Robust Real-time Traffic Light Detector on Small-Form Platform for Autonomous Vehicles

Gelayol Golcarenarenji[a,b], Ignacio Martinez-Alpiste[a], Qi Wang[a] and Jose Maria Alcaraz-Calero[a]

[a]The School of Computing, Engineering and Physical Sciences, University of the West of Scotland, High St, Paisley, UK; [b]School of Computing, University of Portsmouth, Portsmouth, UK

CONTACT Gelayol Golcarenarenji. Email: gollygolkar@gmail.com
Ignacio Martinez-Alpiste. Email: ignacio.alpiste@uws.ac.uk
Qi Wang. Email: qi.wang@uws.ac.uk
Jose Maria Alcaraz-Calero. Email: jose.alcaraz-calero@uws.ac.uk

ABSTRACT
Timely and accurate detection and recognition of traffic lights are critical for Autonomous Vehicles (AVs) to avoid crashes due to red light running. This paper integrates a new robust machine learning based solution by combining a Convolutional Neural Network (CNN) with computer vision techniques to achieve a real-time traffic light detector. The proposed detection and recognition algorithm is capable of recognising traffic lights on low-power small-form platforms, which are lightweight, portable, and can be mounted on AVs in daylight scenarios. The LISA open-source dataset is utilised with augmentation methods to increase the accuracy of the solution. The proposed approach achieves 93.42% of accuracy at a speed of 30.01 Frames Per Second (FPS) on an NVIDIA Jetson Xavier platform without using hardware accelerators such as FPGA. This solution is expected to promote the quicker adoption and wider deployment of AVs by increasing the chances of avoiding crashes and ultimately saving lives.

KEYWORDS
Computer Vision; Traffic Light Detection; Autonomous Vehicle; Modified YOLO

1. Introduction

Autonomous driving has gained great momentum in recent years as a propitious solution to various transportation problems (Binangkit & Widyantoro, 2016). To deploy Autonomous Vehicles (AVs) on the roads, it is vital for autonomous vehicles to be able to understand their surroundings and deal with traffic-related information, such as traffic lights (Chen et al., 2021). Red light running caused 683 death and 133,000 injuries only in the USA in 2012 (Iihs, 2012), thereby, making traffic light recognition an essential requirement and a crucial component for AVs. Timely and accurate detection and recognition of traffic lights are imperative for AVs to avoid crashes involved in red light running, which leads to reducing deaths and injuries of people. Therefore, a traffic light detection and recognition system must be adequately capable in terms of both high accuracy and speed, and it also has to be portable and small-sized to be integrated into an AV.

Although there is a lot of existing work on traffic light detection and recognition (Gouda et al., 2022), this Artificial intelligence (AI) problem is still challenging. The root cause of this issue is due to the high power consumption of deep learning methods on small-form platforms which are resource-constrained for self-driving vehicles. Furthermore, it is due to the challenges in recognising traffic lights at greater distances with different illumination, to name a few (Hirabayashi et al., 2019; John et al., 2014; Possatti et al., 2019; Saini et al., 2017). Hence, there is a need for a robust real-time traffic light recognition system that is able to improve traffic light detection from distance on low-power small-form devices.

Traditional computer vision approaches such as image segmentation, threshold, and edge detection are rudimentary techniques that have been used in traffic light detection. They can achieve high detection speed due to the simple logic behind these techniques. However, they require fine-tuning of many parameters and are not accurate enough for complex outdoor environments (Ouyang et al., 2020). Traditional machine learning approaches such as Histogram of Oriented Gradients (HOG) in combination with classification techniques such as Support Vector Machines (SVM) and Adaptive Boosting (AdaBoost) are also suitable for traffic light detection scenarios due to their model size and effectiveness. However, the accuracy is still low in outdoor environments (Hirabayashi et al., 2019).

Deep learning-based object detectors, such as "You Only Look Once" (YOLO) (Redmon & Farhadi, 2018), "Faster R-CNN" (Ren et al., 2015), and "Single Shot Detector" (SSD) (Liu et al., 2016), are very efficient techniques for detection of objects, as more features can be learnt by convolutional layers than traditional methods. These three are classified as CNN-based one-stage object detectors. Nevertheless, they are still computationally expensive for real-time scenarios, such as traffic light detection running on computationally restricted devices embedded in autonomous vehicles.

As a result, to overcome the low accuracy of traditional computer-vision techniques and low speed of CNN-Based methods, this paper proposes a light algorithm based on standard Tiny-YOLOv3 algorithm with an extra output layer (He et al., 2019) integrated with Partial Residual Networks-based (PRN-based) (Wang et al., 2019) architecture. In addition, this architecture has also been equipped with OpenCV segmentation algorithms for state recognition (red, yellow, green) deployed on an embedded platform (NVIDIA Jetson AGX Xavier). This architecture accelerates traffic light detection and enables small traffic lights with high accuracy. In summary, our main contributions are as follows:

(1) Propose an improved algorithm for enhanced accuracy and speed for traffic light detection and recognition.
(2) Design and prototype a novel fast yet accurate traffic light detector processing pipeline suitable for embedded car devices.
(3) Deployment and validation of the system in an NVIDIA Jetson AGX Xavier embedded platform suitable to be installed in cars.

The rest of the paper is structured as follows. Section II presents the related work. Section III explains the design of the proposed approach. This is followed by the experimental setup in Section IV. The results of the proposed approach are discussed in Section V. Section VI concludes the article.

2. Related work

This section reviews state-of-the-art work in relation to traffic light detection systems and algorithms, with a focus on techniques used in this article to achieve a real-time traffic light detector.

2.1. Object detection

As mentioned, traffic light detection approaches have been classified into three different categories: traditional computer-vision based methods, traditional machine-learning methods, and deep-learning methods. In the first category, the detector uses a heuristically defined threshold in a selected color space. However, as the color changes in different scenes due to different illumination, it is integrated with shape and structural information to perform the detection (Jensen et al., 2017). In the second method, image feature extraction approaches, such as HOG, are combined with classifiers, such as SVM. These approaches are computationally expensive and low in speed which are not suitable for real-time scenarios. In the third method, CNN approaches are used for the detection and recognition of the states of the traffic lights.

These CNN-based methods are generally classified into two main categories: two-stage and one-stage object detectors. The two-stage detectors (Dai et al., 2016; Girshick, 2015; Ren et al., 2017) extract the Region Of Interest (ROI) and perform the classification at the second stage. These detectors are accurate, but computationally expensive due to the double stage available in their design and thus not suitable for resource-limited devices. One-stage detectors such as YOLOv3, YOLOv4 and YOLOv7 (Bochkovskiy et al., 2020; Redmon et al., 2016; Redmon & Farhadi, 2017, 2018; Van Etten, 2018; Wang et al., 2022) select and detect the objects simultaneously (Liu et al., 2016) in the same stage. This makes them faster in speed but lower in accuracy.

2.2. Previous work

This subsection provides an overview of various research work related to traffic light detection in literature together with a comparative analysis, summarised in Table 1, to analyse the published results related to traffic light detection and compare them with our contribution in order to distinguish our contribution from the state of the art.

It can be seen how the vast majority of research work carried out in this field has been designed, optimised, and validated in traditional personal computer platforms,

Table 1. Comparison of previous works

Ref	Algorithm	HW	Platform	Accuracy (%)	Speed (FPS)	HW Accel.	Power (W)	Distance (m)	Resolution	Dataset
Binangkit & Widyantoro (2016)	Seg/SVM	PC	NG	0.88(F1-score)	NG	NG	300*	NG	1290x960	LISA
Hirabayashi et al. (2019)	SSD	PC	Cafee	90	58.8	GPU+ FPGA	300* 0-120	1368x1096	Private	
Saini et al. (2017)	SVM	PC	Caffe	94.47	39	GPU	300*	NG	1280x800	Private
Kim et al. (2018)	Faster R-CNN	PC	NG	38.47	37.24	NG	300*	NG	1280x720	Private/ Bosch/ VIVA
Weber et al. (2016)	Custom CNN/R-FCN	PC	Caffe	88.1(F1 score)	34.72	GPU	300*	NG	640x480/ 1280x960	Private/ LaRA
John et al. (2016)	Seg+ Custom CNN	PC	NG	100	50	NG	300*	NG	NG	Private
Chen & Huang (2016)	PCAnet+ SVM	PC	NG	95.7	3	CPU	300*	NG	1920x1080	WPI
Ozcelik et al. (2017)	Seg+ SVM	PC	NG	95.00/88.00	22.22	NG	300*	NG	900x506	Private
J. Kim et al. (2018)	SSD	PC	Caffe	41.15 red 10.05 yellow 69.6 green	NG	GPU	300*	NG	1280x720	Private
Müller & Dietmayer (2018)	Custom-SSD	PC	NG	95	8.4	NG	300*	NG	2048x512	DriveU
Weber et al. (2018)	Alexnet	PC	Caffe	80.7	23.868	NG	300*	NG	640x480/ 1280x960	Private/ Bosch
Weber et al. (2018)	GoogLeNet	PC	Caffe	87.5	11.37	NG	300*	NG	640x480/ 1280x960	Private/ Bosch
Weber et al. (2018)	VGG	PC	Caffe	76	10.6	NG	300*	NG	640x480/ 1280x960	Private/ Bosch
Fernández et al. (2018)	Custom CNN	PC	Keras	90.3	142.86	GPU	300*	NG	64×32	BOSCH/ CITY/ LARA/ LISA/ WPI
Pon et al. (2018)	Faster-RCNN	PC	Tensorflow	44	66.66	GPU	300*	NG	NG	Bosch/ Tsinghua-Tencent
Yudin & Slavioglo (2018)	Custom-CNN	TX2	Keras	0.59 (F1score)	20	GPU	10	NG	256x455	Nexar
Ouyang et al. (2020)	Tiny-Yolo v3	Tx1/Tx2	Darknet	IOU 38.13 / 32.47 5 Recall	5.2	GPU	10	0-50	1290x960	WPI/ VIVA/ YBY
Ouyang et al. (2020)	Tiny-Yolo v3	Tx1/Tx2	Darknet	IOU 17.71 / 5.59 5 Recall	14.5	GPU	10	0-50	608x608	WPI/ VIVA/ YBY
Ouyang et al. (2020)	Heuristic + Custom CNN	PC/Tx1 /Tx2	Darknet	91.2 (IOU 40.45 / 31.4 5 Recall)	10.2	GPU/FPGA	10	0-50	1290x960	WPI/ VIVA/ YBY
Ou et al. (2022)	Transformer +Resnet	PC		88.1	NG	NG	300*	NG	NG	COCO
Vaidya & Paunwala (2019)	CNN+	Jetson Nano	Tensorflow	98.26	90.9	GPU	NG	NG	64 × 64	German/ Belgium
	Transformer+ STN			99.94					32 × 32	
TP	Own Approach	Xavier	Darknet	93.42 (IOU 70.18/93.00 5 Recall)	30.015	GPU	30	0-50	416x416	LISA
TP	Own Approach	Xavier	Darknet	93.42 (IOU 70.18/93.00 5 Recall)	10.23	GPU	10	0-50	416x416	LISA
TP	Own Approach	PC	Darknet	93.42 (IOU 70.18/93.00 5 Recall)	191.18	GPU	10	0-50	416x416	LISA

TP = This Paper; NG = Not Given; *=Desktop PC Minimum Power Supply commercially available is 300W

which have no limitation in power consumption, with a significant computational budget available and with the possibility to use GPU acceleration (Binangkit & Widyantoro, 2016; Chen & Huang, 2016; Fernández et al., 2018; Hirabayashi et al., 2019; John et al., 2016; Kim et al., 2018; J. Kim et al., 2018; Müller & Dietmayer, 2018; Ozcelik et al., 2017; Pon et al., 2018; Saini et al., 2017; Weber et al., 2016, 2018). These algorithms may not be suitable to be deployed on low-performance platforms such as Jetson AGX Xavier for real-time scenarios. Moreover, the detection distance has not been specified in any of these studies except for (Hirabayashi et al., 2019).

Some studies such as Binangkit & Widyantoro (2016), Pon et al. (2018), and Kim et al. (2018) did not report speed as a necessary metric for evaluation (Martinez-Alpiste et al., 2019, 2020). In addition, studies such as Chen & Huang (2016), Ozcelik et al. (2017), Müller & Dietmayer (2018), and Weber et al. (2018), indicated that the reported speed is not real-time on a PC platform.

Hirabayashi et al. (2019) used a Field-Programmable Gate Array (FPGA) to accelerate deep learning for traffic light detection on AVs. They used a combination of a traditional computer-vision based model and a custom CNN to extract the RoI and to classify the traffic light state. Their approach is fast and accurate. However, FPGA chips to accelerate the model are very expensive and not easy to be used.

Regarding solutions tailored for embedded resource constrained-devices, Yudin & Slavioglo (2018) developed a custom-CNN algorithm for traffic light detection on the NVIDIA Jetson TX2 platform. They achieved a high speed for their detector. Meanwhile, the accuracy has room for improvement as shown in Table 1.

Ou et al. (2022) developed a model based on Transformer and Resnet convolutional neural network using COCO data to perform traffic signal light recognition. Although achieved a high accuracy of 88%, the model size and speed were not reported. In addition, Vaidya & Paunwala (2019) developed a novel Architecture based on Convolutional Neural Network (CNN), color Transformer Network and Spatial Transformer Network (STN). The performance of the algorithm is compared with German and Belgium Traffic Sign datasets. The model was also deployed on Jetson Nano. The high accuracy of 98.26 and 99.94 are achieved. The speed was 90.9 FPS on Jetson Nano with input size of 64. Ouyang et al. (2020) provided a combination of customized CNN and a heuristic algorithm to perform the classification of the traffic lights for resource-constrained devices. They tested their algorithms on NVIDIA Jetson TX1 and TX2, and achieved a very high accuracy 91.2% with a very acceptable 10.2 frames per second (FPS) when running on a TX2 platform with 2 FPGA for available low-power (10W). This accuracy has been achieved by training their algorithms with three databases including the LISA dataset using high-resolution images 1280×960. They did a very comprehensive comparison of their solution with other algorithms including Tiny-YOLOv3, which is the foundation of our research work. It is worth noting that the Intersection over Union (IoU), the recall, and the speed reported for this algorithm were 38.13% and 32.45% and 5.2 FPS respectively in their study for LISA dataset with the resolution of 1280×960 using FPGA in their Jetson platform.

Our solution has achieved a high accuracy with an input size of 416×416 for small objects suitable for low-resolution scenarios, which are highly likely to be the case. Moreover, it satisfies the requirement of real-time performance on low-performance platforms without using any hardware accelerators such as FPGA. Accordingly, this paper proposes a light traffic light detection and state recognition system with high accuracy that meets the requirements of mobile platforms such as Jetson AGX Xavier in terms of speed.

3. Design of Proposed Algorithm

In this section, the architecture of our approach is defined. By applying our approach, we aim to achieve high accuracy for the detection of small objects such as traffic lights on low-power platforms and improve the speed of detection.

3.1. Tiny-YOLOv3-3L-PRN (Our approach)

Figure 1 illustrates the schematic of the proposed design. As stated in section 1, standard YOLOv3, as an accurate one-stage detector, is computationally expensive. Therefore, a simplified version of YOLOv3 called Tiny-YOLOv3 has been introduced. It is the fastest model that can perform detection at more than 200 FPS. However, the standard Tiny-YOLOv3 with just two output layers limits the detection of small objects. Hence, similar to YOLOv3, an extra output layer was added to the standard Tiny-YOLOv3 to increase the accuracy of small object detection (Alexey, 2020; He et al., 2019). Therefore, the standard Tiny-YOLOv3 with three output layers (Tiny-YOLOv3-3L) is deployed in this study to enhance the detection of traffic lights from greater distances. To speed up the CNN while preserving the accuracy, PRN (Partial Residual Network) technique is a useful scheme (Wang et al., 2019). This assists the networks with fewer layers to obtain more information about the system. In this method, skip connections are used to combine various numbers of channels. In addition, the number of channels for partial residual connection is 50% of the whole channel number. It has been designed to work with feature maps with different channel numbers. It spreads the combination of the gradients instead of features in the training phase which increase the learning information of layers. Tiny models can benefit from this design technique and are applied in use cases where real-time is a requirement, such as traffic systems.

The PRN technique is used in standard Tiny-YOLOv3 and standard YOLOv3. When this technique is applied in standard YOLOv3, the frame rates are doubled. The frame rates are also increased by 12 FPS in standard Tiny-YOLOv3 when running on the NVIDIA Jetson TX2 platform (Wang et al., 2019). Hence, we combine the PRN technique with Tiny-YOLOv3 with an extra output layer (Tiny-YOLOv3-3L) to have a fast algorithm while preserving the accuracy of Tiny-YOLOv3-3l. Inspired by standard Tiny-YOLOv3-2L-PRN (Wang et al., 2019), an extra partial residual connection is added to Tiny-YOLOv3-3l. As apparent from Figure 1, the convolution with 128 filters is reduced to 64 with three shortcuts added to the last output layer. The hyper-parameters of the Tiny-YOLOv3-2L-PRN are further optimised using grid search to improve the accuracy of detecting traffic lights.

We chose one-stage CNN detectors for traffic light detection due to the importance of not missing any traffic lights. The coordinates obtained from the detection stage are then employed to perform state recognition. The OpenCV library is utilised to recognise the color of the traffic light. The image color code is converted from RGB to HSV color space and the specific range of H-S-V values are defined to recognise green, yellow, and red color as this color space explains colors similarly to how the human eye recognises colors. The masks are generated to determine the regions that define the detected colors and the black pixels which are the values out of the threshold range. The detected traffic light is divided into three separate sections corresponding to different colors in order to tackle the challenges of state recognition due to overlap of the colors, illumination, and low resolution and to ensure a correct recognition and to detect turned-off traffic lights. If the portion of black pixels on the top section is less than the threshold, the red will be detected and the image will be updated. If it lies on the middle part and the portion of black pixels on the top section is less than the threshold, yellow will be detected. At last, the green color will be detected if the portion of black pixels on the bottom section is less than the threshold; otherwise, nothing will be detected. Figure 2 illustrates the algorithm of the color segmentation stage.

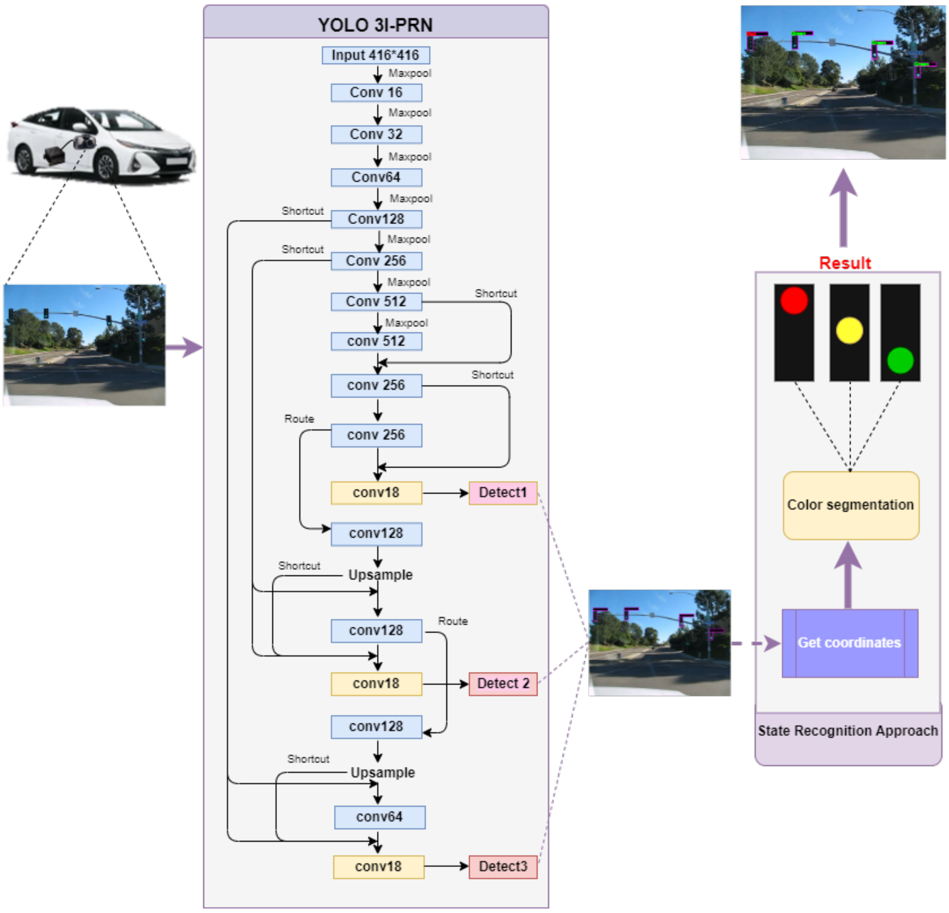

Figure 1. Design of the Proposed Detection and Recognition Approach

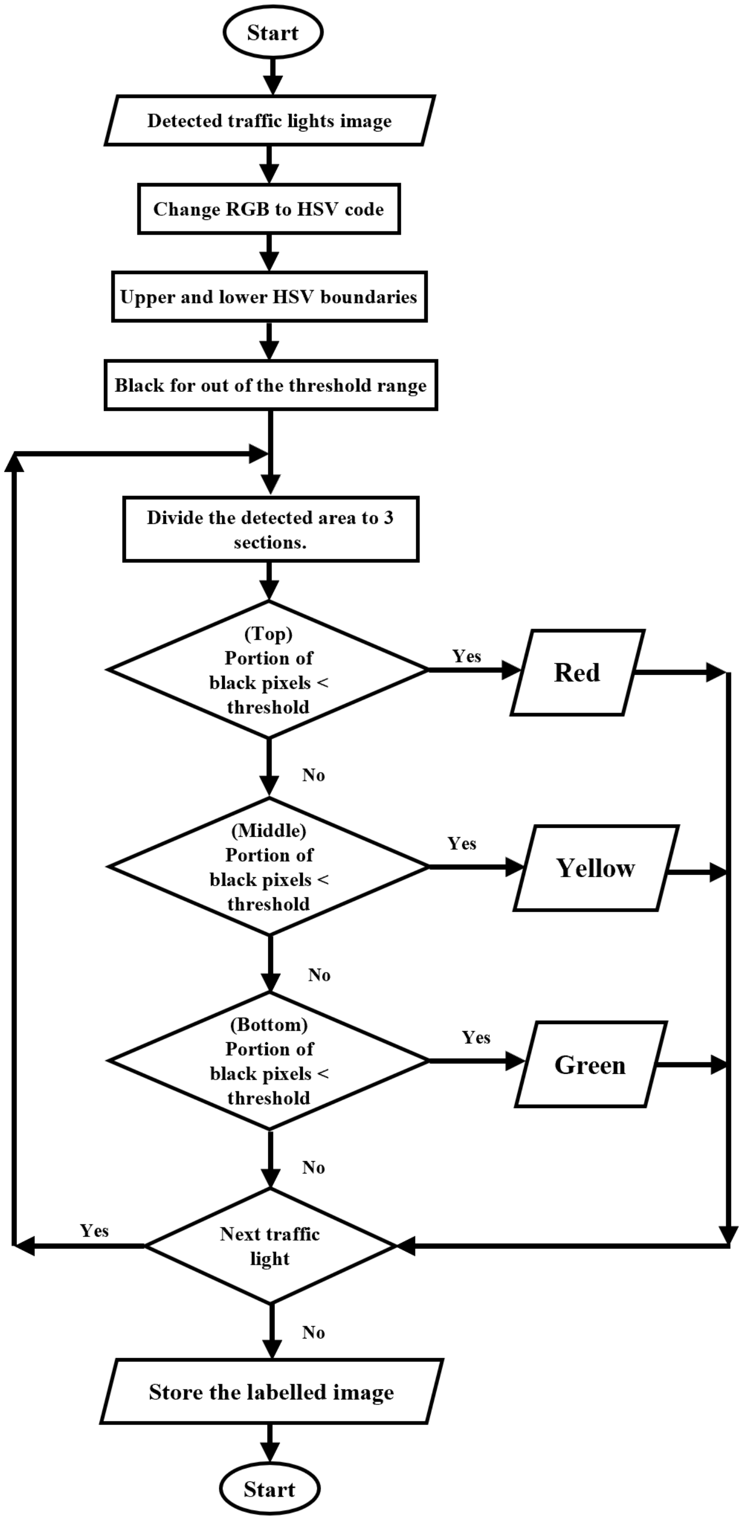

Figure 2. Flowchart of the Proposed Recognition Approach.

4. Experimental setup

4.1. Dataset

We have applied and evaluated our proposed traffic light detector on the LISA dataset (Mogelmose et al., 2012). This dataset contains collected traffic lights during both day and night. The image resolution of 1280×960 pixels were used in this dataset. The day-light images were used in this study and 7354 images were extracted and the related annotation CSV files were converted to YOLO-Mark2 (Alexey, 2017) format to be executed on the Darknet platform. As the data-set was not balanced (same number of positive and negative images), the negative images were extracted and augmented using methods such as Additive Gaussian noise, blurring, contrast adjustment, dropouts, etc. The images with a certain redundancy (homogeneity) were eliminated from the dataset.

4.2. Hyper-parameters

To manage the trade-off between inference time and detection accuracy, 416×416 pixels were adopted for input images.

We have evaluated the performance of each model based on two metrics: mean Average Precision (mAP) and Frame Per Second (FPS). mAP is a standard measure for comparing algorithms. In addition, FPS provides the speed of the models. For the training purpose, the number of iterations was set to 15,000. As previously mentioned in section 3, the initial learning rate was optimised and set to 0.008 using grid search. We used Stochastic Gradient Descent with Warm Restarts (SGDR) (Loshchilov & Hutter, 2017) solver with a momentum coefficient of 0.9 as learning policy. Finally, we have also optimised the weight decay and set it to 0.001 to avoid over-fitting. The batch and the subdivision (the number of mini-batches in one batch) were set to 128 and 2 respectively. Transfer learning (Kohavi, 1995) was another technique employed in these experiments. Pre-trained weights from the COCO dataset (Lin et al., 2014) were used for transfer learning to further increase the accuracy. Anchor boxes were recalculated for the new dataset using the k-means technique (Alexey, 2020).

4.3. Execution Environment

The algorithm was trained and executed on Darknet (Alexey, 2020), which is an open source framework for neural networks. It is written in C and CUDA, and enables execution on CPUs or GPUs. The execution system is held in an NVIDIA Jetson Xavier platform which has integrated an embedded NVIDIA GPU able to operate at different consumption powers. It has a 512-core Volta GPU with Tensor Cores as GPU and 8-core ARM v8.2 64-bit CPU as CPU. Ubuntu 18.04 is the operating system where the object recognition pipeline is executed. In addition, the speed of the proposed model was also tested on a computer with an Intel(R) Xeon(R) E5-2630 v4 at 2.20GHz with 20 cores and 32 GB RAM, running Ubuntu 20.04 with a kernel version of 5.11.0 and an NVIDIA Titan X GPU with 12 GB RAM.

5. Results and Discussions

To test the validity of the proposed traffic light detection and recognition algorithm, a set of unseen images were randomly selected from the LaRA dataset (INRIA's project-team IMARA, 2009) with low-resolution (640 × 480) as well as our on-road testing. Our approach was compared with Tiny-YOLOv3-2L and Tiny-YOLOv3-2L-PRN, Mobilenet-v2, Custom-CNN (Ju et al., 2019), Tiny-YOLOv3-3L, (Alexey, 2020; He et al., 2019), Tiny-YOLOv4-2L and Tiny-YOLOv7-3L in terms of mAPs and integrated with our state recognition approach.

5.1. Quantitative results

The first evaluation metric to examine our traffic light detector is the mAP of the detection model based on the IoU threshold of 0.5. The results are shown in Figure 3. The comparison of different models is summarised in Table 2. Based on Table 2, Tiny-YOLOv3-2L, and Tiny-YOLOv3-2L-PRN are almost the same in terms of accuracy (76.45%, and 76.03% respectively) as expected since the PRN technology accelerates the CNN-based algorithm without sacrificing the accuracy. Mobilenet-v2 has higher accuracy than Tiny-YOLOv3-2L and Tiny-YOLOv3-2L-PRN (84.69%). Our approach is also compared with Ju et al. (2019), due to the proposed CNN being a light yet accurate approach for small object detection. As can be seen from the results, the proposed model (Tiny-custom-CNN) has higher accuracy than previous models (86.65%). The results of our model were also compared with the latest tiny versions of the YOLO, YOLOv4, and YOLO 7. The LISA dataset was compared to Tiny-YOLOv4-2L, and Tiny-YOLOv7 (Which has 3 detection output layers). The accuracy of 77.01 and 90.35 were achieved for YOLOV4-2L (2 layers) and tiny YOLOV7 (3 layers) respectively. The results show that the tiny version of YOLOv4-2L is better than the tiny YOLOv3-2L in terms of accuracy and speed and less than our proposed model. Tiny YOLOv7 with 3L is almost the same as our accuracy.

Table 2. Analysis results of traffic light detector configurations.

Analysis Results				
Index	Model	Accuracy	FPS(@MAXN)	FPS(@10w)
1	Tiny-YOLOv3-2L	76.45	30.71	9.40
2	Tiny-YOLOv3-2L-PRN	76.03	32.18	10.82
3	Tiny-YOLOv4-2L	77.01	32.46	10.91
4	Mobilenet-v2	84.69	29.79	8.01
5	Tiny-Custom-CNN	86.65	27.17	7.30
6	Tiny-YOLOv7-3L	90.35	30.12	10.32
7	Tiny-YOLOv3-3L	90.10	27.06	8.43
8	Our approach	90.01	30.01	10.27

The accuracy of Tiny-YOLOv3-3L and our approach are almost the same (90.10% and 90.01%, respectively). Due to many applications of traffic detection concerning more about avoiding the red light, it is imperative not to miss any traffic lights. As a result, the accuracy of our approach was also calculated at the IOU threshold 0.3

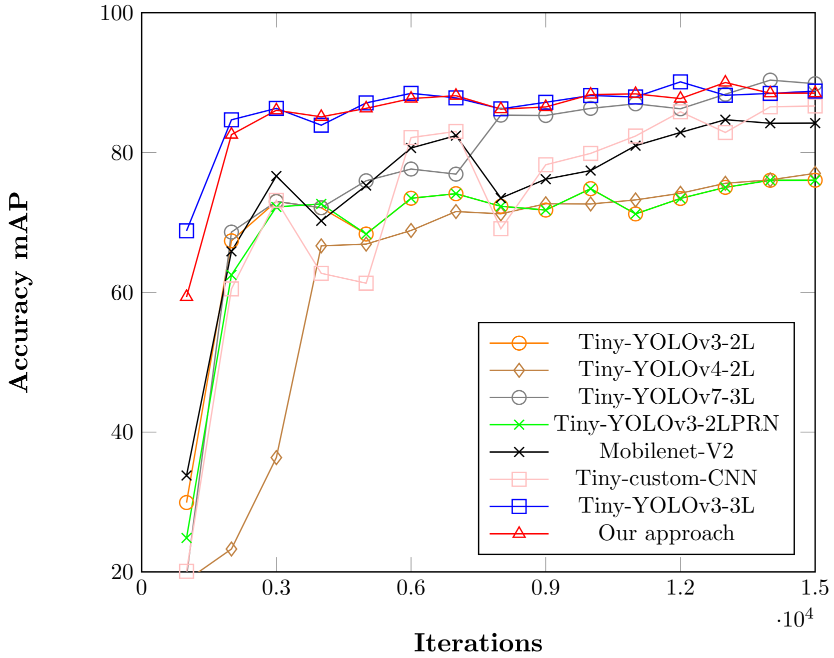

Figure 3. Accuracy (mAP) over 15000 training iterations

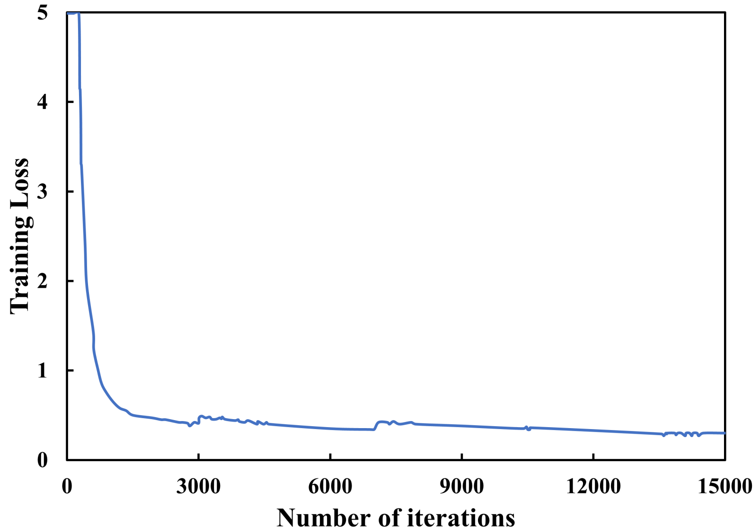

Figure 4. The training loss of our approach over 15,000 training iterations.

(93.42%) and used suitable for real environments. In addition, our proposed state recognition algorithm almost recognised all the colors correctly on the testing videos.

In terms of speed, Tiny-YOLOv3-2L-PRN was faster than Tiny-YOLOv3-2L on Jetson Xavier (32.18 FPS versus 30.71 FPS @MAXN and 10.82 versus 9.40 @10W) due to integrating the PRN technology. Mobilenet-v2 was slower than Tiny-YOLOv3-2L-PRN and Tiny-YOLOv3-2L(29.79 @MAXN and 8.01 @10W). The Tiny-custom-CNN model was slower than previous models (27.17 @MAXN and 7.30 @10W). Tiny-YOLOv3-3L-PRN was faster than Tiny-YOLOv3-3L (30.01 FPS versus 27.06 FPS

@MAXN mode and 10.27 versus 8.43 @10W) on Jetson AGX Xavier as a result of benefiting from the PRN technology. The speed of 32.46 and 30.12 were also achieved for YOLOV4-2L (2 layers) and tiny YOLOV7 (3 layers) respectively. Tiny YOLOv7 with 3L has almost the same as our accuracy and speed. The Training loss of our approach was also illustrated in Figure 4 over 15,000 iterations. In order to calculate the total FPS for the traffic light detection and state recognition, the processing time of the state recognition was also calculated and added to the processing time of the CNN-based traffic light detector and was calculated for Tiny-YOLOv3L as the basis of our model as well as our approach. The results in Figure 5 illustrate that standard Tiny-YOLOv3L and Tiny-YOLOv3-PRN combined with traffic light state recognition achieved 27.06 FPS (33.22 FPS for detection and 145.82 FPS for recognition) and 30.01 FPS (37.78 FPS for detection and 145.82 FPS for recognition) respectively.

In addition, it is also apparent in Figure 6 that by lowering the power consumption to 10 watt mode (@ GPU max frequency of 512) Jetson AGX Xavier, the integration of Tiny-YOLOv3-3L and Tiny-YOLOv3-3L-PRN with state recognition algorithm achieved 8.43 FPS (9.7 FPS for detection and 64.41 FPS for recognition) and 10.27 (12.22 FPS for detection and 64.41 FPS for recognition) respectively, which meets the low-power requirements on Jetson Xavier. As expected, the recognition state method was faster when compared with the CNN-based detection method when the power was adjusted to both 10W and MAXN mode.

These results state that our approach is highly accurate while maintaining the real-time requirement for the use cases on small traffic detection and state recognition on mobile platforms (Jetson AGX Xavier). Our model is comparable with Tiny-YOLOv7 in terms of accuracy and speed which is proven to be a better model compared to Tiny-YOLOV3, and Tiny-YOLOv4. Furthermore, it is faster than Tiny-YOLOv3 due to using PRN technology which makes a difference in power-constrained devices.

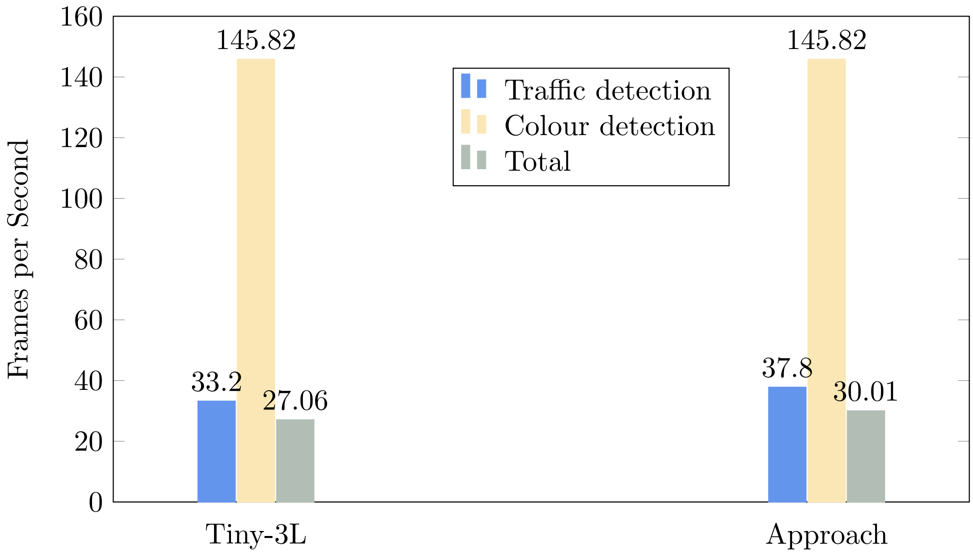

Figure 5. Detection and Recognition rate @MAXN mode.

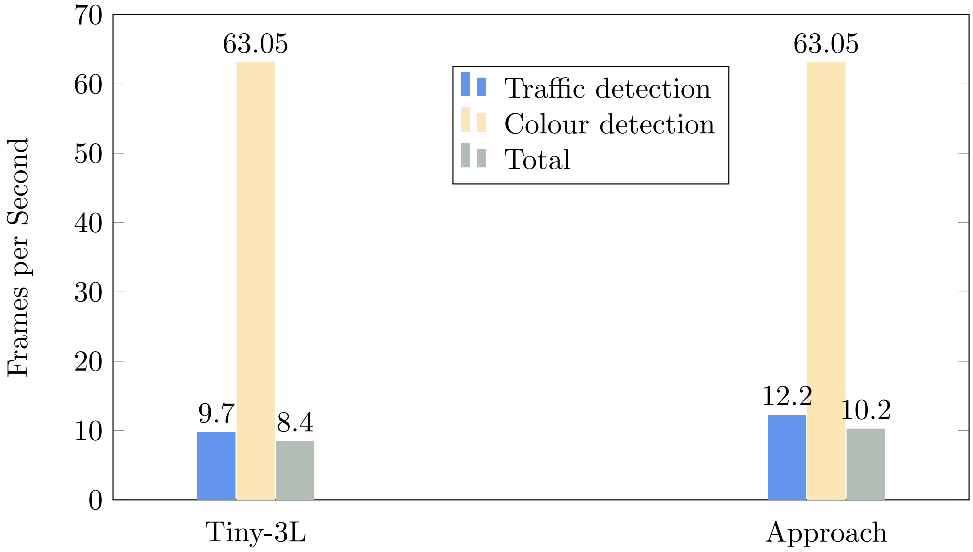

Figure 6. Detection and Recognition rate @10W mode.

5.2. Qualitative results

Our approach was deployed and tested on unseen random videos for the purposes of validation. Figure 7 shows four different screenshots of traffic light detection with different states at different locations. Figure 7a displays the yellow light detection and state recognition. In Figure 7b, the detection and recognition were performed for green and red light. Figure 7c presents detection and state recognition for green lights. Figure 7d shows detection and state recognition for red lights. As apparent in the results, the system successfully detected traffic lights and their states and no cases of false negatives were observed, thereby yielding robust performance in all the test cases.

6. Concluding Remarks and Future Work

In this work, we have proposed an integration of a light convolutional neural network based detector with a traditional computer vision technique suitable for small traffic light detection on low-power platforms such as NVIDIA Jetson AGX Xavier. Our traffic light detector achieved an average detection accuracy of 93.42 % at 30.01 FPS on the NVIDIA AGX Jetson Xavier platform. The proposed approach is suitable to be used in AVs for traffic light detection and state recognition in daylight scenarios thanks to its real-time speed and high accuracy performance. Further performance improvement could be achieved with the collection of a comprehensive customised data set containing more training data suitable for small traffic light detection. In addition, detection of various shapes of traffic lights, detection in night time (Tsai & Lin , 2022) development of a hybrid model of CNN and Transformers are areas that will be considered in future work.

(a) Detection of yellow traffic light. (b) Detection of green and red traffic light.

(c) Detection of green traffic light. (d) Detection of red traffic light.

Figure 7. Sample of detection and recognition system.

Disclosure statement

The authors declare no conflicts of interest.

References

Alexey. (2017). Yolomark. Retrieved from https://github.com/AlexeyAB/Yolomark.

Alexey. (2020). Darknet. Retrieved from https://github.com/AlexeyAB/darknet.

Binangkit, J. L. & Widyantoro, D. H. (2016). Increasing accuracy of traffic light color detection and recognition using machine learning. *2016 10th International Conference on Telecommunication Systems Services and Applications (TSSA)*, 1-5.

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y.M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004, 10934*.

Chen, X., Chen, Y., & Zhang, G. (2021). A computer vision algorithm for locating and recognizing traffic signal control light status and countdown time. *Journal of Intelligent Transportation Systems*, *25(5)*, 533-546.

Chen, Z., & Huang, X. (2016). Accurate and reliable detection of traffic lights using multiclass learning and multiobject tracking. *IEEE Intelligent Transportation Systems Magazine*, *8(4)*, 28-42.

Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object Detection via Region-based Fully Convolutional Networks. (online). Retrieved from http://arxiv.org/abs/1605.06409.

Fernández, C., Guindel, C., Salscheider, N.-O., & Stiller, C., (2018). A deep analysis of the existing datasets for traffic light state recognition. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 248-254.

Girshick, R. (2015). Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, *2015 Inter*, 1440-1448.

Gouda, M., Epp, A., Tilroe, R., & El-Basyouny, K., (2022). Fast Traffic sign extraction using deep hierarchical feature learning and mobile light detection and ranging (LiDAR) data on rural highways. *Journal of Intelligent Transportation Systems*, 1-22.

He, W., Huang, Z., Wei, Z., Li, C., & Guo, B. (2019). TF-YOLO: An Improved Incremental Network for Real-Time Object Detection. *Applied Sciences*, *9(16)*, 3225-3240.

Hirabayashi, M., Sujiwo, A., Monrroy, A., Kato, S., & Edahiro, M. (2019). Traffic light recognition using high-definition map features. *Robotics and Autonomous Systems*, *111*, 62-72.

Iihs. (2012). Red light running (Online). Retrieved from https://www.iihs.org/topics/red-light-running

INRIA's project-team IMARA, Mines ParisTech's CAOR. (2009). Traffic light recognition dataset (Online). Retrieved from http://www.lara.prd.fr/benchmarks/trafficlightsrecognition.

Jensen, M. B. Nasrollahi, K., & Moeslund, T. B. (2017). Evaluating state-of-the-art object detector on challenging traffic light data. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 9-15.

John, V., Yoneda, K., Qi, B., Liu, Z. & Mita, S. (2014). Traffic light recog-nition in varying illumination using deep learning and saliency map. *17th International IEEE Conference on Intelligent Transportation Systems (ITSC), IEEE2014*, 2286-2291.

John, V., Zheming, L., & Mita, S. (2016). Robust traffic light and arrow detection using optimal camera parameters and GPS-based priors. *2016 Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, 204-208.

Ju, M., Luo, J., Zhang, P., He, M., & Luo, H. (2019). A simple and efficient network for small target detection. *IEEE Access*, *7*, 85771-85781.

Kim, H.-K., Park, J. H., & Jung, H.-Y. (2018). An efficient color space for deep-learning based traffic light recognition. *Journal of Advanced Transportation*, *2018*.

Kim, J., Cho, H., Hwangbo, M., Choi, J., Canny, J., & Kwon, Y. P. (2018). Deep Traffic Light Detection for Self-driving Cars from a Large-scale Dataset. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 280-285.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (online)*, Retrieved from http://robotics.stanford.edu/ronnyk. *5* 7.

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. *European conference on computer vision*, *Springer*, 740-755.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9905 LNCS*, 21-37.

Loshchilov, I. & Hutter, F. (2017). SGDR: Stochastic Gradient Descent with Warm Restarts. *ICLR*.

Martinez-Alpiste, I., Casaseca-de-la Higuera, P., Alcaraz-Calero, J., Grecos, C., & Wang, Q. (2019). Benchmarking Machine-Learning-Based Object Detection on a UAV and Mobile Platform. *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, 1-6.

Martinez-Alpiste, I., Casaseca-de-la Higuera, P., Alcaraz-Calero, J., Grecos, C., & Wang, Q. (2020). Smartphone-based object recognition with embedded machine learning intelligence for unmanned aerial vehicles. *Journal of Field Robotics*, *37(3)*, 404-420.

Mogelmose, A., Trivedi, M. M. & Moeslund, T. B. (2012). Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *IEEE Transactions on Intelligent Transportation Systems*, *13(4)* 1484-1497.

Müller, J., & Dietmayer, K., (2018). Detecting traffic lights by single shot detection. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 266-273.

Ouyang, Z., Niu, J., Liu, Y., & Guizani, M., (2020). Deep cnn-based real-time traffic light detector for self-driving vehicles. *IEEE transactions on Mobile Computing*, *19(2)* 300-313.

Ou, Y., Sun, Y., Yu, X., & Yun, L. (2022). Traffic Signal Light Recognition Based on Transformer. *Proceedings of the 12th International Conference on Computer Engineering and Networks*, 1354-1361.

Ozcelik, Z., Tastimur, C., Karakose, M., & Akin, E. (2017). A vision based traffic light detection and recognition approach for intelligent vehicles. *2017 International Conference on Computer Science and Engineering (UBMK)*, 424-429.

Pon, A. Adrienko, O. Harakeh, A. & Waslander, S.L. (2018). A hierarchical deep architecture and mini-batch selection method for joint traffic sign and light detection. *2018 15th Conference on Computer and Robot Vision (CRV)*, 102-109.

Possatti, L. C., Guidolini, R., Cardoso, V. B., Berriel, R. F., Paixão, T. M., Badue, C., De Souza, A. F., & Oliveira-Santos, T. (2019). Trafficlight recognition using deep learning and prior maps for au-tonomous cars. *2019 International Joint Conference on NeuralNetworks (IJCNN). IEEE 2019*, 1-8.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 779-788.

Redmon, J. & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 6517-6525.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement (Online). Retrieved from http://arxiv.org/abs/1804.02767.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advancesin neural information processing systems*, 91–99.

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39(6)*, 1137-1149.

Saini, S., Nikhil, S., Konda, K. R., Bharadwaj, H. S. & Ganeshan, N. (2017). An efficient vision-based traffic light detection and state recog-nition for autonomous vehicles. *2017 IEEE Intelligent Vehicles Symposium (IV). IEEE2017*, 606-611.

Tsai, M.-Y., & Lin, H.-Y. (2022). Nighttime Traffic Light Detection with Arrow Signal Recognition Using a Vision and Learning Based Approach. *2022 IEEE 31st International Symposium on Industrial Electronics (ISIE)*, 594-599.

Van Etten. A. (2018). You Only Look Twice: Rapid Multi-Scale Object Detection In Satellite Imagery (Online). Retrieved from http://arxiv.org/abs/1805.09512.

Vaidya, B. & Chirag N. Paunwala, C.N. (2019). Traffic Sign Recognition Using Color and Spatial Transformer Network on GPU Embedded Development Board. *Proceedings of Computer Vision and Image Processing: 4th International Conference, CVIP 2019*, 82-93.

Wang, C.-Y., Liao, H.-Y.M., Chen, P.-Y., & Hsieh, J.-W. (2019). Enriching variety of layer-wise learning information by gradient combination. *Proceedings of the 2019 International Conference on Computer Vision Workshop, ICCVW 2019*, 2477-2484.

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y.M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207, 02696.*

Weber, M., Wolf, P., & Zöllner, J. M., (2016). DeepTLR: A single deep convolutional network for detection and classification of traffic lights. *2016 IEEE Intelligent Vehicles Symposium (IV)*, 342-348.

Weber, M., Huber, M., & Zöllner, J. M. (2018). HDTLR: A CNN based Hierarchical Detector for Traffic Lights. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 255-260.

Yudin, D., & Slavioglo, D. (2018). Usage of fully convolutional network with clustering for traffic light detection. *2018 7th Mediterranean Conference on Embedded Computing (MECO)*, 1-6.