

Development and performance evaluation of object and traffic light recognition model by way of deep learning

Shweta Bali¹, Tapas Kumar¹, Shyam Sunder Tyagi²

¹Department of Computer Science & Engineering, Manav Rachna International Institute of Research and Studies, Faridabad, India

²Department of Computer Science & Engineering, IIMT College of Engineering, Greater Noida, India

Article Info

Article history:

Received Sep 17, 2021

Revised Mar 9, 2022

Accepted Mar 23, 2022

Keywords:

Autonomous driving

Deep learning

Faster R-CNN

SSD

Traffic light detection

ABSTRACT

Deep learning models have shown incredible achievement in the field of autonomous driving, covering different aspects ranging from recognizing traffic signs and traffic lights, vehicle detection, license plate detection, pedestrian detection. Most of the algorithms perform better when the traffic lights are bigger in size, but the performance degrades in case of small-sized traffic lights. In this paper, the main emphasis is on evaluating two most promising deep learning architectures: single shot detector (SSD) and faster region convolutional network (Faster R-CNN) on “la route automatisée (LaRA) traffic light dataset” which contains small traffic lights as objects. The strengths and weaknesses are evaluated based on different parameters. The performance is compared in terms of mean average Precision (mAP@0.50) and average recall. The impact of data augmentation on the two architectures is also analyzed. ResNet50 V1 as feature extractor for Faster R-CNN achieved 96% mAP (mean average precision) which performed better than Original ResNet50 V1 Faster R-CNN pipeline. Also, different parameters such as batch size, learning rate and optimizer are tuned for detecting and classifying small traffic lights into different categories.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Shweta Bali

Department of Computer Science and Engineering, Faculty of Engineering and Technology

Manav Rachna International Institute of Research and Studies

Faridabad, India

Email: bali82.shweta@gmail.com

1. INTRODUCTION

Recognizing traffic lights is an important and widely researched area in the field of autonomous driving. Conventional traffic light detection techniques face challenges in detecting small objects due to small representations, low resolution, deformable property, significant overlapping, and fewer pixels. They also suffer from false positives in complex backgrounds. Recently, deep learning algorithms have addressed the real-time tasks related to autonomous driving such as detecting traffic signals, traffic signs, and pedestrians. Deep learning for object detection has attracted the sight of the researchers with the evolution of region convolutional network (R-CNN) [1]. Deep learning techniques are aimed to extract the features automatically and are highly accurate than the traditional machine learning techniques. Previous deep learning algorithms frequently encounter challenges when detecting small objects due to the disagreement between the spatial details and semantic information of deep convolutional neural networks (DCNNs). In complicated scenarios with identical backdrop objects and/or opacity, such as remote sensing imagery, the problem can be more difficult. Traffic lights occupy small area of the images. Moreover, small traffic light images are difficult to detect, and the background examples consists of trees, road, cars, and sky occupy large portion of the images. Li *et al.* [2] proposed that prior knowledge related to the context of traffic lights were

used to eliminate computational redundancy for traffic light recognition. The authors suggested a set of enhanced approaches such as aggregate channel feature method by altering each channel for traffic light type and further constructing a fusion detection mechanism. They elaborated an inter-frame information analysis method that employs previous frame detection information to alter initial proposal regions, thereby improving accuracy. Experiments were conducted on vision for intelligent vehicles and applications (VIVA) data set, the model performed well in contrast to previous traffic light identification methods. Kim *et al.* [3] presented an approach that comprised of two stages, segmentation technique that used connected component based labelling with an 8-connected neighbourhood to find the coordinates of the bounding boxes for finding the potential regions and convolutional network.

The simulations showed that the presented technique outperformed traditional Faster R-CNN. Kim *et al.* [4] explored that input video's colour space and the deep learning network model play a vital role in designing the algorithm for detecting traffic lights. The authors used six colour spaces following Faster R-CNN and R-FCN deep learning architectures. They conducted the experiments on traffic light dataset with images having 1280 * 720 resolution and the simulation results suggested that using the RGB colour space alongwith Faster R-CNN technique together yields good results. These findings can be used to develop a comprehensive traffic light detecting system design guide. Jensen *et al.* [5] presented different challenges in the traffic light recognition (TLR) research. They examined different TLR systems and proposed the evaluation process for such systems. They also created a public dataset of U.S. roadways footage which comprised of video sequences shot with a stereo camera in various lighting and weather situations for comparing TLR systems. Munoz-Organero *et al.* [6] developed a novel mechanism for automatically detecting traffic lights, street crossings, and roundabouts for producing street maps. Janahiraman and Subuhan [7] evaluated TensorFlow object recognition framework [8] to handle different challenges. They developed technique for traffic light detection using MobileNetV2 single shot multibox detector (SSD), and Faster R-CNN. The results of the experiments showed that Faster R-CNN outperformed SSD by 38.806%. Wang *et al.* [9] investigated identification of the traffic lights based on deep learning.

They proposed a region proposal technique based on different parameters such as color, intensity and geometry for detecting the traffic lights. They tested the model on 6804 photographs of diverse circumstances, the model achieved an average accuracy of 99.6%, recall of 99.2% and accuracy of detection as 98.5%. Wang *et al.* [10] designed simultaneous detection and tracking mechanism to ascertain color as well as position of traffic lights. Wang and Zhou [11] proposed a technique in which traffic light detection is performed on dark frames. In bright frames, this dual-channel method can fully utilize undistorted colour and shape information whereas it uses context in light frames. Hassan *et al.* [12] examined two approaches for detecting small objects. The first approach used traditional color-based segmentation method to recognize objects based on hue, saturation, and value parameters, whereas the second employs mask R-CNN to identify traffic light. In this paper, Hui *et al.* [13] presented a novel method that includes the generation of candidate traffic light regions based on genetic optimization. Localization and classification of traffic lights is based on deep neural networks that perform different tasks ranging from feature region extraction, parameter sampling of candidate region, and parameter optimization of the genetic algorithm. Gokul *et al.* [14] researchers examined model architecture and parameters of Faster R-CNN and you only look once (YOLO) detectors to recognise and categorise images in Bosch small traffic light dataset.

In terms of precision, the experiment indicates that the Faster R-CNN model outperforms the YOLO model whereas YOLO, on the other hand, outperforms the competition when it comes to real-time deployment. Lu *et al.* [15] developed an attention model that produced a small number of probable regions that help small targets to be easily detected and further categorised. They curated a Tencent street dataset comprising of 15K instances with a wide diversity of traffic lights in street settings with different lighting situations, as well as forms than those found in the laboratory for intelligent and safe automobiles (LISA) dataset for traffic signal recognition. It was found that the discussed algorithm outperformed the conventional Faster R-CNN object detection framework. Muller and Dietmayer [16] implemented experiments on inception V3 model and employed non-maximum suppression algorithm on DriveU traffic light dataset (DTLD) for multiple detections avoidance on small objects. Cao *et al.* [17] proposed a novel enhanced loss function based on intersection over union (IoU) and used bilinear interpolation to improve the position information. There is improvement in the recall as well as the accuracy values for the small objects. Patel and Thakkar [18] discussed the role of AI in different application areas. Jasm *et al.* [19] implemented the image classification using convolutional neural network (CNN) on Canadian Institute for Advanced Research, 10 classes (CIFAR-10) dataset. Kadim *et al.* [20] used pre-trained CNN alongwith fully connected layers to handle challenges for dealing with changes in appearance. Also different hyperparameters, learning rate and ratio of training samples are optimized for tracking in night conditions. Liu and Yan [21] discussed the use of capsule networks for traffic light detection. In literature it has been found that detecting the small objects is a challenging problem due to small no of the pixels, varying background, small resolution which need to be researched. The existing deep learning architectures perform better on larger objects but for smaller objects

there is a performance issue. In this paper experiments are performed on the SSD pipeline with feature pyramid networks (FPN) and Faster R-CNN using ResNet50 V1 as feature extractor and it is found that after adding the data augmentation to the Faster R-CNN with ResNet50 V1 as pretrained model performed better than existing techniques.

2. RELATED WORK

2.1. Two stage object detection algorithm: Faster R-CNN

RCNN families of the algorithms were the first designed algorithms. Faster R-CNN is improvisation over Fast R-CNN. Fast R-CNN [1] used selective search algorithm for generating regions in which object could be present. In case the dataset is large using Fast R-CNN resulted in huge number of regions being generated. Faster R-CNN [22] is a neural network-based algorithm which framework incorporates both region proposal network (RPN) and Fast R-CNN algorithms for region generation and object detection respectively. It utilizes RPN to create object proposals and reduces proposal generation time taken by R-CNN algorithms using selective search algorithm of 2s to 10ms per image. It also allows sharing of layers between region proposal stage and detection stages, thus enhancing the feature representation. The architecture [19] for the model is shown in Figure 1.

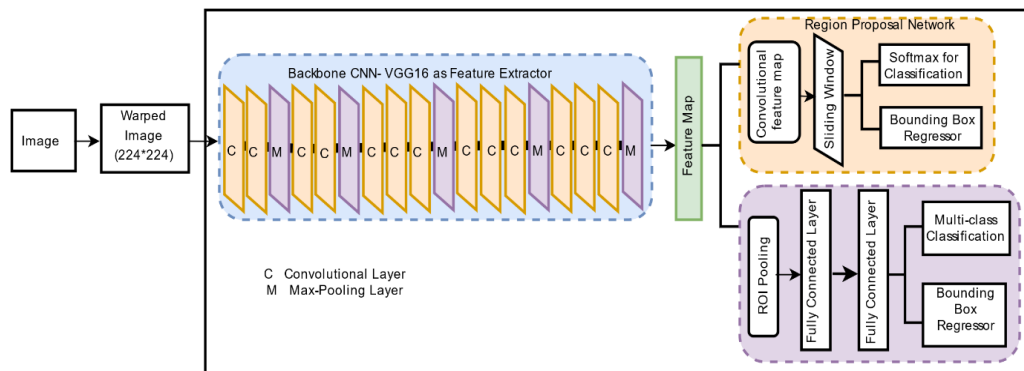


Figure 1. Faster R-CNN deep learning neural network architecture [20]

2.1.1 Convolutional neural network

CNN are used as feature extractor/backbone to which an image is passed. As discussed in the paper [19], there are two possible CNNs: VGG-16 (13 shareable convolutional layers) or ZFnet (5 shareable convolutional layers) which have network stride of 16 that means if the input image has (1000x600) dimensions then it generates $(1000/16 \times 600/16) \approx (62 \times 37)$ sized feature map. In Figure 1 VGG-16 is shown as the feature extractor that comprises of a series of 3x3 convolutional layers with stride of 1 and padding of 1 and 2x2 Maxpooling with stride of 2. Layer 13 is the last convolutional layer that is used as input to the (RPN) network.

2.1.2 Region proposal network

The input to the RPN is the convolutional feature map generated by the backbone network and output are the anchors (potential bounding box candidates where there is a possibility of object presence) obtained by applying sliding window convolution. For each and every point on the output feature map, the networks outputs if an object is present or not by placing $k=9$ (default) anchors in three distinct scales with box region 128^2 , 256^2 , 512^2 and three aspect ratios of 1:1, 1:2 and 2:1. Therefore for a convolutional feature map ($w \times h$), a total of $n = w \times h \times 9$ anchors are generated. These anchors are additionally refined to generate bounding boxes. Then these region proposals are passed into a 3x3 convolution layer with padding of 1 and generates 512 dimensional feature map for every location for VGG-16. RPN network generates the region proposals from anchors. The output generated is passed to two 1x1 convolution layers. One is the classification layer with output having size ($h, w, 18$) which outputs probabilities of the bounding box to have an object or not. The total number of the output parameters are $2*n$ ($w * h * (2*k)$). The other is the regression layer with output having size ($h, w, 36$) that generated 4 regression coefficients for all the 9 anchors related to every point present in backbone feature map. The total number of the output parameters are $4*n$ ($w * h * (4*k)$). Due to sharing computation on convolutional features, RPN reduces the cost,

improves accuracy, and reduces running time while also avoiding the generation of superfluous proposal boxes improves the accuracy of Faster R-CNN but slows down its processing speed. The output feature map comprises of 40×60 locations and a total of $40 \times 60 \times 9$ anchors. An anchor is considered to contain an object if IoU (Intersection over Union) is more than 0.7 value with any ground-truth box. An anchor does not contain an object if its IoU with all groundtruth boxes is less than 0.3. The total number of the anchors is $40 \times 60 \times 9 = 21,600$ which is very large and in order to reduce this count, anchors that are out of the boundary of an image are ignored and non-maximum suppression algorithm is also applied which results in reducing the number of anchors to 2000 which are used for training. The loss in RPN is calculated in (1):

$$L(p_i, t_i) = \frac{1}{N_c} \sum_i L_c(p_i, p_i^*) + \lambda \frac{1}{N_r} \sum_i L_r(t_i, t_i^*) \quad (1)$$

where N_c : No of anchors in minibatch(256), N_r : No of anchors(~2000), L_c : Cross entropy(Softmax) loss for classifier, L_r : Smooth loss for bounding box regressor that gets triggered in case anchor contains an object, p_i : predicted probability of anchors contains an object or not, p_i^* : ground truth probability(0: for negative anchors, 1: for positive anchors), t_i : predicted box coordinates, t_i^* : ground truth predicted box.

2.1.3 Fast-RCNN (detector)

The detection algorithm constitutes a pretrained CNN for feature extraction, region of interest (ROI) pooling layer, fully connected layers, softmax layer for classification and regression layer. ROI pooling is combined with Fast-RCNN for making the detection pipeline. ROI pooling layer makes the different sized region proposals that are generated by RPN into fixed-size feature map of size $(7 \times 7 \times D)$ ($D=256$ for VGG-16, $D=512$ for ZFnet). This feature map is further directed to two fully connected layers. One among them is the classification layer or the softmax layer which has $n+1$ parameters (n : number of classes) that generates the classification score that depicts the probability for each class proposals. The regression layer has $4 \times n$ parameters and uses coefficients to improve the predicted bounding boxes. ROI are considered as object proposals if IoU is greater than or equal to 0.5. ROI that are positive for a class are labelled as belonging to that class ($u=1 \dots n$), ROI that belong to the background class have $u=0$. The multi-task loss for each ROI is calculated in (2) as combination of two losses: classification loss and regression loss.

$$\text{Fast R - CNN loss} = L_c(p, u) + \lambda L_r(t^u, v) \quad (2)$$

The classification loss $L_c(p, u)$ is calculated for every ROI over $(n+1)$ classes. $L_c(p, u) = -\log(p_u)$: log loss for true class u . $L_r(t^u, v)$: Smooth L1 loss for regression. The regression layer generates regression offsets, t_i^k where $i = (x, y, w, h)$, (x, y) refers to the top left corner and w, h refers to bounding box width and height respectively v_i refer to true bounding box regression targets.

2.2. One stage object detection algorithm: single shot detector

One stage detector solves the problem of object detection as a simple regression problem by taking an input image and learning the class probabilities and bounding box coordinates. The architecture of SSD is explained below.

2.2.1 Backbone network

The main goal of backbone network is feature extractions. There are two versions SSD300 and SSD500. In this paper [23], SSD300 is considered. VGG-16 architecture which is pretrained on ImageNet dataset for image classification is used for feature extraction and generation of the feature maps. The different feature maps used to piece together SSD300 with VGG16 base network are: conv4_3, fc7, conv8_2, conv9_2, conv10_2 and conv11_2.

2.2.2 Convolutional layers for prediction

Convolutional feature layers are used to augment the shortened base network. Convolutional predictors prevent the loss of the spatial information given by the feature maps and also generate less no of the parameters as the predictions produced are of the size $38 \times 38 \times \text{number_of_classes}$. SSD uses the auxiliary layers to extract the features at the multiple scales and these layers gradually reduce the size of the Input. The architecture [20] in Figure 2 generates bounding boxes and scores for objects present in those boxes using a feed-forward convolutional network, then applies the non-maximum suppression (NMS) algorithm to obtain final result. The first layers in the network are based on a standard architecture for high-quality image classification. These layers gradually reduce in size, allowing detections at various scales to be predicted. To forecast detections, a new convolutional model is utilised for each feature layers. Using convolutional filters at each layer generates 8732 predictions per object. The detections are performed at

different scales, fc6 layers and fc7 layers of VGG16 networks are changed into convolutional layers, pool5 layer's pool size changed from (2,2) to (3,3) with stride of 1. To the conv4_3 layer of VGG16, L2 normalization is added. SSD training objective is the weighted sum of confidence loss (L_c) and localization loss (L_r) as depicted in (3), (4), and (5) respectively.

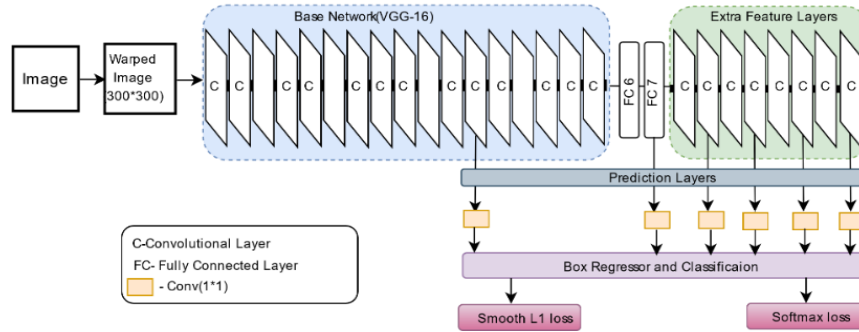


Figure 2. Single shot detector architecture [19]

$$L(x, c, l, g) = \frac{1}{N} (L_c(x, c) + \alpha L_r(x, l, g)) \tag{3}$$

$$L_r(x, l, g) = \sum_{i \in POS}^N \sum_{m \in \{cx, cy, w, h\}} x_{ij}^k smooth_{L1}(l_i^m - \hat{g}_j^m) \tag{4}$$

where

$$\begin{aligned} \hat{g}_j^{cx} &= (\hat{g}_j^{cx} - d_i^{cx}) / d_i^w \\ \hat{g}_j^{cy} &= (\hat{g}_j^{cy} - d_i^{cy}) / d_i^w \\ \hat{g}_j^w &= \log(g_i^w / d_i^w) \\ \hat{g}_j^h &= \log(g_i^h / d_i^h) \\ L_c(x, c) &= -\sum_{i \in POS}^N x_{ij}^p \log(\hat{c}_i^p) - \sum_{i \in Neg}^N \log(\hat{c}_i^0) \text{ where } \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)} \end{aligned} \tag{5}$$

where N: number of matched default boxes, c: number of classes, α : weight term, localization loss, L_r is the smooth L1 loss between predicted box (l) and ground box (g). The transformations are then performed for bounding box corrections to center (cx, cy) of the default box (d) with width (w) and height (h).

3. RESEARCH METHOD

In this paper, ResNet50 V1 model is used as the feature extractor in the Faster R-CNN framework as shown in Figure 3. Different parameters such as batch size, learning rate, epochs, step-size are fine tuned for the traffic lights present in the dataset. Different data augmentation techniques applied are brightness, contrast, hue and saturation to increase the diversity of the dataset and improve the performance on the small traffic lights in the dataset.

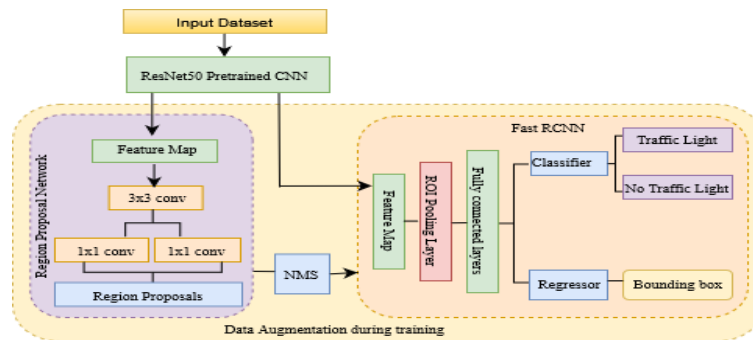


Figure 3. Data augmented Faster R-CNN architecture

4. RESULTS AND DISCUSSION

4.1. Experimental setup

The experiments were performed on Google Colab using Tesla K80 GPU. The software used are Tensorflow object detection API framework, Anaconda virtual environment included CUDNN 7.6, CUDA 10, Python 3.8. The evaluation is performed on subset of la route automatisée (LaRA) traffic light dataset provided [23]. The training of the different architectures had been run for 6000 steps each with a batch size of 4. The object detection results are evaluated on the metrics: mean average precision and recall. The training and testing images are resized to resolution 640×640 pixels as per the requirement of different frameworks. For experimentation different data augmentation transformations applied to training dataset such as random brightness with value of 0.2, random contrast in range (0.7,0.11), random hue with value of 0.01 and random saturation in the range (0.75,1.15). For conducting experiments of deep convolutional neural network-based object detection, subset of the benchmark dataset, LaRA traffic light dataset is used. The original dataset [24], [25] contains 11179 8-bit red, green, blue (RGB) traffic light images and 9168 annotated traffic lights with resolution of 640x480 divided into four classes namely go, stop, warning and ambiguous. In Figure 4, different types of the images are presented depicting different challenges like blur as shown in Figure 4(a), multiple objects as shown in Figure 4(b), small-size objects as shown in Figure 4(c) while performing detection.

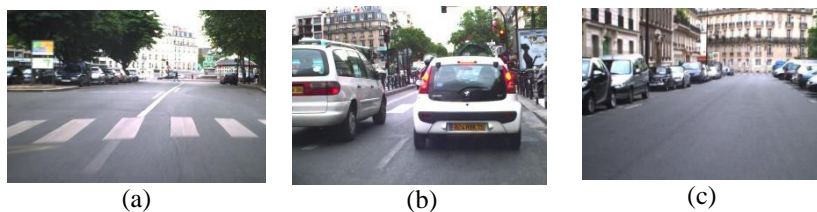


Figure 4. Examples of frames from LARA traffic light dataset; (a) blurred image, (b) multiple objects, and (c) small-sized object

4.2. Evaluation metrics

In evaluating the object detection model for traffic light detection, precision in (6) is defined as the ratio of total true positives traffic light samples to total number of positive predictions and Recall in (7) refers to ratio of total true positives traffic light samples to total number of actual/relevant samples. These two parameters have trade-off relationship which means that one improves at the expense of other.

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

IoU is the amount of overlap between predicted boundary box with ground truth box for object detection. A prediction is marked as true positive if IoU is above a certain threshold. IoU varies between 0.5 and 0.95. Average precision (AP) is a metric used to calculate accuracy of object detection model. It indicates how well the model handles the positives. For common objects in context (COCO), AP is computed by taking average over several IoU AP@ [.5:.95] is defined as average AP for IoU from the range 0.5 to 0.95 with a step size of 0.05. Mean average precision (mAP) is calculated as mean of AP score over all classes across all IoU thresholds.

4.3. Discussion of the results

In this paper, ResNet50 V1 pretrained model on ImageNet database is used as the feature extractor and fine-tuned on the LaRA traffic light dataset. RPN is trained on mini-batch, and then RPN and base network parameters are updated. Then, both the positive and negative proposals generated by RPN are used for training and further updating the classifier. The loss function and parameterization of coordinates for bounding box regression are kept as in original Faster R-CNN. Stochastic gradient descent (SGD) is used as optimizer with initial learning rates of the RPN and classifier set to 0.04 with the learning rate decay of 0.0005 per batch. In Figure 5-7 the results of the different frameworks such as SSD Resnet50 V1 FPN, Faster R-CNN ResNet50 V1, data augmented Faster R-CNN ResNet50 V1 architecture are shown respectively. The network is trained for 6000 steps. Different loss graphs have been illustrated such as classification loss in

Figure 5(a), localization loss in Figure 5(b), regularization loss in Figure 5(c), total loss in Figure 5(d) for SSD Resnet50 V1 FPN architecture, in Figure 6(a) BoxClassifier: classification loss is presented, BoxClassifier: localization loss is presented in Figure 6(b), RPN: objectness loss is presented in Figure 6(c), RPN: localization loss is presented in Figure 6(d), total loss in Figure 6(e), regularization loss for Faster R-CNN ResNet50 V1 architecture, in Figure 7(a) BoxClassifier: classification loss is presented, BoxClassifier: localization loss is presented in Figure 7(b), RPN: objectness loss is presented in Figure 7(c), RPN: localization loss is presented in Figure 7(d), total loss in Figure 7(e), regularization loss for data augmented modified Faster R-CNN ResNet50 V1 architecture, where x-axis represent the total number of the steps and the y-axis represent different losses respectively. The loss is decreasing continuously with each step till value of 0.2. Table 2 depicts the results of the different architectures in terms of mean average precision score and it is experimentally found that the Faster R-CNN based model performs better by 11% increase in the mean average precision score than SSD based model. Also there is an increase in 2% mean average precision score after data augmentation is performed.

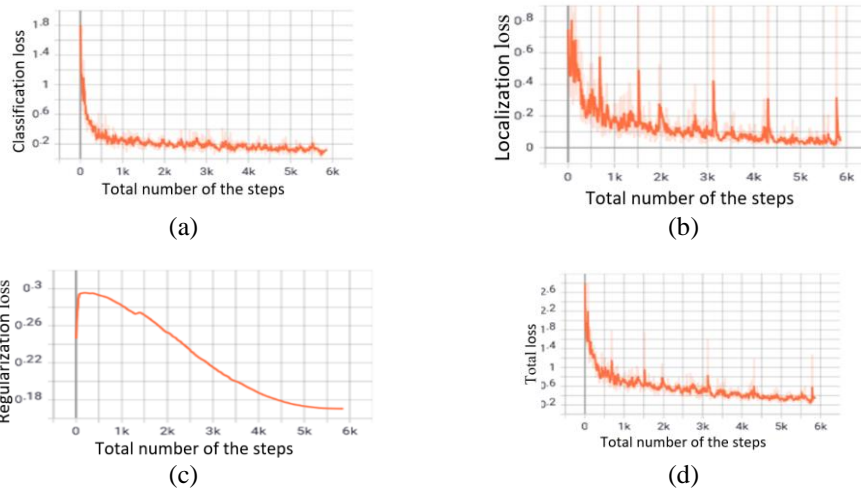


Figure 5. Losses in SSD Resnet 50 V1 FPN detection framework for (a) classification loss, (b) localization loss, (c) regularization loss, and (d) total loss

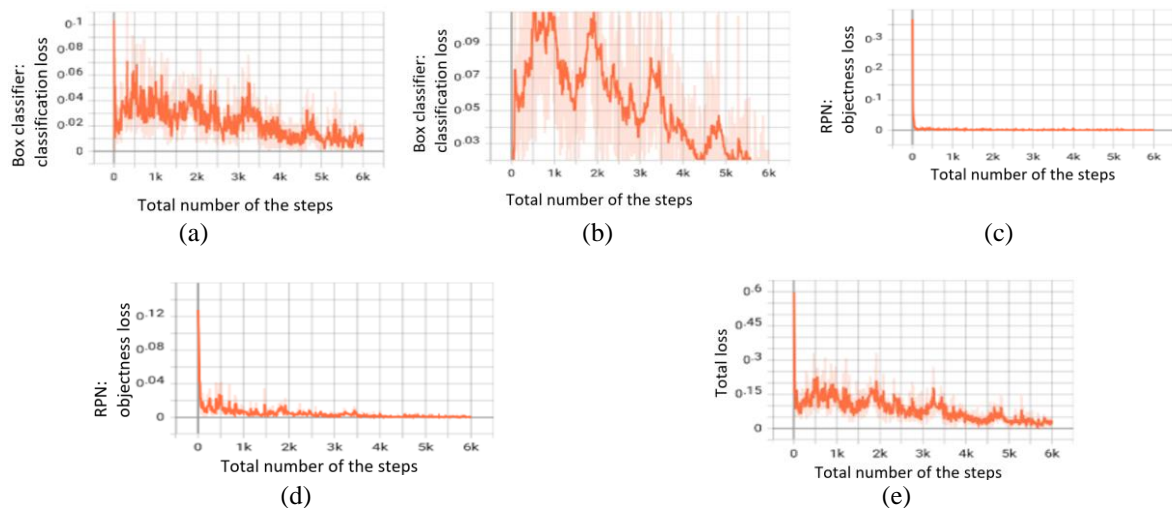


Figure 6. Losses in Faster R-CNN Resnet50 V1 detection framework for (a) BoxClassifier: classification loss, (b) BoxClassifier: localization loss, (c) RPN: objectness loss, (d) RPN: localization loss, and (e) total loss

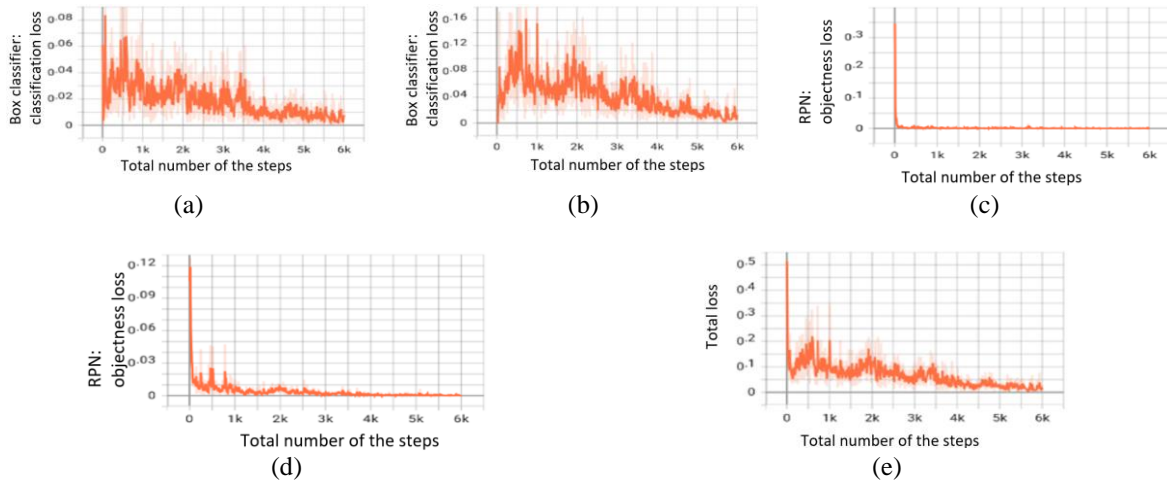


Figure 7. Losses in modified Faster R-CNN Resnet50 V1 detection framework for (a) BoxClassifier: classification loss, (b) BoxClassifier localization loss, (c) RPN: objectness loss, (d) RPN: localization loss, and (e) total loss

Table 2. Results for different models on LaRa traffic light dataset

Model	Resolution (in pixels)	mAP (in %)	AR (in %)
SSD Resnet50 V1 FPN	640×640	83	62
Faster R-CNN Resnet50 V1	640×640	94	43
Data augmented +Faster R-CNN Resnet50 V1	640×640	94	40

5. CONCLUSION

In this paper the problem of detecting the small traffic light in the images on the challenging dataset is addressed. Also a comparison is performed between two existing architectures namely SSD and Faster R-CNN with ResNet50 V1 as the pretrained feature extractor. It has been found that Faster R-CNN ResNet50 V1 achieved a higher mAP of 94% as compared to SSD ResNet50 FPN V1 of 83%. Further data augmentation is applied to the dataset during the training process and an improvement of 2% in mAP score of Faster R-CNN ResNet50 V1 is achieved. The results together with the systematic study led to performance gains for detecting the small traffic lights. In future further different architectures can be explored for performance on the dataset.




REFERENCES

- [1] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.
- [2] X. Li, H. Ma, X. Wang, and X. Zhang, "Traffic light recognition for complex scene with fusion detections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 199–208, Jan. 2018, doi: 10.1109/TITS.2017.2749971.
- [3] H.-K. Kim, K.-Y. Yoo, J. H. Park, and H.-Y. Jung, "Traffic light recognition based on binary semantic segmentation network," *Sensors*, vol. 19, no. 7, p. 1700, Apr. 2019, doi: 10.3390/s19071700.
- [4] H.-K. Kim, J. H. Park, and H.-Y. Jung, "An efficient color space for deep-learning based traffic light recognition," *Journal of Advanced Transportation*, vol. 2018, pp. 1–12, Dec. 2018, doi: 10.1155/2018/2365414.
- [5] M. B. Jensen, M. P. Philipsen, A. Mogelmoose, T. B. Moeslund, and M. M. Trivedi, "Vision for looking at traffic lights: issues, survey, and perspectives," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1800–1815, Jul. 2016, doi: 10.1109/TITS.2015.2509509.
- [6] M. Munoz-Organero, R. Ruiz-Blaquez, and L. Sánchez-Fernández, "Automatic detection of traffic lights, street crossings and urban roundabouts combining outlier detection and deep learning classification techniques based on GPS traces while driving," *Computers, Environment and Urban Systems*, vol. 68, pp. 1–8, Mar. 2018, doi: 10.1016/j.compenvurbusys.2017.09.005.
- [7] T. V. Janahiraman and M. S. M. Subuhan, "Traffic light detection Using Tensorflow object detection framework," in *2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*, Oct. 2019, pp. 108–113, doi: 10.1109/ICSEngT.2019.8906486.
- [8] J. Huang et al., "Speed/accuracy trade-offs for modern convolutional object detectors," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 3296–3297, doi: 10.1109/CVPR.2017.351.
- [9] C. Wang, G. Zhang, W. Zhou, Y. Rao, and Y. Lv, "Traffic lights detection based on deep learning feature," in *IoTaaS 2019: IoT as a Service*, vol. 316 LNCS, 2020, pp. 382–396.
- [10] K. Wang, X. Tang, S. Zhao, and Y. Zhou, "Simultaneous detection and tracking using deep learning and integrated channel feature for ambient traffic light recognition," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 1, pp. 271–281, Jan. 2022, doi: 10.1007/s12652-021-02900-y.




- [11] J.-G. Wang and L.-B. Zhou, "Traffic light recognition with high dynamic range imaging and deep learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 4, pp. 1341–1352, Apr. 2019, doi: 10.1109/TITS.2018.2849505.
- [12] N. Hassan, K. W. Ming, and C. K. Wah, "A comparative study on HSV-based and deep learning-based object detection algorithms for pedestrian traffic light signal Recognition," in *2020 3rd International Conference on Intelligent Autonomous Systems (ICoIAS)*, Feb. 2020, pp. 71–76, doi: 10.1109/ICoIAS49312.2020.9081854.
- [13] X. Hui, G. Yu'ang, C. Chaoyi, X. Qing, and L. Keqiang, "Traffic light detection based on genetic optimization and deep learning," *Automotive Engineering*, vol. 41, no. 8, pp. 960–966, 2019, doi: 10.19562/j.chinasae.qcgc.2019.08.001.
- [14] R. Gokul, A. Nirmal, K. Bharath, M. Pranesh, and R. Karthika, "A comparative study between state-of-the-art object detectors for traffic light detection," in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, Feb. 2020, pp. 1–6, doi: 10.1109/ic-ETITE47903.2020.449.
- [15] Y. Lu, J. Lu, S. Zhang, and P. Hall, "Traffic signal detection and classification in street views using an attention model," *Computational Visual Media*, vol. 4, no. 3, pp. 253–266, Sep. 2018, doi: 10.1007/s41095-018-0116-x.
- [16] J. Muller and K. Dietmayer, "Detecting traffic lights by single shot detection," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2018, pp. 266–273, doi: 10.1109/ITSC.2018.8569683.
- [17] C. Cao *et al.*, "An improved Faster R-CNN for small object detection," *IEEE Access*, vol. 7, pp. 106838–106846, 2019, doi: 10.1109/ACCESS.2019.2932731.
- [18] P. Patel and A. Thakkar, "The upsurge of deep learning for computer vision applications," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, p. 538, Feb. 2020, doi: 10.11591/ijece.v10i1.pp538-548.
- [19] D. A. Jasm, M. M. Hamad, and A. T. H. Alrawi, "Deep image mining for convolution neural network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 1, p. 347, Oct. 2020, doi: 10.11591/ijeecs.v20.i1.pp347-352.
- [20] Z. Kadim, M. A. Zulkifley, and N. Hamzah, "Deep-learning based single object tracker for night surveillance," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 4, p. 3576, Aug. 2020, doi: 10.11591/ijece.v10i4.pp3576-3587.
- [21] X. Liu and W. Q. Yan, "Traffic-light sign recognition using capsule network," *Multimedia Tools and Applications*, vol. 80, no. 10, pp. 15161–15171, Apr. 2021, doi: 10.1007/s11042-020-10455-x.
- [22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-Time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.
- [23] W. Liu *et al.*, "SSD: single shot multiBox detector," in *Computer Vision – ECCV 2016*, 2016, pp. 21–37.
- [24] R. de Charette and F. Nashashibi, "Real time visual traffic lights recognition based on spot light detection and adaptive traffic lights templates," in *2009 IEEE Intelligent Vehicles Symposium*, Jun. 2009, pp. 358–363, doi: 10.1109/IVS.2009.5164304.
- [25] "Traffic lights recognition (TLR) public benchmarks:2015." Robotics centre of mines paristech and imara team of INRIA, [Online]. Available: <http://www.lara.prd.fr/benchmarks/trafficlightrcognition>.

BIOGRAPHIES OF AUTHORS






Shweta Bali    is a research scholar in department of Computer Science and Engineering (CSE), FET, Manav Rachna International Institute of Research and Studies (MRIIRS), Faridabad. She has 10 research publications to her credit. Her area of interest are Machine learning and Computer Vision. She can be contacted at email: bali82.shweta@gmail.com.



Dr. Tapas Kumar    is presently working as Professor and Head, Computer Science & Engineering (CSE), FET Manav Rachna International Institute of Research and Studies (MRIIRS), Faridabad. Dr. Tapas Kumar is currently guiding 6 Ph.D scholars, 08 research scholars have completed their P.hD under his guidance. He has the credit of publishing 45 Research Articles in refereed International Journals and 15 research Papers published in various International and National Conferences. He has to his credit as found through Google Scholar with 361 citations has got the h-index of 08. His area of interests includes artificial intelligence, machine learning, pattern recognition and digital image processing. He can be contacted at tapaskumar.fet@mriu.edu.in.



Dr. Shyam Sunder Tyagi    is presently working as a Professor and Director, IIMT College of Engineering, Greater Noida. He is having experience of more than 29 years of academic/teaching/research. He is guiding/guided Ph.D. Scholars in field of ad hoc networks, software defined networking, cloud computing, and wireless security. There are more than 100 research publications to his credit published in reputed International/National Journals and in the proceedings of International and National Conferences. He can be contacted at shyamtyagi@hotmail.com.