IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 40, No. 6, December, 2023, pp. 2577-2586
Journal homepage: http://iieta.org/journals/ts

Integrating Recurrent Neural Networks with Convolutional Neural Networks for Enhanced Traffic Light Detection and Tracking

Riadh Ayachi[1], Mouna Afif[1], Yahia Said[2], Mohamed Atri[3*], Abdessalem Ben Abdelali[1]

[1] Laboratory of Electronics and Microelectronics, Faculty of Sciences, University of Monastir, Monastir 5019, Tunisia
[2] Electrical Engineering Department, College of Engineering, Northern Border University, Arar 91431, Saudi Arabia
[3] College of Computer Science, King Khalid University, Abha 61421, Saudi Arabia

Corresponding Author Email: Matri@kku.edu.sa

https://doi.org/10.18280/ts.400620

Received: 30 January 2023
Revised: 7 July 2023
Accepted: 10 August 2023
Available online: 30 December 2023

Keywords:
convolutional neural network, recurrent neural network, traffic light detection, and tracking, mobile systems

ABSTRACT

Detecting and tracking traffic lights within an urban landscape, from a camera affixed to a moving vehicle, poses a substantial challenge. It necessitates the development of a reliable traffic light detection and tracking system that strikes an optimal balance between precision and real-time processing capabilities. Driven by this imperative, this study puts forward a novel traffic light detection methodology that harnesses the synergistic power of convolutional neural networks (CNN) and recurrent neural networks (RNN). The CNN, renowned for its self-learning capability, is employed for effective feature extraction, while the RNN is leveraged to retain information from video frames, thereby facilitating more reliable predictions. These two types of neural networks are amalgamated into a singular, expansive neural network, working in unison. The input video is initially processed through the CNN, leading to the extraction of spatial features. Subsequently, it is fed into the RNN for the extraction of temporal features. The final determination of the traffic light state is based on the combined usage of these extracted features. The addition of temporal features significantly bolsters overall performance without escalating computational complexity. The proposed methodology was trained and evaluated using two distinct datasets: the DriveU Traffic Light Dataset and the Bosch Small Traffic Lights Dataset, both of which include video frames captured by a camera mounted on a moving vehicle. Demonstrating high detection precision while operating in real-time, the proposed approach exhibits significant potential for practical application.

1. INTRODUCTION

Advanced driver assistance systems (ADAS) and autonomous vehicles have transitioned from the realm of fantasy into practical reality, marking significant strides in vehicular technology. A plethora of perception-based technologies has been seamlessly incorporated into contemporary vehicles, empowering them to perform routine or repetitive tasks such as highway navigation and parking with ease. However, the complex dynamics of urban driving present a unique set of challenges that necessitate innovative solutions. The technologies deployed must effectively emulate human perception and interaction with diverse elements such as pedestrians, vehicles, and bicycles.

One notable component in this technological ensemble is traffic light detection, which holds paramount importance for both ADAS and autonomous vehicles. It interprets the state of traffic lights, thereby informing decision-making processes. Despite the inherent challenges such as occlusion, varying perspectives, weather conditions, and the similarity with car backlights, a robust and reliable traffic light detection system is indispensable for real-world driving scenarios.

The objective of developing a Traffic Light Detection System for ADAS is to augment the capabilities of autonomous vehicles, equipping them with real-time data on traffic lights. ADAS technologies aim to supplement human drivers and enhance overall road safety. The following encapsulates the primary goals of creating a Traffic Light Detection System for ADAS:

Traffic Light Recognition: The system is designed to accurately detect and recognize traffic lights in real-time, providing crucial information for autonomous vehicles to understand traffic light states and make appropriate navigational decisions.

Enhanced Safety: Recognizing and detecting traffic lights allows the system to guide autonomous vehicles in navigating intersections and responding to signal changes, thereby minimizing the risk of collisions with other vehicles or pedestrians.

Traffic Efficiency: The Traffic Light Detection System for ADAS can optimize traffic flow by providing precise information about signal timings. It enables vehicles to adjust their speed and plan their movements in anticipation of traffic light changes, reducing congestion and promoting overall traffic efficiency.

Decision Making and Planning: Armed with traffic light

information, the system assists autonomous vehicles in making informed decisions such as when to halt, slow down, or proceed through an intersection. It can also facilitate route planning by considering traffic light sequences to optimize travel paths.

Reducing Driver Workload: By automating the detection and interpretation of traffic lights, the system diminishes the cognitive load on human drivers in semi-autonomous vehicles, allowing them to concentrate on other aspects of driving and reducing the likelihood of human errors.

Integration with ADAS Systems: The Traffic Light Detection System can be integrated with the broader ADAS framework, working in conjunction with other sensors such as cameras, lidar, or radar to provide a comprehensive perception of the environment, thereby enhancing the overall capabilities of the ADAS system.

In essence, the primary intention and goals of developing a Traffic Light Detection System for Advanced Driver Assistance Systems (ADAS) are centered around enhancing safety measures, optimizing traffic efficiency, aiding in decision-making processes, and relieving driver workload. By accurately identifying and interpreting traffic lights, the system plays a pivotal role in the progression of autonomous driving technology, fostering the creation of safer and more efficient transportation systems.

The conundrum of traffic light detection within Advanced Driver Assistance Systems (ADAS) hinges on the precise and reliable identification of traffic light states in real-time. This is crucial in assisting autonomous vehicles in making apt driving decisions. The problem emerges from the complexity and variability inherent to traffic light scenarios, which encompass variations in lighting conditions, occlusions, and the presence of other objects within the scene. The challenge lies in devising robust and efficient methodologies capable of detecting and recognizing traffic lights, thereby ensuring the safety and efficiency of autonomous vehicles at intersections.

The research domain of traffic light detection in Advanced Driver Assistance Systems (ADAS) integrates disciplines such as computer vision, machine learning, and sensor fusion techniques. Computer vision algorithms are enlisted to process image or video data captured by vehicular-mounted cameras, while machine learning techniques are harnessed to train models that can accurately detect and classify traffic lights. Sensor fusion, on the other hand, involves the amalgamation of data from multiple sensors, including cameras, lidar, and radar, to bolster the reliability and accuracy of traffic light detection.

Within the realm of traffic light analysis, the conventional paradigm involves searching for the traffic light within the image, detecting it, and subsequently discerning its state and tracking its position. Recently, self-learning models such as Convolutional Neural Networks (CNN) [1] and Recurrent Neural Networks (RNN) [2] have been successfully deployed to tackle these tasks. A myriad of detection models have been proposed, including but not limited to Faster R-CNN [3], You Only Look Once (YOLO) [4], Single-Shot MultiBox Detection (SSD) [5], and Feature Pyramid Network (FPN) [6]. Most of these detection models have been utilized for traffic-related tasks, but the results achieved often fall short for real-world driving scenarios. The models are typically either accurate or fast, but for safety purposes, an optimal balance between speed and accuracy is essential.

The recent success of deep learning models for computer vision tasks [7, 8], inclusive of traffic-related tasks [9, 10], has catapulted performance to a new level. This boost can be attributed to the development of deep neural networks and their training on expansive datasets. To achieve high reliability for real-world driving scenarios, many optimizations must be considered. In a bid to construct a robust traffic light detection system, numerous works based on deep learning models have been proposed. These approaches typically enhance either the accuracy or the processing speed. Striking a delicate balance between accuracy and processing speed is a formidable challenge that must be surmounted to establish a robust traffic light detection system for autonomous vehicles.

Guided by the techniques proposed, researchers continually strive to develop traffic light detection systems that are accurate, reliable, and capable of real-time operation. These methodologies ensure that autonomous vehicles can effectively perceive and interpret traffic lights, thereby facilitating safe and efficient navigation at intersections. The relentless pursuit of research and improvements in this field contributes to the evolution of advanced ADAS systems, inching us closer to the ultimate goal of fully autonomous vehicles.

In our investigation into the task of traffic light detection, we introduce a novel hybrid neural network that leverages the strengths of Convolutional Neural Networks (CNN) [1] and Recurrent Neural Networks (RNN) [2]. This hybrid model is designed to take advantage of the feature extraction prowess inherent in convolutional neural networks, and the memory retention capabilities of recurrent neural networks. The amalgamation of these two neural networks has resulted in the desired performance, striking an optimal balance between speed and accuracy.

In our proposed model, the features extracted from the Convolutional Neural Network are concatenated and subsequently passed onto a Recurrent Neural Network. This hybrid neural network is predicated on an object detection model, followed by a Recurrent Neural Network. We employed the Single Shot MultiBox Detector (SSD) with MobileNet v2 [3] as the backbone model for detection, and Gated Recurrent Units (GRU) [4] as a Recurrent Neural Network at the decision-making level. The detection model was initially pre-trained on the COCO [11] dataset and subsequently fine-tuned for traffic light detection. The memory capabilities of the GRU were utilized to retain the previous state of the traffic light, which proves beneficial in predicting the subsequent state. For instance, if the current state is red or green, the next state will likely be orange, and if the current state is orange, the next state will likely be red or green. This information contributes to more accurate predictions.

Generally, the performance of deep learning models can be significantly amplified when larger datasets are utilized. Additionally, data collected under unconstrained conditions can augment the generalization power of the model. For traffic light detection, several datasets have been proposed, with the most recent ones being the DriveU Traffic Light Dataset [5] and the Bosch Small Traffic Lights Dataset [6]. Both datasets comprise a set of annotated video frames recorded under real-world conditions including day and night, sunny and cloudy weather, etc. Our proposed model, trained and evaluated on these datasets, achieved high performance, outperforming the current state-of-the-art.

The crux of this endeavor lies in developing a hybrid neural network for traffic light detection in video streams. In this hybrid neural network, features were extracted from various

levels of convolutional layers and fed into the recurrent neural network. This strategy proved advantageous for the task at hand, as it facilitated the seamless integration of spatial and temporal features. As corroborated by our experimental results, the proposed hybrid neural network achieves superior performance in terms of both accuracy and processing speed, surpassing the existing state-of-the-art works.

The key findings of the proposed hybrid neural network include:

- The implementation of a detection model predicated on a convolutional neural network.
- The fusion of features at different levels, which significantly bolstered the performance of the traffic light detection system.
- The deployment of Gated Recurrent Units (GRU) at the decision-making level, which amalgamated low, mid, and high-level features for precise prediction.
- The evaluation performed using two distinct datasets, which attested to the efficiency of the proposed network.

The remainder of the paper is structured as follows: Section 2 is devoted to related works. The proposed approach is described in detail in Section 3. Section 4 presents and discusses the experimental results. Finally, Section 5 draws conclusions from the study.

2. RELATED WORKS

Traffic light detection is a critical component for Advanced Driver Assistance Systems (ADAS) and autonomous vehicles. To establish a reliable traffic light detection system, numerous methodologies have been proposed, particularly recent ones that leverage deep learning techniques. For a more holistic review of traffic light detection, readers are encouraged to refer to studies [11, 12]. Deep learning and neural networks have been successfully deployed to enhance the performance of traffic light detection systems.

In this vein, Kulkarni et al. [13] proposed a traffic light detection and recognition system underpinned by the Region-Based Convolutional Neural Network (R-CNN) [14] with the Inception v2 backbone [15]. They utilized the Selective Search method [16] for region proposal. Each proposed region underwent processing by Inception v2, and a Support Vector Machine (SVM) classifier was employed for traffic light recognition. This method was tested on the Indian Traffic Lights Dataset, and it achieved low accuracy and slow processing speed.

Another traffic light detection methodology for autonomous cars was put forth in the study [17]. This was predicated on Adaptive Thresholding [18] and a modified version of the VGG-16 Convolutional Neural Network [19]. The input data underwent preprocessing, and the region of interest was proposed using Adaptive Thresholding and morphological operations. Further processing was performed using the VGG-16 Convolutional Neural Network for traffic light detection and recognition. The proposed method was trained and tested on the LISA Traffic Light Dataset. It achieved an accuracy of 89.6% for the recognition task and an accuracy of 92.67% for the detection task. However, the processing time of the proposed method was exceedingly slow, rendering it unsuitable for real-world application.

Ouyang et al. [20] introduced a traffic light detection system that leverages a combination of a heuristic detector module and a convolutional neural network classifier. The heuristic detection module was used to generate candidate regions potentially containing traffic lights, while the convolutional neural network was employed to classify each proposed candidate region. The proposed approach was trained and evaluated on several public traffic light datasets, resulting in a rather low recall of 31.4% due to the generation of many false-negative samples.

The YOLOv3 [21] model was integrated with prior maps for traffic light detection in the study [22]. YOLOv3, an acclaimed state-of-the-art object detection model, was initially trained on the MS COCO dataset [23]. The transfer learning technique was applied to the YOLOv3 model to repurpose it for traffic light detection. This model was then fine-tuned on two datasets, namely the DriveU Traffic Light Dataset and the LISA Traffic Light Dataset. The output of the detection model was synthesized with prior maps to select the relevant traffic light. The approach was evaluated on a custom video sequence, revealing that the proposed method achieved acceptable accuracy (60% to 80%) across different datasets, albeit with a quite slow processing speed.

Feng et al. [24] proposed a multi-scale attention network for traffic light detection. This multi-scale network comprised a base neural network, an attention module, and a detection module. ResNet 101 [25] served as the base network for feature extraction. Three attention modules at different scales were deployed to combine low-level features from the base network with layers from the up-sampled layers. The attention module's primary function was to construct feature maps laden with rich information for traffic light detection. The detection module consisted of three stages, drawing inspiration from the YOLOv3 model [21]. The proposed method was trained on public datasets such as LISA Traffic Light Dataset and Bosch Small Traffic Lights Dataset, and then evaluated on a custom-made dataset. The resulting performance indicated a low accuracy of 41.17% on the proposed dataset, with an acceptable processing time of 45 ms.

Wang et al. [26] augmented the YOLO v4 model [27] for traffic light detection. The model was enhanced to detect small-sized traffic lights by improving the bottom of the backbone to extract more relevant features. Furthermore, uncertainty prediction was introduced to improve the bounding box prediction, based on a Gaussian model applied on the coordinates. The proposed improvement bolstered the detection accuracy compared to the original YOLO v4.

Another traffic light detection system was proposed that integrated handcrafted features and the YOLO model [28]. A compilation of handcrafted features such as color, shape, and texture were fused based on an integral channel feature. Subsequently, the fused features were injected into the YOLO model to detect the state of the traffic light. The proposed system was evaluated on the Bosch Small Traffic Light Dataset and achieved acceptable results.

Most of the existing methods grapple with low precision or slow processing speed, which are not suitable for real-world applications that demand high precision and real-time processing. To address these challenges, we propose a novel hybrid neural network that strikes an optimal balance between precision and processing speed. More details about the proposed method are presented in the subsequent section.

3. PROPOSED APPROACH

Considering the requirements of traffic light detection and tracking and the different features collected using different neural network models, we propose to combine convolutional neural networks and recurrent neural networks to detect and track traffic lights in video. The CNN was used due to its power in extracting spatial features that enable a high detection precision. The RNN has deployed thanks to its ability in extracting temporal features that enable the tracking of the traffic light. The final prediction is generated based on the combination of the spatial and temporal features. The proposed approach is composed of an object detection model and a recurrent neural network that predict the location and the state of the traffic light by fusing different predictions. As a detection model, we propose to use the SSD model with the mobile v2 backbone. GRU network was used for the decision-making level. Figure 1 present a detailed flowchart of the proposed approach for detecting and tracking traffic based on the combination of the SSD model based on the mobileNet backbone and the GRU network.

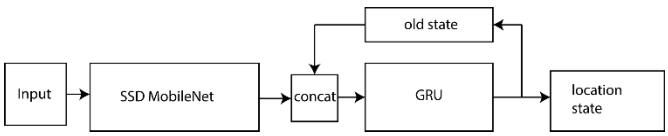

Figure 1. Flowchart of the proposed approach for detecting and tracking traffic light

Based on the concise requirements of object detection, the SSD was deployed due to its high detection accuracy and real-time processing. Besides, we integrated a lightweight backbone to make the proposed approach suitable for mobile devices. Then, the GRU was attached to extend the tracking power of the network and enhance the final prediction accuracy. The final decision is generated after processing every 10 frames and fusing the decisions over those frames. The pipeline of the proposed approach is presented in algorithm 1.

Algorithm 1

```
Initializing the input frames from the camera
For frames <= 10, frames ++, do
  If SDD does not detect traffic light then
    Final prediction = no traffic light
  Else
    SSD predict the location of the traffic light
    SSD predict the state of the traffic light
    GRU track the traffic light
    GRU predict the state of the traffic light
    Final prediction = location of the traffic light
    Final prediction = state of the traffic light
  End If
End For
```

To balance the performance and the computation complexity, a balancing index was proposed based on the accuracy improvements of different models. The balancing index can be computed as (1).

$$B_x = a_h - \max(a_{SSD}, a_{GRU}) \tag{1}$$

where, a_h is the accuracy of the hybrid model, a_{SSD} is the accuracy of the SSD model and a_{GRU} is the accuracy of the GRU network. To maximize the accuracy improvement and achieve the best balance, relation in (2) must be respected.

$$\max_{a_{SSD}, a_{GRU}} B_x = \frac{(1-2m_{SSD}+\varepsilon)^2}{4+ m_{SSD}+\varepsilon} \tag{2}$$

where, m_{SSD} is the miss rate of the SSD model and ε is a positive calibration variable smaller than one. The best condition can be achieved by respecting the condition in (3).

$$a_{SSD} = a_{GRU} = (1 - 2m_{SSD} + \varepsilon)/2 \tag{3}$$

To prove the condition presented in (3), we analyzed the accuracy improvement for different conditions. The accuracy improvement can be computed as (4).

$$\begin{cases} -a_{SSD}\, a_{GRU} + a_{SSD} + a_{SSD}\varepsilon + (1-2a_{SSD})m_{GRU} - \varepsilon, a_{SSD} \geq a_{GRU} \\ (\varepsilon - a_{GRU} - 2m_{GRU})a_{SSD} + m_{GRU} + a_{GRU} - \varepsilon, a_{SSD} < a_{GRU} \end{cases} \tag{4}$$

Since $-a_{SSD} \leq 0$ and $\varepsilon - a_{GRU} - 2m_{GRU} < 0$ for both conditions of the accuracy improvements. The balancing index reach each maximum for $a_{SSD} = a_{GRU}$. The maximum value of the balancing index can be computed as (5).

$$\max_{a_{SSD}, a_{GRU}} B_x = -\left(a_{SSD} - \frac{(1-2m_{SSD}+\varepsilon)}{2}\right)^2 + \frac{(1-2m_{SSD}+\varepsilon)^2}{4} + m_{SSD} - \varepsilon \tag{5}$$

The maximum value of the balancing index is a quadratic polynomial with a_{SSD} as the main parameter and when the condition in (4) establishes, the maximum value in (2) is achieved.

The model fusion method achieves a good accuracy improvement for the condition $0 < m_{GRU} < \varepsilon < 0.3$. knowing that $m_{SSD} \leq m_h$ (m_h is the miss rate of the hybrid model) and to achieve the highest balance between the performances and the computation complexity, we fix $m_{SSD} = 0.05$ and $\varepsilon = 0.1$. considering the conditions mentioned earlier, the maximum accuracy improvement can be achieved for $a_{SSD} = a_{GRU} = 0.5$. Based on the analyses performed in the precedent formula, the improvement of the miss rate can be computed as (6).

$$\begin{gathered} B_m = m_{SSD} - m_h \\ B_m = m_{SSD}\,(a_{GRU} - \varepsilon) - (1 - 2m_{SSD})m_{GRU} \\ B_m = 0.5m_{SSD} - 0.05 \end{gathered} \tag{6}$$

where, m_{GRU} is the miss rate of the GRU network. Considering the formula presented in (7), the miss rate of the hybrid model is smaller than the miss rate of the SSD model. By summarizing the theoretical analyses presented above, the best balancing index can be achieved for the following numerical conditions in (7).

$$\begin{gathered} m_{SSD} \approx 0.05, m_{GRU} > 0.1 \\ a_{SSD} = a_{GRU} = 0.5, \varepsilon \approx 0.1 \end{gathered} \tag{7}$$

In this work, we looked for achieving the best balance between performances and computation complexity while considering the available computation resources. Thus, we combined the mentioned neural network models to perform the traffic light detection task.

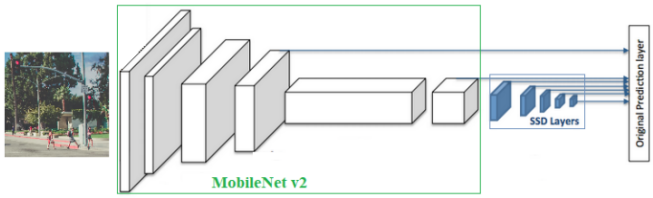

Figure 2. Architecture of the SSD model

Generally, the SSD model is composed of a backbone model, MobileNet v2 in our case, and SSD layers which are based on convolution layers. Figure 2 presents the architecture of the SSD model.

The SSD model was designed for real-time object detection. It uses predefined anchors to eliminate the need for a region proposal mechanism. Thus, it speeds up the processing time but drops the accuracy. To recover the accuracy, the SSD model applies some techniques such as multi-scale features and computes the size of the predefined anchors based on the input data. Those techniques allow the recovery of the accuracy and the maintenance of the real-time processing speed using lower resolution images compared to other object detection models.

To detect objects, the SSD model computes the location parameters and the class score using 3×3 convolution filters. Each computed filter generates N+4 channels where N is the number of classes and 4 is the bounding box parameters (x, y, h, w). Then it predicts each location. The SSD model uses multi-scale feature maps to detect objects of different sizes. Low-resolution feature maps are used to detect small objects and high-resolution feature maps are used to detect large objects. Figure 3 presents an illustration of the use of low-resolution and high-resolution feature maps.

The five SSD layers are considered multi-scale feature maps for object detection. As illustrated in Figure 2, six predictions are made by the SSD model where the first one is from the middle of the backbone network and the five SSD layers. The use of multi-scale feature maps significantly improves accuracy. Besides, the choice of predefined anchors has a big role in the improvement of accuracy.

Instead of using a random box and optimize them in the training process, the SSD model carefully selects the predefined anchors using the k-means cluster algorithm. The data and its annotation were fed to a k-men cluster then the sizes of the ground truth bounding box were clustered to get a close overview of the desired sizes. The centroid of each cluster was considered as a predefined anchor. The SSD model uses four or six predefined anchors based on the input data.

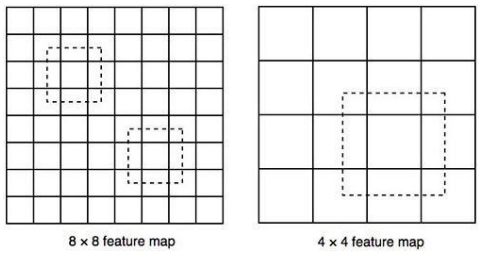

Figure 3. High-resolution (left) vs low-resolution (right)

The predefined anchors are chosen manually and this is a very sensitive step that must be performed carefully. The SSD model assigns each feature map of the multi-scale feature maps with a scale value. Feature map of the backbone gets the smallest scale value 0.1 then this value increases linearly until achieving its maximum 0.9 at the last feature map. To compute the high and the width of the predefined anchor, the scale value and the aspect ratio were combined. The high (h) and the width (w) of the predefined anchor are computed as (8).

$$h = scale \times \sqrt{aspect\ ratio}$$
$$w = \frac{scale}{\sqrt{aspect\ ratio}} \quad (8)$$

In real life, objects have a fixed shape of different sizes. In this work, we propose to detect traffic lights with having a fixed rectangular shape with a vertical or horizontal orientation and different sizes based on the distance separating the acquisition camera from the target traffic light. Figure 4 presents an illustration of traffic lights at a different orientation. Also, traffic lights do not occupy more than 10% of the total size of the image. So, after computing the k-mean cluster on the proposed datasets, we get six predefined anchors for each dataset where three are used to detect horizontal traffic lights and the three others are used to detect vertical traffic lights.

To generate bounding box prediction, the SSD model introduces the matching strategy which classifies predictions as positive or negative matches. Only positive matches are sent for further processing. If the IoU (intersection over the union of ground truth box and a redefined anchor) value is greater than 0.5 then the prediction is considered as a positive match. Otherwise, it is a negative match. This matching strategy ensures to predict shapes close to the predefined anchors. Thus, the predictions are more diverse and the training is fast and stable.

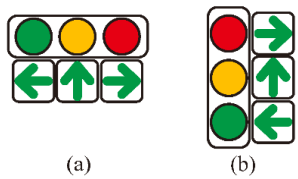

Figure 4. (a) Horizontal traffic light, (b) Vertical traffic light

The detection model was based on the MobileNet v2 backbone which is a lightweight convolutional neural network model designed for implementation on mobile embedded devices. The MobileNet achieves high accuracy with a minimum of computation resources due to the use of inverted residual blocks. An inverted residual block is composed of a 1×1 convolution with a ReLU6 activation function followed by a 3×3 depthwise convolution (Dwise) and 1×1 convolution without any non-linear activation function. The ReLU6 is an activation function that transforms any negative activation to 0 and any activation greater than 6 to 6 and maintains other values. This activation function was used to prevent the weight from becoming too high to optimize memory usage. Figure 5 presents the inverted residual blocks architecture. The main idea behind removing the non-linear activation function at the output level of the inverted residual block was as they claimed [7] that the use of the ReLU6 again in the block will make the deep network act as a linear classifier at the non-zero part of the output domain.

For the downsampling process, the MobileNet uses similar blocks to the inverted residual blocks but without residual connection and with a stride of 2. As proved in the study [25], using stride convolution instead of max pooling is more efficient for convolutional neural networks designed for

embedded implementation.

The MobileNet achieves high performance in many applications such as image classification, object detection, and instance segmentation using fewer computing resources compared to state-of-the-art models. Based on the advantages of the MobileNet for embedded implementation, we propose to use it as a backbone for the SSD model for traffic light detection.

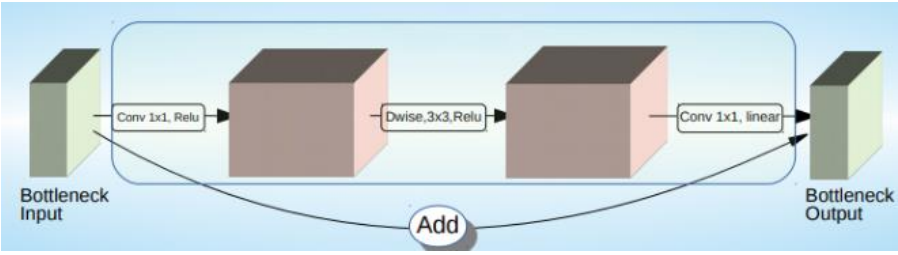

Figure 5. Inverted residual block

After detailing the detection model, we will describe the decision-making level based on the GRU network. The GRU is the last generation of recurrent neural networks. It is composed of a reset gate and an update gate. It used the hidden state to transfer information. An illustration of the GRU is presented in Figure 6. The update gate is used to decide what information to eliminate and what information to store. The reset gate is used to decide on the amount of past information to forget. The GRU has few parameters to learn. So, it can be processed fast for real-time processing and grantee high performance.

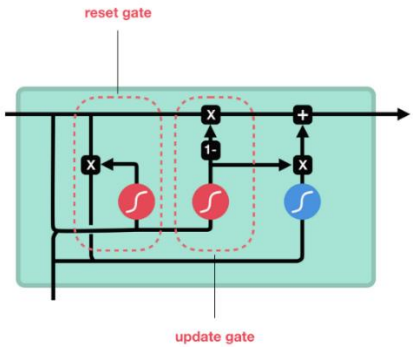

Figure 6. Gated recurrent unit (GRU)

To make predictions based on GRU, we propose a two layers network where each layer is composed of 1280 GRU. The proposed GRU network is illustrated in Figure 7. The features extracted from the middle layer of the backbone network and the features extracted from the SSD layers are concatenated and fed to the GRU network. The GRU network store information from consecutive frames and use them to generate predictions. The state of the traffic light can be predicted from more than one sequence for more trusted predictions. The old state of the traffic light can be used as an important feature. Figure 6 present the proposed GRU network.

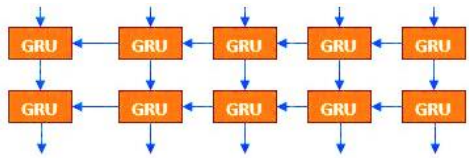

Figure 7. Proposed GRU network for decision-making level

This helps to reduce prediction probabilities and achieve better performance. The current state can be predicted and the next state can estimate based on the current state. As a result, the processing time can be reduced and the prediction accuracy can be enhanced. At the experimental results, those guesses will be proved with empirical results.

The proposed traffic light detection system is based on two components. The first one is the object detection model which is the SSD model and the second one is a recurrent neural network which is the GRU network. The proposed model for traffic light detection is illustrated in Figure 8. The combination of those components allows detecting traffic lights in videos at real-time processing. The SSD model with MobileNet backbone was used for feature extraction and to make predictions and the GRU network was used to store temporal information for better predictions. The MobileNet was used to process the input videos in real-time with low computation complexity. The SSD model improved the detection accuracy using a set of techniques such as the matching strategy, the multi-scale feature maps, the use of convolution layers instead of fully connected layers to generate predictions, and other techniques to reduce the computation complexity.

In this work, we the combination of spatial and temporal features for traffic light detection and tracking. Compared to existing methods, the proposed approach has many advantages such as the lightweight model based on the mobileNet and the use of the GRU network for final predictions which consider the state change sequence.

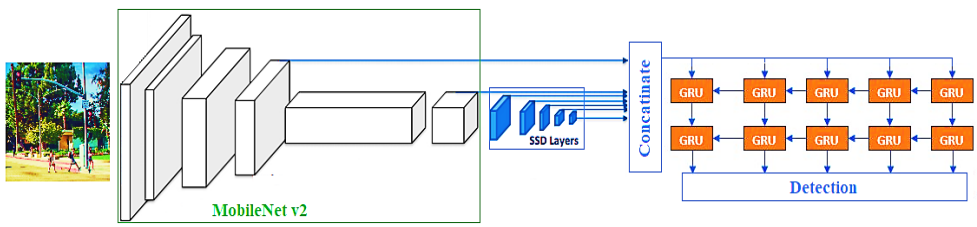

Figure 8. Proposed end-to-end convolutional neural network recurrent neural network for traffic light detection

4. EXPERIMENT AND RESULTS

In this section, we provide an overview of the experimental environment used for the implementation of the proposed approach then the achieved results will be presented and discussed.

The experimental environment used for the implementation of the proposed approach is composed of a desktop with a Linux operating system equipped with an Intel i7 CPU with 32 GB of RAM and an Nvidia GTX960 GPU with 2048 CUDA cores and 4 GB of memory. The proposed model was developed based on the TensorFlow deep learning framework. The CUDA library was used for GPU acceleration and the open cv library was used for image manipulation. To train and evaluate the proposed approach, we propose to use two challenging datasets. The first is the DriveU traffic light dataset [5] and the second is Bosch Small Traffic Lights Dataset [6].

The DriveU traffic light dataset was designed to provide a huge number of traffic lights with a focus on high diversity in the visual appearance of the collected samples. The dataset was collected by recording videos in 11 German cities with a frequency of 15 FPS using a camera of 2 megapixels. It contains 232,039 annotated traffic lights from 344 labels by taking into account the state, the orientation, the relevance, pictogram, number of units, and many others.

Bosch Small Traffic Lights Dataset was designed for tracking, detecting, and classifying traffic lights in a real environment. The dataset contains more than 13427 images at a resolution of 1280×720. The images were collected using an RGB camera in El Camino Real in the San Francisco Bay Area in California. 24000 traffic lights were labeled using bounding boxes with the identification of the state of each traffic light. At the testing process, only 4 labels (red, yellow, green, off) were considered.

As evaluation metrics, we propose to use different metrics for different datasets. For the DriveU traffic light dataset, the recall (R) was defined as an evaluation metric. The recall is the true positive rate which provides an idea on the percentage of total relevant outputs correctly predicted. The recall is defined as (9).

$$R= TP/(TP+FN) \quad (9)$$

where, *TP* is the true positive and *FN* is the false negative.

For the Bosch Small Traffic Lights Dataset, the mean average precision was used as an evaluation metric. The precision provides information on the percentage of the relevant predictions. The precision is defined as (10). The mean average precision (mAP) is the sum of the average precisions of all classes divided by the number of classes.

$$P = TP/(TP+FP) \quad (10)$$

where, *TP* is the true positive and *FP* is the false positive.

The defined evaluation metrics were proposed to perform a fair comparison against state-of-the-art models trained and tested on the same datasets used in this work. The performance of the traffic light detection system was evaluated using the precision of relevant state prediction for each input frame. The confusion matrix that computes the difference between the target state and the predicted state was reported.

The proposed model was trained for 50000 iterations on the Bosch Small Traffic Lights Dataset and for 100000 iterations for the DriveU traffic light dataset because of its large amount of training data. The SSD model was pre-trained on the MS COCO dataset then it was finetuned on the proposed datasets. The evaluation of the proposed model based on the proposed evaluation metrics results in a recall of 94.7% on the DriveU traffic light dataset and an mAP of 65.3% on the Bosch Small Traffic Lights Dataset. Table 1 present a comparison against state-of-the-art works on the DriveU traffic light dataset in term of recall and processing speed.

Table 1. Comparison against state-of-the-art on the DriveU traffic light dataset

Model	Recall (%)	Speed (ms)
TL-SSD [5]	92.10	111
IARA [22]	91.05	97
Ours	94.70	85

Table 2. Comparison against state-of-the-art models

Model	mAP (%)	Speed (ms)
Faster RCNN + ResNet101 [29]	45.07	2638.73
YOLO v2 [29]	30.63	543.32
RetinaNet [30]	54.25	108
Ours	65.30	85

As shown in Table 1, the proposed approach achieves Hight performance and outperforms state-of-the-art models in both recall and processing speed tested on the same dataset.

Table 2 shows a comparative study between the proposed approach and state-of-the-art models tested on the Bosch Small Traffic Lights Dataset.

Table 1 compares three different models, TL-SSD [5], IARA [22], and the proposed model tested on the DriveU traffic light dataset, based on their recall percentage and processing speed.

The TL-SSD model achieves a recall rate of 92.10%, indicating that it successfully detects and captures 92.10% of the actual traffic lights present in the scene. It operates at a speed of 111 ms, which indicates the time taken by the model to process each frame.

IARA: The IARA model achieves a recall rate of 91.05%, which indicates a robust performance in detecting traffic lights. It operates at a speed of 97 ms, similar to the TL-SSD model.

The proposed model achieves the highest recall rate of 94.70%, demonstrating its robustness in accurately detecting a higher percentage of traffic lights in the scene. It operates at a speed of 85 ms, indicating a faster processing time compared to the other models.

The experimental results suggest that the proposed model outperforms the TL-SSD and IARA models in terms of recall percentage, indicating its effectiveness in accurately capturing a higher number of traffic lights. Additionally, the proposed model maintains a faster processing speed, making it suitable for real-time applications.

These results demonstrate the robustness of the traffic light detection system based on the SSD model, showcasing high recall rates and fast processing times. The system effectively detects and captures a significant percentage of the traffic lights present, ensuring accurate perception for ADAS applications.

By testing on the Bosch Small Traffic Lights Dataset, the Faster RCNN + ResNet101 achieves an mAP of 45.07%, indicating a moderate performance in accurately detecting and localizing traffic lights. However, it operates at a relatively

slow speed of 2638.73 ms, which may impact real-time applications.

The YOLO v2 model achieves an mAP of 30.63%, which is lower compared to other models. It operates at a faster speed of 543.32 ms, making it suitable for real-time applications that prioritize speed over accuracy.

The RetinaNet model achieves an mAP of 54.25%, indicating a relatively high level of accuracy in detecting traffic lights. It operates at a speed of 108 ms, which is significantly faster compared to the previous two models.

The proposed model achieves the highest mAP of 65.30%, showcasing its robustness in accurately detecting and localizing traffic lights. It operates at a speed of 85 ms, which is faster compared to other models, making it suitable for real-time applications.

The experimental results proved that the SSD-based model outperforms the other models in terms of both accuracy (mAP) and speed. It demonstrates robustness in accurately detecting traffic lights while maintaining a fast-processing time. These results indicate the effectiveness of the SSD model for traffic light detection in ADAS applications, providing a balance between accuracy and real-time performance.

In the Bosch Small Traffic Lights Dataset, the average width of traffic lights per image is 9.43 pixels. So, most of the proposed models fail in detecting those small traffic lights. To overcome this issue and enhance the detection precision, we use the proposed model with the multi-scale feature maps to take advantage of using information from different feature maps and generate an average prediction and we use small predefined anchors to fit the size of the detected traffic lights. The combination of those features results in a significant improvement in terms of precision without hearting the processing speed.

As reported in Table 1 and Table 2, the proposed approach achieves state of art on both datasets in terms of precision/recall and processing speed. The proposed approach based on the combination of an object detection model and a GRU network was very effective for traffic light detection in real-time. The high performance was achieved thanks to many factors. First, using a lightweight convolutional neural network as a backbone for the object detection model speed up the processing speed. Second, the use of multi-scale feature maps enables the possibility of detecting traffic lights at different scales without increasing the processing speed because of the use of high-resolution images. Third, the custom predefined anchors for each dataset allow us to detect very small traffic lights. Finally, the use of the GRU network helps to store more information from previous frames and use them for the prediction of the current frame. An average prediction of many frames results in improving the detection accuracy and allows the tracking of the state of the traffic light and use it for the prediction of the next state.

The experimental results prove the efficiency of the proposed approach based on combining convolutional neural networks with recurrent neural networks. The proposed approach was suitable for mobile devices through the use of a lightweight backbone. That was very important for vehicles processing units equipped with limited computations and energy. The use of the GRU network has enhanced detection precision through collecting temporal features. Considering the old state of the traffic light the final decision will be more trusted. It is critical to achieving high precision to guarantee safety. The combination of the lightweight model and the GRU makes the traffic light system useful due to achieving real-time processing and high precision. Besides, being suitable for mobile devices makes the proposed method applicable for real-world applications.

5. CONCLUSIONS

Ensuring the safety of the urban environment requires a strong emphasis on respecting traffic lights. To enhance this safety aspect, it is essential to integrate a reliable traffic light detection system into ADAS, which leverages intelligent processing of camera data. To build such a robust system, we recommend combining a convolutional neural network (CNN)-based object detection model with a recurrent neural network (RNN).

For the object detection model, we propose utilizing the SSD model with the MobileNet architecture as its backbone. This combination has shown promising results in accurately detecting and tracking the state of traffic lights, even under challenging conditions. To further enhance the system's performance, we suggest employing a GRU network with two layers, each comprising 1280 units, as the RNN component.

To validate the effectiveness and robustness of our proposed approach, extensive evaluations were conducted on two datasets. The reported results substantiate the reliability of our traffic light detection system, showcasing its capability to detect even the smallest traffic lights in various conditions, including different speeds, occlusions, and varying perspectives.

Moreover, to optimize the system's efficiency, we suggest exploring compression techniques such as quantization and pruning. Applying these techniques can help improve the processing speed and reduce the model size, ensuring seamless integration with mobile devices without compromising performance.

By following these recommendations, we can enhance the safety of the urban environment by effectively detecting and respecting traffic lights. The proposed system, combining a CNN-based object detection model with an RNN, showcases promising results and can be further optimized for real-world deployment. The reported results proved that our traffic detection system is reliable and can be used to detect very small traffic lights at different conditions such as moving speed, occlusion, a different point of view, etc. In future work, we propose to implement the proposed system in a mobile device and tested in a real environment. Besides, more compression techniques such as quantization and pruning can be applied to the proposed model to speed up the processing speed and reduce the model size to better fit the mobile device.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through large group Research Project under grant number RGP .2/408/44.

REFERENCES

[1] Albawi, S., Mohammed, T.A., Al-Zawi, S. (2017). Understanding of a convolutional neural network. In 2017 International Conference on Engineering and

Technology (ICET), Antalya, Turkey, pp. 1-6. https://doi.org/10.1109/ICEngTechnol.2017.8308186
[2] Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena, 404: 132306. https://doi.org/10.1016/j.physd.2019.132306
[3] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, pp. 4510-4520. https://doi.org/10.1109/CVPR.2018.00474
[4] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078. https://doi.org/10.48550/arXiv.1406.1078
[5] Fregin, A., Muller, J., Krebel, U., Dietmayer, K. (2018). The DriveU traffic light dataset: Introduction and comparison with existing datasets. In 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, pp. 3376-3383. https://doi.org/10.1109/ICRA.2018.8460737
[6] Behrendt, K., Novak, L., Botros, R. (2017). A deep learning approach to traffic lights: Detection, tracking, and classification. In 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, pp. 1370-1377. https://doi.org/10.1109/ICRA.2017.7989163
[7] Afif, M., Ayachi, R., Said, Y., Atri, M. (2020). Deep learning based application for indoor scene recognition. Neural Processing Letters, 51: 2827-2837. https://doi.org/10.1007/s11063-020-10231-w
[8] Afif, M., Ayachi, R., Said, Y., Pissaloux, E., Atri, M. (2020). An evaluation of retinanet on indoor object detection for blind and visually impaired persons assistance navigation. Neural Processing Letters, 51: 2265-2279. https://doi.org/10.1007/s11063-020-10197-9
[9] Ayachi, R., Afif, M., Said, Y., Atri, M. (2020). Traffic signs detection for real-world application of an advanced driving assisting system using deep learning. Neural Processing Letters, 51: 837-851. https://doi.org/10.1007/s11063-019-10115-8
[10] Ayachi, R., Said, Y.E., Atri, M. (2019). To perform road signs recognition for autonomous vehicles using cascaded deep learning pipeline. Artificial Intelligence Advances, 1(1): 1-10. https://doi.org/10.30564/aia.v1i1.569
[11] Jensen, M.B., Philipsen, M.P., Møgelmose, A., Moeslund, T.B., Trivedi, M.M. (2016). Vision for looking at traffic lights: Issues, survey, and perspectives. IEEE Transactions on Intelligent Transportation Systems, 17(7): 1800-1815. https://doi.org/10.1109/TITS.2015.2509509
[12] Diaz, M., Cerri, P., Pirlo, G., Ferrer, M.A., Impedovo, D. (2015). A survey on traffic light detection. In: Murino, V., Puppo, E., Sona, D., Cristani, M., Sansone, C. (eds) New Trends in Image Analysis and Processing -- ICIAP 2015 Workshops. ICIAP 2015. Lecture Notes in Computer Science(), vol 9281. Springer, Cham. https://doi.org/10.1007/978-3-319-23222-5_25
[13] Kulkarni, R., Dhavalikar, S., Bangar, S. (2018). Traffic light detection and recognition for self driving cars using deep learning. In 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, pp. 1-4. https://doi.org/10.1109/ICCUBEA.2018.8697819
[14] Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, pp. 580-587. https://doi.org/10.1109/CVPR.2014.81
[15] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 2818-2826. https://doi.org/10.1109/CVPR.2016.308
[16] Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W. (2013). Selective search for object recognition. International Journal of Computer Vision, 104: 154-171. https://doi.org/10.1007/s11263-013-0620-5
[17] Vitas, D., Tomic, M., Burul, M. (2020). Traffic light detection in autonomous driving systems. IEEE Consumer Electronics Magazine, 9(4): 90-96. https://doi.org/10.1109/MCE.2020.2969156
[18] Roy, P., Dutta, S., Dey, N., Dey, G., Chakraborty, S., Ray, R. (2014). Adaptive thresholding: A comparative study. In 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kanyakumari, India, pp. 1182-1186. https://doi.org/10.1109/ICCICCT.2014.6993140
[19] Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. https://doi.org/10.48550/arXiv.1409.1556
[20] Ouyang, Z., Niu, J., Liu, Y., Guizani, M. (2019). Deep CNN-based real-time traffic light detector for self-driving vehicles. IEEE Transactions on Mobile Computing, 19(2): 300-313. https://doi.org/10.1109/TMC.2019.2892451
[21] Redmon, J., Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767. https://doi.org/10.48550/arXiv.1804.02767
[22] Possatti, L.C., Guidolini, R., Cardoso, V.B., Berriel, R. F., Paixão, T.M., Badue, C., De Souza, A.F., Oliveira-Santos, T. (2019). Traffic light recognition using deep learning and prior maps for autonomous cars. In 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, pp. 1-8. https://doi.org/10.1109/IJCNN.2019.8851927
[23] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L. (2014). Microsoft coco: Common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8693. Springer, Cham. https://doi.org/10.1007/978-3-319-10602-1_48
[24] Feng, Y., Kong, D., Wei, P., Sun, H., Zheng, N. (2019). A benchmark dataset and multi-scale attention network for semantic traffic light detection. In 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, pp. 1-8. https://doi.org/10.1109/ITSC45078.2019.9086430

[25] Ayachi, R., Afif, M., Said, Y., Atri, M. (2020). Strided convolution instead of max pooling for memory efficiency of convolutional neural networks. In: Bouhlel, M., Rovetta, S. (eds) Proceedings of the 8th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT'18), Vol.1. SETIT 2018. Smart Innovation, Systems and Technologies, vol 146. Springer, Cham. https://doi.org/10.1007/978-3-030-21005-2_23

[26] Wang, Q., Zhang, Q., Liang, X., Wang, Y., Zhou, C., Mikulovich, V.I. (2021). Traffic lights detection and recognition method based on the improved YOLOv4 algorithm. Sensors, 22(1): 200. https://doi.org/10.3390/s22010200

[27] Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934. https://doi.org/10.48550/arXiv.2004.10934

[28] Hassan, E., Khalil, Y., Ahmad, I. (2023). Learning deep feature fusion for traffic light detection. Journal of Engineering Research, 100066. https://doi.org/10.1016/j.jer.2023.100066

[29] Gokul, R., Nirmal, A., Bharath, K.M., Pranesh, M.P., Karthika, R. (2020). 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), Vellore, India, pp. 1-6. https://doi.org/10.1109/ic-ETITE47903.2020.449

[30] Aneesh, A.N., Shine, L., Pradeep, R., Sajith, V. (2019). Real-time traffic light detection and recognition based on deep RetinaNet for self driving cars. In 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kannur, India, pp. 1554-1557. https://doi.org/10.1109/ICICICT46008.2019.8993293