TEPIAN Vol. 5 No. 4 (December 2024) 143-149 p-ISSN 2721-5350 e-ISSN 2721-5369
https://doi.org/10.51967/tepian.v5i4.3213 © 2024 Tepian Politeknik Pertanian Negeri Samarinda

Implementation of Deep Learning Algorithm for Vehicle Count Monitoring System

M Ridwan Dwi Septian *
Informatics, Gunadarma University, Depok, 16424, Indonesia
ridwandwiseptian@staff.gunadarma.ac.id
**Corresponding Author*

Agustine Hana Masitoh
Information Systems, Gunadarma University, Depok, 16424, Indonesia
agustinehana@staff.gunadarma.ac.id

Intan Meutia Sari
Information Systems, Gunadarma University, Depok, 16424, Indonesia
intan_meutia@staff.gunadarma.ac.id

Submitted: 2024-11-30; Accepted: 2024-12-06; Published: 2024-12-08

Abstract— Vehicle detection plays a crucial role in various applications such as traffic surveillance, license plate recognition, and the development of autonomous vehicles. The You Only Look Once (YOLO) object detection method is renowned for its high-speed real-time object detection capabilities. In this study, YOLO is employed to detect vehicles in images and videos. YOLO treats object detection as a direct regression problem for bounding boxes and class predictions. The aim of this research is to develop a vehicle counting system using the YOLO method. The Midpoint algorithm is utilized to calculate the midpoint between two points in a coordinate plane. Another objective is to analyze the strengths and weaknesses of the method and algorithm in the context of vehicle detection while identifying related research trends. The test results indicate that the system is capable of detecting vehicles with an average accuracy of 92.42% across four different time periods. In the morning, the system detected 156 vehicles (manual count: 147, accuracy: 94.23%); at midday, it detected 246 vehicles (manual count: 225, accuracy: 91.46%); in the evening, 377 vehicles were detected (manual count: 351, accuracy: 93.10%); and at night, the system identified 526 vehicles (manual count: 225, accuracy: 92.58%). This study contributes to the development of a more effective vehicle counting system for smart city applications while also paving the way for further research on vehicle detection under varying lighting and environmental conditions.

Keywords— Accuracy, CNN, Hungarian, Midpoint, Yolo

I. Introduction

Depok City is a municipality located in West Java Province, Indonesia (Widyawan & others, 2019). It is part of the Jabodetabekpunjur metropolitan area and lies to the south of the Special Capital Region of Jakarta. Depok serves as a supporting city for activities in Jakarta and Bogor Regency, with relatively heavy transportation conditions. Increased production in Depok can accelerate economic growth, which, in turn, impacts mobility. However, the city's development and the rising use of motor vehicles have introduced new challenges, particularly in the transportation sector (Widyawan & others, 2019).

According to the West Java Central Statistics Agency (BPS) website in 2023, the number of motor vehicles in Depok City has reached 1,139,603 units (Barat, 2023) The increasing vehicle volume is undoubtedly the primary cause of current traffic congestion. Toll roads are among the routes frequently used by four-wheeled and larger vehicles (L. A. Nugroho et al., 2024).

Vehicle monitoring is crucial for understanding traffic volume, allowing for evaluations and improvements in Depok City. Monitoring systems are highly needed in various fields today, aiming to enhance productivity and security in specific sectors. The implementation of monitoring systems typically relies on object detection through images or videos (L. A. Nugroho et al., 2024).

Various algorithms are used for object detection, including traditional approaches that rely on statistical methods such as Haar Cascade Classifier, Histogram of Oriented Gradients (HOG), Support Vector Machine (SVM), and Deformable Part Model (DPM). In addition, Deep Learning algorithms using Convolutional Neural Network (CNN)-based approaches have become the standard for object detection. Examples include Region-based Convolutional Neural Networks (R-CNN), Single Shot MultiBox Detector (SSD), You Only Look Once (YOLO), RetinaNet, and EfficientDet (Syarif & others, 2023).

Several researchers have conducted related studies. For example, Fitria Rachmawati and Dahlia Widhyaestoeti, in their research titled " Deteksi Jumlah Kendaraan di Jalur SSA Kota Bogor Menggunakan Algoritma Deep Learning YOLO" employed YOLOv3 for detection. Data collection was conducted through video recording, which was then segmented into images. Detection accuracy was influenced by lighting quality, camera-object distance, and object size. For instance, large buses were sometimes detected overlapping with other vehicles, such as trucks. YOLOv3 proved effective in detecting vehicles based on type and size (Rachmawati & Widhyaestoeti, 2020).

Research by Muhammad Fauzan Arif, Ahmad Nurkholis, Sootomosi Laia, and Perani Rosyani, titled "Deteksi Kendaraan Dengan Metode YOLO" focused on developing YOLO to overcome its limitations and enhance overall vehicle detection accuracy. YOLO excels in real-time vehicle detection with high efficiency. Challenges include detecting small and overlapping objects, necessitating further research. YOLO continues to evolve, with potential for integration with other technologies to improve accuracy and reliability(Arif et al., 2023).

Another study by Amar Andra Saifullah, Ega Hegarini, and Bheta Agus Wardijono, titled "Sistem Penghitung Jumlah Kendaraan dan Pendeteksi Kecepatan pada Ruas Jalan Menggunakan Metode Haar Cascade Classifier" aimed to develop a system for detecting the number of vehicles and measuring their speed on roads using the Haar Cascade Classifier method. In real-world testing with traffic videos, out of 40 vehicles passing, only 16 were successfully detected, yielding a detection success rate of 40%. Variations in vehicle speeds were also calculated and visualized (Hegarini et al., 2024).

Based on these introductions, monitoring systems are essential for tracking the number of four-wheeled vehicles. Researchers conducted a study to assess vehicle volume on toll roads and tested this system on the Depok toll road, specifically the Cinere-Jagorawi Toll located on Insinyur Haji Juanda Street (Jl. Ir. H. Juanda), at different times of the day (morning, afternoon, evening, and night).

For this monitoring system, researchers applied a Deep Learning algorithm using the YOLOv8 architecture, incorporating the Hungarian Algorithm for object tracking. The Intersection over Union (IoU) Matching method was used to measure the overlap between two bounding boxes (ground truth and predicted). After obtaining the predicted box, the next step was to use the Midpoint Circle Algorithm to determine the center point of the predicted box. The purpose of this algorithm is to identify the center point, and when this point intersects with a predefined line, the vehicle count increases. These findings were implemented into an application.

II. RESEARCH METHODS

A. *Deep Learning*

Deep learning is a specialized method within machine learning that uses neural networks in sequential layers to learn from data iteratively. It is particularly useful for understanding patterns in unstructured data. The complex neural networks in deep learning are designed to mimic the functioning of the human brain, allowing computers to be trained to handle abstract concepts and problems that are not well-defined (Aryanto et al., 2023).

Figure 1 illustrates that a neural network in deep learning has three main layers: the input layer, which receives the input data; the hidden layer, which processes and extracts important features; and the output layer, which generates predictions or the final result (Aryanto et al., 2023).

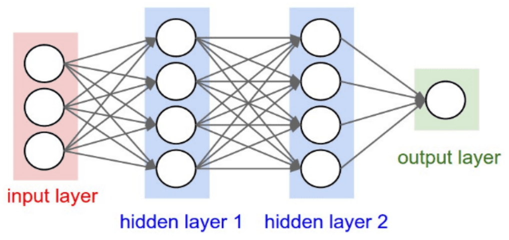

Figure 1. Architecture Neural Network

B. *Convolutional Neural Network*

CNN (Convolutional Neural Network) is a specialized architecture in Deep Learning that is highly effective in processing image data and tasks related to Computer Vision. Unlike regular Neural Networks, where all neurons are connected to each other, CNN employs a more efficient approach by connecting each neuron only to specific parts of the previous layer (Rasywir et al., 2020).

In Figure 2, the main components of CNN are the convolutional layer, pooling layer, and Rectified Linear Unit (ReLU). The convolutional layer applies filters to the input data to extract important features. The pooling layer performs down sampling to extract dominant features. ReLU is a non-linear activation function used to introduce non-linearity into the model. In the CNN architecture, the convolutional layers are arranged sequentially, followed by optional pooling layers. This architecture is repeated several times until the important features are effectively captured. Fully connected layers follow the convolutional layers, and the SoftMax function is used for probability estimation (Christian & Al Idrus, 2023).

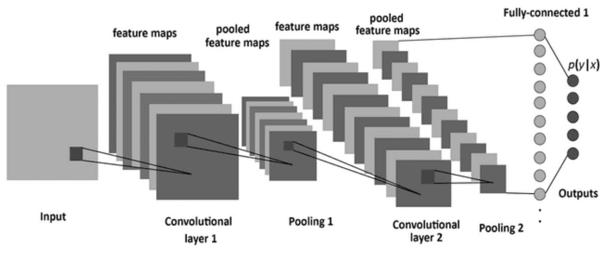

Figure 2. Architecture CNN

C. *YOLOv8*

YOLOv8 is the latest model in the YOLO family developed by Ultralytics, creating a significant difference from YOLOv5 in terms of architecture and developer experience. Since its first launch in 2015, the YOLO model has gained popularity due to its high accuracy and small model size (Yanto et al., 2023). Over time, YOLO has garnered attention from the computer vision community, with several previous versions maintained in the C language using Darknet. The development of YOLOv8 began when its creator, Glenn Jocher, followed YOLOv3 in PyTorch and eventually launched YOLOv5, which became "State-of-the-Art" thanks to the flexible Python structure. YOLOv8 is the result of research and development over the past six months and was officially launched in January 2023 (Wahyuni & Sulaeman, 2022).

In the anchor box-free approach, the model directly predicts the center of the object from the objects present in the image without using offsets from predefined anchor boxes. This approach helps reduce the complexity in making box predictions and can improve the model's performance by avoiding issues that arise from selecting the right anchor box for each object in the image (Putri, 2023).

The YOLOv8 model does not use anchor boxes, but directly predicts the center of the object. As a result, the model does not need to compute offsets from anchor boxes to determine the object's position. This approach helps reduce the number of box predictions and speeds up the Non-Maximum Suppression (NMS) process, which is an advanced processing stage that filters candidate detections based on confidence scores and avoids excessive detections (Kim et al., 2023).

Additionally, there have been improvements in the convolutional structure of YOLOv8 that positively impact the model's performance. In the system, the 6x6 convolution in YOLOv5 is replaced with a 3x3 convolution. The main building block has also changed, with C2f replacing C3 (Zhang, n.d.). This module contains components such as Conv, BatchNorm, and SiLU (Swish Activation Function). In C2f, all outputs from the Bottleneck are combined, whereas in C3, only the output from the last Bottleneck is used (Niu et al., 2022).

Although the Bottleneck in YOLOv8 is similar to YOLOv5, the size of the first convolution kernel has changed from 1x1 to 3x3. This indicates a return to the ResNet block defined in 2015. In the 'neck' section, features are directly merged without forcing the same channel dimensions, reducing the number of parameters and the overall tensor size (Sama & Sharma, 2023) (refer Figure 3).

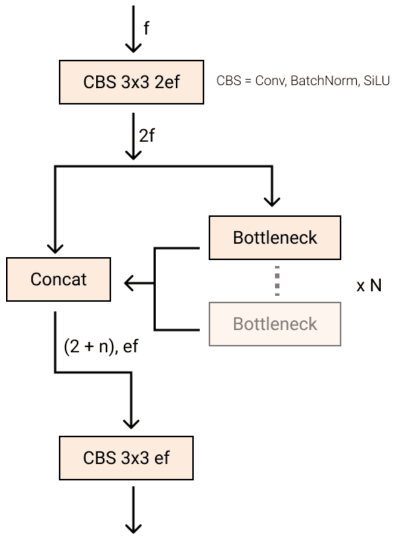

Figure 3. Anchor Box-Free Visualization in YOLOv8

D. Hungarian Algoritm

he Hungarian Algorithm in the context of Object Tracking is a method used to match objects detected in the current video frame with objects detected in the previous frame. This algorithm is commonly used in Multi-Object Tracking (MOT) systems to ensure that each object maintains a consistent identity across frames. The basic principle of the Hungarian Algorithm solves the optimal assignment problem. In the context of object tracking, this problem involves matching between (Ünlü, 2021) :

1. New Detections: objects detected in the current frame.
2. Existing Tracks: object identities determined from the previous frame.
3. It can use metrics such as:
4. Euclidean Distance: for location,
5. IoU (Intersection over Union): for bounding boxes,
6. Appearance Features: if using visual features.

E. IoU (Intersection over Union)

In object detection, the output generated is a bounding box (bounding box) that represents the system's prediction of the position of a predefined object. This bounding box represents the position of the object in an image. To evaluate the object detection model that we have trained, there are several methods, one of which is using the Intersection Over Union (IoU) method. IoU utilizes the bounding boxes present in Figure 4 (Yu et al., 2021).

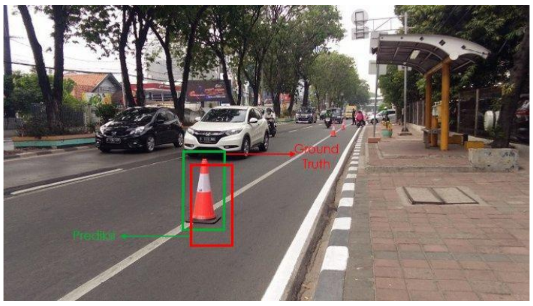

Figure 4. Bounding Box

Intersection Over Union (IoU) is a value based on the statistical similarity and diversity of sample sets, with the aim of evaluating the overlapping area (the intersecting area) between two bounding boxes: the predicted bounding box and the ground truth bounding box. Therefore, the requirement for applying IoU is having both bounding boxes. By applying IoU, we can determine other evaluation metrics such as precision, recall, and so on. The formula for intersection over union is as follows (Yu et al., 2021) equation (1):

$$IOU = \frac{area(BB_{prediction} \cap BB_{groundTruth})}{area(BB_{prediction} \cup BB_{groundTruth})} \tag{1}$$

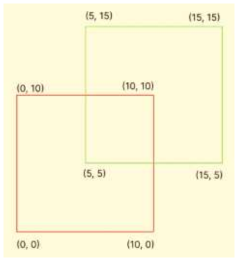

Figure 5. Bounding Box with Coordinates

The explanation from Figure 5 shows a bounding box with coordinates where the Area of Overlap is: 5 * 5 = 25 and the Area of Union is: Area of the Red Bounding Box

+ Area of the Green Bounding Box - Area of Overlap = 10 * 10 + 10 * 10 - 25 = 175. The IoU (Intersection over Union) is calculated as: Area of Overlap / Area of Union = 25 / 175 ~ 0.14, indicating a poor IoU (Ramadhani et al., 2024).

Based on the illustration above, the equation to obtain the IoU value is simply a comparison of the area of intersection divided by the area of union. By dividing these two areas, we get the Intersection Over Union (IoU) score. After obtaining the IoU score, the rule for assessing whether the IoU score is good or bad is that the more the bounding boxes overlap, or the closer the distance between the predicted bounding box and the ground truth bounding box, the better the score. More clearly, Figure 6 illustrates the IoU scores, which we categorize as Poor, Good, and Excellent.(Yu et al., 2021).

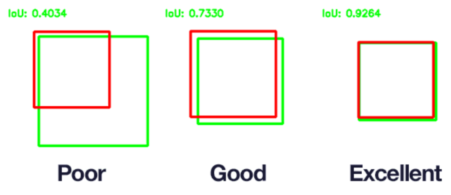

Figure 6. An example of computing Intersection over Unions for various bounding boxes.

From the illustration above, it can be concluded that the score will be higher if the distance between the predicted bounding box and the ground truth bounding box becomes smaller (the overlapping area between the two bounding boxes increases) (Adianto, 2021).

F. Midpoint Algoritm

The Midpoint Algorithm is one of the computer graphics algorithms used to draw lines, circles, or curves with a discrete approach on a pixel grid. This algorithm works iteratively to determine which pixels should be drawn based on the midpoint's position relative to the desired shape. The computational procedure takes several values or sets of values as input and processes them to produce an output, making the algorithm a sequence of computational steps that transform input into output. The Midpoint Algorithm for Line Drawing was developed by Pitteway in 1967. A brief overview of its steps: (G. S. Nugroho & Hazmin, 2022):

1. Plot the first pixel (x_1, y_1).
2. Determine the sign of the decision variable. If the decision variable is positive, increment both x and y dan add $(a + b)$ to the decision variable. Otherwise, increment x and y, and add (a) to the decision variable.
3. *Plot the pixel at position* (x, y).
4. Repeat step 2 until the last pixel (x_2, y_2).

Steps/Process stages in the solution:

This algorithm starts by determining the radius and the center point at (x_k, y_k) (Figure 7). The algorithm generates pixels in 8 directions from the 4 straight points (0°, 90°, 180°, 270°). Each of these 4 points has two pixels generation directions determined by the algorithm. The loop stops when the closest angle is a multiple of 45° (45°, 135°, 225°, 315°), or when the condition $x = y$ is met."

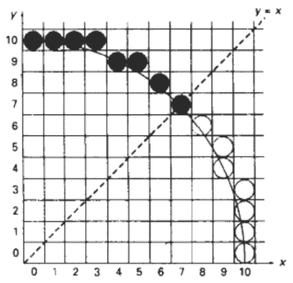

Figure 7. Circle

The midpoint algorithm uses a function to determine the next pixel with the following equation (2).

$$f_{circle}(x, y) = x^2 + y^2 - r^2 \tag{2}$$

The formula applies the basic mathematical formula for a circle, equation (3), which is :

$$(x - x_c)^2 + (y - y_c)^2 = r^2 \tag{3}$$

To determine the next point using the function, the parameters used are $x_k + 1, y_k - \frac{1}{2})$ These parameters represent the position of the midpoint for the decision point in determining the next pixel. The value of x will always be incremented (+1), However, the next value of y is determined by the position of the midpoint, which is obtained from the function's equation. If the midpoint is inside the circle $(f(x, y) < 0)$ or exactly on the circle, Then the value of y will not be changed. However, if the midpoint is outside the circle, $(f(x, y) > 0)$ the value of y will be decreased to stay on the circle's path (Tamsir et al., 2021).

G. Implementation

For the implementation, the researcher explains the flow of the process in the implementation. Figure 8 shows the stages.

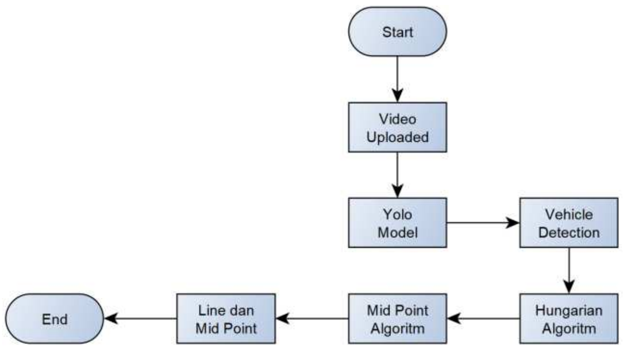

Figure 8. System Implementation Flow

The initial process begins by inputting the video results from the research. Figure 9 shows the result of the video capture.

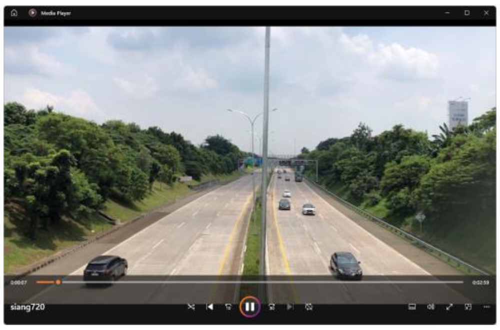

Figure 9. Video Uploaded

The next step is that the uploaded video enters the YOLO Model architecture stage to detect objects. After the objects are detected, the next task is to separate the objects, selecting only vehicles with four or more wheels (Vehicle). Figure 10 displays the results of object detection using YOLOv8.

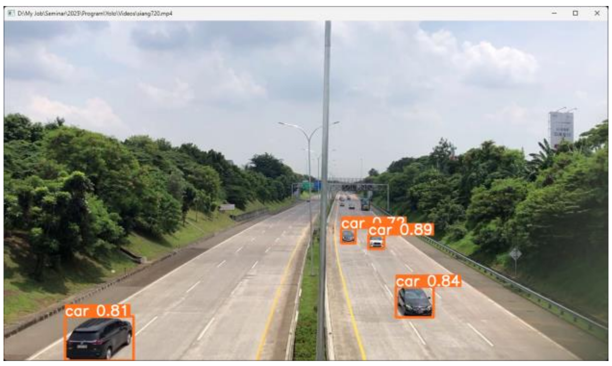

Figure 10. Get Vehicle from YOLOv8

After detecting the vehicle objects, the next step is to enter the Hungarian Algorithm process stage with the IoU method using equation (4), and the results are displayed in Figure 11.

$$
\begin{aligned}
x_1, y_1, x_2, y_1 &= Overlap \\
x_1, y_1, x_2, y_1 &= x_1, y_1, x_2, y_1 \qquad (4) \\
w, h &= x_2 - x_1, y_2, - y_1,
\end{aligned}
$$

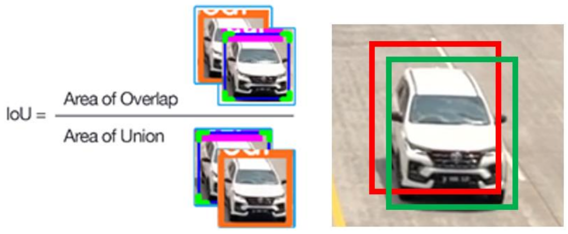

Figure 11. Bounding Box with Hungarian Algorithm

After obtaining the predicted bounding box, the next step is to use the Midpoint algorithm to find the center point of the bounding box, using equation (5) :

$$cx, cy = \frac{x_1 + w}{2}, \frac{y_1 + h}{2} \qquad (5)$$

Figure 12 shows the result of the calculation using the Midpoint Algorithm

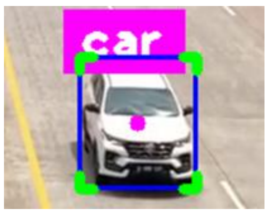

Figure 12. Midpoint Implementation

The final stage of the implementation is to count the number of vehicles crossing the predetermined line. If the predicted bounding box, which has been processed by the Midpoint Algorithm, touches the line, the count is automatically incremented. Figure 13 shows the result of the intersection between the Midpoint Algorithm and the predefined line.

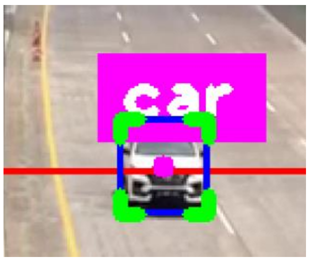

Figure 13. Result of Implementation

Figure 14 illustrates the results of the implementation applied to the application. The program also displays the number of four-wheeled or larger vehicles crossing the designated line, automatically counting the vehicles that pass through.

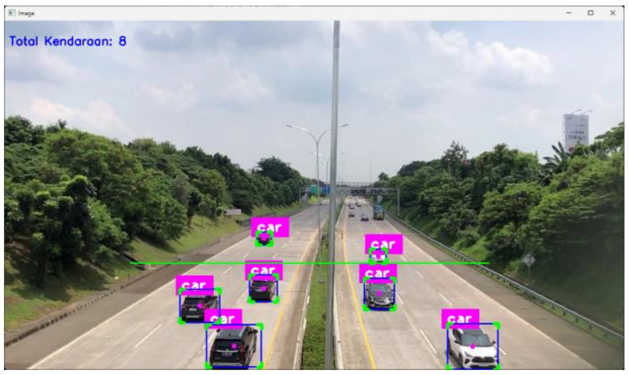

Figure 14. Result of Implementation

III. Results and Discussion

In this section, the researcher discusses the results of the vehicle count monitoring system implementation. During the testing process, it was conducted at different times of the day, including morning, afternoon, and late afternoon. Table 1 is the result of the implementation.

Table 1. Implementation

Time	Durations	Result		
		Manual	Program	Accuracy
Morning	185 s	147	156	94.23%
Afternoon	186 s	225	246	91.46%
Late Afternoon	173 s	351	377	93.10%
Night	187 s	487	526	92.58%
			Average	92.42%

Based on table 1, the research time was conducted 4 times, namely morning, afternoon, evening and night. From the results of the research that has been done when monitoring the number of vehicles in the morning using the program, 156 vehicles were obtained, when manually calculated, 147 vehicles were obtained and from the results of testing in the morning, an accuracy value of 94.23% was obtained. The results of the study during the day using the program obtained 246 vehicles, when manually calculated, 225 vehicles were obtained and from the results of testing during the day, an accuracy value of 91.46%. The results of the study in the afternoon using the program obtained 377 vehicles, when manually calculated, 351 vehicles were obtained and from the results of testing in the afternoon, an accuracy value of 93.10%. And the results of the study at night using the program obtained 526 vehicles, when manually calculated, 225 vehicles were obtained and from the results of testing at night, an accuracy value of 92.58%. The average accuracy value of the four times is 92.42%.

IV. Conclusion

This study developed a vehicle count monitoring system using deep learning algorithms, specifically the YOLOv8 method for object detection, the Hungarian algorithm for tracking, and the Midpoint algorithm for vehicle counting. The system was tested on traffic videos at different times (morning, afternoon, evening, and night) with an average accuracy of 92.42%. The results demonstrated reliable vehicle detection capabilities, with the highest accuracy of 94.23% observed in the morning. The system outperformed manual counting methods in accuracy and showed great potential for smart city applications in real-time traffic monitoring. The study also recommended further development to include vehicle type classification and traffic pattern analysis to enhance system efficiency.

Suggestions for further research include exploring the integration of YOLOv8 with real-time data streaming systems for continuous monitoring in various traffic environments. In addition, extending the system to classify vehicles by type and analyse traffic patterns over a long period of time will provide a more comprehensive understanding of traffic dynamics. This approach can further enhance the scalability and utility of vehicle counting systems in smart city applications.

References

Adianto, F. K. (2021). *Deteksi Kantuk Menggunakan Pendekatan Deep Learning Secara Real-time*. Institut Teknologi Sepuluh Nopember.

Arif, M. F., Nurkholis, A., Laia, S., & Rosyani, P. (2023). Deteksi Kendaraan Dengan Metode YOLO. *AI Dan SPK: Jurnal Artificial Intelligent Dan Sistem Penunjang Keputusan*, *1*(1), 20–27.

Aryanto, R., Rosid, M. A., & Busono, S. (2023). Penerapan Deep Learning untuk Pengenalan Tulisan Tangan Bahasa Aksara Lota Ende dengan Menggunakan Metode Convolutional Neural Networks. *Jurnal Informasi Dan Teknologi*, 258–264.

Barat, B. P. S. J. (2023). Jumlah Kendaraan Bermotor Menurut Kabupaten/Kota dan Jenis Kendaraan di Provinsi Jawa Barat (unit), 2023. In *https://jabar.bps.go.id/id/statistics-table/3/VjJ3NGRGa3dkRk5MTlU1bVNFOTVVbmQyVURSTVFUMDkjMw==/jumlah-kendaraan-bermotor-menurut-kabupaten-kota-dan-jenis-kendaraan-di-provinsi-jawa-barat–unit—2023.html?year=2023*.

Christian, J., & Al Idrus, S. I. (2023). Introduction to Citrus Fruit Ripens Using the Deep Learning Convolutional Neural Network (CNN) Learning Method. *Asian Journal of Applied Education (AJAE)*, *2*(3), 459–470.

Hegarini, E., Saifullah, A. A., & Wardijono, B. A. (2024). Sistem Penghitung Jumlah Kendaraan dan Pendeteksi Kecepatan pada Ruas Jalan Menggunakan Metode Haar Cascade Classifier. *Jurnal SIKOMTEK*, *14*(01), 95–100.

Kim, J.-H., Kim, N., & Won, C. S. (2023). High-speed drone detection based on yolo-v8. *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–2.

Niu, H., Liu, J., Yu, Z., Zheng, D., He, P., & Wang, F. (2022). Real-time object tracking system using PTZ camera. *2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, 471–478.

Nugroho, G. S., & Hazmin, G. (2022). Perbandingan Algoritma untuk Mereduksi Noise pada Citra Digital. *Journal of Information Technology Ampera*, *3*(2), 159–174.

Nugroho, L. A., Latifa, E. A., & Maulani, E. O. (2024). Dampak jumlah kendaraan besar terhadap kemacetan lalu lintas di jalan tol. *JURNAL TEKNIK SIPIL CENDEKIA (JTSC)*, *5*(2), 915–928.

Putri, V. H. (2023). *Detecting Incoming and Outgoing Passengers on Intelligent Car (iCar Its) Using Computer Vision*. Institut Teknologi Sepuluh Nopember.

Rachmawati, F., & Widhyaestoeti, D. (2020). Deteksi Jumlah Kendaraan di Jalur SSA Kota Bogor Menggunakan Algoritma Deep Learning YOLO. *Prosiding LPPM Uika Bogor*.

Ramadhani, F., Satria, A., & Dewi, S. (2024). Identifikasi Kendaraan Bermotor pada Dashcam Mobil

Menggunakan Algoritma YOLO. *Hello World Jurnal Ilmu Komputer*, *2*(4), 199–206.

Rasywir, E., Sinaga, R., & Pratama, Y. (2020). Evaluasi pembangunan sistem pakar penyakit tanaman sawit dengan metode deep neural network (DNN). *Jurnal Media Informatika Budidarma*, *4*(4), 1206–1215.

Sama, A. K., & Sharma, A. (2023). Simulated uav dataset for object detection. *ITM Web of Conferences*, *54*, 2006.

Syarif, H., & others. (2023). *Evaluasi Tingkat Visual Attention Mahasiswa pada Pembelajaran Online dengan Meeting Zoom Menggunakan Metode Histogram of Oriented Gradients (HOG)= Evaluation Of Students Visual Attention Levels In Online Learning With Meeting Zoom Using The Histogram Of Oriented Gradients (Hog) Method*. Universitas Hasanuddin.

Tamsir, N., Soetikno, Y. J. W., & others. (2021). Aplikasi Penjualan Baju Kaos Berbasis Web dan Android. *SISITI: Seminar Ilmiah Sistem Informasi Dan Teknologi Informasi*, *10*(1), 1–8.

Ünlü, Ü. C. (2021). *Improvement on motion-guided siamese object tracking networks using prioritized windows*. Izmir Institute of Technology (Turkey).

Wahyuni, S., & Sulaeman, M. (2022). Penerapan algoritma deep learning untuk sistem absensi kehadiran deteksi wajah di PT Karya Komponen Presisi. *Jurnal Informatika SIMANTIK*, *7*(1), 12–21.

Widyawan, S., & others. (2019). Analisis Kinerja Simpang Bersinyal Untuk Meningkatkan Keselamatan Pada Simpang Depok Kota Depok. *AIRMAN: Jurnal Teknik Dan Keselamatan Transportasi*, *2*(1), 30–38.

Yu, J., Xu, J., Chen, Y., Li, W., Wang, Q., Yoo, B., & Han, J.-J. (2021). Learning generalized intersection over union for dense pixelwise prediction. *International Conference on Machine Learning*, 12198–12207.