The International Journal of Engineering and Information Technology

journal homepage:www.ijeit.misuratau.edu.ly

MISURATA UNIVERSITY

An Automated Detection Model of Vehicle Identification Number Using YOLOv5

Alkhadafe, H.
Faculty of IT,
Sebha University,
has.alkhadafe@sebhau.edu.ly

Khalleefah, Z.
Faculty of IT,
Sebha University,
za.khalifa@sebhau.edu.ly

Nasir, I.
Faculty of Engineering,
Sebha University,
ibr.nasir@sebhau.edu.ly

Abstract— **The identification plate of vehicles is not an effective way of identifying vehicles because it can be stolen, removed, or altered. Therefore, the present research suggests utilizing the advantages of using the vehicle identification number (VIN) instead of vehicle plates in determining vehicle identities.**
This paper proposes an automated detection model of vehicle identification number using YOLOv5.
The detection model was tested using Roboflow dataset, which consists of 2797 images of quality 256. The experimental results reach 80.3% in mAP 0.5:0.95 and 99.4% in mAP 0.5.
Index Terms— **vehicle identification, automated detection, VIN, YOLOv5,**

I. INTRODUCTION

For official identification purposes, vehicles are required to have a vehicle registration plate or license plate, which is a metal or plastic plate attached to the front and back of the vehicle in order to identify vehicles. All countries require registration plates for road vehicles such as cars, trucks, and motorcycles. For example, a license plate is shown in Fig 1. The license plate can be deliberately forged and removed in cases of fraud or replaced with stolen plates; moreover, due to the current circumstances in Libya, some vehicles can also be driven on the roads without plates, resulting in difficulty in identifying vehicles.

Figure 1 License plates

Also, all vehicles have a vehicle identification number (VIN), which is a distinctive and unique number assigned to each vehicle when it is manufactured and it can be used to identify motor vehicles. For example, a vehicle identification number is shown in Fig. 2.

Received 30 Apr , 2024; revised 08 May, 2024; accepted 15 Mar 2024.
Available online 08 Aug, 2024

The vehicle identification number (VIN) consists of 17 alphanumeric characters. It does not include the letters I, O, Q, or Z to avoid confusion between numbers and letters [1]. The advantage of using the VIN to identify a vehicle is that it is difficult to altered or removed in comparison with a vehicle license plate, which can be easily removed and replace with a stolen plate [2]. Due to the advantages of the vehicle identification number (VIN), the present research suggests using an automated system based on the VIN to identify vehicles.

Figure 2 vehicle identification number (VIN)

In general, an automatic recognition VIN system consists of several steps, namely preprocessing, image detection, and recognition. Pre-processing involves filtering or histogram equalization, which is used to improve contrast in images. Using image detection techniques, the area of interest (ROI), which is a portion of an image, can be determined and extracted during the image detection step. In the recognition step, the VIN is recognized. This research uses digital image processing techniques in an effort to discover a new approach for VIN recognition. Developing a detection system that uses object detection techniques to find and detect the VIN is the primary goal.

Many different sectors have made substantial use of detecting object approaches, including robotics [3], self-driving cars [4], monitoring [5], and healthcare [6]. There are two primary categories into which the most advanced detection methods can be divided: detectors with two stages and one stage When compared to one-stage detectors, two-stage detectors often yield improved accuracy but at a larger computational cost.[7]. Two-stage object detectors first generate region proposals referred to as regions of interest (RoI), then classify and refine those proposals. This two-step approach leads to high accuracy but is computationally more expensive.

The two-stage detector's initial algorithmic work is Region CNN (RCNN)[8], followed by Fast Region-Based CNN (Fast-RCNN) [9], and then it was enhanced by Faster Region-Based CNN (Faster-RCNN)[10]. In the detection of objects field, Faster-RCNN remains a highly competitive algorithm because of its exceptional performance. Following that, algorithms such as Feature Pyramid Network (FPN)[11] and Mask RCNN[12] have suggested modifications to fix issues with Faster RCNN, which improves its performance and significantly enriches its components. The one-stage detector generates the object's category probability and position coordinate values directly, without requiring a separate area proposal stage like the two-stage detector. This means that the final detection result can be obtained immediately after just a single detection. As a result, the one-stage detector has a faster overall detection speed compared to the two-stage approach. [13]. One of these types of algorithms is You Only Look Once (YOLO)[14], following that Single Shot MultiBox Detector (SSD)[15] then it was enhanced by Retinanet [16]. Following that enhanced to YOLOv2[17] YOLOv3[18], YOLOv4 [19], and YOLOv5 are based on YOLO. While the two-stage object detection technique yields higher prediction accuracy, more recently, literature has emerged that offers findings about how well YOLOv5 does in object detection compared with SSD Like drone detection[20],[21], Mask Detection[22], Person Identification Using Ear-Biometrics[23], Object Detection for Street-level[24], Weed Detection in Mulched Onions[25]. Due to the advantages associated with the yolov5 model for detection of objects such as license plates of cars [26], [27], and Unmanned Aerial Vehicles (UAVs)[28]. the proposed approach in this paper adapted this technique for the detection of VIN.

II. YOLO NETWORK STRUCTURE

YOLOv5 is a model from the YOLO family of computer vision models that is widely used for object detection due to its high stability and reliability. It is known for its ease of implementation and training, as well as its exceptional precision in object detection, placing it among the top-performing single-stage algorithms in terms of real-time accuracy. For this reason, the YOLOv5 detector has been adopted in the present approach to detect the relevant features.

The network structure of YOLOv5 generally follows the previous YOLO models. The backbone feature extraction network uses a CSPDarknet architecture [29], and the input is modified by incorporating a focus structure, slicing the input image, reducing its size, and increasing the depth. These modifications are aimed at speeding up the feature extraction process. Additionally, the CSP2 structure is implemented in the neck part of the network to enhance the fusion capability of the network features. The optimization function uses a combination of Adam [30] and SGD [31], and the network structures contain convolution kernels and residual components. This allows the network depth to be changed by controlling the number of residual components in the convolutional layers, while the network width can be adjusted by controlling the number of convolution kernels. As a result, YOLOv5 has developed four model variations, ranging from small (YOLOv5s) to large (YOLOv5x), to accommodate different size requirements.

The overall network structure of YOLOv5 consists of an input layer, a backbone for feature extraction, and a head for feature aggregation. Fig 3 shows the YOLOv5 structure diagram.

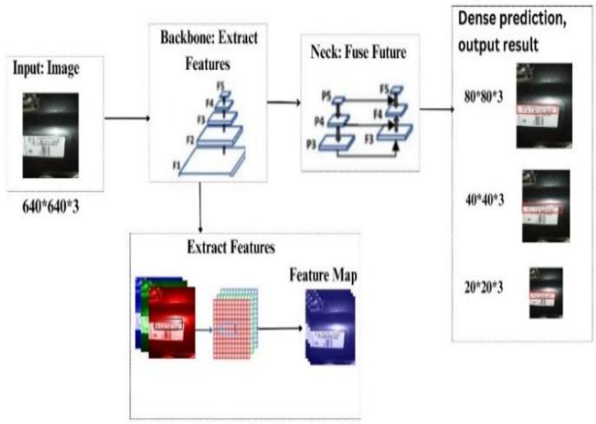

Figure 3 YOLOv5 structure diagram

A. Input Layer and Backbone

The input layer uses a focus structure to process the input images for optimization and analysis. This structure resizes the input images and enhances their feature depth, resulting in accelerated feature extraction. The backbone is responsible for extracting features from the input image. The core network of YOLOv5 is based on CSPDarknet, which makes use of CSP blocks to reduce computing costs while improving feature representation.

B. Neck

combines Spatial Pyramid Pooling Fusion (SPPF) and a modified Cross Stage Partial Path Aggregation Network (CSP-PAN). This fusion enhances feature integration, enabling accurate object detection across various scales and complexities in the image.

C. Output

Non-Maximum Suppression (NMS) is used as a screening criterion for box regression prediction. The algorithm takes into account the overlap area, aspect ratio, and center point distance between the predicted and real boxes. By applying NMS, redundant boxes belonging to the same detected object are filtered out, resulting in accurate and concise object detection outcomes.

III. PROPOSED APPROACH

Fig4 shows the framework of the proposed approach for the detection of VIN which consists of several stages, including data collection, pre-processing, and data augmentation. The overall process, as illustrated in Fig. 4, aims to train YOLOv5 using a specific configuration.

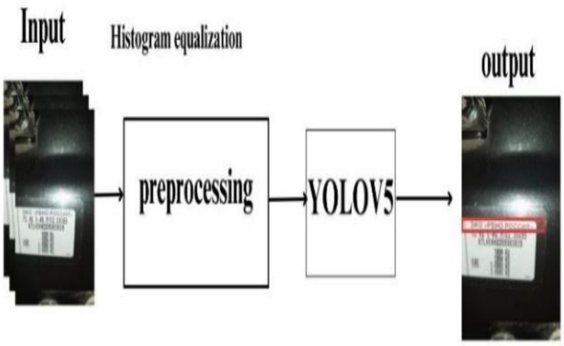

Figure 4 the overall framework of the proposed VIN detection method

1. Dataset

A dataset consisting of 3395 VIN images was obtained and labeled from Roboflow[32]. Roboflow offers tools to prepare and annotate image/video datasets for training computer vision models, including data augmentation, labeling, and dataset management. The platform's dataset capabilities are central to simplifying the computer vision development lifecycle. the data set of VIN is divided into 2797 training images, 494 validation images, 104 test and quality images 256*265. Dataset comprised samples captured from diverse perspectives within the vehicle shown in Fig5. The pictorial conditions exhibited significant variability, including instances affected by noise, rain, divergent lighting, and differing levels of reflection.

Figure 5 different VINs

2. Pre-processing

Prior to the training process, the data underwent preprocessing procedures, which included improving the visual appearance of images by equalizing the distribution of pixel values using Histogram equalization a technique used in image processing to enhance contrast.

3. YOLOv5 Network

In this paper, to detect the VIN the network structure, which is explained in the previous section is used. It is real-time model inference [33],[34], making it well-suited for applications that require fast and accurate object detection.

III. EXPERIMENT AND RESULT

I. EXPERIMENT ENVIRONMENT

The Colab-hosted Jupyter Notebook is used in this work; the experimental setup includes Python 3.10.12, and Torch 2.1.0+cu121. All model is trained, validated, and tested using the same hyper parameters on the T4 GPU. Table 1 shows the experiment's exact parameters.

Table 1Training parameters

Training Parameters	Value(category)
Epoch	200
Batch Size	32
Image Sizes	256*256

II. EVALUATION CRITERIA

The typical metrics used to evaluate the effectiveness of object detection algorithms include precision, recall, average precision (AP), mean average precision (mAP), model parameter count, floating-point operations (FLOPs), and frames per second (FPS) of processed images [35]. In this study, the evaluation focused on two specific metrics: mAP0.5 and mAP0.5:0.95.

Average Precision (AP), also known as mean Average Precision (mAP), is a widely adopted metric for object detection assessment. It calculates the average detection precision across varying recall levels, allowing for performance comparisons between different object detection models [35].

Intersection over Union (IoU) is a measure used in object detection to quantify the degree of overlap between predicted and ground truth bounding boxes. It represents the ratio of the intersection area to the union area of the two boxes Fig 6. The AP metric incorporates IoU to assess the quality of the predicted bounding boxes [36].

The mAP0.5 metric represents the mean average precision at an IoU threshold of 0.5, while mAP0.5:0.95 represents the mean average precision across a range of IoU thresholds from 0.5 to 0.95. These metrics provide a comprehensive assessment of the model's ability to accurately detect and localize objects within the images.

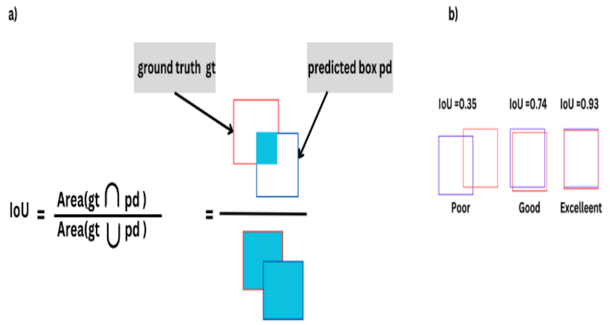

Figure 6 Intersection over Union (IoU). a) The IoU is calculated by dividing the intersection of the two boxes by the union of the boxes; b) examples of three different IoU values for different box locations

III. MODEL SELECTION EXPERIMENTS

In this study, comparison tests were performed using input image size settings 256 on the VIN dataset. The mAP@0.5 value for the YOLOv5 model is shown in Fig 7 Image results showed the performance of the YOLOv5 model in object detection at an IoU threshold of 0.5. The value of mAP@0.5 increased as the model was trained, reaching a stable level around epoch 30. The mAP is shown as 0.5:0.95.

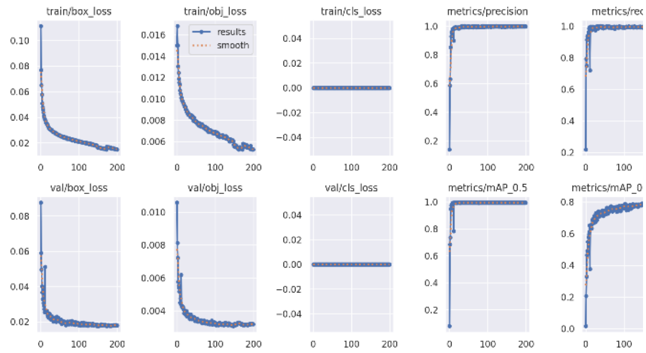

Figure 7 result training yolov5

For the model trained as the detection rate reached 80.3%.The mAP50 is exceptionally high at 99.4%, indicating outstanding performance at the 0.5 IoU threshold. This finding suggests that the model accurately positions the objects, demonstrating a substantial agreement between the predicted bounding boxes and the ground truth bounding boxes. It indicates that the model's predictions closely match the actual location and size of the objects in the images, showcasing impressive performance and implying suitability for tasks requiring accurate object detection. However, the mAP50-95 of 80.3% is much lower compared to the mAP50, indicating that objects that are partially obscured, far away, or have slight differences in appearance may pose challenges for the model. To evaluate whether the best model will detect VIN and interpret them as VIN, predictions were made for the additional test dataset using the trained YOLOv5 models of 104 images of sizes 256 for VIN card, 102 images (98%) were detected. And 2 images (2%) are missing detection (false negatives), which indicates how many VINs will be missed in images recorded by the system. Fig 8. Shows the normalized confusion matrix of predictions of unfamiliar VIN, which is present in the test dataset based on the YOLOv5 model.

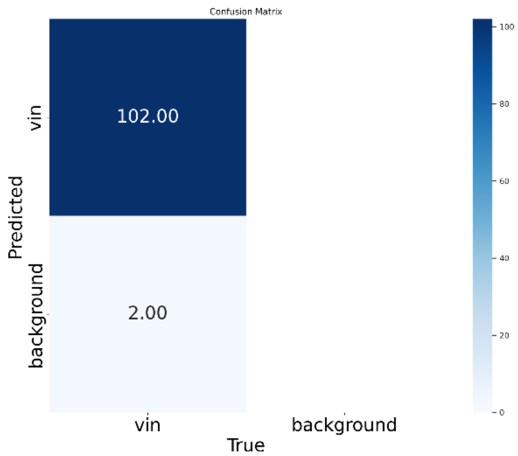

Figure 8 Normalised confusion matrix of predictions of unfamiliar VIN in the test dataset, based on the best YOLOv5 model. The y-axis (True) is the broad taxonomic class of unfamiliar VIN

Evaluation of the model's performance for different image sizes 256, 384, 512, and 640. Table A shows that the model performs better at the smaller image size of 256 in terms of mAP. But not as fast as larger images.

Image size	mAP 0.5	mAP 0.5:0.95	Time m/s
256	99.4%	80.3%	3.5
384	98.3%	79.5%	1.4
512	88.9%	71.2%	1.8
640	73%	56%	5.7

Table 2 performance for different images

To evaluate the model's suitability for real-world inference, a set of images that were not included in the training dataset were selected for testing. As an example, the images shown in Fig 9 illustrate the type of data used for this evaluation. The diverse set of vin image examples was used to assess the model's real-world inference capabilities beyond the training data.

Measuring the accuracy of VIN detection across this challenging test set provided valuable insights into the model's detection and identification performance.

The comprehensive evaluation of this out-of-sample data validated the model's ability to generalize and perform reliably in practical, real-world scenarios.

(a) (b) (c) (d)

(e) (f)

Figure 9 The pictorial conditions (a) rain ,(b)reflection, (c)(d) divergent lighting,(e)(f) noise .

IV.CONCLUSION

This research paper proposes a system for detecting VIN codes using the YOLOv5 model. The study focused on training the model using VIN from roboflow data to improve its performance and robustness. The model proved to be effective in successfully detecting VIN.

The experiment's results confirmed the usefulness of the proposed approach, demonstrating the model's capacity to precisely locate items. This demonstrates the system's potential for real-time applications requiring accurate and efficient object recognition.

The utilization of the YOLOv5 model provided a solid foundation for the development of the VIN detection system. Its efficiency and real-time inference capabilities make it well-suited for deployment in various applications.

REFERENCES

[1] J. Celko, "Vehicle Identification Number (VIN)," *Joe Celko's Data, Meas. Stand. SQL*, pp. 175–178, 2010, doi: 10.1016/B978-0-12-374722-8.00023-2.

[2] V. Selvarajan and P. Matheswaran, "Number Plate and VIN

Recognition Using Artificial Neural Networks," vol. 9, no. 4, pp. 11–13, 2020.

[3] G. Zhaoxin, L. Han, Z. Zhijiang, and P. Libo, "Design a Robot System for Tomato Picking Based on YOLO v5," *IFAC-PapersOnLine*, vol. 55, no. 3, pp. 166–171, Jan. 2022, doi: 10.1016/J.IFACOL.2022.05.029.

[4] B. Mahaur and K. K. Mishra, "Small-object detection based on YOLOv5 in autonomous driving systems," *Pattern Recognit. Lett.*, vol. 168, pp. 115–122, Apr. 2023, doi: 10.1016/J.PATREC.2023.03.009.

[5] H. Liu and W. Zhang, "A YOLOv5-based method for detecting vehicles in overhead surveillance video," pp. 811–817, Dec. 2023, doi: 10.1109/CCDC58219.2023.10326536.

[6] S. M. Omer, K. Z. Ghafoor, and S. K. Askar, "Lightweight improved yolov5 model for cucumber leaf disease and pest detection based on deep learning," *Signal, Image Video Process.*, pp. 1–14, Nov. 2023, doi: 10.1007/S11760-023-02865-9/METRICS.

[7] J. Lee and K. il Hwang, "YOLO with adaptive frame control for real-time object detection applications," *Multimed. Tools Appl.*, vol. 81, no. 25, pp. 36375–36396, Oct. 2022, doi: 10.1007/S11042-021-11480-0/FIGURES/12.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016, doi: 10.1109/TPAMI.2015.2437384.

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

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[11] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature Pyramid Networks for Object Detection," *2017 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 936–944, 2016, [Online]. Available: https://api.semanticscholar.org/CorpusID:10716717

[12] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *2017 IEEE Int. Conf. Comput. Vis.*, pp. 2980–2988, 2017, [Online]. Available: https://api.semanticscholar.org/CorpusID:206771194

[13] P. Shf and C. Zhao, "Review on deep based object detection," *Proc. - 2020 Int. Conf. Intell. Comput. Human-Computer Interact. ICHCI 2020*, pp. 372–377, Dec. 2020, doi: 10.1109/ICHCI51889.2020.00085.

[14] J. S. D. R. G. A. F. Redmon, "(YOLO) You Only Look Once," *Cvpr*, vol. 2016-Decem, pp. 779–788, Dec. 2016, doi: 10.1109/CVPR.2016.91.

[15] J. Yi, P. Wu, and D. N. Metaxas, "ASSD: Attentive single shot multibox detector," *Comput. Vis. Image Underst.*, vol. 189, p. 102827, Dec. 2019, doi: 10.1016/J.CVIU.2019.102827.

[16] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," *2017 IEEE Int. Conf. Comput. Vis.*, pp. 2999–3007, 2017, [Online]. Available: https://api.semanticscholar.org/CorpusID:47252984

[17] 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.

[18] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *ArXiv*, vol. abs/1804.0, 2018, [Online]. Available: https://api.semanticscholar.org/CorpusID:4714433

[19] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv Prepr. arXiv2004.10934*, 2020.

[20] S. M. Alkentar, B. Alsahwa, A. Assalem, and D. Karakolla, "Practical comparation of the accuracy and speed of YOLO, SSD and Faster RCNN for drone detection," *J. Eng.*, vol. 27, no. 8, pp. 19–31, Aug. 2021, doi: 10.31026/J.ENG.2021.08.02.

[21] M. Phadtare, V. Choudhari, R. Pedram, and S. Vartak, "Comparison between yolo and ssd mobile net for object detection in a surveillance drone," *Int. J. Sci. Res. Eng. Man*, vol. 5, pp. 1–5, 2021.

[22] R. Iyer, P. S. Ringe, and K. P. Bhensdadiya, "Comparison of YOLOv3, YOLOv5s and MobileNet-SSD V2 for real-time mask detection," *Artic. Int. J. Res. Eng. Technol*, vol. 8, pp. 1156–1160, 2021.

[23] S. Hossain, H. Anzum, and S. Akhter, "Comparison of YOLO (V3, V5) and MobileNet-SSD (V1, V2) for Person Identification Using Ear-Biometrics," *Int. J. Comput. Digit. Syst.*, vol. 15, no. 1, pp. 1259–1271, 2024.

[24] M. G. Naftali, J. S. Sulistyawan, and K. Julian, "Comparison of Object Detection Algorithms for Street-level Objects," Aug. 2022, Accessed: Jun. 09, 2024. [Online]. Available: https://arxiv.org/abs/2208.11315v1

[25] P. R. Sanchez, H. Zhang, S. S. Ho, and E. De Padua, "Comparison of One-Stage Object Detection Models for Weed Detection in Mulched Onions," *IST 2021 - IEEE Int. Conf. Imaging Syst. Tech. Proc.*, 2021, doi: 10.1109/IST50367.2021.9651352.

[26] S. Raj, Y. Gupta, and R. Malhotra, "License Plate Recognition System using Yolov5 and CNN," *8th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2022*, pp. 372–377, 2022, doi: 10.1109/ICACCS54159.2022.9784966.

[27] K. A, B. D. J, S. P. Gandu, J. Hinn T O, R. Kennedy C, and A. Immanuel J, "Systematic Number Plate detection using improved YOLOv5 detector," pp. 1–6, Jun. 2023, doi: 10.1109/VITECON58111.2023.10157727.

[28] P. Sun and X. Ding, "UAV image detection algorithm based on improved YOLOv5," *2022 IEEE 5th Int. Conf. Inf. Syst. Comput. Aided Educ. ICISCAE 2022*, pp. 757–760, 2022, doi: 10.1109/ICISCAE55891.2022.9927693.

[29] 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.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv Prepr. arXiv1412.6980*, 2014.

[31] S. Zheng *et al.*, "Asynchronous stochastic gradient descent with delay compensation," in *International conference on machine learning*, 2017, pp. 4120–4129.

[32] K. Sergeev, "Vin detection Dataset," *Roboflow Universe*. Roboflow, Feb. 2023. [Online]. Available: https://universe.roboflow.com/kirill-sergeev/vin-detection

[33] Y. Zhang, Z. Guo, J. Wu, Y. Tian, H. Tang, and X. Guo, "Real-time vehicle detection based on improved yolo v5," *Sustainability*, vol. 14, no. 19, p. 12274, 2022.

[34] R. Nachet and T. B. Stambouli, "A Real Time Object Detection System for Infant Safe Sleep Based on YOLOv5 Algorithm," *2022 3rd Int. Conf. Embed. Distrib. Syst. EDiS 2022*, pp. 16–21, 2022, doi: 10.1109/EDIS57230.2022.9996513.

[35] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimed. Tools Appl.*, vol. 82, no. 6, pp. 9243–9275, Mar. 2023, doi: 10.1007/S11042-022-13644-Y/TABLES/7.

[36] I. Ahmad *et al.*, "Deep Learning Based Detector YOLOv5 for Identifying Insect Pests," *Appl. Sci. 2022, Vol. 12, Page 10167*, vol. 12, no. 19, p. 10167, Oct. 2022, doi: 10.3390/APP121910167.