the convolutional layers and RPN modules. YOLOv2 model and YOLO9000 used for real-time detection systems for detecting and classifying objects in video records have been used here [6]. They have used GPU to increase speed and processes at 40 frames per second. The computation, processing speed, and efficiency in identifying the objects in the video record have been improved.

The first dedicated dataset for aerial survey of railways has been created by collecting images from Google and frames from YouTube videos, used as dataset for training the CNN to detect the obstacles. This paper [7] uses two versions of the faster R-CNN, i.e., faster R-CNN inception V2 model and faster R-CNN ResNet inception V2 model. Among these models, they took the most efficient one for the application.

### 3 Proposed Work

There are many sectors in our society like agriculture, transport, pharmaceutical, and many more. Agriculture introduced some techniques like fertility detection and many methods for the welfare of this sector. With pharmaceuticals, market fix modeling is the method introduced using machine learning models to promote the medicines in the markets. Therefore, machine learning and deep learning have invented new technologies for the benefit of humankind and the field. They transported one topic that is left idle because this is the field where these technologies are missing. Among transports, one such area that has to be noted is the railways. This is the sector that is lagging in terms of its technologies. They require technology for monitoring various activities, like monitoring the driver, signaling the driver, proper railway crossings, and many more. One such activity that is very essential is the proper vigilance of railway tracks, because this activity may cost people's lives because of a lack of monitoring of railway tracks. Hence, this project helps the driver to be aware of the objects in the tracks in advance by designing a deep learning model for the early detection of objects on the tracks.

## 4 Implementation

Implementing the project is carried out on Google Collab with GPU, where both the models are trained and tested on the custom dataset. The dataset creation involves collecting images and annotating them using the “labeling” tool. The proposed work comprises three modules such as data preprocessing, prediction and classification, and comparative study of faster R-CNN and YOLO. A brief explanation of each module is as follows:

### 4.1 Dataset Description

The dataset that is used is the customized one, where the images were collected from the Web, based on the classes chosen for detection. The custom dataset that is created comprises various classes of images categorized under the labels, namely animal, branch, boulder, iron rod, vehicle, and person. 1050 samples contribute to the training and testing sets for the object detection process. Out of which, the training set comprises 880 samples, and the testing set contains 170 samples. Since there are two models involved in the detection process, the faster R-CNN uses Pascal VOC which gives the annotation details in the “.xml” file, whereas the YOLO uses its own YOLO format and saves its annotation details in the “.txt” file.

### 4.2 Detection Using Faster R-CNN

It based the implementation on TensorFlow, which is an end-to-end open-source platform for machine learning. Install all the packages like pillow, lxml, Cython, OpenCV-Python, Matplotlib, pandas, etc., using the pip install command. Here, the “pandas” and “OpenCV-Python” packages are used in Python scripts to generate TFRecords. This TFRecord will contain the image info as NumPy arrays and the labels as a string. The model that we used for training is the faster R-CNN inception v2 coco model, downloaded from the TensorFlow detection model zoo [8] repository. In the config file of the downloaded model, changes should be made based on the created train and test datasets, label map, and record files. The hyper-parameters of the model, such as weight values and learning rate, are set as default. It should train the model for at least 60,000 steps and until the loss becomes less than 0.05 (Fig. 4). Once it is done, it will save all the trained models in the respective folder. Now, the last step is to generate the frozen inference graph (.pb file) with which the detection is to be made (Fig. 1). Finally, we can test our model for detecting objects in the input image, which will be annotated with its class name and detection score.

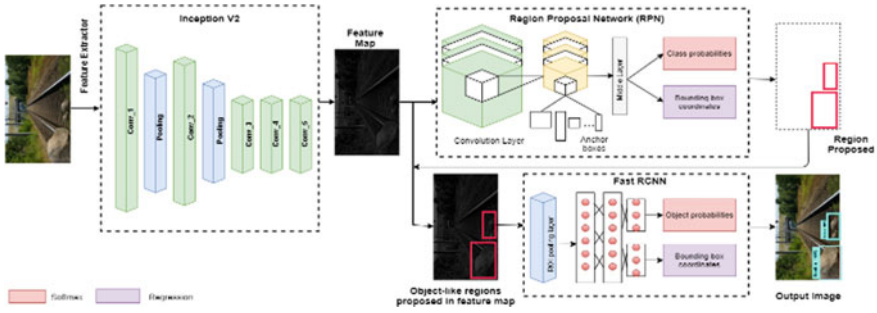


Fig. 1 Architecture diagram of faster R-CNN

### 4.3 Detection Using YOLO

For YOLO, start by cloning the darknet folder [9]. The next step is to create a data file that contains information about the location of the dataset files and the details of the bounding box. Then, split the dataset into train and test text files (80% for training and 20% for testing) which contain the filename of the images. darknet 53. conv.74 is a pre-trained model which should be further trained on the custom dataset. Changes should be made in the config file of YOLO\_v3 concerning the training and testing parameters such as batch, subdivisions, and learning rate. It contains 3 YOLO layers, where the number of classes should be changed and in the preceding convolutional layer, change the value of the number of filters used according to the number of classes (Fig. 2). Start the training with the help of the created data file for the custom dataset and by using the darknet function. Once the training is completed, the epoch VS loss graph is plotted (Fig. 4). We then used the trained model for testing. For testing, images are given as inputs and the output image contains the objects detected with the bounding boxes, the classes it belongs to the detection score, and the time taken for the prediction.

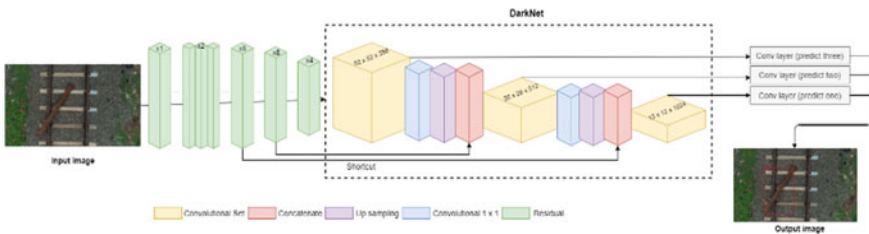


Fig. 2 Architecture diagram of YOLOv3

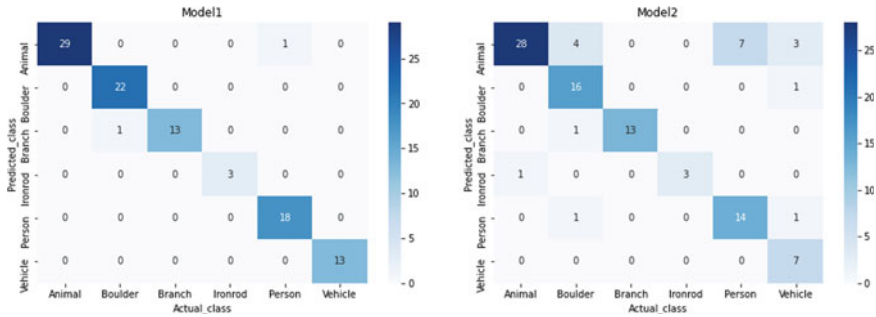


Fig. 3 Confusion matrix of faster R-CNN and YOLO

### 4.4 Multi-class Classification Testing

Once, the training of faster R-CNN and YOLO is successfully completed, then comes the testing phase, where the evaluation is done by giving an input image to the trained model. Therefore, the results obtained during the testing of both the models should be noted down on separate excel sheets.

To perform the computation for comparison of performance metrics, necessary packages such as pandas and NumPy should be imported. Then, “confusion matrix” is built, for both the models based on the excels created, using “crosstab ()” taken from panda library (Fig. 3). The confusion matrix consists of axis-like, where the horizontal one is the “actual class” and the vertical one corresponds to the “predicted class.” So, this confusion matrix helps in finding out various parameters such as “true positive” (TP), “false positive” (FP), “false negative” (FN), and “true negative” (TN). The TP is nothing but the diagonal elements of the confusion matrix, the FP is found by considering all the columns except the values at diagonal, FN is identified by considering all the rows except the ones which have the same class label as the actual class, TN is obtained by summing up all the elements of the confusion matrix and subtracting it from all the above parameters. Once, the above parameters are calculated for each class, the precision, recall, and F1 score are calculated. After the values are computed, the classification report is generated for both the models using the function classification\_report(). From this report, the model with the best accuracy will be chosen for the object detection purpose.

## 5 Result and Analysis

From the comparative study made on faster R-CNN and YOLO, it is evident that faster R-CNN performs better than YOLO in terms of accuracy (faster R-CNN = 98%, YOLO = 81%) and other performance metrics.

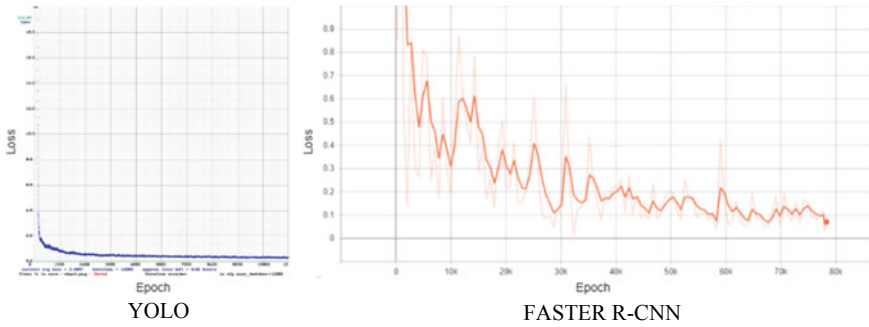


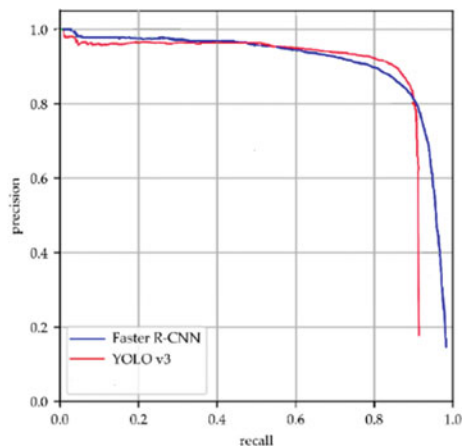
Fig. 4 Epoch versus loss graph

During an epoch, the loss function is calculated across every data item and give the quantitative loss measure at the given epoch. But plotting curve across iterations gives the loss on a subset of the entire dataset. So, epoch versus loss graph is plotted for both YOLO and faster R-CNN (Fig. 4).

Precision versus recall graph is plotted for both the model (Fig. 5). The mean average precision (mAP) was calculated for all networks with a pre-defined IoU threshold. Both models have a mAP above 80% on the integrated test set, illustrating that these methods were able to achieve favorable result. Faster R-CNN demonstrated the highest mAP (90.4%) than YOLO. A preliminary analysis suggested that the network inception V2, it performs with fast inference on low computing power, consuming a small amount of memory, playing a fundamental role in the detection accuracy improvement of faster R-CNN.

As output of both the models, we will be obtaining an output image with objects detected along with its class name and the detection score (Fig. 6).

Fig. 5 Precision versus recall graph



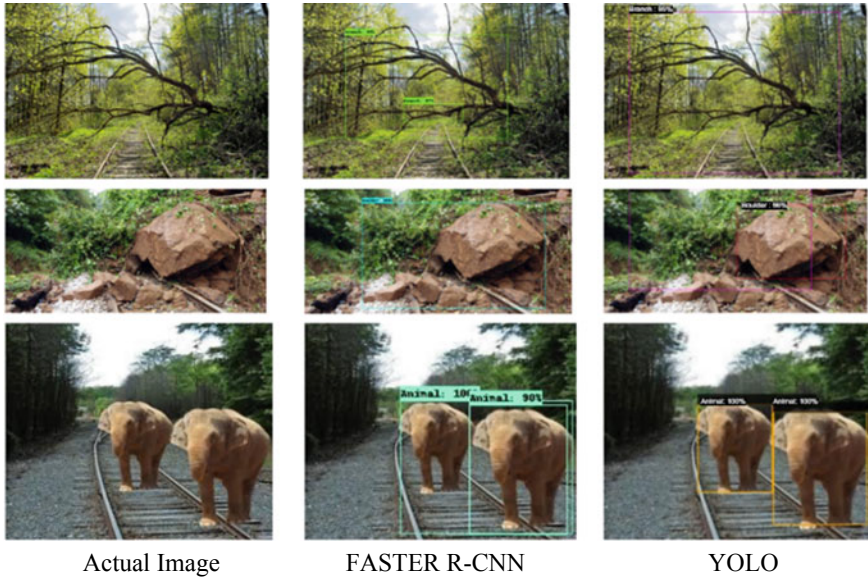


Fig. 6 Prediction of obstacle by faster R-CNN and YOLO

## 6 Conclusion and Future Work

The project was divided into three phases: The first phase is all about faster R-CNN, where the Pascal VOC dataset is created, augmented, and annotated for the model training. Once the data processing is done, the faster R-CNN model is trained and tested with an input image. The second phase is the implementation of YOLO, where the data processing is similar to faster R-CNN and the model is trained and tested on the dataset. Then, the final phase of the project is the comparison of both the models with the help of test results. Here, different performance metrics are calculated for each model and found that the model, “faster R-CNN” is the best one for the object detection on railway tracks.

As future work, we will try to detect objects (obstacles on tracks) in live videos and once the object is detected, an alerting mechanism like alarms can be added to alert the loco pilot. Further, the dataset can be improvised by capturing some real-time images and collecting frames from live videos.



## References

1. Zhao Z-Q, Zheng P, Xu ST, Wu X (Nov 2019) Object detection with deep learning: a review. IEEE transactions on neural networks and learning systems 30(11). <https://doi.org/10.1109/TNNLS.2018.2876865>
2. Galvez, Bandala A, Dadios P, Vicerra P Object detection using convolutional neural networks. TENCON 2018–2018 IEEE region 10 conference. <https://doi.org/10.1109/TENCON.2018.8650517>
3. Ren S, He K, Sun J (1 June 2017) Faster R-CNN: towards real-time object detection with region proposal network. IEEE Trans Pattern Anal Mach Intell 39:6. <https://doi.org/10.1109/TPAMI.2016.2577031>
4. Kafetzis D, Fourfouris I, Argyropoulos S, Koutsopoulos I UAV-assisted aerial survey of railways using deep learning. 2020 international conference on unmanned aircraft systems (ICUAS). <https://doi.org/10.1109/ICUAS48674.2020.9213928>
5. Liu B, Zhao W, Sun Q Study of object detection based on faster R-CNN. 2017 Chinese automation congress (CAC). <https://doi.org/10.1109/CAC.2017.8243900>
6. Jana AP, Biswas A, Mohana YOLO based detection and classification of objects in video records. 2018 3rd IEEE international conference on recent trends in electronics, information and communication technology (RTEICT). <https://doi.org/10.1109/RTEICT42901.2018.9012375>
7. Abbas SM, Dr. Singh SN Region-based object detection and classification using faster R-CNN. 2018 4th international conference on computational intelligence and communication technology (CICT). <https://doi.org/10.1109/CICT.2018.8480413>
8. [https://github.com/tensorflow/models/blob/master/research/object\\_detection/g3doc/tf1\\_detection\\_zoo.md](https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf1_detection_zoo.md), Tensorflow Model Zoo
9. Darknet <https://github.com/AlexeyAB/darknet>
10. Pathak AR, Pandey M, Rautaray S Application of deep learning for object detection. Int Conf Comput Intell Data Sci (ICCIDS). <https://doi.org/10.1109/ICIS.2017.7960069>
11. Fang F, Li L, Zhu H, Lim J-H (22Oct 2019) Combining faster R-CNN and model-driven clustering for elongated object detection. IEEE Trans Image Proc 29. <https://doi.org/10.1109/TIP.2019.2947792>
12. Liu L, Ouyang W, Wang X, Fieguth P (2020) Deep learning for generic object detection: a survey. 31 October 2019 Int J Comput Vision 128:261–318. <https://doi.org/10.1007/s11263-019-01247-4>
13. Shah M, Kapdi R Object detection using deep neural networks. 2017 international conference on intelligent computing and control systems (ICICCS). <https://doi.org/10.1109/ICCONS.2017.8250570>
14. Liu R, Yu Z, Mo D, Cai Y An improved faster RCNN algorithm for object detection in remote sensing images. 2020 39th Chinese control conference (CCC). <https://doi.org/10.23919/CCC50068.2020.9189024>
15. Mane S, Mangale S Moving object detection and tracking using convolutional neural networks. 2018 Second international conference on intelligent computing and control systems (ICICCS). <https://doi.org/10.1109/ICCONS.2018.8662921>
16. Wei W. Small object detection based on deep learning. 2020 IEEE international conference on power, intelligent computing and systems (ICPICS). <https://doi.org/10.1109/ICPICS50287.2020.9202185>