Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.060928

ARTICLE

Optimized Convolutional Neural Networks with Multi-Scale Pyramid Feature Integration for Efficient Traffic Light Detection in Intelligent Transportation Systems

Yahia Said[1,2,*], **Yahya Alassaf**[3], **Refka Ghodhbani**[4], **Taoufik Saidani**[4] and **Olfa Ben Rhaiem**[5]

[1]Department of Electrical Engineering, College of Engineering, Northern Border University, Arar, 91431, Saudi Arabia
[2]Center for Scientific Research and Entrepreneurship, Northern Border University, Arar, 73213, Saudi Arabia
[3]Department of Civil Engineering, College of Engineering, Northern Border University, Arar, 91431, Saudi Arabia
[4]Faculty of Computing and Information Technology, Northern Border University, Rafha, 91911, Saudi Arabia
[5]College of Science, Northern Border University, Arar, 91431, Saudi Arabia

*Corresponding Author: Yahia Said. Email: Yahia.said@nbu.edu.sa

Received: 12 November 2024; Accepted: 17 December 2024; Published: 17 February 2025

ABSTRACT: Transportation systems are experiencing a significant transformation due to the integration of advanced technologies, including artificial intelligence and machine learning. In the context of intelligent transportation systems (ITS) and Advanced Driver Assistance Systems (ADAS), the development of efficient and reliable traffic light detection mechanisms is crucial for enhancing road safety and traffic management. This paper presents an optimized convolutional neural network (CNN) framework designed to detect traffic lights in real-time within complex urban environments. Leveraging multi-scale pyramid feature maps, the proposed model addresses key challenges such as the detection of small, occluded, and low-resolution traffic lights amidst complex backgrounds. The integration of dilated convolutions, Region of Interest (ROI) alignment, and Soft Non-Maximum Suppression (Soft-NMS) further improves detection accuracy and reduces false positives. By optimizing computational efficiency and parameter complexity, the framework is designed to operate seamlessly on embedded systems, ensuring robust performance in real-world applications. Extensive experiments using real-world datasets demonstrate that our model significantly outperforms existing methods, providing a scalable solution for ITS and ADAS applications. This research contributes to the advancement of Artificial Intelligence-driven (AI-driven) pattern recognition in transportation systems and offers a mathematical approach to improving efficiency and safety in logistics and transportation networks.

KEYWORDS: Intelligent transportation systems (ITS); traffic light detection; multi-scale pyramid feature maps; advanced driver assistance systems (ADAS); real-time detection; AI in transportation

1 Introduction

Intelligent transportation systems, including ADAS and autonomous cars, have benefited greatly from recent developments in AI and big data. By incorporating computer software that can perceive and disrupt traffic conditions in an urban setting, intelligent transportation systems aim to enhance vehicle safety and efficiency. There are a lot of obstacles that make it difficult to simulate human visual data analysis on a computer. These include things like backdrop complexity, object similarity, intraclass variance, and different perspectives. Computers need to be able to detect and identify a wide variety of things, including humans, other cars, traffic signs, and traffic signals. Among them, the traffic lights stand out as an essential component of cityscapes for managing vehicular traffic. More cars on the road means more complicated traffic conditions, which means more accidents. The development of artificial systems capable of independently detecting and recognizing things in their environment and making decisions in accordance with traffic regulations is, thus, viewed as a potential answer.

A high-performance traffic light detector is the subject of our investigation in this study. Due to their substantial influence on the intelligent transportation system, such applications need a balance between speed and accuracy. The standard geometric shape for traffic lights is a rectangle oriented either vertically or horizontally, with a predetermined number of colors (red, yellow, and green) and directions (straight, left, and right) to alert cars. Having said that, traffic lights only take up a tiny fraction of a real-life scene and need to be identified from more than 100 m away. Even in a complex, faraway landscape, a human motorist may easily notice and recognize the status of the traffic signal. It takes a lot of processing power and robust cameras for an artificial system to figure out what the traffic light status is. Estimating the recognition distance is as simple as finding the distance between the vehicle's braking point and the stop line. The braking distance is approximately 98 m if the top speed in an urban setting is 50 KM/H. We need to talk about the trade-off between the suggested method's performance and the capabilities of the hardware utilized to execute it if we assume a recognition distance of 100 m.

A convolutional neural network equipped with multi-scale pyramid feature maps is our suggested solution to tackle these issues. The self-learning and generalizability capabilities of convolutional neural networks [1] have made them very effective in several computer vision applications [2–6]. The majority of the convolutional neural networks that are suggested are repurposed for tackling different types of problems through the process of transfer learning. By transferring the training weights and fine-tuning the data of the new application, the transfer learning approach enables the usage of an existing convolutional neural network to handle new problems. Numerous traffic-related applications have shown the efficacy of this method, including the identification of traffic signs [6], cars [7], and a plethora of others [8,9]. Because most current convolutional neural networks only use one dimensional for feature extraction, downsampling the input picture makes it harder to localize and identify traffic lights in high-resolution photos. As a solution, we suggest integrating local and global characteristics using multi-scale pyramid feature maps. Building a multi-scale feature extraction network on top of the feature pyramid network was the primary goal in order to acquire semantic characteristics of tiny traffic lights [10]. We suggest ResNet 101 [11], a basic model for convolutional neural networks because it ranks fifth with an error of 4.60% in the ILSVRC 2015 classification challenge. The ResNet replaces standard layers with residual blocks, which are connected to the input and output of the block using skip layers. To avoid complexity explosions and mitigate the impact of vanishing gradient, residual blocks are utilized in the construction of extremely deep convolutional neural networks (with above 100 layers). Therefore, it facilitates the effortless training of extremely deep convolutional neural networks. To recognize traffic signals at multiple sizes, the multi-scale feature extraction network was used in conjunction with the suggested ResNet 101 model. Proposals for regions were generated using the region-based method [12]. On top of that, we suggest using ROI Align [13] to create fixed feature map dimensions from the combined feature maps. To further improve the detection results, we also included the usage of the soft-NMS algorithm [14] for seeking the best-fit bounding box.

Training and evaluating convolutional neural networks requires massive amounts of data. Accordingly, the LISA traffic light dataset is what we recommend using [15]. The film consists of 43,007 continuous frames with 113,888 annotated traffic signals. The suggested method was trained and tested using the dataset. By producing excellent results, the assessment of the suggested method demonstrates its efficacy.

In order to implement a convolutional neural network, one should aim for the graphics processing unit (GPU). Convolutional neural networks can now be integrated into embedded devices, thanks to advancements in embedded GPU technology. The latest embedded GPU from Nvidia, the Drive AGX, has enormous capabilities and little power consumption. To put the suggested method to the test, we create an experimental setting that matches the performance of the Nvidia Drive AGX.

This work primarily contributes three things: (1) a method to fix the dimensions of feature maps using region-of-interest alignment; (2) a network to extract features from multiple scales in high-resolution images; and (3) an application of transfer learning to the ResNet 101 model for traffic light detection.

The remainder of the paper is organized as follows. Related works are discussed in Section 2. In Section 3, the proposed approach is detailed. Section 4 is reserved for experimental results and discussions. Conclusions and future work are presented in Section 5.

2 Related Works

Traffic light detection was and still is an important research field. There have been several attempts to address the issue of traffic light detection using various ways. Deep learning and convolutional neural network-based efforts will constitute the main emphasis of this paper. Using Global Positioning System (GPS) data, John et al. [16] suggested a convolution neural network that can identify traffic signals in a variety of lighting situations.

Behrendt et al. [17,18] proposed a YOLO-based convolutional neural network for traffic light detection, leveraging the YOLO framework for object detection in high-resolution images. During the training phase, input data was generated from randomly selected crops of the original images. In the testing phase, only three crops from the top portion of the image were used, as traffic lights are typically located in this region. Additionally, the system monitored the traffic light's status in real time using stereo vision and vehicle odometry.

To train and evaluate the proposed technique, a dedicated dataset was introduced. Approximately 5000 images were collected along El Camino Real in the San Francisco Bay Area of California for training purposes. For testing, 8334 images were extracted from a stereo video sequence recorded along University Avenue in Palo Alto, California. While the proposed method demonstrated real-time processing capabilities, its accuracy was limited, achieving less than 60%.

In reference [19], a convolutional neural network-based encoder-decoder. The encoder network was ResNet 101 [11], whereas the decoder network was constructed using deconvolution blocks with skip connections. In order to anticipate boundary boxes for traffic light detection, a detector network was also utilized. The YOLO framework [18] served as the basis for the detector network. Two datasets, the Bosch small [17] traffic light dataset and the LISA traffic lights dataset [15], were used to assess the suggested method. According to the results that were provided, the performance was better than what is currently considered state-of-the-art.

The status of traffic lights can be effectively detected and recognized using a combination of a convolutional neural network (CNN) detector and pre-existing maps, as described in [20]. The pre-existing maps are utilized to locate the traffic light, while the CNN identifies its current state. The proposed method was evaluated using videos captured with a 2D camera that met specific requirements. The results demonstrate that the method accurately detects the traffic light's condition along the tested trajectory. However, while the method performs well, it is limited in applicability without the availability of pre-existing maps.

Using LIDAR data and a convolutional neural network, Yeh et al. [21] suggest a method for traffic light detection. There are two parts to the suggested method. In the first, we used two cameras with varying focus lengths to identify traffic lights at various distances. In the second, we used the same lights to determine their status. The initial step was finding the traffic light in the picture using the YOLOv3 model [22].

After identifying the traffic light status (red, green, yellow, or arrows) using the YOLOv3 small model [22], the second stage was using the LeNet model [23] to calculate the arrow's direction. A dataset based on Taiwanese road data was constructed for the purpose of training and evaluating the suggested method. A massive amount of computational work is required by the suggested method, yet it achieves an average mean accuracy of 67%.

In [24], a two-stage method was suggested for traffic light detection. There were two parts to the suggested method: detection and recognition. The detection step was built using the MobileNet v2 model [25] and the single-shot multi-box detection framework [26]. A suggested bespoke convolutional neural network was used for the recognition step. At that moment, the two-stage strategy outperforms the one-stage method.

While prior studies have explored traffic light detection using convolutional neural networks, most focus on high-resolution or simple environments, with limited attention to small, occluded, or low-resolution traffic lights in complex urban settings.

Existing models often lack optimization for real-time performance on embedded systems, making them impractical for deployment in real-world Intelligent Transportation Systems (ITS) and Advanced Driver Assistance Systems (ADAS). In effect, insufficient integration of advanced techniques like multi-scale feature extraction, dilated convolutions, and Soft Non-Maximum Suppression (Soft-NMS) to improve detection accuracy and reduce false positives. Also, there is a lack of comprehensive benchmarking against diverse datasets that represent real-world variability in traffic scenarios.

To address these gaps, this study proposes an optimized CNN framework specifically designed for real-time, high-accuracy traffic light detection in complex urban environments.

The traffic light detection problem has been studied extensively, but no reliable solution has been developed for practical implementation. Finding the optimal trade-off between speed and accuracy should be the primary goal of improving state-of-the-art performance. Here, we present a traffic light detector with an excellent trade-off between accuracy and speed. What follows is a more in-depth description of the strategy that has been suggested.

3 Proposed Approaches

The method for traffic light detection that has been suggested is described in this section. The feature extraction convolutional neural network is introduced first. The next section elaborates on the network for extracting features from many scales, and the last section describes the detection and recognition module.

In a convolutional neural network, the features extraction network is its strongest component. In picture processing, it is fundamental to the convolutional neural network's performance. In contrast to traditional approaches that relied on manually crafting features, the efficacy of automatic feature extraction straight from the input image was demonstrated. An existing model of a convolutional neural network, ResNet 101, is suggested for usage in this article [11]. One hundred and ten layers of feature extraction make up this neural network model. In the ILSVRC 2015 picture classification competition, it obtained a top-5 error rate of 4.61%. ResNet 101 deviates from the conventional convolutional layer approach by using residual blocks as an alternative. Making skip links between the block's input and output is the fundamental concept of residual blocks. The preceding block's output and input are therefore passed on to the following block. Fig. 1 shows a schematic of the ResNet101 residual block. Using this method, the computational complexity may be kept from exploding even when deep neural networks with over 100 layers are constructed.

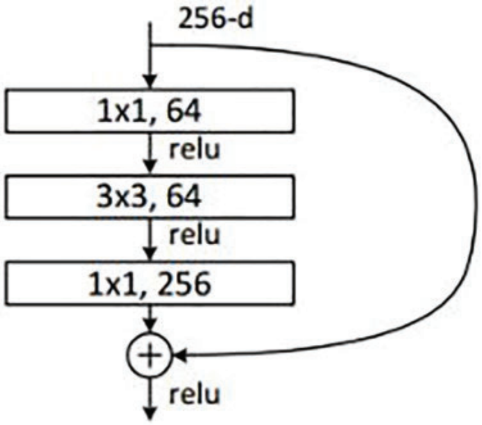

Figure 1: Residual block proposed for ResNet 101

It is feasible to concentrate on a wide receptive field in order to provide predictions for the picture categorization issues. However, zeroing in on the item's location's receptive field is essential for object detection. Given the relatively tiny size of the traffic light in relation to the input picture, the semantic information is either insufficient for traffic light recognition or may even vanish in the later layers of the feature extraction network. And there can be a variety of sizes of traffic lights in the supplied image. In order to identify traffic signals of varying sizes, it must extract features from feature maps of varying scales. Combining high-resolution and low-resolution feature maps to obtain high semantic information is critical for detecting minor traffic signals. In order to do this, we suggest modifying the classic feature pyramid network to create a multi-scale feature extraction network. Layers for convolution and upsampling make up the suggested multi-scale feature extraction network, which enables the integration of ResNet 101 layers.

The usage of upsampling layers in the architecture of the multi-scale feature extraction network will lead to severe errors in the convolutional neural network [27]. The upsampling layers of the multi-scale feature extraction network will not be able to reconstruct any traffic light with fewer than 16 pixels, assuming that ResNet 101 is trained with 4 downsampling layers. When features are pooled, their spatial structure is lost, and the connection between the traffic light's position and global features is damaged. We suggest replacing pooling layers with strided convolution layers [28] to fix the problem. Conventional convolution layers with a stride of 2 are known as strided convolution layers. In addition, to enhance the receptive field for detecting tiny traffic lights in high-resolution photos, the dilated convolution [29] was used in lieu of specific convolution layers. With a dilation ratio of 1, traditional convolution layers are referred to as dilated convolution layers. Dilated convolution with three distinct dilation ratios (D = 1, D = 2, and D = 3) is shown in Fig. 2.

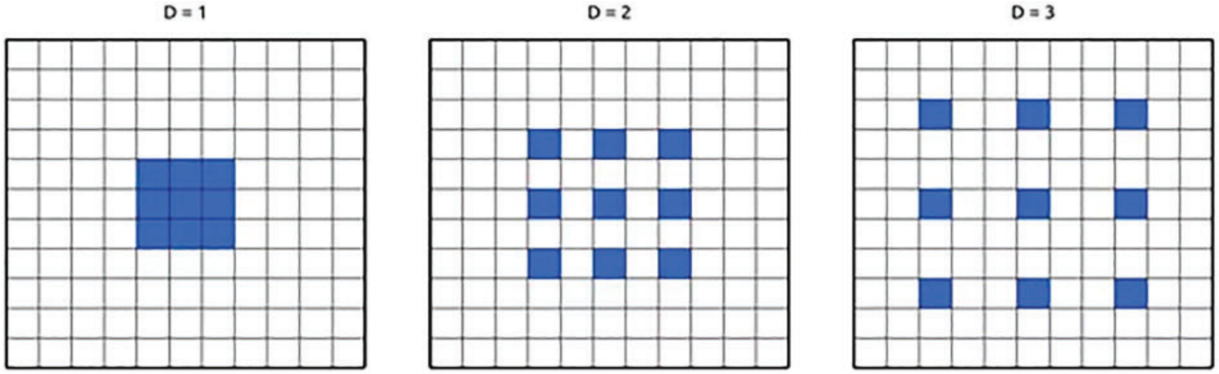

Figure 2: Dilated convolution layers with different dilation ratio

We use a dilated convolution with a dilation ratio of 2 and a D-weight of 4 for ResNet 101. Because of this, we obtain a bigger receptive field by utilizing 5 × 5 kernels for D = 2 and 15 × 15 kernels for D = 4, respectively, rather than 3 × 3 kernels. In conclusion, although dilated convolution exhibits an exponential growth in receptive field, conventional convolution layers have a linear correlation with the quantity of layers. The enhanced ResNet model using strided and dilated convolutions is summarized in Table 1.

Table 1: Improved ResNet 101 with strided convolution and dilated convolution

Layers	Name	Kernel	Stride	Filters	Output size
Convolution 1	S1	7 × 7	2	64	320 × 320
Convolution 2		3 × 3	2	64	160 × 160
3× residual block 1	S2	1 × 1 3 × 3 1 × 1	1 1 1	64 64 256	160 × 160
Residual block 2	S3	1 × 1 3 × 3 1 × 1	2 1 1	128 128 512	80 × 80
3× residual block 3		1 × 1 3 × 3 1 × 1	2 1 1	128 128 512	40 × 40
23× residual block 4 *dilated, D = 2	S4	1 × 1 3 × 3* 1 × 1	1 1 1	256 256 1024	40 × 40
3× residual block 5 **dilated, D = 4	S5	1 × 1 3 × 3** 1 × 1	1 1 1	512 512 2048	40 × 40

Starting with upsampling ResNet 101's final convolution layers, we merge the network's low-level high-resolution feature maps with its top-level high-sematic features to construct the multi-scale feature extraction network. A 1 × 1 convolution layer was utilized to guarantee a coupling of the ResNet 101 layers with the multi-scale feature extraction network. To generate region ideas for traffic signals, the region proposal network (RPN) [12] was put into action. All of the RPN's neural connections are convolutional. By checking a predetermined set of anchors at each feature map region, it produces region recommendations. The next step is to use a binary classifier to determine if an item is present in the tested region. If an object is found, the parameters for the predicted bounding box are generated. The suggested method for traffic light identification, which utilizes ResNet 101 and the multi-scale feature extraction network, is shown in Fig. 3.

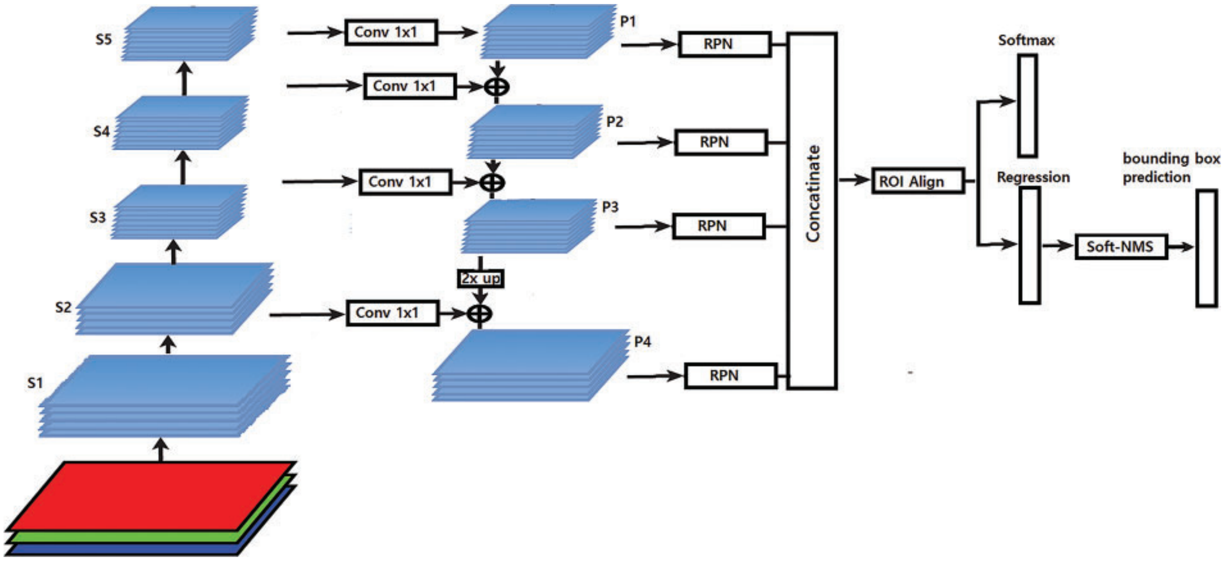

Figure 3: Proposed approach for traffic light detection

The ResNet 101 model's feature maps (S1, S2, S3, and S4) were subjected to a 1 × 1 convolution in order to construct the layers of the multi-scale feature extraction network. In order to construct the P1 layer, the channel dimension is reduced using a 1 × 1 convolution applied to the S5 feature maps. To get the P2, we take the feature maps from the preceding P1 and add them element by element, then run a 1 × 1 convolution on the S4 maps. Similarly, P3 is generated by adding elements of P2 and then performing a 1 × 1 convolution. P4 is created by applying a 1 × 1 convolution on S2 and then adding the upsampled P3 element-wise by a factor of 2. The feature maps P1, P2, P3, and P4 were subjected to a 3 × 3 convolution in order to mitigate the aliasing impact of the upsampling procedure.

Our statistics show that a 640 × 640 picture can only accommodate traffic lights no larger than 60 pixels. Most traffic signals have a maximum size that ranges from sixteen to thirty-six pixels. We recommend applying a convolution layer to feature maps P1, P2, P3, and P4 using a parameter optimization method including various scales and ratios in order to extract reliable information from such a small size. In this way, it is possible to get region suggestions that may be adjusted to the identified traffic lights' fluctuation range. The RPN is able to reduce processing time by calculating a lower number of region proposals at the final decision step, thanks to the suggested procedures, which enable it to create more trustworthy region proposals.

The RPN is designed as a fully convolutional network (FCN). This structure ensures that the operations for feature extraction, anchor box generation, and objectness scoring are conducted entirely through convolutional layers without any fully connected layers. This design aligns with the principles described in the Faster R-CNN framework by Ren et al. [12], where RPNs were introduced. Ren et al. [12] highlighted the efficiency and scalability of using convolutional layers exclusively in RPNs for generating region proposals.

The majority of RPN-based techniques used ROI pooling to combine the convolutional neural network's extracted feature maps with the RPN's region proposals. This allowed the feature maps to be sized according to the predicted bounding box's coordinate position. After that, fully linked layers receive the produced feature maps and process them further. There is a pixel-level classification difficulty in identifying a traffic light that fills less than 2% of the input picture. The recognition job considers each pixel value as essential information. A linear regression layer, which calculates floating-point values, yields the bounding box prediction; nevertheless, completely linked layers necessitate a fixed value. Consequently, two improvements are required in the ROI pooling layer. When identifying large items, the optimization does not affect the final judgment, but for tiny objects, the optimizations could cause the bounding box to lose its coordinate location. This might cause an increase in the false-positive rate by influencing the semantic resolution of detected pixels.

Instead of ROI Pooling, we suggest using ROI Align to set the sizes of feature maps before connecting them to completely linked layers. This should eliminate this issue. In terms of balancing detection efficiency with accuracy, the ROI Align method is tops. Prior to being mapped to the feature map, region suggestions are not quantized in the pictures. The floating-point value is held on the boundary of each of the n × n bins. The bilinear interpolation procedure was employed to determine the coordinate values of the locations of the k sample points. So, using the gradient descent approach, we can optimize the loss function L at the ROI align as Eq (1).

$$\frac{\partial L}{\partial x_i} = \sum_r \sum_j [d(x_i, x_i^* (r, j) < 1](1 - \Delta h)(1 - \Delta w)\frac{\partial L}{\partial y_{rj}} \tag{1}$$

$x_i^* (r, j)$ is a floating-point value referring to the coordinate of the sampling point calculated through the mapping function in the feed-forward process. Each point coordinate less than 1 and to different to $x_i^* (r, j)$ in the feature, the map must accept the gradient of y_{rj} returned by the same point. The d(.) function defines the distance between 2 points in the feature map. Δh and Δw are respectively the horizontal and vertical difference between the coordinate of 2 points x_i and $x_i^* (r, j)$. The ROI Align process is presented in Fig. 4.

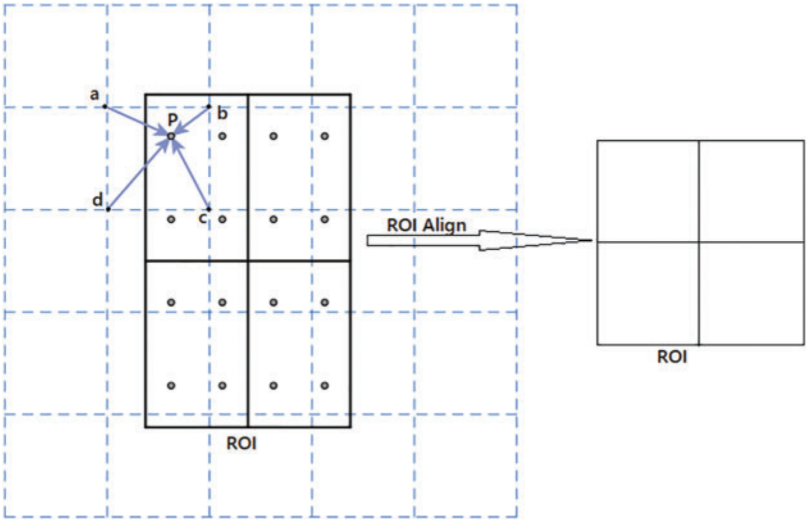

Figure 4: ROI align algorithm

Assuming that the sampling factor is 4 and each region proposal is divided into $n \times n$ bins and each bin is divided into 4 squares, P is a pixel in a certain square, and a, b, c, d are the corners of the grid around P, by applying the linear interpolation algorithm from 2 directions the 4 points pixels take the value of P.

Applying the non-maximum suppression (NMS) method to the output on the regression layer yields the final bounding box prediction. Finding the most appropriate detecting bounding box and removing inaccurate predictions is the primary function of this approach. The traditional NMS method uses a decreasing sorting of the predicted bounding boxes and then chooses the one with the greatest detection score. Any additional bounding boxes that overlap with the best one, according to a preset threshold, are removed. Because of the camera's perspective, using the NMS algorithm to identify traffic lights can result in the removal of the bounding boxes of other observed lights. We suggest switching from the NMS method to the Soft-NMS algorithm to address this issue. One key distinction between the NMS method and the Soft-NMS algorithm is that the latter gradually decreases the score until it finds the best prediction, rather than eliminating all bounding boxes with a fixed overlap value with the best bounding box. The projected bounding boxes (BI), their accompanying score (SI), and the final bounding box predictions (BF) are all taken into consideration. Here is how the Soft-NMS operates: The following steps are taken by the Soft-NMS: (1) compile all the scores in S; (2) choose the bounding box with the highest score, bf_i; (3) transfer the selected box from BI to BF; (4) recalculate the score of b_i if the IoU of a predicted bounding box b_i from BI and bf_i is greater than a predefined threshold t. The score function is smoothed using a Gaussian weight function. Accordingly, the attenuation function is used to recalcine the scores of neighboring bounding boxes, where the size of the overlap region between the boxes determines the function. The formula for the function that recalculates scores is calculated as Eq (2).

$$S_i = \begin{cases} S_i, IOU\left(bf_i, b_i\right) < t \\ 0, IOU\left(bf_i, b_i\right) \geq t \end{cases} \tag{2}$$

$$S_i = S_i e^{-\frac{IOU\left(bf_i, b_i\right)^2}{\sigma}}, \forall b_i \notin \mathrm{BF}$$

$IOU\left(bf_i, b_i\right)$ defines the overlap value between the highest score bounding box bf_i and a bounding box b_i. Using the continuous Gaussian function, boxes with no overlap keep it score but overlapped bounding boxes will be demonstratively guarded. Unlike traditional NMS, the Soft-NMS can avoid eliminating bounding boxes of partially occluded traffic lights by another bounding box. It results in improving the detection performance of the model.

To prove the efficiency of the proposed approach many experiments have been conducted and the achieved results prove the performance. More details are presented in the next section.

4 Experiments and Results

A hybrid system with an Intel i7 processor, 16 GB of RAM, and an Nvidia GTX 960 graphics processing unit would be ideal for our experimental needs. In many ways, the NVidia Drive AGX Pegasus is an inspiration for the planned system. In Table 2, we can see how the NVidia Drive AGX Pegasus stacks up against the suggested setup.

Table 2: Comparison between the proposed system and the NVidia Drive AGX pegasus

	Proposed system	**Nvidia drive AGX pegasus**
CPU	Intel i7, 8 cores	ARM, 8 cores
RAM	16 GB	16 GB
GPU	Nvidia GTX 960, 2084 cores	2 Xavier GPU, 2048 cores

With the help of the Nvidia acceleration libraries CUDA and cuDNN, the convolutional neural networks were built using the TensorFlow deep learning framework. For picture editing, we used the open cv library. The pre-trained weight from the ImageNet dataset was used to initialize the ResNet 101 model.

Optimizing the loss function was done using the gradient descent approach with momentum. From a configuration standpoint, we establish the following values: momentum = 0.9, learning rate = 0.001, and weights decay = 0.1. The attenuation technique was used to maximize the learning rate and select the optimal one. The output layers were started randomly with a standard deviation of 0.01 and mean values of 0. All other bounding boxes will be taken as trustworthy forecasts, except for the one with an IOU, which will be ignored if its value is less than 0.6.

Using the LISA traffic light dataset, we trained the suggested program. It's a freely accessible dataset that researchers use to evaluate various traffic light recognition methods and determine which ones are the most advanced. This dataset is made up of video frames that were captured in real-time while driving in San Diego, California, USA. The collection includes 113,888 annotated traffic signals and 43,007 frames overall. The scenes were recorded using a stereo camera that was positioned on top of a moving car. The weather and lighting conditions varied throughout the day and night. For testing and training purposes, we have deactivated the stereo functions and restricted ourselves to using only the left camera view. This LISA traffic light dataset contains photos with a 66° Field of View with a resolution of 1280 × 960. Separate training and testing sets were created from the dataset. The remaining footage serves as a test set, while thirteen clips shot throughout the day and five clips shot at night make up the training set. We take into account seven categories in the dataset: begin, warning, stop, go left, begin, warning left, stop left, and go forward.

We suggest using the average precision and the mean average precision as metrics to assess the efficacy of the suggested method. How good the samples that were accurately predicted are may be inferred from the precision. The suggested method's efficacy may be demonstrated using this assessment criteria.

Due to the restricted memory of the employed GPU, the suggested technique has been trained for 200,000 iterations with a batch size of 16 photos. We refined the loss function and got it down to 0.004. In Table 3, you can see the average accuracy for each class as well as the overall accuracy for the whole dataset. A mean average accuracy of 96.73% was reached by the suggested technique when assessed on the test set.

We used the same dataset settings to ensure a fair comparison with state-of-the-art studies. In Table 4, we can see how our results compare to those of previous studies using the LISA traffic light dataset. On the LISA traffic light dataset, the suggested strategy achieves better results than state-of-the-art approaches, as shown in Table 4.

Table 3: Achieved average precision per class and mean average precision

Traffic light class	Average precision (%)
Go	98.45
Warning	97.56
Stop	97.12
Go left	95.35
Warning left	96.84
Stop left	96.45
Go forward	95.39
Mean	**96.73**

Table 4: Comparison against existing works on the LISA traffic light dataset

Method	Mean average precision (%)
YOLO [19]	58.3
TTTL [30]	84.95
BSSNet-full-size & TLC3Net [31]	78.31
[32]	92.6
Ours	96.73

Our model achieves an accuracy of 96.73% which outperforms [32] by 4.13% in urban scenarios with complex backgrounds. This improvement can be attributed to our use of multi-scale pyramid feature maps and Soft Non-Maximum Suppression (Soft-NMS).

Unlike [30], which struggles with detecting small and occluded traffic lights, our method demonstrates a detection rate of 96.73% for these objects, highlighting the effectiveness of our dilated convolutions. The improved performance of our model is rooted in its architectural innovations.

By leveraging pyramid feature maps, the network captures context at multiple resolutions, significantly enhancing its ability to detect traffic lights in varying scales and orientations. This directly addresses limitations identified in other works.

The adoption of Soft-NMS mitigates issues of false positives, especially in densely populated urban environments, a recurring challenge in prior studies such as [19].

Our focus on parameter optimization ensures that the model operates seamlessly in real-world settings, unlike the computationally intensive approaches of [31].

The robustness of our framework in handling diverse environmental conditions underscores its potential for deployment in Intelligent Transportation Systems (ITS) and Advanced Driver Assistance Systems (ADAS). This scalability is a notable advancement over previous work, which often lacks real-world applicability.

With a processing time of 7 FPS, the suggested technique offers a decent balance between speed and accuracy. Depending on the vehicle's speed and the distance from the traffic light, this period might be thought of as real-time processing. The maximum speed in an urban context is 60 KM/H, which is 16 m/s.

The system requires a minimum of two frames per second to detect and identify the traffic light, considering the ten-meter distance between the car and the light. Thanks to the finished processing time, the system may function in real-time.

Our proposed convolutional neural network, utilizing multi-scale pyramid feature maps, has demonstrated remarkable efficiency based on the presented findings. The incorporation of multi-scale pyramid feature maps has significantly improved detection and classification accuracy. The method effectively detects and identifies small traffic signals even in complex backgrounds. The use of an ensemble of RPN networks has further enhanced the detection rate and expedited processing by generating highly confident region proposals, thereby reducing the number of regions requiring evaluation. This optimization has notably improved the performance of the traffic light detection application. With the availability of advanced hardware, the proposed approach is viable for practical implementation in ADAS systems.

5 Conclusion

Traffic light detection is one of the most critical applications of Advanced Driver Assistance Systems (ADAS). However, developing a reliable detector poses significant challenges due to complex backgrounds, small traffic lights, occlusion, and other factors. This work focused on designing an efficient and dependable traffic light detection system. To achieve this, we proposed a convolutional neural network (CNN) leveraging multi-scale pyramid feature maps for robust feature extraction.

The CNN incorporates residual links to enhance feature learning, while multi-scale pyramid feature maps are utilized to accurately detect small traffic lights against complex backgrounds. Regional proposal generation is facilitated by the Region Proposal Network (RPN), and ROI Align is employed to preserve pixel-level feature information and minimize the impact of quantization deviations. The bounding box selection and refinement processes are further optimized using Soft-NMS algorithms.

In testing, the proposed approach demonstrated exceptional performance, effectively recognizing and identifying small traffic signals in high-resolution images. The results highlight the cutting-edge functionality of our method. However, the model remains sensitive to extreme lighting conditions and adverse weather. Future work will focus on incorporating adaptive mechanisms and domain generalization techniques to overcome these limitations and further enhance system robustness.

Acknowledgement: The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University for funding this work through research group No. (RG-NBU-2022-1234).

Funding Statement: This research was funded by the Deanship of Scientific Research at Northern Border University, Arar, Saudi Arabia through research group No. (RG-NBU-2022-1234).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Yahia Said, Yahya Alassaf; data collection: Taoufik Saidani, Refka Ghodhbani; analysis and interpretation of results: Taoufik Saidani, Olfa Ben Rhaiem, Refka Ghodhbani; draft manuscript preparation: Yahia Said, Yahya Alassaf, Taoufik Saidani. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data will be made available on request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Xing J, Wei D, Zhou S, Wang T, Huang Y, Chen H. A comprehensive study on self-learning methods and implications to autonomous driving. IEEE Trans Neural Netw Learn Syst. 2024;1–20.
2. Pereira R, Barros T, Garrote L, Lopes A, Nunes UJ. A deep learning-based global and segmentation-based semantic feature fusion approach for indoor scene classification. Pattern Recognit Lett. 2024;179(14):24–30. doi:10.1016/j.patrec.2024.01.022.
3. Wei H, Zhang Q, Qin Y, Li X, Qian Y. YOLOF-F: you only look one-level feature fusion for traffic sign detection. Vis Comput. 2024;40(2):747–60. doi:10.1007/s00371-023-02813-1.
4. Lin KX, Cho I, Walimbe A, Zamora BA, Rich A, Zhang SZ, et al. Benefits of synthetically pre-trained depth-prediction networks for indoor/outdoor image classification. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision; 2023; Waikoloa, HI, USA. p. 360–9.
5. Zhao X, Cheah CC. BIM-based indoor mobile robot initialization for construction automation using object detection. Autom Constr. 2023;146(1):104647. doi:10.1016/j.autcon.2022.104647.
6. Zeng G, Wu Z, Xu L, Liang Y. Efficient vision transformer YOLOv5 for accurate and fast traffic sign detection. Electronics. 2024;13(5):880.
7. Bakirci M. Enhancing vehicle detection in intelligent transportation systems via autonomous UAV platform and YOLOv8 integration. Appl Soft Comput. 2024;164:112015.
8. Rezwana S, Lownes N. Interactions and behaviors of pedestrians with autonomous vehicles: a synthesis. Future Trans. 2024;4(3):722–45.
9. Yang Y, Lee YM, Madigan R, Solernou A, Merat N. Interpreting pedestrians' head movements when encountering automated vehicles at a virtual crossroad. Trans Res Part F: Traffic Psychol Behav. 2024;103:340–52.
10. Wang J, Chen Y, Gu Y, Yan Y, Li Q, Gao M, et al. A lightweight vehicle mounted multi-scale traffic sign detector using attention fusion pyramid. J Supercomput. 2024;80(3):3360–81. doi:10.1007/s11227-023-05594-5.
11. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016; Las Vegas, NV, USA. p. 770–8. doi:10.1109/CVPR.2016.90.
12. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell. 2017;39(6):1137–49. doi:10.1109/TPAMI.2016.2577031.
13. He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision; 2017; Venice, Italy. p. 2961–9.
14. Bodla N, Singh B, Chellappa R, Davis LS. Soft-NMS—improving object detection with one line of code. In: Proceedings of the IEEE International Conference on Computer Vision; 2017; Venice, Italy. p. 5561–9.
15. Jensen MB, Philipsen MP, Møgelmose A, Moeslund TB, Trivedi MM. Vision for looking at traffic lights: issues, survey, and perspectives. IEEE Trans Intell Transp Syst. 2016;17(7):1800–15. doi:10.1109/TITS.2015.2509509.
16. John V, Yoneda K, Qi B, Liu Z, Mita S. Traffic light recognition in varying illumination using deep learning and saliency map. In: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC); 2014; Qingdao, China: IEEE. p. 2286–91. doi:10.1109/ITSC.2014.6958056.
17. Behrendt K, Novak L, Botros R. A deep learning approach to traffic lights: detection, tracking, and classification. In: 2017 IEEE International Conference on Robotics and Automation (ICRA); 2017; Singapore: IEEE. p. 1370–7. doi:10.1109/ICRA.2017.7989163.
18. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016; Las Vegas, NV, USA. p. 779–88. doi:10.1109/CVPR.2016.91.
19. Lee E, Kim D. Accurate traffic light detection using deep neural network with focal regression loss. Image Vis Comput. 2019;87(8):24–36. doi:10.1016/j.imavis.2019.04.003.
20. Possatti LC, Guidolini R, Cardoso VB, Berriel RF, Paixão TM, Badue C, et al. Traffic light recognition using deep learning and prior maps for autonomous cars. In: 2019 International Joint Conference on Neural Networks (IJCNN) 2019; Budapest, Hungary: IEEE. p. 1–8. doi:10.1109/IJCNN.2019.8851927.

21. Yeh TW, Lin SY, Lin HY, Chan SW, Lin CT, Lin YY. Traffic light detection using convolutional neural networks and Lidar data. In: 2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS); 2019; Taipei, Taiwan: IEEE. p. 1–2. doi:10.1109/ISPACS48206.2019.8986310.
22. Redmon J, Farhadi A. YOLOv3: An incremental improvement. arXiv:1804.02767. 2018.
23. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. 1998;86(11):2278–324. doi:10.1109/5.726791.
24. Cai Y, Li C, Wang S, Cheng J. DeLTR: a deep learning based approach to traffic light recognition. In: International Conference on Image and Graphics; 2019; Beijing, China. p. 604–15. doi:10.1007/978-3-030-34113-8_50.
25. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. MobileNetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018; Salt Lake City, UT, USA. p. 4510–20. doi:10.1109/CVPR.2018.00474.
26. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. SSD: single shot multibox detector. In: European Conference on Computer Vision; 2016; Amsterdam, The Netherlands. p. 21–37. doi:10.1007/978-3-319-46448-0_2.
27. Dumoulin V, Visin F. A guide to convolution arithmetic for deep learning. arXiv:1603.07285. 2016.
28. Ayachi R, Afif M, Yahia S, Mohamed A. Strided convolution instead of max pooling for memory efficiency of convolutional neural networks. In: International Conference on the Sciences of Electronics, Technologies of Information and Telecommunications; 2018. p. 234–43. doi:10.1007/978-3-030-21005-2_23.
29. Yu F, Koltun V, Funkhouser T. Dilated residual networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017; Honolulu, HI, USA. p. 472–80. doi:10.1109/CVPR.2017.75.
30. Lu Y, Lu J, Zhang S, Hall P. Traffic signal detection and classification in street views using an attention model. Comput Vis Media. 2018;4(3):253–66. doi:10.1007/s41095-018-0116-x.
31. Kim HK, Yoo KY, Park JH, Jung HY. Traffic light recognition based on binary semantic segmentation network. Sensors. 2019;19(7):1700. doi:10.3390/s19071700.
32. Vitas D, Tomic M, Burul M. Traffic light detection in autonomous driving systems. IEEE Consum Electron Mag. 2020;9(4):90–6. doi:10.1109/MCE.2020.2969156.