2022 17th International Conference on
Control, Automation, Robotics and Vision (ICARCV)
December 11-13, 2022. Singapore

2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV) | 978-1-6654-7687-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICARCV57592.2022.10004379

# Vehicle Detecting and Tracking Application Based on YOLOv5 and DeepSort for Bayer Data

Chuheng Wei*
Warwick Manufacturing Group
University of Warwick
Coventry, UK
* Corresponding author: Chuheng.Wei@warwick.ac.uk

***Abstract*—Recently, with the advancement of vehicle electronics hardware and the rapid development of artificial intelligence technology, intelligent transportation technology has begun to play a very significant role in the development of vehicle technology. Vehicle tracking technology based on deep learning has also received more and more attention, especially since the rapid development of deep neural networks. A significant contribution of this paper is an evaluation of the performance of neural networks in relation to the type of input data, and an exploration of future development possibilities. Based on YOLOv5 and DeepSort, a vehicle detection and tracking algorithm was chosen to test the algorithm's performance on the Bayer data. The results are discussed and analyzed.**

***Keywords-vehicle tracking; object detection; Bayer data; YOLOv5; DeepSort***

## I. Introduction

Considering the amount of image and video data present in traffic scenes, the application of computer vision plays a significant role in both road scene monitoring systems and in-vehicle autonomous driving assistance systems [1]. Vehicle detection and tracking is one of the fundamental tasks in the field of computer vision, and is also the basic task of monitoring technology in intelligent transportation systems. With the development of deep learning in recent years, vehicle identification and detection technology based on deep neural networks has made significant progress.

The majority of recent research has centered on techniques based on deep neural networks, focusing on the structure of the backbone [2] and the fusion of features [3] to find algorithms with enhanced performance. Meanwhile, some research institutions have opted to combine visual information with millimeter wave Radar or LiDAR information in order to obtain more accurate detection results [4, 5]. However, relatively few attempts have been made to modify the input image type of neural networks. The objective of this paper is to apply the YOLOv5 object detection technique and the DeepSort tracking technique, which have so far produced better experimental results, to implement the vehicle tracking technique, and to explore the possibilities of their applications on the vehicle tracking technique using the original image (Bayer data) captured by the camera. The following subsections describe the vehicle detection and tracking tasks and Bayer image data types.

### A. *Vehicle detection*

The term "object detection" refers to determining the class and location of a target of interest within an image or video frame, and is a fundamental computing task [6]. In object detection, since the number, size, and shape of objects in each image vary, and often have occlusion truncations, object detection technology is also highly challenging, and has been one of the most studied areas of research scholars since its inception[7]. With the advancement of electronic hardware and the development of artificial intelligence theory, the concept of intelligent transportation has been widely promoted in recent years, and more and more research is being devoted to detecting moving vehicles. As of now, the most widely used and best-performing algorithm is a deep learning-based algorithm[8], and the YOLOv5 algorithm [9] selected in this paper is a deep learning-based algorithm.

### B. *Vehicle Tracking*

As the name suggests, target tracking refers to predicting the state of an object in subsequent frames based on the initial position and size of that object in a video sequence [10]. Tracking does not only involve tracking an object in a video, but also includes context modeling and spatiotemporal information, as well as prediction of target motion information in continuous video images[1]. It provides data security for analysis and understanding of the semantic content of the object in order to support more advanced vision tasks. By doing so, the tracking component is able to parse and understand the target motion state. The topic of vehicle tracking is of great interest in computer vision and intelligent transportation systems, and is closely related to people's everyday lives [11].

Target tracking algorithms can be divided into three main categories according to the research content[12, 13]: namely, classical tracking methods, correlation filter-based vehicle tracking and deep learning-based tracking methods. The DeepSort algorithm [14] selected in this paper is a deep learning-based tracking algorithm.

978-1-6654-7687-4/22/$31.00 ©2022 IEEE

### C. *Bayer data*

In most computer vision algorithms, digital images acquired by digital cameras are analyzed and processed to determine output information, so digital cameras are the main vehicle for computer vision to acquire images. In digital cameras, the image sensor is the most crucial component, and its main function is to acquire images [15]. In currently used digital cameras, there two types of sensors, namely Charged Coupled Devices (CCD) [16] and Complementary Metal Oxide Semiconductors (CMOS) [17]. As monochrome electronic components, CCD and CMOS imaging sensors cannot detect color information, only light brightness. To obtain a color image, a Color Filter Array (CFA) must be applied to the monochrome sensor [18]. The CFA allows each pixel point to sense only one color of light arriving at the sensor, thereby excluding other colors of light [19]. As a result of the addition of the CFA, each pixel point on the sensor can only capture one of the three color components: Red (R), Green (G), and Blue (B).

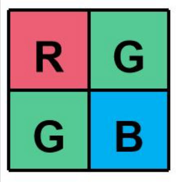


Fig 1. The minimum unit of Bayer CFA

Bayer CFA was developed by Kodak engineer Bayer in 1976 [19], and is the most widely used CFA today. In the Bayer CFA, an alternating set of green and red filters and a set of green and blue filters process incident light. As shown in Figure 1, each 2*2 minimum unit has one red pixel, two green pixels, and one red pixel. The green light accounts for the most light since it is closer to the spectral response band of the human visual system [20], so the human eye is more sensitive to green and is able to distinguish more details, which further enhances the clarity of the recovered color image. The process of Bayer image acquisition is shown in Figure 2.

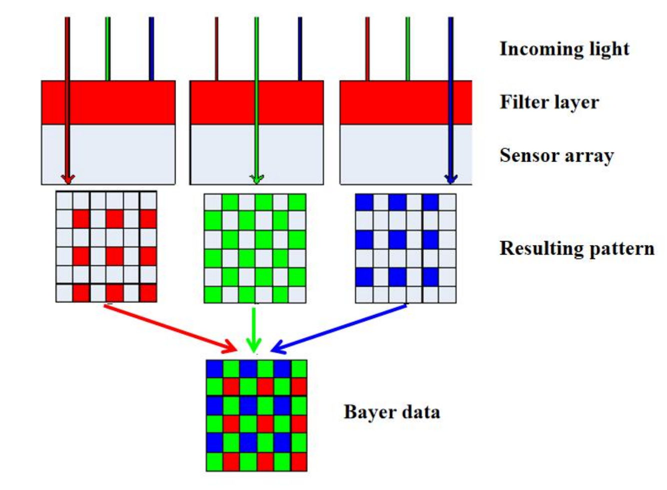


Fig 2. Bayer image acquisition process

The Bayer color filter array produces an image in which each pixel point has only one-color component of red, green, and blue, such an image is called a Bayer data (Bayer image). Figure 3 presents a Bayer image of the RobotCar dataset [21] and a zoomed-in view of its local details.

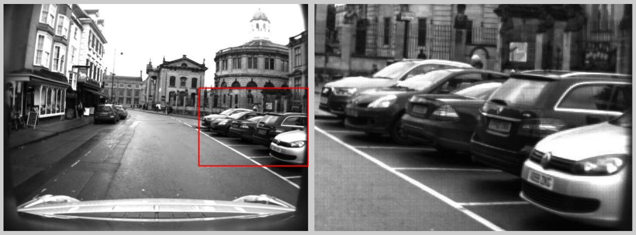


Fig 3. An example of Bayer data [21]

From the figure, it is obvious that although it may illustrate the simple outline and attributes of the image, the details are not smooth, similar to a mosaic, thus resulting in a less than satisfactory visual effect. Thus, both in everyday life and in the field of computer vision, it is necessary to process the data with demosaicing techniques in order to obtain RGB images that are close to human visual perception. Figure 4 briefly illustrates the entire process of image acquisition by the camera.

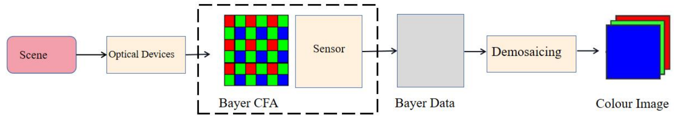


Fig 4. A simplified pipeline figure of the image acquisition

### D. *Research implications*

In this experiment, the main objective is to discover the possibilities of Bayer data for vehicle tracking. The two primary reasons for the importance of this experiment are as follows:

First of all, it is more difficult for humans to understand how deep neural networks work due to their internal convolution process, which is more challenging to read than the convolution process for shallow neural networks[22]. Most institutions engaged in the development of neural networks use RGB images as inputs, primarily because RGB images are more closely related to human perception. However, it remains to be determined whether RGB is the most suitable format for neural networks.

Furthermore, there may be a correlation between the input layer of YOLOv5 and the Bayer image. At the time of acquiring the Bayer image [19], each unit has a minimum of two red sensors, two green sensors, and one blue sensor. As a result, the four pixels in each cell on the Bayer image correspond to three different color components. The focus mechanism of YOLOv5 [23] divides the four points in the 2x2 cell of each image into four separate images and feeds them into the neural network. According to the connection between YOLOv5 and Bayer format images, if the Bayer images are used as the input for YOLOv5, then the segmented images in the focus layer can represent an image that includes the information of each image's color channel.

## II. Methodology

### A. Overview

In order to achieve vehicle tracking, the entire algorithm must consist of two parts, two models for detecting and tracking vehicles. The YOLOv5-s mechanism is used in the vehicle detection part of the model, and the DeepSort algorithm is used in the tracking part. When the two models have been trained, the video sequence composed of Bayer images is input to the model and the output effect is determined.

### B. YOLOv5

YOLOv5 [9] is a relatively new algorithm in the YOLO family [24-27] of algorithms, which is both lightweight and incredibly efficient. According to YOLOv5, the model size of YOLOv5 in the limit can be 90% smaller than that of the YOLOv4 algorithm [27], and both algorithms are equally accurate. The structure of YOLOv5 is shown in Figure 5. Among the main improvements of YoloV5 over previous generations of the Yolo algorithm are the following.

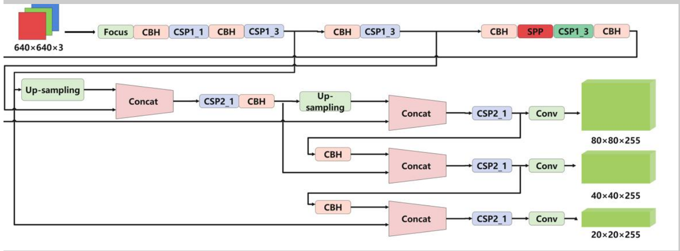


Fig 5. Architecture of YOLOv5

#### 1) Focus mechanism

YOLOv5 proposes a Focus structure within the first layer of the backbone network, and the schematic diagram is shown in Figure 6. The input image size is assumed to be 4×4×3, and after slicing, channel stitching, and convolution, it is transformed into a 2×2×12 image. By adding the Focus structure, the high-resolution image information is transformed from the spatial dimension to the channel dimension, which retains the input information to the maximum extent, but also reduces the size of the input, helping to improve network training and inference.

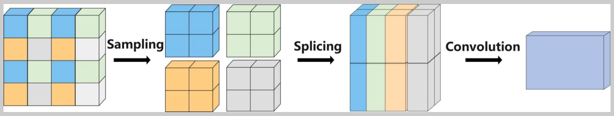


Fig 6. Focus mechanism of YOLOv5

#### 2) CSPNet

The YOLOV5 network structure borrows the design idea of CSPNet [28] and implements the CSP structure in the backbone network. It uses different CSP structures in different parts of the network.As shown in Figure 7, Yolov5 uses the CSP structure with residual components in the backbone network and replaces the residual components with convolution operations in the neck. In the CSP structure, the feature map is divided into two parts, one part of which is used to continue the convolution process and the other part of which is combined with the previously convolutioned feature map. This cross-stage processing reduces the computation and memory space requirements while ensuring detection accuracy.

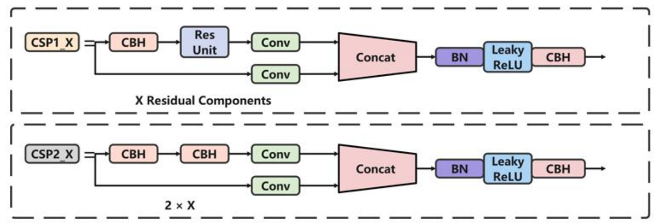


Fig 6. CSPNet of YOLOv5

#### 3) Mosaic Data Augmentation

In addition to other data augmentation methods used for object detection, YOLOv5 makes use of Mosaic data augmentation Firstly, four images are randomly selected, then each image is flipped, scaled, its brightness changed, its saturation changed, and so on. Then the four images are repositioned in the order of top-left, bottom-left, bottom-right, top-right, and then After that, a new image is formed by intercepting the fixed area of each image with a matrix, and then combining them.

### C. DeepSort

DeepSort (Simple online and realtime tracking with a deep association metric) algorithm [14] builds upon Sort (Simple online and realtime tracking) [29].

As the performance of the detection algorithm directly affects the accuracy of the tracking algorithm, improving the performance of the detection algorithm is also a viable option. With the Sort algorithm, the Kalman filter algorithm is combined with the Hungarian matching algorithm, resulting in highly accurate results with improved accuracy and precision. However, the Sort algorithm only uses the Kalman filter to estimate the target position frame by frame and the Hungarian matching algorithm to assign IDs, which can be effective for low-speed motion and simple scenes, but ignores the representational properties of the object as a whole and will fail to track objects in slightly complex scenes. In addition, since the Sort algorithm must balance tracking accuracy and speed, it only uses IOU for matching and does not address target occlusion, resulting in serious problems with ID switching in complex scenarios.

The DeepSort algorithm can be viewed in Figure 7, which improves on the Sort algorithm in the following ways.

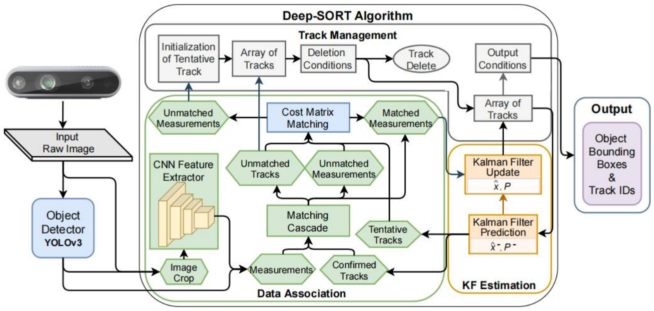


Fig 7. Overview of the DeepSort algorithm [30]

1. In order to track the target when it reappears in an occlusion situation, the DeepSort algorithm adds appearance information to the metric of the Sort algorithm.
2. The Sort algorithm uses an 8-dimensional state matrix to characterize the target state information, and DeepSort processes and manages the target trajectory based on this information, adding new, terminated, and matching states to the trajectory.
3. DeepSort algorithm innovatively uses cascade matching algorithm, adds Marxian distance and cosine distance, and introduces depth features to improve the variability of features.

As a result of these improvements, the detection performance of this algorithm is further improved for detection networks trained on large-scale target tracking datasets. Another advantage of the DeepSort algorithm is low computational power requirements, high real-time performance, and no convolutional neural networks are used for tracking, which further increases its speed..

## III. Experiment and Results

### A. Datasets

In reviewing the mainstream open source datasets on vehicle detection and tracking [31-33], it was found that no open source datasets contained Bayer images as data images.

Therefore, this experiment uses the KITTI dataset [34] based on RGB images as a training set for the vehicle detection and tracking model.

Having trained the neural network, the Bayer images in the RobotCar dataset [21] are then synthesized into a video sequence and the effects of the model are examined on the video sequence.

### B. Apparatus

The parameters of the training platform and validation platform used in this experiment are shown in Table 1

Table 1. Apparatus of the experiments

| **Parts** | **Model** |
|---|---|
| CPU | AMD Ryzen 9 5900HX |
| RAM | 32 GB |
| GPU | NVDIA GeForce RTX 3070 |
| VRAM | 16GB |
| Operating System | Windows 11 64bit |
| Deep learning framework | 1.7.1 |

### C. Results

In this experiment, three types of Bayer image sequences were selected and tested, and some of the vehicle trajectories within the video could be tracked in all three cases. Nevertheless, the specific tracking effect varies widely, and Figure 8 shows the detection result of four consecutive frames in three different input videos. On the figure, the rectangular box represents the detected vehicle, and the irregular line segment of the same color is the moving trajectory at the center of the vehicle, which is the tracking path.

Each of the three videos selected for the experiments was synthesized with a 25 frames per second video using 1500 images taken from the RobotCar dataset. A brief overview of the specific detection results obtained in the neural network for these three different input images is given below.

#### 1) Gray Bayer images

First, we detect RobotCar's original image, which is a single-channel gray Bayer image. Based on the detection results, we found that:

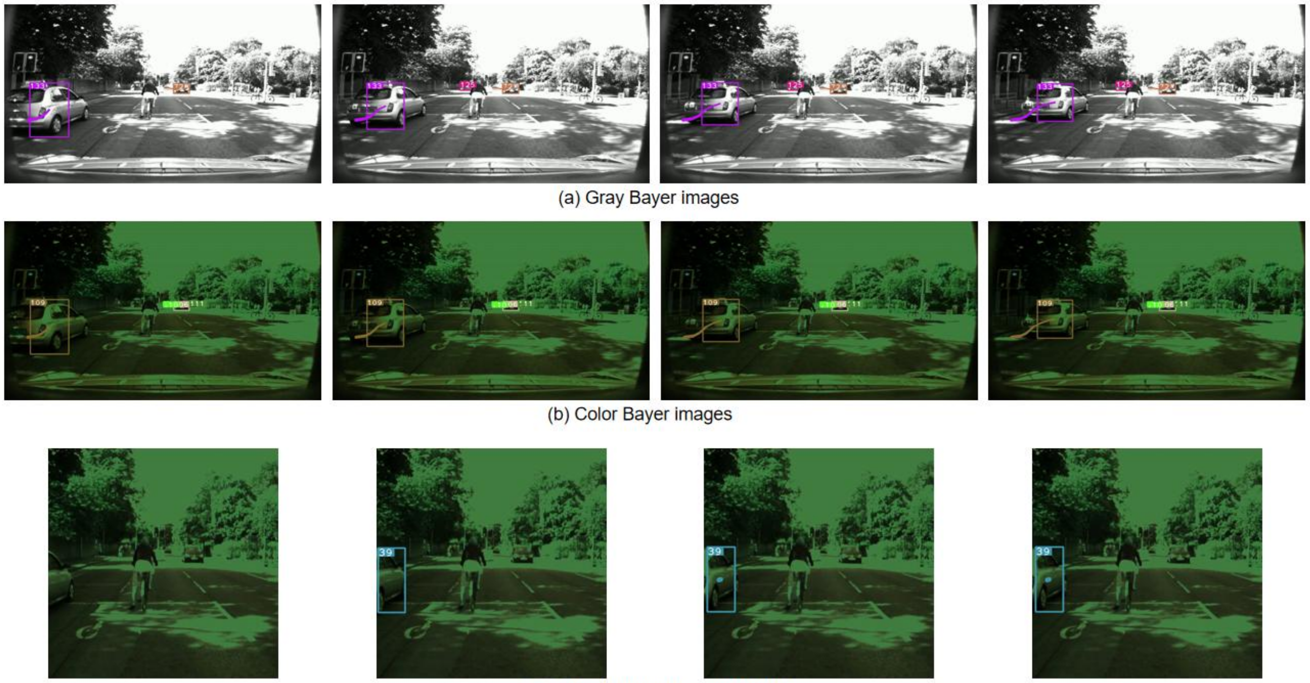
(1) The model is capable of detecting and tracking most of the vehicle motion information in the gray Bayer image.

(2) The model does a better job of tracking obvious vehicles near the image, but the trajectory may be interrupted by changes in light during the tracking process.

(3) The model can detect distant vehicles in gray Bayer images, but vehicle trajectory in complex environments may be more chaotic.

#### 2) Color Bayer images

Due to the use of RGB images to train the deep neural network used in this experiment, color information may affect the effectiveness of detection during the training phase. Consequently, we convert the original Bayer image into a color-meaningful Bayer image by mapping the grayscale information of the pixel points representing red to an RGB color pixel with only red values, and similarly for the pixel points representing green and blue. Following the conversion, the image acquired has certain color characteristics, and the specific detection effects in this case can be summarized as follows.



(a) Gray Bayer images

(b) Color Bayer images

(c) Color Bayer images 640*640

Fig 8. Detection and tracking results of three types of input

(1) This model tracks vehicles in close proximity well and almost never interrupts the tracking trajectory, under almost all incomplete occlusion conditions, without the target disappearing from sight.

(2) The model is not capable of detecting distant vehicles in this type of image.

### 3) *640*640 Color Bayer images*

Due to the similarity of the meaning of each pixel in the Bayer image and the Force mechanism of YOLOv5, this paper speculates that both may provide better performance at the pixel level. Therefore, we intercepted a 640*640 image of interest on a color Bayer image in our experiment. However, the detection effect of video generated by this type of image on neural network is not as good as that of other types of input images, which is specifically reflected in the following two aspects.

(1) The model can detect most vehicles in close proximity, however the vehicle trajectory may be misaligned.

(2) The success rate of this model in recognizing vehicles on this type of image, both near and far, is low compared to the first two types of images, and even complete vehicles that appear clear at a distance are sometimes difficult to distinguish.

A possible explanation for the above problems is that the training set is not a square image. The proportion of the image changes when the image size is modified after being imported into the neural network. If the square image is input, the vehicle may not be recognized because the proportion is different from the previous training image

## IV. Conclusion and Discussion

The purpose of this paper is to attempt to apply Bayer images to a deep learning based target tracking algorithm. In the paper, the vehicle detection algorithm is divided into two parts. The first part of the algorithm consists of a YOLOv5 implementation, whereas the second part is a DeepSort implementation. As there is no Bayer image training set, the experiment uses RGB images as the training set to train the vehicle tracking model. After the model has been trained usingth RGB images, the model's performance is tested using several types of Bayer dataset.

According to the intuitive detection results, image sequences containing Bayer type images can also be used for vehicle tracking, and a relatively satisfactory performance is obtained. Therefore, it can be concluded that Bayer images have the potential to be used for vehicle tracking. If it is feasible to replace RGB images with Bayer images as inputs to the neural network, then on one hand the hardware cost of the camera used in the tracking algorithm and the processing time can be reduced and on the other hand the neural network may be able to perform better in terms of speed or accuracy.

Despite the fact that the results from this experiment indicate to some extent that the application of Bayer images to vehicle tracking is a worthwhile area for further study, the study presented in this paper has some shortcomings.

1. Since the experimental results were not tested on the labeled training set, the experimental results were analyzed only qualitatively and not quantitatively.
2. Due to the lack of a suitable Bayer dataset with labels, the training set used in the proposed model is of RGB type, and better results may be obtained if Bayer images are used as the training set.
3. As there is no Bayer image and RGB image in the same scene, there are no control experiments to compare the two images as a whole.

## V. Future Work

Despite the limitations of the experiment, it does at least demonstrate the potential of Bayer images for target detection and tracking. Future research can further enhance experimental conclusions by enhancing data and conducting rigorous experiments. Regarding the performance effect of Bayer images on vehicle tracking, the following aspects can be further improved experimentally or studied in depth in the future.

1. To find a solution to the training set problem we will need to: (1) manually label the Bayer image dataset. (2) reverse mosaic the existing labeled dataset to convert it into Bayer images and use it as a training dataset.
2. As well as target tracking, neural networks have a wide range of applications in other fields. There is the possibility of using Bayer data for more YOLOv5-based vision tasks.
3. Bayer type images can be used not only for one algorithm of YOLOv5, but also for other algorithms in the YOLO series, or target detection algorithms that use Bayer type images as input data.

## References

[1] S. S. Dukare, D. A. Patil, and K. P. Rane, "Vehicle tracking, monitoring and alerting system: a review," *International Journal of Computer Applications,* vol. 119, no. 10, 2015.

[2] M. N. Chan and T. Tint, "A Review on Advanced Detection Methods in Vehicle Traffic Scenes," in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, 2021: IEEE, pp. 642-649.

[3] Y. Chen *et al.*, "Retracted: Multiscale fast correlation filtering tracking algorithm based on a feature fusion model," *Concurrency and Computation: Practice and Experience,* vol. 33, no. 15, p. e5533, 2021.

[4] N.-E. El Faouzi, H. Leung, and A. Kurian, "Data fusion in intelligent transportation systems: Progress and challenges–A survey," *Information Fusion,* vol. 12, no. 1, pp. 4-10, 2011.

[5] M. Li, X. Tian, Y. Zhang, K. Xu, and D. Zheng, "A review of vision-based vehicle detection and tracking techniques for intelligent vehicle," in *2015 international conference on intelligent systems research and mechatronics engineering*, 2015: Atlantis Press, pp. 402-405.

[6] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems,* vol. 30, no. 11, pp. 3212-3232, 2019.

[7] R. Padilla, S. L. Netto, and E. A. Da Silva, "A survey on performance metrics for object-detection algorithms," in *2020 international conference on systems, signals and image processing (IWSSIP)*, 2020: IEEE, pp. 237-242.

[8] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *arXiv preprint arXiv:1905.05055,* 2019.

[9] Glenn Jocher. "YOLOv5." https://github.com/ultralytics/yolov5 (accessed June 15, 2022).

[10] D. Forsyth and J. Ponce, *Computer vision: A modern approach*. Prentice hall, 2011.

[11] A. Shukla and M. Saini, "“Moving Object Tracking of Vehicle Detection”: A Concise Review," *International Journal of Signal Processing, Image Processing and Pattern Recognition,* vol. 8, no. 3, pp. 169-176, 2015.

[12] Z. Pan, S. Liu, and W. Fu, "A review of visual moving target tracking," *Multimedia Tools and Applications,* vol. 76, no. 16, pp. 16989-17018, 2017.

[13] M. Kumar and S. Mondal, "Recent developments on target tracking problems: A review," *Ocean Engineering,* vol. 236, p. 109558, 2021.

[14] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*, 2017: IEEE, pp. 3645-3649.

[15] M. Mancuso and S. Battiato, "An introduction to the digital still camera technology," *ST Journal of System Research,* vol. 2, no. 2, 2001.

[16] W. S. Boyle and G. E. Smith, "Charge coupled semiconductor devices," *Bell System Technical Journal,* vol. 49, no. 4, pp. 587-593, 1970.

[17] M. Bigas, E. Cabruja, J. Forest, and J. Salvi, "Review of CMOS image sensors," *Microelectronics journal,* vol. 37, no. 5, pp. 433-451, 2006.

[18] S. Yadav and V. Dalal, "An overview of lossless compression scheme for Bayer color filter array images," in *Proceedings of the International Conference and Workshop on Emerging Trends in Technology*, 2010, pp. 1003-1003.

[19] B. E. Bayer, "Color imaging array," 1976.

[20] L. Wang and G. Jeon, "Bayer pattern CFA demosaicking based on multi-directional weighted interpolation and guided filter," *IEEE Signal Processing Letters,* vol. 22, no. 11, pp. 2083-2087, 2015.

[21] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The Oxford RobotCar dataset," *The International Journal of Robotics Research,* vol. 36, no. 1, pp. 3-15, 2017.

[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[23] D. Thuan, "Evolution of yolo algorithm and yolov5: the state-of-the-art object detection algorithm," 2021.

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *arXiv e-prints,* p. arXiv: 1506.02640, 2015.

[25] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263-7271.

[26] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767,* 2018.

[27] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934,* 2020.

[28] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 390-391.

[29] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE international conference on image processing (ICIP)*, 2016: IEEE, pp. 3464-3468.

[30] R. Pereira, G. Carvalho, L. Garrote, and U. J. Nunes, "Sort and Deep-SORT Based Multi-Object Tracking for Mobile Robotics: Evaluation with New Data Association Metrics," *Applied Sciences,* vol. 12, no. 3, p. 1319, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/3/1319.

[31] J. Guo, U. Kurup, and M. Shah, "Is it safe to drive? an overview of factors, metrics, and datasets for driveability assessment in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems,* vol. 21, no. 8, pp. 3135-3151, 2019.

[32] M. Shafiq, Z. Tian, A. K. Bashir, A. Jolfaei, and X. Yu, "Data mining and machine learning methods for sustainable smart cities traffic classification: A survey," *Sustainable Cities and Society,* vol. 60, p. 102177, 2020.

[33] R. A. Hadi, G. Sulong, and L. E. George, "Vehicle detection and tracking techniques: a concise review," *arXiv preprint arXiv:1410.5894,* 2014.

[34] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research,* vol. 32, no. 11, pp. 1231-1237, 2013.