Enhanced Object Detection by Integrating Camera Parameters into Raw Image-Based Faster R-CNN

Chuheng Wei, *Graduate Student Member*, *IEEE*, Guoyuan Wu, *Senior Member*, *IEEE*, Matthew Barth, *Fellow, IEEE,* Pak Hung Chan, Valentina Donzella, and Anthony Huggett

***Abstract*—The rapid progress in intelligent vehicle technology has led to a significant reliance on computer vision and deep neural networks (DNNs) to improve road safety and driving experience. However, the image signal processing (ISP) steps required for these networks, including demosaicing, color correction, and noise reduction, increase the overall processing time and computational resources. To address this, our paper proposes an improved version of the Faster R-CNN algorithm that integrates camera parameters into raw image input, reducing dependence on complex ISP steps while enhancing object detection accuracy. Specifically, we introduce additional camera parameters, such as ISO speed rating, exposure time, focal length, and F-number, through a custom layer into the neural network. Further, we modify the traditional Faster R-CNN model by adding a new fully connected layer, combining these parameters with the original feature maps from the backbone network. Our proposed new model, which incorporates camera parameters, has a 4.2% improvement in mAP@[0.5,0.95] compared to the traditional Faster RCNN model for object detection tasks on raw image data.**

Keywords—Raw image, object detection, faster R-CNN, deep neural networks

I. Introduction

Among various technologies, enabling intelligent vehicles, computer vision plays a vital role in achieving real-time perception and decision-making for autonomous driving [1]. In this context, camera-generated images, which closely resemble what human beings see, are widely used as input data for deep neural networks (DNNs) to perform tasks such as object detection and semantic segmentation [2]. However, the use of RGB images presents certain challenges, as the camera systems used to capture these images require complex image signal processing (ISP) steps, such as demosaicing, color correction, and noise reduction [3]. These steps consume a significant amount of computational resources and increase the overall processing time, which may hinder the real-time performance of the perception system [4].

A. Background

In recent years, there has been growing interest in optimizing the ISP to decrease the time and cost associated with computer vision tasks [5, 6]. One potential approach is to use raw camera images, i.e., images without ISP, as input for DNNs [7]. Despite the potential benefits, studies have shown that using raw data directly in object detection tasks yields inferior performance compared to RGB images, with mAP@[0.5,0.95] differences exceeding 38% [7]. Although it is challenging to apply raw data directly to deep neural networks with good results, many studies [5, 6] have been trying to reduce the steps involved in the ISP algorithm. However, even simplified ISP processes can consume some processing time and hardware resources [4], without major improvement in performance. Therefore, there has been growing interest in exploring alternative methods to further optimize these processes.

Some studies [8-10] have demonstrated that the ISP process can be implemented within neural networks, enabling the conversion of raw images into ISP-processed images and the direct application of the generated images in neural network-based tasks. This suggests that, in principle, it is feasible to bypass the traditional ISP pipeline and use the raw data as input to a single, complex neural network for direct computer vision tasks. Such an "end-to-end" approach could potentially reduce processing time, enhance the real-time performance of autonomous driving systems, and lower hardware costs, thereby promoting the widespread adoption of intelligent vehicles.

B. Research Gaps

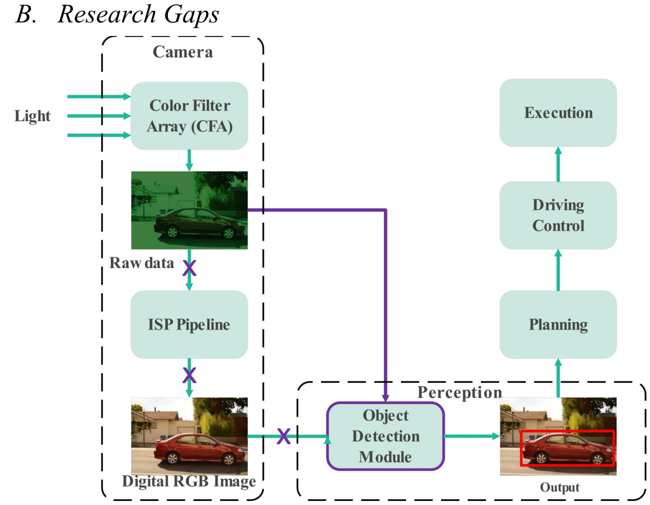

Figure 1. Object detection task pipeline for automated Vehicle .The green arrows represent the traditional method of signal propagation in conventional algorithms, while the purple marks depict the aspect that this paper aims to improve.

Although some research efforts have ensured improved performance of images in neural networks[11, 12], they have not completely addressed the issues of time and energy

Chuheng Wei, Dr Guoyuan Wu and Dr Matthew Barth are with the Department of Electrical and Computer Engineering, the University of California at Riverside, Riverside, CA 92507 USA (*Corresponding author: Chuheng Wei, e-mail: chuheng.wei@email.ucr.edu*)

Pak Hung Chan and Valentina Donzella are with WMG, University of Warwick, Coventry, CV4 7AL, UK.

Anthony Huggett is with ON Semiconductor, Greenwood House, Bracknell, RG12 2AA, UK

consumption caused by the ISP process in autonomous vehicles. Previous studies primarily concentrated on improvements at the ISP level, without modifying the neural network itself. If a neural network more suitable for raw images can be found, it would be possible to eliminate the ISP of the camera entirely, thereby enhancing the efficiency of computer vision systems and reducing the energy consumption of cameras. Fig. 1 provides a more concise pipeline proposed in this study for object detection tasks, where the step of converting raw images to RGB images is removed, and the raw Bayer data is directly used as input to the object detection module.

C. Contributions

This paper presents a novel approach to enhancing object detection in neural networks by innovatively integrating camera parameters as an integral part of the input data. The modifications to the Faster R-CNN algorithm aim to address the limitations of using raw image data directly, improving the overall accuracy of the model. The contributions of this work can be delineated in two significant aspects:

1. Creative Input Integration: In a departure from traditional methods, we incorporate camera parameters as part of the neural network input alongside raw image data. With this integration, the network can leverage both raw pixel information and ISP-related parameters.

2. Modification of Faster R-CNN: This paper introduces a modification to the Faster R-CNN [13] algorithm, particularly in the feature map section. The backbone-generated image features and the features generated by camera parameters are merged, establishing more comprehensive and effective feature fusion.

II. Related work

A. RAW Image Data and ISP

Raw image data, particularly Bayer data [14], plays a critical role in various applications, including automotive vision systems. The Bayer filter mosaic, named after its inventor Bryce Bayer, is the most widely adopted Color Filter Array (CFA) arranged in a specific pattern on the image sensor of a digital camera [14]. The pattern comprises alternating rows of green (G) and red (R) filters, followed by alternating rows of green (G) and blue (B) filters. This configuration is designed to closely emulate the human eye's natural sensitivity to green light, which is why there are twice as many green filters as red or blue ones [14]. The arrangement can be represented as Figure 2.

Figure 2. Bayer CFA pattern

To transform the raw image data into a full-color image suitable for display or further processing, a technique called demosaicing is applied [15]. Demosaicing algorithms interpolate the missing color values at each photosite, considering the neighboring photosites' values, to reconstruct a complete RGB image[16]. However, the reconstructed image often requires further refinement to achieve optimal results. The Image Signal Processor (ISP) plays a crucial role in this context, handling additional processing stages to improve image quality and correct any imperfections.

In the context of automotive vision systems, the ISP plays a significant role in both the manufacturing cost and overall performance of the camera [6]. The need for real-time image processing in these systems demands a high level of computational efficiency from the ISP, which can increase costs due to the requirement for specialized hardware and optimized algorithms [4]. Furthermore, the time consumed by the ISP in processing the images can introduce latency [7], which may impact the system's responsiveness in safety-critical applications. Balancing the trade-offs between processing time, image quality, and manufacturing costs is a critical challenge for designers and engineers working on automotive vision systems.

B. ISP Revisal

Buckler et al. [17] experimentally validated that demosaicing, denoising, and gamma correction are the most critical processes for the performance of computer vision tasks. However, the images used to verify the experiment's effect in their paper were reversely generated by the authors using ISP, not genuine raw images. Recognizing the limitation of their work, Lubana et al. [18] proposed a two-step preprocessing pipeline involving only gamma compression and pixel merging. The images processed through this pipeline showed a significant improvement in detection accuracy, compared to RAW images. Shi et al. [5] proposed a framework based on evolutionary algorithms to find a compact set of ISP configurations for high-level vision tasks. The framework did not strictly dictate the ISP steps to retain and discard, but to eliminate different redundant modules within the ISP based on different datasets and computer vision tasks.

Apart from reducing ISP steps, researchers have also attempted to modify the ISP pipeline to make it more adaptable to neural networks. Wu et al. [6] proposed an ISP specifically for computer vision called *VisionISP*. Although the output it generates looks entirely different from traditional RGB, it performs better in object detection tasks in autonomous driving scenarios.

C. ISP Removal

In traditional computer vision systems, ISP consumes a significant amount of computational resources, processing time, and energy [17, 18]. Removing the ISP process could facilitate the development of vehicular vision systems. In recent years, with the development of deep neural networks, some researchers believe that the capabilities of neural networks can directly handle raw data without pre-processing, i.e., the end-to-end approach. Ratnasingam [2] demonstrated the feasibility of bypassing the hardware image processing step and directly using neural networks to generate RGB images, which, to some extent, validates the image processing capabilities of neural networks. Hansen et al. [4] empirically proved that the process of ISP does influence the accuracy of classification. They also showed that compared to complex neural networks, ISP consumes less memory and computational cost. However, in their experiment, they merely compared the effects of ISP and Convolutional Neural Networks (CNN) on image processing, without verifying whether the CNN that replaces the ISP can be directly integrated into the model.

Experiments by Chan et al. [11] verified that regardless of the type of Bayer image, it is challenging to achieve the performance of RGB images in object detection tasks. However, with some simple padding, filling the blank channels on the original image with the pixel values of the neighboring same-color pixels can make the performance of the Bayer image very close to that of the RGB image. However, the raw images generated by the experiment are not the original images captured by the camera, but are reversely generated from RGB images. However, it is very challenging to fully eliminate ISP impact from the reverse process [18].

III. Methodology

A. Down-sampling RAW Images

In the instances where high-resolution images are being processed, it becomes necessary to perform down-sampling to align with existing computer vision workflows. However, traditional down-sampling methods, such as bilinear interpolation or nearest-neighbor interpolation, are not capable of preserving the Bayer pattern of the original image [7]. Therefore, this paper proposes a specialized method for tenfold down-sampling of PASCAL RAW dataset [19].

The original image size in the PASCAL RAW dataset is 4012 × 6034 pixels (*N=4012, M=6034*), while the provided PASCAL RAW labels have a resolution of 400 × 600 pixels [19]. To match the label resolution, we perform a 10-fold down-sampling (*D=10*) on the images. To maintain compatibility with the RGGB color filter array, we ignore the first 6 rows ($I_r=6$) and the last 6 rows, as well as the first 16 columns ($I_c=6$) and the last 18 columns of the image. This ensures that the ignored rows and columns are multiples of even numbers, since the minimum unit of RGGB is a 2 × 2 cell. The down-sampling formula for each channel (R, G, and B) can be represented as follows:

Channel R and B (i=j):

$$P_{i,j} = \frac{1}{(D/2)^2} \sum_{x=0}^{(D/2)-1} \sum_{x=0}^{(D/2)-1} P_{i\times D+2x,\, j\times D+2y}$$

Channel G (i≠j):

$$P_{i,j} = \frac{1}{2(D/2)^2} \sum_{x=0}^{(D/2)-1} \sum_{x=0}^{(D/2)-1} P_{i\times D+2x+1,\, j\times D+2y} + P_{i\times D+2x,\, j\times D+2}$$

As shown in Fig. 3, the down-sampling process involves mapping each 20×20-pixel block in the original image to a 2 ×2-pixel block in the down-sampled image. We divide each 20 × 20-pixel block into four 10 × 10-pixel regions, corresponding to the four corners of the block. For each 10× 10 region, we compute the average of the same relevant channel values, considering the 25 (5×5) RGGB 2×2 cells.

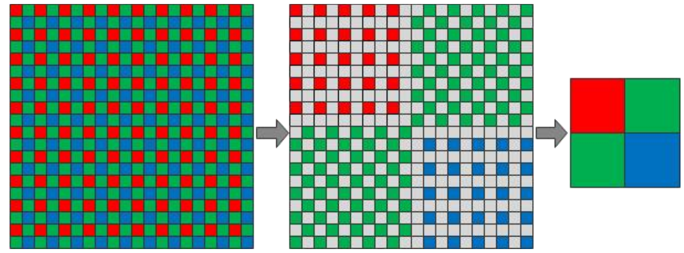

Figure 3. Down-sampling method for Bayer-pattern raw data

By applying this down-sampling technique, we reduce each 20 × 20-pixel block to a 2 × 2-pixel block, resulting in a 10-fold reduction in image size. The final down-sampled image has a resolution of 400 × 600 pixels, which is consistent with the provided PASCAL RAW labels.

The pixels removed from the edges account for 0.6% of the width and 0.3% of the height. Ignoring these pixels leads to a sampling deviation within 3 pixels. As there is already inherent error in the object detection dataset annotation, the label deviation caused by our sampling method can be considered negligible, compared to the inaccuracies introduced by manual annotation.

B. Faster R-CNN Incorporating Camera Parameters

To enable the neural network to implicitly learn an elementary ISP model, we present our methodology for incorporating camera parameters into the *Faster R-CNN* [13] object detection model. This approach leverages the information provided by the camera parameters to potentially improve the model's performance in object detection tasks. To incorporate camera parameters into the Faster R-CNN model, the following modifications are introduced: Fig. 4 illustrates the modified schematic diagram of Faster R-CNN for raw data post-modification.

Input Layer: The model takes both the image and four camera parameters as inputs.

Preprocessing Layer: In the preprocessing stage, the image information is processed using the original image down-sampling scheme proposed in this paper, which down-samples the image to a size of 400 × 600 pixels. The specific dimensions of 400 × 600 pixels have been chosen as they strike a balance between preserving important visual details and reducing the overall data size.

On the other hand, the camera parameters are normalized, which involves scaling and shifting them to a standardized range or distribution. This normalization step ensures that the camera parameters are on a consistent scale, enabling fair comparisons and facilitating the learning process in subsequent stages of the model.

Convolutional Layer: In the convolutional layers for the image, a combination of ResNet-50 [20] and Feature Pyramid Network (FPN) [21] is selected as the backbone for extracting image features. ResNet-50 is a deep convolutional neural network composed of 50 convolutional layers. It employs residual blocks that effectively address the issues of vanishing and exploding gradients. This architecture enables the network to be deeper and easier to train [22]. In addition, FPN is a method used to address scale invariance and multi-scale

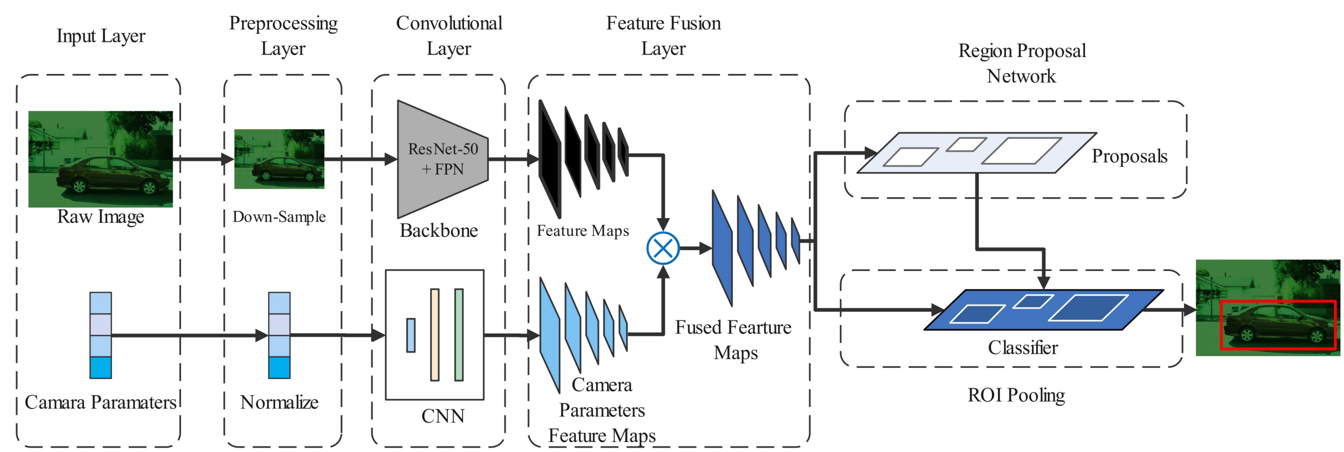

Figure 4. Architecture of the Proposed Faster R-CNN

feature representation in object detection. It introduces up-sampling and fusion operations in the feature maps of ResNet-50 to generate a feature pyramid with different scales. This approach captures object information at different levels, enabling a more comprehensive feature representation.

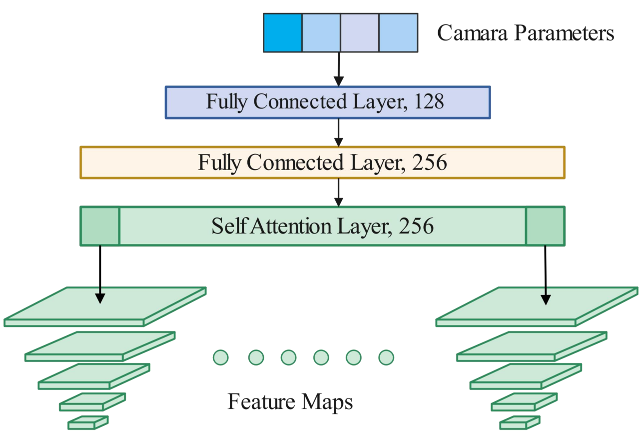

Figure 5. CNN layer of camera parameters

The convolutional part of the camera parameters is depicted in Figure 5. In the CNN layer for camera parameters, we choose a convolutional layer composed of two fully connected layers and one self-attention layer. Through two convolutional operations, the dimensionality of the camera parameters is expanded to 256, matching the channel number of the pyramid features outputted by the image backbone. During training, it was observed that after several iterations, some feature values may become close to zero or very small, while others become extremely large. To address this, a self-attention layer is introduced. Self-attention captures long-range dependencies, redistributes feature importance, and mitigates the impact of problematic values, preventing vanishing or exploding gradients. Finally, each of the 256 feature values is directly expanded to cover the entire feature pyramid layer.

Feature Fusion Layer: After expanding the camera parameter features, a multiplication operation is performed between the expanded camera parameter features and the corresponding feature maps from the ResNet-50 and FPN backbone. The element-wise multiplication operation enables the fusion of the camera parameter information with the spatial information encoded in the feature maps. By incorporating the camera parameter features through multiplication, the resulting feature maps contain both image-based and camera parameter-based information, capturing the joint influence of these factors on the target task.

Other Parts: After feature fusion, the remaining steps in the proposed approach align with the conventional Faster R-CNN framework, which consists of Region Proposal Networks (RPN) [13] and ROI (Region of Interest) pooling [23].

IV. Experiments

A. Dataset

In this experiment, in order to verify the algorithm's performance on raw images from the real world, we selected the PASCAL RAW dataset [19], which includes 4259 high-resolution (4012×6036) 12-bit grayscale RGGB images [19]. The dataset has 6550 objects annotated in accordance with the original PASCAL VOC guidelines [24], including 1,765 cars, 4,077 persons, and 708 bicycles. It should be noted that the PASCAL RAW dataset, despite consisting of images of vehicles, persons, and bicycles, does not contain images captured from a car's perspective, nor does it include complex traffic scenes. However, despite the limitations in viewpoint and content, our experiment focuses more on the type of input data, hence we chose this dataset in this paper.

All images used in the experiment have a size of 400×600. As the PASCAL RAW dataset also provides processed images in jpg format, we used these directly as the RGB dataset for the experiment. The RAW Bayer images are derived from the 4012×6036 Nikon Electronic Format (NEF) images [25] in the PASCAL RAW dataset, which are raw images captured by a Nikon camera, and down-sampled. The camera parameters required in the experiment were extracted from the NEF files.

B. Camera Parameters Selection

Given that all images within the PASCAL RAW dataset are captured utilizing the same camera and lens, the majority of the parameters within the NEF files exhibit uniformity. We have elected to focus on four parameters that display variability, to serve as inputs for our model. The significance of these parameters and their potential impact on the model are as follows:

ISO Speed Rating is a measure of the sensitivity of the camera's sensor to light. Higher ISO values can capture images in darker conditions but at the risk of increasing image noise. Including the ISO speed rating as an input feature could help the network adjust its predictions for objects that might be harder to detect due to noise or low-light conditions.

Exposure Time (Shutter Speed) controls how long the camera's sensor is exposed to light. Longer exposure times can increase the brightness of the resulting image but can also introduce motion blur if objects in the scene or the camera itself is moving. By considering the exposure time, the model might better account for the effects of motion blur or brightness levels on object detection.

Focal Length is a measure of the camera lens's ability to magnify distant subjects' images. Different focal lengths can change the relative sizes of objects in an image and their sharpness. By incorporating the focal length into the model, it could become more robust to variations in object size and detail resulting from changes in the camera's zoom level.

F-number (Aperture) is a measure of the size of the aperture. A lower F-number corresponds to a larger aperture, which lets more light in but reduces the depth of field, making the foreground and background blurrier. Incorporating the F-number into the model may make it more robust to changes in depth of field and the effect it has on object sharpness at different distances from the camera.

In the context of integrating these parameters into Faster R-CNN model, the objective appears to compensate for the absence of an ISP that ordinarily manages these camera settings. While processing raw Bayer data, these parameters significantly influence the final image's appearance. By merging these parameters as features with the Faster R-CNN's backbone features, the model essentially gains awareness of the conditions under which each image is captured. Consequently, this might enhance the model's adaptability in object detection strategies across a broader range of imaging conditions, potentially boosting detection performance.

C. *Quantitative Results*

In this paper, all models were developed, trained, and assessed using the PyTorch framework in Python We trained the models for 500 epochs using a stochastic gradient descent (SGD) optimizer for a batch size of 4, a learning rate of 0.002, and a batch size of 4.

We used three metrics to evaluate the performance of the model: mAP@0.5, mAP@0.75, and mAP@[0.5:0.95]. mAP@0.5 and mAP@0.75 denote the mean average precision calculated using Intersection over Union (IoU) thresholds of 0.5 and 0.75, respectively. As these values increase, the model is required to make predictions that are closer to the ground truth. Moreover, mAP@[0.5:0.95] is a more stringent measure. It tests the performance of the model at varying IoU threshold increments (in this case, from 0.5 to 0.95, with an increment of 0.05) and averages the performance across these increments.

We allocated 80% of the entire dataset for model training, while the rest for model validation. The results of two different models are presented in Table 2 after 500 iterations of training and evaluating different datasets. According to the COCO detection benchmark [26], the results are expressed in terms of mean average precision (mAP). Considering the small number of images and objects for validation, the mAP after 500 epochs is hard to compare with models trained on large datasets.

TABLE I. EVALUATION RESULTS ON THE PASCALRAW DATASET AFTER 500 TRAINING EPOCHS

Datasets	Model	mAP		
		mAP $_{0.5}$	mAP $_{0.75}$	mAP $_{[0.5:0.95]}$
RGB	Faster R-CNN	0.6349	0.5163	0.4345
Raw Bayer Data	Faster R-CNN	0.5892	0.4607	0.3938
Raw Bayer Data	Our Method	0.6213	0.5039	0.4104

As shown in TABLE I, from the evaluation metrics chosen for the experiment, the original Faster R-CNN model performed inferiorly on RAW images than RGB images. However, when we incorporated camera parameters as features into the neural network, there was a clear improvement in mAP. This potentially indicates that camera parameters can compensate for some weaknesses caused by the absence of ISP in the model.

D. *Qualitative Results*

Fig.6 presents a comparative display of the detection results. We showcase the detection results of three representative images from the PASCAL RAW dataset using different algorithms. The three images in (a) are RGB images, while the three images in (b) and (c) are raw Bayer images. The results in (a) and (b) are based on the conventional Faster R-CNN algorithm, while the results in (c) are based on our proposed algorithm. The bounding boxes for the three different objects are represented with different colors, and the top left corner displays the object category and confidence scores. From the presented images, it can be observed that Faster R-CNN on raw data tends to overlook some small objects compared to RGB images, as well as the confidence scores for some objects are lower Furthermore, it can also be observed that the performance of our algorithm on RGB images is close to that of Faster R-CNN directly applied to RGB images.

Furthermore, our algorithm eliminates the need for hardware-intensive Image Signal Processors (ISP) in cameras, opting instead to increase the computational load on the vehicle's processor. This results in reduced energy consumption. Moreover, the cost of ISP hardware is a significant factor in the overall cost of the camera. By ensuring accuracy while reducing vehicular cameras' hardware expenses, our approach makes automotive vision systems more affordable and therefore more widespread.

(a) RGB - Faster R-CNN (b) Raw Data - Faster R-CNN (c) Raw Data - Our Method

Figure 6. Object detection results of Faster R-CNN and our proposed algorithm on RGB images and Raw image data. (The image numbers of the three images from top to bottom corresponding to PASCAL RAW are 2014_000061, 2014_000114 and 2014_000111).

V. CONCLUSION

According to our review of related studies, Bayer images tend to perform worse than RGB images for object detection tasks, possibly due to the lack of an Image Signal Processing (ISP) stage. We therefore incorporated the camera parameter features into the neural network input to compensate for the ISP process in the neural network to a certain extent by using the modified Faster R-CNN model.

Empirical results demonstrated that, compared to the original Faster R-CNN model, our model significantly improved the performance of raw images, bringing their detection effectiveness closer to that of RGB images. This validates the feasibility of compensating for the camera's ISP process within the neural network in deep learning-based tasks. It underscores the viability of utilizing raw data directly in on-board object detection tasks without ISP processing, which could reduce camera costs and power consumption, and enhance real-time responsiveness.

REFERENCES

[1]S. Thrun, "Toward robotic cars," *Commun. ACM,* vol. 53, no. 4, pp. 99–106, 2010, doi: 10.1145/1721654.1721679.

[2]S. Ratnasingam, "Deep Camera: A Fully Convolutional Neural Network for Image Signal Processing," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 27-28 Oct. 2019 2019, pp. 3868-3878, doi: 10.1109/ICCVW.2019.00480.

[3]E. R. Fossum, "CMOS image sensors: Electronic camera-on-a-chip," *IEEE transactions on electron devices,* vol. 44, no. 10, pp. 1689-1698, 1997.

[4]P. Hansen *et al.*, "ISP4ML: The Role of Image Signal Processing in Efficient Deep Learning Vision Systems," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 10-15 Jan. 2021 2021, pp. 2438-2445, doi: 10.1109/ICPR48806.2021.9411985.

[5]Y. Shi, S. Li, X. Jia, and J. Liu, "Refactoring ISP for High-Level Vision Tasks," in *2022 International Conference on Robotics and Automation (ICRA)*, 23-27 May 2022 2022, pp. 2366-2372, doi: 10.1109/ICRA46639.2022.9812052.

[6]C. T. Wu *et al.*, "VisionISP: Repurposing the Image Signal Processor for Computer Vision Applications," in *2019 IEEE International Conference on Image Processing (ICIP)*, 22-25 Sept. 2019 2019, pp. 4624-4628, doi: 10.1109/ICIP.2019.8803607.

[7]W. Ljungbergh, J. Johnander, C. Petersson, and M. Felsberg, "Raw or Cooked? Object Detection on RAW Images," Cham, 2023: Springer Nature Switzerland, in Image Analysis, pp. 374-385.

[8]Y. Cheng, H. Yue, and Y. Mao, "A Lightweight Convolutional Neural Network for Camera ISP," in *2021 IEEE 21st International Conference on Communication Technology (ICCT)*, 13-16 Oct. 2021 2021, pp. 1346-1350, doi: 10.1109/ICCT52962.2021.9658007.

[9]A. Ignatov *et al.*, "AIM 2019 Challenge on RAW to RGB Mapping: Methods and Results," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 27-28 Oct. 2019 2019, pp. 3584-3590, doi: 10.1109/ICCVW.2019.00443.

[10]K. H. Uhm, S. W. Kim, S. W. Ji, S. J. Cho, J. P. Hong, and S. J. Ko, "W-Net: Two-Stage U-Net With Misaligned Data for Raw-to-RGB Mapping," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 27-28 Oct. 2019 2019, pp. 3636-3642, doi: 10.1109/ICCVW.2019.00448.

[11]P. H. Chan, C. Wei, A. Huggett, and V. Donzella, "Raw camera data object detectors: an optimisation for automotive processing and

transmission," 2023， in processing.

[12]C. Wei, "Vehicle Detecting and Tracking Application Based on YOLOv5 and DeepSort for Bayer Data," in *2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 11-13 Dec. 2022 2022, pp. 843-849, doi: 10.1109/ICARCV57592.2022.10004379.

[13]S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence,* vol. 39, no. 06, pp. 1137-1149, 2017, doi: 10.1109/tpami.2016.2577031.

[14]B. Bayer, "Color imaging array," *United States Patent, no. 3971065,* 1976.

[15]D. Menon and G. Calvagno, "Color image demosaicking: An overview," *Signal Processing: Image Communication,* vol. 26, no. 8-9, pp. 518-533, 2011.

[16]X. Li, B. Gunturk, and L. Zhang, "Image demosaicing: A systematic survey," in *Visual Communications and Image Processing 2008*, 2008, vol. 6822: SPIE, pp. 489-503.

[17]M. Buckler, S. Jayasuriya, and A. Sampson, "Reconfiguring the Imaging Pipeline for Computer Vision," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 22-29 Oct. 2017 2017, pp. 975-984, doi: 10.1109/ICCV.2017.111.

[18]E. S. Lubana, R. P. Dick, V. Aggarwal, and P. M. Pradhan, "Minimalistic Image Signal Processing for Deep Learning Applications," in *2019 IEEE International Conference on Image Processing (ICIP)*, 22-25 Sept. 2019 2019, pp. 4165-4169, doi: 10.1109/ICIP.2019.8803645.

[19]A. Omid-Zohoor, D. Ta, and B. Murmann, "PASCALRAW: raw image database for object detection," ed: Stanford Digital Repository, 2014.

[20]K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[21]T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117-2125.

[22]C. Lee, H. J. Kim, and K. W. Oh, "Comparison of faster R-CNN models for object detection," in *2016 16th International Conference on Control, Automation and Systems (ICCAS)*, 16-19 Oct. 2016 2016, pp. 107-110, doi: 10.1109/ICCAS.2016.7832305.

[23]R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440-1448.

[24]M. Everingham and J. Winn. "Pascal visual object classes challenge 2007 (voc2007) annotation guidelines." http://host.robots.ox.ac.uk/pascal/VOC/voc2007/guidelines.html (accessed 04, 27, 2023).

[25]"Nikon Electronic Format (NEF)." https://www.nikonusa.com/en/learn-and-explore/a/products-and-innovation/nikon-electronic-format-nef.html (accessed 05,03, 2023).

[26]T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 2014: Springer, pp. 740-755.