Jurnal Infotel
Vol. 16, No. 2, May 2024, pp. 289–301. doi:10.20895/infotel.v16i2.1105

Research Article

A Generic Model of Traffic Monitoring and Vehicle Counting Systems Based on Edge Computing

Hery Heryanto[1,*], Maclaurin Hutagalung[2], Yoyok Yusman Gamaliel[3], Dina Angela[4], Dionisius Pratama[5], Inge Martina[6], and Tunggul Arief Nugroho[7]

[1,5,6]Department of Informatics, Institut Teknologi Harapan Bangsa, Bandung 40132, Indonesia
[2,3]Department of Computer Engineering, Institut Teknologi Harapan Bangsa, Bandung 40132, Indonesia
[4,7]Department of Electrical Engineering, Institut Teknologi Harapan Bangsa, Bandung 40132, Indonesia

*Corresponding email: hery_heryanto@ithb.ac.id

Received: January 31, 2024; Revised: March 12, 2024; Accepted: March 17, 2024.

Abstract: One of the leading global issues, particularly in urban areas, is traffic congestion. The number of vehicles continues to increase and is less balanced by the development of transportation infrastructure, especially landlines, causing more complex problems. A government ministry overseeing and regulating transportation needs an intelligent application system that can provide knowledge to unravel congestion. The problem is how to perform edge computing to reduce latency so that the highway monitoring application system runs in real-time. This research proposes a basic design for a vehicle monitoring application system that can accurately recognize vehicles, count the number of vehicles, and propose an edge computation that brings computation directly to the data source. The dataset is a video of traffic in Bandung and Jakarta. The images in the dataset consist of 4,890 training images, 467 validation images, and 231 testing images. In the proposed model, the YOLOv5 and YOLOv7 architectures accurately detect and count vehicles. The test results show a Mean Average Precision (mAP) value of 99.1 % with an Intersection over Union (IoU) threshold of 50 %. Other results include a precision value of 96.2 % and a recall of 97.7 %. The proposed model can accurately monitor vehicles and reduce latency with an edge computing approach.

Keywords: convolutional neural network, edge computing, internet of things, vehicle counting, traffic monitoring, vehicle counting, yolo.

1 Introduction

The President of the Republic of Indonesia, Joko Widodo, issued Presidential Instruction (Instruksi Presiden or Inpres) Number 2 of 2022 on March 30, 2022. In the Presidential Instruction, one of the issues discussed is accelerating the increase in domestic products in implementing government procurement of goods/services [1]. This policy is a challenge to producing domestic technology to monitor traffic density, vehicle counts, and visualization of vehicle movement data.

On the other hand, many studies have proposed utilizing edge computing to reduce latency by bringing computing closer to data sources, known as edge computing [2–4]. These studies have certainly been the basis for edge computing applications in the current era. The goal is to keep up with the development of deep learning models in real-time object detection or recognition. The obstacle or challenge in Indonesia is the limited infrastructure and internet network for complex computing models. Bandwidth limitations and latency have become an issue, especially in vehicle counting and traffic monitoring applications on the highway. Edge computing can answer problems with the help of Internet of Things (IoT) devices and deep learning models to perform real-time computing through computer vision technology in the form of artificial intelligence (AI) [5]. The deep learning approach in applying highway traffic density, volume monitoring, and vehicle counters can produce high accuracy. By utilizing cameras and drones, the resulting accuracy, when compared to ground truth, reaches 98.26 % [6]. Using drones, the Root Mean Square Error (RMSE) value produced is 17.49 [7]. The resulting RMSE value for satellite images has reached 31.06 for large vehicles and 1,238.61 for small vehicles [8].

The YOLOv5 architecture was used by Song *et al.* [9] to detect objects by combining radar and imagery. The experimental results showed the highest mAP value of 88.8 %. On the other hand, Yu *et al.* [10] used another YOLO architecture, YOLOv7. His research aimed to recognize moving objects in the form of unmanned aerial vehicles (UAVs). The data used as input was in the form of images. The highest mAP value produced in his research was 92.2 %. YOLOv5 and YOLOv7 architectures were the reference deep learning architectures in the study. The mAP value was used as a measure for comparison. One of the research challenges is getting the vehicle detection accuracy rate close to 100 %. It is because a model that detects vehicles with maximum accuracy, or 100 %, is required to count vehicles accurately.

Another activity that affects the accuracy of object detection is image annotation. Research by Zheng *et al.* [11] and Tang *et al.* [12] showed that annotation dramatically affects the accuracy rate, increasing accuracy by more than 8 %. Limited datasets during training and some factors that caused objects to be obstructed by other objects can cause object detection failure.

The high accuracy produced by some previous studies must be accompanied by data processing speed to apply deep learning models in real life. Accuracy is not enough; it needs to be fast or real-time so that the application system response can immediately provide feedback. The large size of the model is another issue, so it is necessary to choose the exemplary deep learning architecture in the vehicle counter application system. The devices working near the sensors do not have specifications as sophisticated as laptops or servers in offices or homes.

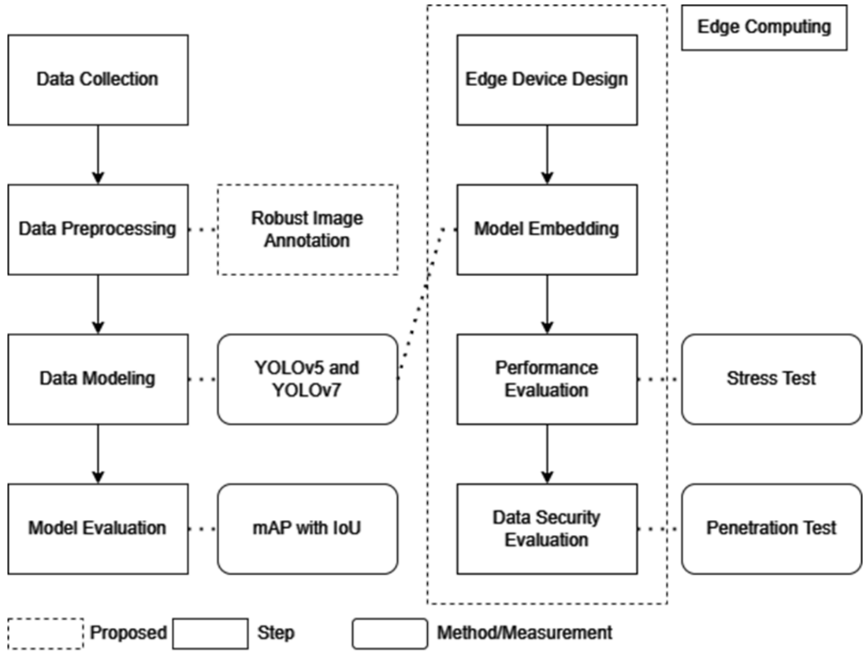

Figure 1: Research flow diagram.

Based on previous research and emerging issues, this study aims to create a basic design of a traffic monitoring and vehicle counting application system using edge computing. The main contributions of this work are concluded as follows:

(a) The proposed application system will be able to monitor traffic density, calculate passing vehicles, visualize data, and even predict vehicle movements throughout Indonesia.

(b) The proposed model supports decisions to unravel congestion, especially in big cities.

This paper uses an iterative application system development methodology based on a deep learning model and edge computing device technology. Section 2 describes the design procedure of the proposed model, both in terms of YOLO architecture and edge computing devices, along with the evaluation techniques. Section 3 provides an overview of the experimental results through visualization and text. The experimental results show that the accuracy produced by the model is close to 100 %, and the application system runs in real-time. A discussion of the results is presented in Section 4, followed by conclusions and future research plans in Section 5.

2 Research Method

This research uses YOLOv5 and YOLOv7 architectures to compare the performance of each architecture. The architecture selection has been based on several related studies. The design of the proposed application system can be seen in Figure 1. The research contributions

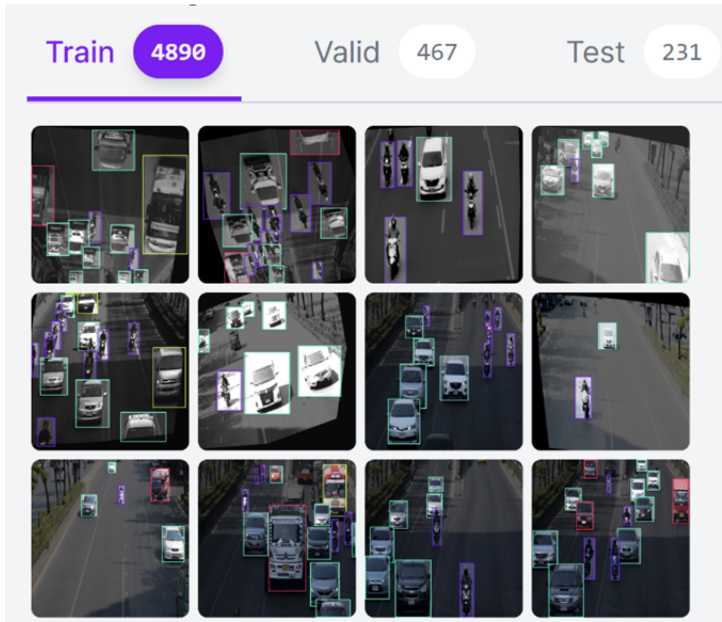

Figure 2: Research datasets for training, validation, and testing.

are shown with dotted lines. Image annotations in preprocessing data and a model application scheme have been integrated into a device capable of performing edge computation. It aims to directly recognize objects (compute) input data from sensors through cameras. The results are then sent to a server that serves as a data centre.

The following is a description of the methods used during the research, the construction of deep learning models, and the design of edge computing devices.

2.1 Dataset

The data collection process was carried out by recording vehicle activity on the highway and several other roads in Bandung City. The sensor used in the research is a camera, capturing data in the morning and afternoon. Lighting conditions are adjusted according to actual circumstances or real needs in the field. For the training process, videos from several places have the characteristics of traffic activity datasets in Bandung and Jakarta.

The number of datasets used in the research is 4,890 images for training, 467 images for validation, and 231 images for testing. Illustrations or examples of datasets can be seen in Figure 2.

2.2 Preprocessing

One of the most critical steps in this research is image annotation. The machine learning technique used in this research is supervised learning. Therefore, annotating vehicle objects is essential before entering the deep learning modelling stage. The annotation technique

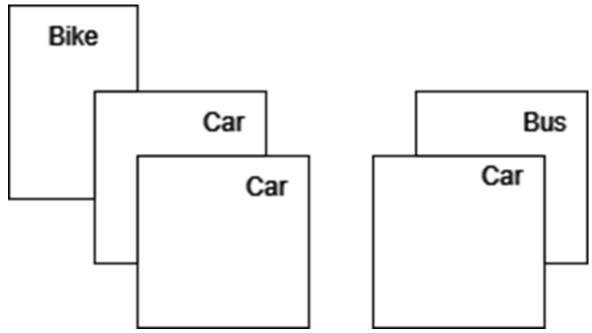

Figure 3: Robust image annotation strategy.

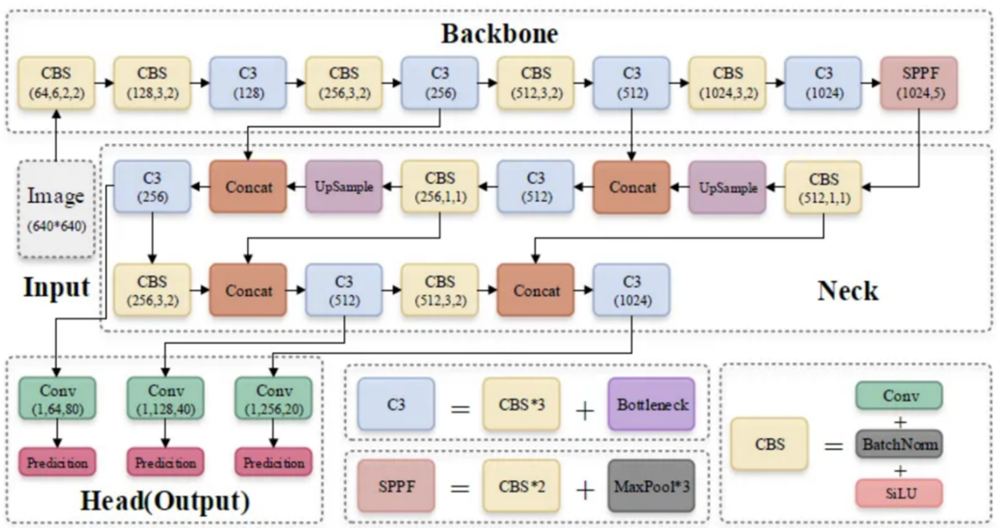

Figure 4: YOLOv5 architecture [13].

used in general is then compared with the proposed robust image annotation technique. The proposed annotation technique uses a stack approach.

This annotation is very effective when the highway is congested. This technique can be seen in Figure 3. The proposed strategy is to annotate the object even if it is obscured by the object in front of it. A simple technique that will be able to handle the detection of vehicles that are obstructed by other vehicles.

Data augmentation has also been carried out in mosaic, shear, crop, and horizontal flipping. The purpose is to accommodate changes in the position of the vehicle or camera.

2.3 Modelling

The model used in the research is a deep learning model with YOLOv5 and YOLOv7 architecture. The YOLO architecture can be seen in Figure 4 [13]. YOLOv5 is divided into three major modules: Backbone, Neck, and Head. The Backbone consists of a stack of multiple

Convolution, Batch Normalization, SiLU (CBS), Concentrated-Comprehensive Convolution (C3), and finally connected with one spatial pyramid pooling-fast (SPPF) module. The CBS module assists the C3 module in feature extraction, while the SPPF module enhances the feature expression capability of the base set. The convolution operation on the CBS module will produce a numerical matrix that depends on the filter size, input size, and convolution steps that have been previously defined. The Neck in YOLOv5 uses Feature Pyramid Networks (FPN) and Pyramid Attention Network (PAN) methods. FPN will upsample the output feature maps (C3, C4, and C5) generated by some convolution dimension reduction operations of the feature extraction network. It is done to generate several new feature maps (P3, P4, and P5) to detect targets of various sizes. The Head in YOLOv5 is the final part of the network. The Head performs object detection and classification based on features that have been processed and extracted through the Backbone and Neck [13].

In other words, convolution is performed within each layer-shaped module. The purpose is to accommodate changes in the position of the vehicle or camera. A significant advantage of the YOLO algorithm over Faster-RCNN is its speed. While Faster-RCNN can recognize only six frames per second, the YOLO algorithm can recognize 19 frames per second [10].

2.4 Evaluation

Intersection over Union (IoU) is a calculation to measure the similarity between the anchor box and the ground-truth bounding box. IoU is calculated by dividing the area of the overlapping area of the two bounding boxes by the combined area of the two bounding boxes. The mathematical formula of IoU is written in (1) [14].

$$J(A,B) = \frac{A \cap B}{A \cup B} \tag{1}$$

The study uses mean average precision (mAP) for the accuracy value. The selection of measuring instruments has followed previous research using the YOLO architecture [9,10]. The mAP was calculated from the accumulated average precision value of the nth object and divided by the total number of objects evaluated. The mathematical formula of mAP is as follows [15]:

$$\mathrm{mAP} = \sum \frac{\mathrm{AP}(n)}{N} \tag{2}$$

The mAP value range is from 0 to 1. A value of 1 means 100 % accuracy. If it is 0.975, the accuracy rate is 97.5 %.

2.5 Edge Computing

Research embedding models built based on the YOLO architecture into edge computing devices. The stages performed in the study can be seen in Figure 5. The purpose of utilizing this technology is to reduce latency, which is a significant obstacle for computer vision applications, especially vehicle counter and traffic monitoring systems. The sensor used is a camera, and the data captured by the sensor is video. If it is processed in the cloud server, it will undoubtedly cause data transmission to be slow because the files or data that must be transferred are large. The edge computing device directly performs vehicle detection

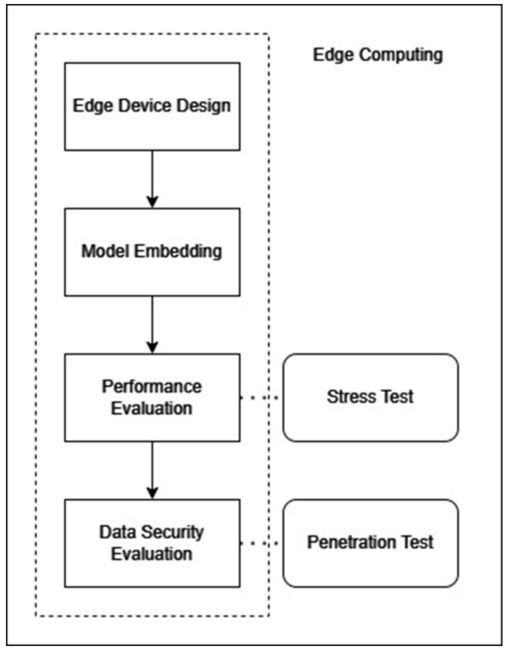

Figure 5: Stages of implementing deep learning models to edge computing devices.

and calculation with an input device as a camera sensor. Data synchronization from the device to the data center is done periodically. The end-to-end encryption method is used for data security to prevent data interception or sniffing by unauthorized parties.

3 Result

Tests were conducted with 60 scenarios. The parameters used are epoch values of 30, 40, 50, 70, and 100. The second parameter is the 16, 32, and 64 batch size. The IoU threshold value is in the range of 50 % and between 50-95 %. Finally, there are two different datasets: baseline system and proposed strategy and robust image annotation. For data augmentation, three combinations were used: 1) flip and mosaic; 2) flip, mosaic, and shear; 3) flip, mosaic, shear, and crop.

3.1 Robust Image Annotation

The first test compared the basic model that uses complete image annotation with the proposed model that considers the overlap between objects or vehicles in a scene (robust image annotation). Figure 6 shows that the proposed model provides a higher mAP value than the base model. The proposed model has offered a mAP value of 0.991 or 99.1 %, while the baseline model only has a mAP value of 0.969, or 96.9 %. The resulting difference is two per cent. It is undoubtedly a significant result. The basis is that it is difficult to increase the mAP value when it has already reached more than 0.95, or 95 %.

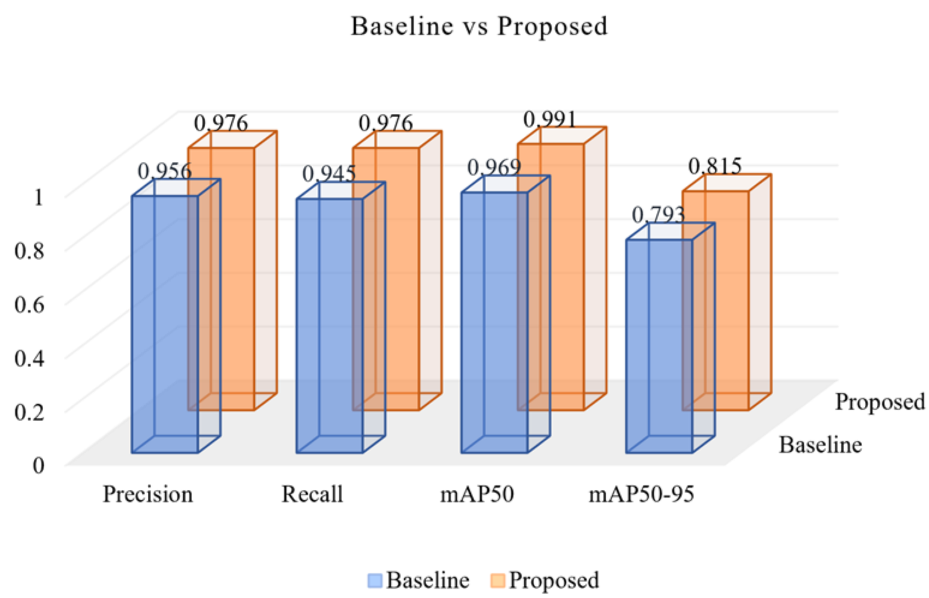

Figure 6: Test results of baseline systems with the proposed model.

The proposed model can be seen in Figure 6, marked in orange. The same phenomenon is seen in the precision and recall values. The IoU value as a threshold also affects the mAP value. For threshold values between 50-95 %, the accuracy value decreased by 18 %, or very significantly, as shown in Figure 6. It is due to vehicles that overlap or are covered when captured by the camera sensor.

3.2 Data Augmentation

Data augmentation in research datasets, especially in vehicle detection and calculation, is needed. This process will help the model to recognize vehicles when image changes occur due to transformations caused by the motion of the object itself, changes in camera angle, or changes in the surrounding environment [16, 17]. In the experiments, three data augmentation scenarios were used. The first combination is the provision of image transformation in the form of flip and mosaic. The second combination is flip, mosaic, and shear. The last scenario is the addition of crop effect to the image.

Based on the observations in Figure 7, the three scenarios do not significantly impact. Interestingly, the first scenario that only provides flip and mosaic effects yields the highest accuracy. The mAP value produced by the model with flip and mosaic augmentation alone is 0.99419. This result differs from the second and third scenarios by a difference of about 0.001, or 0.1 %. The second scenario produces the highest mAP value of 0.99311, and the last scenario is 0.99234.

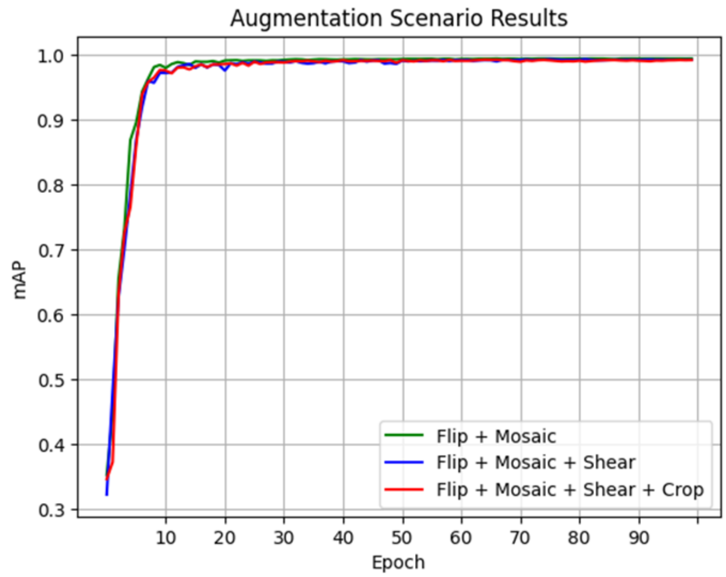

Figure 7: Test results of three data augmentation scenarios against the proposed model.

3.3 YOLO Architecture Hyperparameters

The YOLO model in calculating vehicles is undoubtedly influenced by several hyperparameters [18]. Like other deep learning architectures, a hyperparameter tuning is also needed [19]. The parameters tested in this study are batch size and epoch. Batch size is used to see how fast the model is built, and epoch is used to know the learn.

Figure 6 Test results of baseline systems with the proposed model ing trend of the architecture so that the model information converges and the optimal point is known. The batch sizes tested were 16 and 32. For epoch values, the values tested were 30, 40, 50, 70, and 100. In Figure 8, the hyperparameters that generally affect deep learning models do not significantly impact the YOLO architecture in this study. The significantly different result is caused by using the IoU value as the threshold in vehicle recognition.

3.4 Edge Devices

The edge computing device architecture successfully designed in this research is a device that utilizes a mini PC. This device is directly connected to the camera. The hardware specifications are 1) Core i5 processor, 2) 8 GB of RAM, and 3) 128 GB SSD. An additional device connected is a GPS module that uses a 4G connection as an IoT-based device. The designed architecture can be seen in Figure 9. The device directly processes the data captured by the sensors, and then the YOLO model detects the vehicle and counts it. The data is stored as a data file forwarded to a cloud server. Before transmission, the file is encrypted to keep the data secure. The server in the cloud decrypts the data to be recapitulated and visualized for decision support in road traffic monitoring.

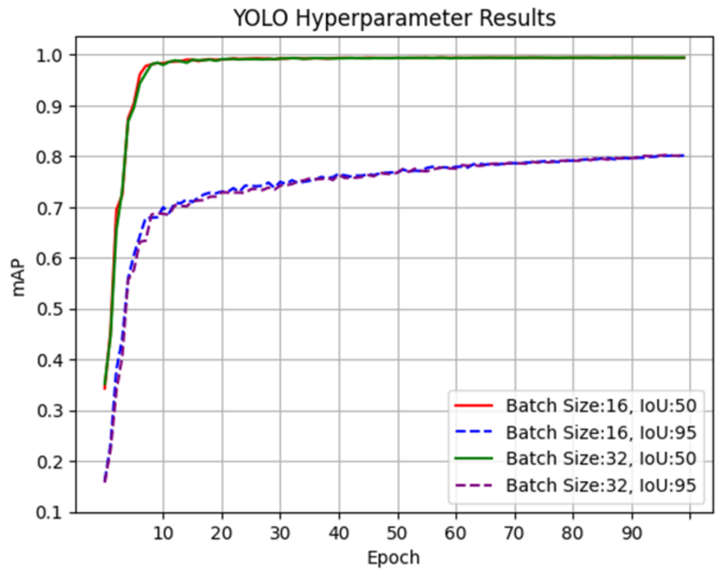

Figure 8: Hyperparameter testing results in the YOLO model.

4 Discussion

Based on the test results, several phenomena have warranted further discussion. The first is the role of image annotation, or image labelling, in the preprocessing stage. The proposed method using the vehicle stack during image labelling can improve accuracy by over two per cent. The measure is the mAP value, which increases from 0.969 to 0.991. It proves that the proposed method contributes to the vehicle counter application system. Supporting data is in Figure 6.

The second thing is the hyperparameters that have no impact. The model has compiled iteratively in the test scenario, but the mAP value has not changed significantly. When viewed from the graphs in Figure 7 and Figure 8, the experimental results show lines that coincide or show no significant change. Therefore, the YOLO architecture has been well established for application, especially in vehicle detection and highway traffic monitoring.

The third thing is the edge computing device model. These devices need to be prepared for reliability, especially when the surrounding environment changes, even tending towards extremes, such as weather changes or other disturbances. This research has not thoroughly discussed the durability and strength of edge computing devices. This research focuses more on the design or basic model only.

The last thing to discuss is the phenomenon of higher mAP values for truck vehicles. The highest accuracy achieved is 0.993, or 99.3 %, with the Truck class. Models more easily detect trucks due to their unique shape than other classes. It is much different from the Bike class, where the shape of the bike is diverse and allows the model to have difficulty recognizing objects in the test dataset.

Table 1 shows that the Truck class has the highest accuracy value in the YOLOv5 or YOLOv7 architecture. This value has been much better than the research conducted by

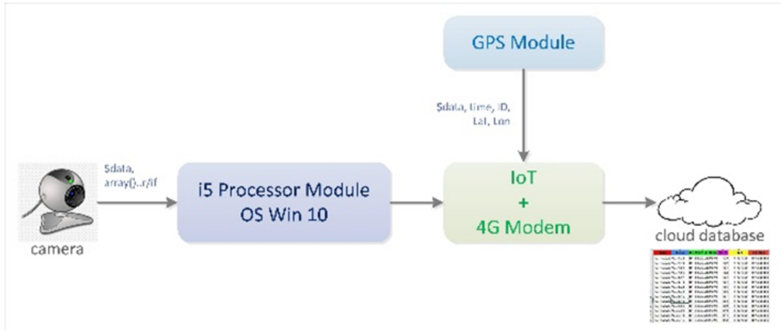

Figure 9: Architecture of the proposed edge computing device.

Table 1: YOLO model performance

Model-Class	Performance			
	Precision	**Recall**	**mAP50**	**mAP95**
v5-All	0.962	0.977	0.990	0.789
v5-Bus	0.961	0.977	0.992	0.833
v5-Car	0.963	0.979	0.991	0.809
v5-Bike	0.960	0.970	0.986	0.696
v5-Truck	0.966	0.982	0.992	0.818
v7-All	0.964	0.975	0.991	0.797
v7-Bus	0.964	0.981	0.992	0.857
v7-Car	0.968	0.974	0.992	0.800
v7-Bike	0.949	0.966	0.986	0.696
v7-Truck	0.975	0.979	0.993	0.835

Xiao *et al.* [20], which only has the highest mAP value of 0.665 or 66.5 %. Further study material for future research is the influence of the IoU value as a threshold that significantly impacts this study. A specific approach is needed to discuss the role of IoU. The difference of more than 0.20 or 20 % needs to be considered.

5 Conclusion

Based on the results of the experiments that have been carried out, several points are concluded in the research. Three issues need to be conveyed. The first issue is that image annotation in the data preprocessing stage has a significant impact. The second dataset increased the mAP value by more than 2 % using the proposed robust image annotation. The highest mAP value in the experiment is 99.1 %, with an epoch parameter value of 70 and an IoU threshold value of 50 %. The YOLOv7 architecture provides a higher mAP value than the YOLOv5 architecture, but the difference is negligible. Based on the experimental results, it can be concluded that the YOLO architecture does not affect the design of the proposed application system. The second is vehicle type, which also showed an influence on accuracy. Trucks always had the highest mAP value or accuracy in all trials, reaching 99.4 %. Last, the design of the proposed application system by embedding the model in the proposed edge computing device can significantly reduce latency. Several obstacles and challenges were encountered during the implementation of the research. Suggestions for future studies or improvements that need to be made include an evaluation to conduct detailed stress and penetration tests. It has been done to test how robust edge computing devices are in carrying out their duties to detect and count vehicles. In detail, the threshold value in the form of IoU shall be investigated in further research to see the optimal threshold value, both for vehicle detection and vehicle recognition.

Acknowledgments

Thanks to the Directorate of Research, Technology, and Community Service, Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia for funding this research. The paper was prepared based on the Research and Community Service Program 2023 study results with contract number 180/E5/PG.02.00.PL/2023.

References

[1] I. P. I. N. . T. 2022, "Instruksi presiden (inpres) nomor 2 tahun 2022 tentang percepatan peningkatan penggunaan produk dalam negeri dan produk usaha mikro, usaha kecil, dan koperasi dalam rangka menyukseskan gerakan nasional bangga buatan indonesia pada pelaksanaan pengadaan barang/jasa pemerintah,"

[2] M. Goudarzi, H. Wu, M. Palaniswami, and R. Buyya, "An application placement technique for concurrent IoT applications in edge and fog computing environments," *IEEE Trans. Mob. Comput.*, vol. 20, pp. 1298–1311, Apr. 2021.

[3] M. De Donno, K. Tange, and N. Dragoni, "Foundations and evolution of modern computing paradigms: Cloud, IoT, edge, and fog," *IEEE Access*, vol. 7, pp. 150936–150948, 2019.

[4] S. N. Shirazi, A. Gouglidis, A. Farshad, and D. Hutchison, "The extended cloud: Review and analysis of mobile edge computing and fog from a security and resilience perspective," *IEEE J. Sel. Areas Commun.*, vol. 35, pp. 2586–2595, Nov. 2017.

[5] F. Liang, W. Yu, X. Liu, D. Griffith, and N. Golmie, "Towards edge-based deep learning in industrial internet of things," *IEEE Internet Things J.*, vol. 7, pp. 4329–4341, May 2020.

[6] S. Li, C. Liu, and F. Chang, "Time-spatial multiscale net for vehicle counting and traffic volume estimation," *IEEE Trans. Cogn. Dev. Syst.*, vol. 14, pp. 740–751, June 2022.

[7] W. Li, H. Li, Q. Wu, X. Chen, and K. N. Ngan, "Simultaneously detecting and counting dense vehicles from drone images," *IEEE Trans. Ind. Electron.*, vol. 66, pp. 9651–9662, Dec. 2019.

[8] G. Gao, Q. Liu, and Y. Wang, "Counting from sky: A large-scale data set for remote sensing object counting and a benchmark method," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, pp. 3642–3655, May 2021.

[9] Y. Song, Z. Xie, X. Wang, and Y. Zou, "MS-YOLO: Object detection based on YOLOv5 optimized fusion millimeter-wave radar and machine vision," *IEEE Sens. J.*, vol. 22, pp. 15435–15447, Aug. 2022.

[10] C. Yu, Y. Liu, W. Zhang, X. Zhang, Y. Zhang, and X. Jiang, "Foreign objects identification of transmission line based on improved YOLOv7," *IEEE Access*, vol. 11, pp. 51997–52008, 2023.

[11] D. Zheng, S. Li, F. Fang, J. Zhang, Y. Feng, B. Wan, and Y. Liu, "Utilizing bounding box annotations for weakly supervised building extraction from remote-sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–17, 2023.

[12] C. Tang, X. Liu, P. Wang, C. Zhang, M. Li, and L. Wang, "Adaptive hypergraph embedded semi-supervised multi-label image annotation," *IEEE Trans. Multimedia*, vol. 21, pp. 2837–2849, Nov. 2019.

[13] H. Liu, F. Sun, J. Gu, and L. Deng, "SF-YOLOv5: A lightweight small object detection algorithm based on improved feature fusion mode," *Sensors (Basel)*, vol. 22, p. 5817, Aug. 2022.

[14] R. Szeliski, *Computer vision*. Texts in computer science, Cham, Switzerland: Springer Nature, 2 ed., Jan. 2022.

[15] M. Hasnain, M. F. Pasha, I. Ghani, M. Imran, M. Y. Alzahrani, and R. Budiarto, "Evaluating trust prediction and confusion matrix measures for web services ranking," *IEEE Access*, vol. 8, pp. 90847–90861, 2020.

[16] K. Seo, H. Cho, D. Choi, and J.-D. Park, "Implicit semantic data augmentation for hand pose estimation," *IEEE Access*, vol. 10, pp. 84680–84688, 2022.

[17] W. Li, C. Chen, M. Zhang, H. Li, and Q. Du, "Data augmentation for hyperspectral image classification with deep CNN," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, pp. 593–597, Apr. 2019.

[18] I. S. Isa, M. S. A. Rosli, U. K. Yusof, M. I. F. Maruzuki, and S. N. Sulaiman, "Optimizing the hyperparameter tuning of YOLOv5 for underwater detection," *IEEE Access*, vol. 10, pp. 52818–52831, 2022.

[19] R. Gonzales-Martínez, J. Machacuay, P. Rotta, and C. Chinguel, "Hyperparameters tuning of faster R-CNN deep learning transfer for persistent object detection in radar images," *IEEE Lat. Am. Trans.*, vol. 20, pp. 677–685, Apr. 2022.

[20] D. Xiao, F. Shan, Z. Li, B. T. Le, X. Liu, and X. Li, "A target detection model based on improved tiny-yolov3 under the environment of mining truck," *IEEE Access*, vol. 7, pp. 123757–123764, 2019.