International journal of science, engineering, and information technology
Volume 08, Issue 02, July 2024
Journal homepage: https://journal.trunojoyo.ac.id/ijseit

Design of YOLOv5 Medium as Unmanned Rail Inspection in Braking Control System Based on Computer Vision

Adiratna Ciptaningrum[a], Andhika Putra Widyadharma[b], Imam Junaedi[c], Rahayu Mekar Bisono[d], R. Akbar Nur Apriyanto[e], R. Gaguk Pratama Yudha[f], Mohammad Erik Echsony[g], Larissa Kartika Putri[h], Edo Zulmi faikhsan[i]

[a,b,c,d,e,f,g,h,i]Department of Engineering, Politeknik Negeri Madiun, Madiun, Indonesia

A B S T R A C T

This research explores the application of computer vision using YOLOv5 Medium for automatic object detection in unmanned rail inspection systems. The proposed technique utilizes image processing to analysing pixel values and detect optical motion vehicles. This detection triggers control system responses, such as activating the inspection train's motor upon vehicle identification. The study demonstrates the effectiveness of YOLOv5 Medium in achieving high accuracy rates. Evaluations at various distances yielded promising results: 97.98% at 3 meters, 100% at 5 meters, 99.49% at 7 meters, and a perfect 100% at 9 meters. These findings suggest optimal system performance at a distance of 9 meters. Overall detection performance across all test distances remained consistently high, with sequential rates of 0.96, 0.97, 0.95, and 0.96. This research emphasizes the crucial role of several factors in maintaining system accuracy and performance. These include the efficacy of the colour segmentation algorithm, ambient lighting conditions, and camera resolution. Furthermore, the importance of extensive testing with a diverse dataset is highlighted to ensure the system's robustness and adaptability to various real-world scenarios.

Keywords: unmanned rail inspection, object detection, YOLOv5 Medium

Article History
Received 00 December 00 Received in revised form 00 January 00 Accepted 00 February 00

1. Introduction

Railway trains and tracks are complex assets that require constant inspection, management and maintenance. Computer vision can be used to control a braking system, detecting and monitoring various aspects of railway infrastructure, and it holds significant relevance for rail inspection technology. Computer vision can be used to detect obstacles on the railway tracks in real-time, allowing for faster and more precise braking compared to human reaction times.

As we know, conventional object detection methods often struggle with specific scenarios, particularly dealing with humans. This conventional method, often involve handcrafted feature extraction, such Viola-Jones Detector, Histogram of Oriented Gradients (HOG) Detector, and Deformable Part-based Model (DPM) [1].

Viola-Jones Detector in Haar-like features, which are simple rectangular features capturing intensity differences, are manual defined. These features are selected based on their ability to differentiate between object and non-object regions [2].

On Histograms of oriented gradients are computed within local cells, capturing edge orientations in the image. These histograms are manually designed to capture the appearance of edges and contours in different orientations [3]. While DPM models consider hierarchical part-based structures, the appearance and geometric models for each part are manually defined. The model's ability to account for deformations and spatial relationships is also designed based on prior knowledge [4].

These traditional approaches have contributed significantly to the development of object detection methods, offering insights into handling different object characteristics and challenges.

Deep learning-based approaches, on the other hand, have gained significant attention and success in recent years. Convolutional Neural Networks (CNNs) are a key technology in deep learning for object detection. They can learn to automatically extract relevant features from images and learn complex patterns that are representative of different object categories. There are two types of object detection architectures based on CNN, one-stage and two-stage detectors [5].

One-stage detectors are designed to directly predict bounding box coordinates and class probabilities for multiple objects in a single pass through the network. These detectors are known for their simplicity and

* *Corresponding author.*
E-mail address: adiratna@pnm.ac.id

E-ISSN 2548-4214

efficiency, as they eliminate the need for a separate proposal generation step. The key idea is to densely sample potential object locations and then predict the presence of an object and its associated bounding box in a single shot. Most relevance one-stage detectors is YOLOv5 [5].

One-stage detectors are known for their speed and real-time capabilities. They perform detection in a single pass, which makes them faster for real-time applications while they ended to have slightly lower accuracy compared to two-stage detectors. They may struggle with detecting small objects and handling object instances with significant size variations [5].

In this method, YOLOv5 Medium a fast and accurate object detection model, has potential applications in a railway inspection system that utilizes a control braking mechanism. Besides that, this approach can distinguish between actual obstacles and irrelevant objects, leading to fewer unnecessary braking events. By automating obstacle detection and braking response, YOLOv5 Medium can potentially improve overall efficiency and reduce operational costs.

2. Methods

2.1. Related Works

A research by Deren Zhu [6] focuses on improving the You Only Look Once version 5 Small model (YOLOv5s) for the specific task of detecting small targets in Unmanned Aerial Vehicle (UAV) imagery. The researchers introduce a model called CCE-YOLOv5s. This is an improved version of YOLOv5s specifically designed for small target detection in UAV imagery. The main improvement involves adding a new layer to the YOLOv5s architecture. This layer, called the small target detection layer, utilizes a 160x160 scale feature map. This allows the model to focus on smaller features in the image, enhancing its ability to detect small objects. This model is significantly better than the previous one. It achieves a mean Average Precision (mAP) of 80%. Additionally, it only takes 0.66 hours to train, which is 43% faster than the original model. In short, this improved model is both more accurate and faster to train, making it ideal for real-time application.

Study by Sougatamoy Biswas [7] focuses on developing a system for recognizing gestures in real-time using an improved version of YOLOv5 model. This study refers to a system that can identify hand gestures from a video stream or camera feed with minimal delay. This is crucial for applications where immediate response to gestures is needed. This model specifically for gesture recognition. These improvements might target accuracy, speed and robustness. In accuracy, enhance the model's ability to correctly identify different gestures, especially subtle variations. In speed, further optimize the model for faster processing to maintain real-time performance. In robustness, improve the model's ability to handle variations in lighting, background clutter, and hand position. Overall, their research demonstrates that the YOLOv5 model achieves mean Average Precision (mAP) of 96.8% indicating excellent real-time performance.

Bowen Zheng [8] conducted a study that refers to identifying and locating objects that are in motion within an image or video sequence. This research, investigates how to leverage YOLOv5's strengths for moving target detection while potentially addressing its limitations. Some areas of improvement might include accuracy, background handling, and real-time performance. Their research enhanced YOLOv5 model achieved a promising average detection accuracy of 82.3%.

2.2. Research Methodology

2.2.1 Design of Unmanned Rail Inspection

Unmanned rail inspection allows automation of tasks related to monitoring of large infrastructure facilities. While unmanned rail inspection offers the benefit of automating tasks for monitoring large infrastructure facilities, the design and material selection for such systems are crucial for ensuring their durability and effectiveness in often harsh environments. Galvalum, a steel sheet coated with a layer of zinc and aluminum, emerges as a compelling choice for several reasons.

Galvalum offers superior corrosion resistance compared to regular steel. The zinc layer acts as a sacrificial anode, protecting the underlying steel from rust and degradation. This is particularly important for unmanned systems that may operate exposed to elements like rain, snow, or coastal environments.

The aluminum layer in galvalum reflects heat, keeping the internal components of the inspection system cooler. This is essential for ensuring proper functioning of electronics and sensors, particularly in hot climates, and hot railway tracks. Incorporating galvalum into the design of unmanned rail inspection systems can contribute to their robustness, longevity,and overall effectiveness in monitoring critical infrastructure assets.

Figure 1. Design of unmanned rail inspection.

2.2.2 You Only Look Once version5 Medium (YOLOv5 Medium)

The You Only Look Once v5 (YOLOv5) model belongs to a family of computer vision models known for their single-stage object detection capabilities. YOLOv5 offers a range of pre-trained variants, including small (s), medium (m), large (l), and extra-large (x) versions. These variants exhibit a progressive increase in accuracy but require correspondingly longer training times. YOLOv5 is designed by a use case of object detection, involves creating features from input image. The architecture of general YOLOv5 can be seen in Figure 2.

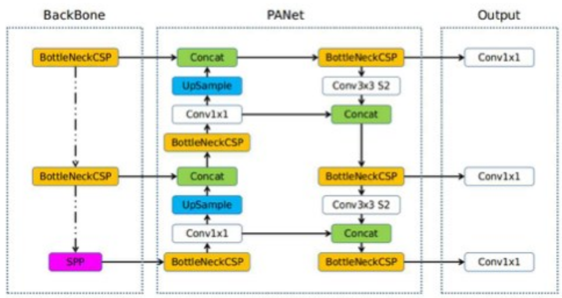

Figure 2. The architecture of YOLOv5.

In Figure 2 can be seen that the architecture contains Backbone (CSPDarknet), Neck (Path Aggregation Network – PAnet), and Head (YOLOv5 layer). In Backbone (CSPDarknet), researcher breaking down into convolution, batch normalization, pointwise convolution (1×1 convolution), and residual connections. In Neck (Path Aggregation Network – PANet), researcher break down into upsampling, convolution, and element-wise addition. In Head (YOLOv5 layer), researcher break down into prediction from feature maps, bounding box prediction, class probability prediction, and loss function .

2.2.2.1 Backbone (CSP Darknet)

- ***Convolution***

Convolution is the core operation for extracting features. The backbone uses multiple convolutional layers, each performing the element-wise multiplication and summation between the filter and the input data (previous layer's output) [9].

$$Output[i,j] = \sum(Filter[m,n] \times Input[i+m,j+n] \qquad (1)$$

It explains that a filter (kernel) $Filter[m,n]$ with learnable weights across the image and performing element-wise multiplication with the underlying image data. The sum ($\sum$) of these products for each location in the filter becomes the corresponding value in the output $Output[i,j]$ feature map.

- ***Batch Normalization***

In Batch Normalization, researcher do normalization and scaling-shifting (learned parameters). Those technique will normalize (X_{hat}) the activation (outputs) of convolutional layers across different mini-batches during training. It helps the network converge faster and improves stability.

$$X_{hat} = \frac{X - \mu_B}{\sigma_B} \qquad (2)$$

$$Y = \gamma \times X_{hat} + \beta \qquad (3)$$

- ***Pointwise Convolution (1×1 convolution)***

This uses a 1×1 filter to perform a linear transformation on the channels of the feature map. It can be used for dimensionality reduction or introducing non-linearity. Similar to regular convolution, but the filter size is 1×1. So, the output at each position depends only on the weighted sum of the channels at that position in the input feature map.

- ***Residual Connection***

The output (y) of a residual block with a shortcut connection can be expressed as:

$$y = F(x) + x \qquad (4)$$

Where (x) represents the input to the residual block, and $F(x)$ represents the transformation learned by the stacked convolutional layers within the block. The input (x) which is passed through a series of convolutional layers $(F(x))$ within the residual block. These layers perform feature extraction and potentially introduce non-linearity through activation functions. The output of these convolutional layers $(F(x))$ is then added element-wise to the original input (x) . This effectively creates a path for the information to flow directly through the block without being altered by the complex transformations. This combination $(F(x) + x)$ becomes the final output (y) of the residual block.

2.2.2.2 Neck (Path Aggregation Network – PANet)

- ***Upsampling (Bilinear Interpolation)***

This method considers the values of four neighboring pixels in the lower-resolution map and calculates a weighted average to create a new pixel value in the upsampled map. The weights are based on the distance of each neighboring pixel to the target location in the upsampled map [10]. It can be defined as:

$$New_{value}(i,j) = \sum(Weight(m,n) \times Old_{value}(i+m,j+n)) \qquad (5)$$

- ***Convolution***

After upsampling, PANet applies convolutions to the feature maps from different stages. These convolutions serve two purposes. First, to Reduce Channels. The number of channels in the feature maps might be high. Convolution with a 1×1 filter can be used to reduce the dimensionality and improve computational efficiency. Mathematically, this involves element-wise multiplication and summation with a learnable filter of size 1×1.

The second one is Feature Refinement. Convolution can also be used to refine the features extracted at each stage. This might involve applying filters of larger sizes (e.g., 3×3) to extract additional information or introduce non-linearity with activation functions like LeakyReLU.

- ***Element-wise Addition***

The core concept of PANet lies in combining information from various stages of the backbone. Once the upsampled and potentially refined feature maps are obtained, PANet performs element-wise addition. This involves adding the corresponding elements from each feature map at the same spatial location. Mathematically:

$$Combined_{feature(i,j)} = Feature_{map}(i,j) + Feature_{map2}(i,j) + \cdots + Feature_{mapN}(i,j) \quad (6)$$

This element-wise addition allows PANet to create a richer feature representation that combines low-level details (from high-resolution, shallow layers) with high-level semantic information (from low-resolution, deep layers) for improved object detection.

2.2.2.3 *Head (YOLOv5 Layer)*

- ***Anchor Boxes and Prior***
  YOLOv5 predefines a set of anchor boxes with various sizes and aspect ratios at each location in the feature map from the neck. These act as initial predictions, and the model refines them to fit the actual objects. Mathematically, each anchor box can be represented as a tuple (cx, cy, w, h), where (cx, cy) represent the center coordinates of the anchor box, and (w, h) represent the width and height of the anchor box.

- ***Bounding Box Prediction with Offset***
  The model predicts offsets for each anchor box to adjust its position and size to match the actual object. These offsets are relative to the predefined anchor box dimensions. Mathematically, the predicted bounding box coordinates (tx, ty, tw, th) are calculated as:

$$Predicted_{box} = (cx, cy, w, h) + (tx, ty, tw, th) \quad (7)$$

Where tx, ty represent the offset adjustments to the anchor box center coordinates, and tw, th represent the offset adjustments (ratios) to the anchor box width and height (taking the logarithm to ensure scale invariance).

- ***Non-Max Suppression***
  After the model predicts bounding boxes for each location, NMS is applied to remove redundant or overlapping boxes. NMS considers the confidence scores of each box and selects the one with the highest score. It then suppresses boxes with significant overlap with the selected box. The math behind NMS involves calculating the Intersection over Union (IoU) between bounding boxes and using thresholds to determine suppression. It's calculated as the area of intersection between the two boxes divided by the area of their union. Each bounding box can be represented by its minimum and maximum coordinates for both width and height:

$$(x_{min}, y_{min}, x_{max}, y_{max}) \quad (8)$$

Then calculate the overlapping width and height of the intersection region:

$$area(box\ 1) = \left(box\ 1_{x_{max}} - box\ 1_{x_{min}}\right) \times \left(box\ 1_{y_{max}} - box\ 1_{y_{min}}\right) \quad (9)$$

$$area(box\ 2) = \left(box\ 2_{x_{max}} - box\ 2_{x_{min}}\right) \times \left(box\ 2_{y_{max}} - box\ 2_{y_{min}}\right) \quad (10)$$

So, the total union area considers both boxes and subtracts the overlapping area to avoid double counting:

$$Union_{area} = area(box1) + area(box2) - Intersection_{area} \quad (11)$$

Simply plug them into main equation that can be seen below:

$$IoU(box\ 1, box\ 2) = \frac{area(intersection(box\ 1, box\ 2))}{area(union(box\ 1, box\ 2))} \quad (12)$$

- ***Class Probability Prediction***
  The model predicts the probability of each object belonging to a specific class within the predefined set of classes (e.g., person, car, truck). A convolutional layer with filters corresponding to each class is used. The output goes through a softmax function to convert it into a probability distribution. Mathematically, the softmax function takes a vector of logits (output from the convolutional layer) and transforms them into probabilities between 0 and 1, ensuring they sum to 1. The formula for softmax is:

$$Softmax(x) = \frac{\exp(x)}{(\sum \exp(x))} \quad (13)$$

- ***Loss Function***
  In this part, the loss function plays a crucial role in training the model to improve its object detection accuracy. It measures the difference between the model's predictions (bounding boxes and class probabilities) and the ground truth labels for each object in the training data. The YOLOv5 typically utilizes a combination of loss functions to address different aspects of the prediction, so researcher using combine losses between bounding box loss and classification loss.
  In bounding box loss focuses on penalizing errors in the predicted bounding boxes. A common choice is the Generalized Intersection over Union (GIOU) loss.

$$GIOU\ Loss = 1 - IoU + \frac{(D^2 - d^2)}{b^2} \quad (14)$$

As we can see in Equation 14, the GIOU loss incorporates not only the overlap (IoU) but also penalizes for excessive areas or poor overlap between the boxes (through the distance terms).

In classification loss, measures the difference between the predicted class probabilities and the ground truth labels. A common choice is the cross-entropy loss:

$$Cross - entropy\ Loss = -\sum\left(y_{true} \times \log\left(y_{pred}\right)\right) \quad (15)$$

The cross-entropy loss penalizes the model for incorrectly predicting class probabilities. YOLOv5 combines these loss functions with a predefined weighting factor (λ) to create the overall loss function:

$$Overall\ Loss = \lambda \times Bounding\ Box\ Loss + Class\ Loss \quad (16)$$

The weighting factor controls the relative importance of each loss term during training. By minimizing the overall loss function through backpropagation during training, the model learns to adjust its predictions to better match the ground truth labels. This continuous optimization process refines the model's ability to accurately detect objects and their corresponding classes in unseen images.

3. Result and Discussion

From this research, the results of YOLOv5 Medium are obtained. However, the results of this research will be divided into two categories. The first category is the result of YOLOv5 Medium by validating the performance and distance in any projection value in rail inspection obtained. And the second, the scope of integration of the YOLOv5 Medium system for rail inspection.

After the design system has been matched with the projection, the YOLOv5 Medium is then integrated with the rail inspection system through the motor, which further regulates the range distance, so that the system will get information on the optimum joint distance. The system will manage the rail inspection in which direction to move through the vehicle object movement on the rail track. If a good result of the YOLOv5 Medium is detected, then the motor condition is on, and vice versa.

In this research, researcher take samples in distance 3, 5, 7, and 9 meters for the accuracy and performance of YOLOv5 Medium in ideal condition. And the class in distance start at a distance of 1-15 meters.

Table 1. Accuracy and performance in distance 3 meter.

	Accuracy (%)	Performance	Distance System to Object in Any Projection
1	90,90909	0,899326865	1,691546
2	100	0,949890467	1,691546
3	100	0,972448902	1,736701
4	100	0,977042456	2,323711
5	100	0,972686289	2,323711
6	100	0,971956094	2,233402
7	100	0,978331889	2,639794
8	100	0,970708237	3,362268
9	100	0,973965923	4,129897
10	100	0,968603098	4,310515
11	100	0,973835746	4,310515
12	100	0,971922181	4,310515
13	100	0,975894094	3,588041
15	100	0,973803011	3,317113
16	100	0,977188696	3,317113
17	100	0,98070703	2,368866
18	100	0,972864359	1,420619
19	100	0,975279707	1,375464

Figure 3. Vehicle detection in distance 3 meters.

Table 3. Accuracy and performance in distance 5 meter.

Frame	Accuracy (%)	Performance	Distance System to Object in Any Projection
1	84,84848	0,850559	6,29732
2	84,84848	0,844887	6,29732
3	96,9697	0,907605	6,161856
4	100	0,930195	6,026392
5	100	0,915528	6,026392
6	96,9697	0,922423	6,161856
7	100	0,93516	6,252165
8	100	0,921102	6,026392
9	100	0,91513	6,026392
10	96,9697	0,906237	5,890928
11	90,90909	0,906352	5,890928
12	96,9697	0,909491	5,890928
13	100	0,920081	5,936082
15	100	0,92177	5,936082
16	100	0,916532	6,116701
17	100	0,924791	6,161856
18	100	0,908984	6,116701
19	96,9697	0,912072	6,116701

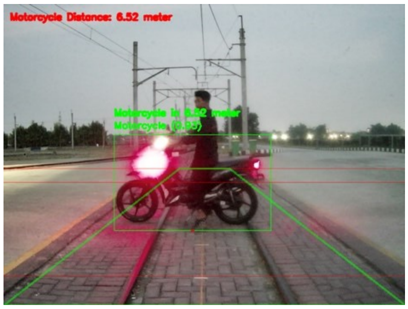

Figure 4. Vehicle detection in distance 5 meters.

Table 4. Accuracy and performance in distance 7 meter.

Frame	Accuracy (%)	Performance	Distance System to Object in Any Projection
1	84,84848	0,850559	6,29732
2	84,84848	0,844887	6,29732
3	96,9697	0,907605	6,161856
4	100	0,930195	6,026392
5	100	0,915528	6,026392
6	96,9697	0,922423	6,161856
7	100	0,93516	6,252165
8	100	0,921102	6,026392
9	100	0,91513	6,026392
10	96,9697	0,906237	5,890928
11	90,90909	0,906352	5,890928
12	96,9697	0,909491	5,890928
13	100	0,920081	5,936082
15	100	0,92177	5,936082
16	100	0,916532	6,116701
17	100	0,924791	6,161856
18	100	0,908984	6,116701
19	96,9697	0,912072	6,116701

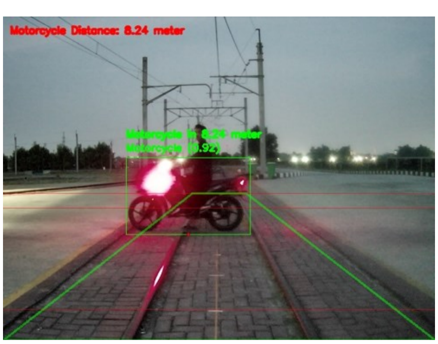

Figure 5. Vehicle detection in distance 7 meters.

Table 5. Accuracy and performance in distance 9 meter.

Frame	Accuracy (%)	Performance	Distance System to Object in Any Projection
3	69,69697	0,713307	6,658557
4	100	0,869228	10
5	100	0,949387	9,445161
6	100	0,9688	9,445161
7	100	0,960908	9,445161
8	100	0,957992	9,445161
9	100	0,96142	9,445161
10	100	0,961074	9,630323
11	100	0,963739	9,445161
12	100	0,96218	9,630323
13	100	0,966155	9,630323
14	100	0,958816	10,00065
15	100	0,96469	9,815484
16	100	0,974385	9,630323
17	100	0,965898	9,815484
18	100	0,95539	10,18581
19	100	0,960274	10,00065

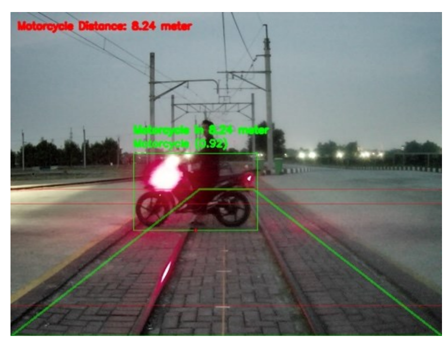

Figure 6. Vehicle detection in distance 9 meters.

After obtaining ideal conditions for object detection using YOLOv5 Medium, the next stage is to set the detection distance to the automatic braking system via computer vision. So, for the braking process the results obtained can be seen in Table 6 below.

Table 6. Optimal distance of motor status activation condition.

Distance (m)	Accuracy (%)	Performance	Motor Status
3	97.98	0.96	ON
5	100	0.97	ON
7	99.49	0.95	ON
9	100	0.96	ON

In Table 6, it can be seen and analyzed that the ideal distance for the system to detect and state the motor ON is at a distance of 3, 5, 7, and 9 metres. It indicates that the graph with the fourth cluster of distances condition shows a good initialization between frame count and accuracy compared to the other distances. However, in the initialization condition or at a distance of 1 meter and 2 meters the motor status is OFF. It happens that the detection of human pose estimation is too close to the distance of the camera, and the range of the camera still cannot see the position of the rail.

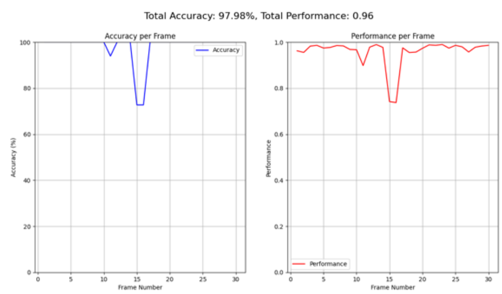

Figure 7. Total accuracy per frame and performance per frame in distance 3 meters.

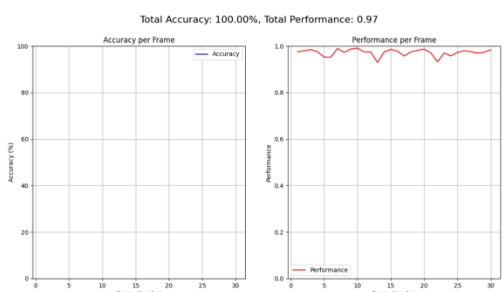

Figure 8. Total accuracy per frame and performance per frame in distance 5 meters.

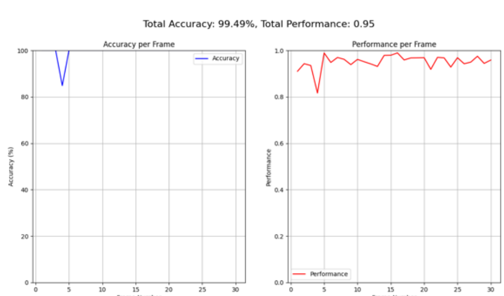

Figure 9. Total accuracy per frame and performance per frame in distance 7 meters.

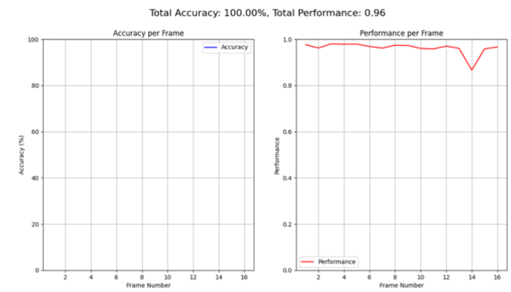

Figure 10. Total accuracy per frame and performance per frame in distance 9 meters.

4. Conclusion

The technique of using computer vision as an automatic control system of unmanned rail inspection is able to generate numerical values of the pixel values of the object detection, the accuracy level, and also the performance. The use of YOLOv5 Medium is used to be able to detect optical motion vehicle so that decisions can be obtained when vehicles are crossing which will be recognized through validation of the detector values obtained which will then be integrated with the control system of the inspection train is identified as crossing then the system will turn on the motor so that the motor is ON, and vice versa. The accuracy rate 97.98% in distance 3 meters, 100 meters in 5 meters, 99.49% in distance 7 meters, and 100% in distance 9 meters. This indicates that at a distance of 9 meters, the system will work ideally. From all the test carried out so that the results obtained the total detection performance rate sequentially 0.96, 0.97, 0.95, and 0.96.The system's accuracy and performance hinge on several factors, the colour segmentation algorithm's effectiveness, ambient lighting conditions, and camera resolution. Additionally, extensive testing with a diverse set of samples is crucial to ensure the system's robustness across various scenarios.

REFERENCES

[1] S. Sumit, D. Rambli, and A. Mirjalili, “Vision-Based Human Detection Techniques: A Descriptive Review,” IEEE Access, vol. 9, January 2021.

[2] Viola, P., & Jones, M. J. (2001). Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition (CVPR) (Vol. I, pp. I-511-I-518). IEEE.

[3] Dalal, N., & Triggs, B. (2005). Histogram of oriented gradients for human detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (Vol. 1, pp. 886-893).

[4] Felzenszwalb, P. F., Girshick, R. B., David, D., & Malik, J. (2010). Object detection with deformable part models. IEEE transactions on pattern analysis and machine intelligence, 32(8), 1407-1417.

[5] Zhao, Q., Pang, Y., Fu, H., Li, Z., & Xu, D. (2023). Recent Advances in Deep Learning-Based Object Detection. IEEE Access, 11, 28238-28264.

[6] D. Zhu, L. Dai and P. Du, "CCE-YOLOv5s: An Improved YOLOv5 Model for UAV Small Target Detection," 2023 IEEE 5th

International Conference on Civil Aviation Safety and Information Technology (ICCASIT), Dali, China, 2023, pp. 824-829

[7] S. Biswas, A. Nandy, A. K. Naskar and R. Saw, "Real time Gesture Recognition using Improved YOLOv5 Model," 2024 11th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2024, pp. 328-333

[8] B. Zheng, P. Li and H. Lu, "Research on Moving Target Detection Technology Based on Yolov5 Model," 2023 2nd International Conference on Artificial Intelligence and Intelligent Information Processing (AIIIP), Hangzhou, China, 2023, pp. 305-309

[9] P. Carrinho and G. Falcao, "Highly accurate and fast YOLOv4-based polyp detection," Expert Syst Appl, vol. 232, p. 120834, 2023, doi: https://doi.org/10.1016/j.eswa.2023.120834.

[10] W. Zhou, W. Bai, J. Ji, Y. Yi, N. Zhang, and W. Cui, "Dual-path multi-scale context dense aggregation network for retinal vessel segmentation," Comput Biol Med, vol. 164, p. 107269, 2023, doi: https://doi.org/10.1016/j.compbiomed.2023.107269.