International Journal of Artificial Intelligence & Robotics (IJAIR)
Vol.7, No.1, 2025, pp.1-9
E - ISSN : 2686-6269

Bolt Detection on Railway Rails Using ResNet-50 V1 with SSD Framework

Helfy Susilawati[1], Ginaldi Ari[2], Firman[3]
[1,2,3]Electrical Engineering Department, Universitas Garut, Indonesia
[1]helfy.susilawati@uniga.ac.id*; [2]ginaldi.ari@uniga.ac.id; [3]firmanfardiansyah2592@gmail.com
*corresponding author

ABSTRACT

One of the parts of a railroad track is a bolt. The role of bolts is significant in railway tracks, namely as a fastener between rails. Considering the importance of bolts on railway tracks, every morning, an officer would be assigned to go to the railway tracks to check the bolts. This inspection is done manually on foot or by driving along the railway tracks. Inspection performed manually has the possibility of errors in recognizing the condition of the bolt. In addition, if performed manually, there will be no record of the condition of the bolt. This data will be used to consider whether the condition of the bolt is still suitable for use or needs to be replaced. The formulation of the problem of the research conducted is how to detect bolts on railroad tracks using deep learning, with the purpose of this study to use a model to recognize and be able to detect bolts on railroad tracks. This study uses deep learning with the deep learning method used SSD Resnet 50 V1. The first step that must be taken in the study is to identify the object of the bolt located on the railway tracks. Further research can be carried out. This research has successfully detected bolts on railway tracks. This study used a dataset of 200 datasets in the first experiment and 300 datasets in the second experiment. The model used in the study is the Resnet 50 V1 SSD model, where using 2,000 steps, the precision value is 92.64%, and the Recall value is 64.87%.

Keywords: Bolt Detection; Railway Rails; Railroad Track; ResNet-50 V1; SSD Framework.

Article History

Received : Dec, 09th 2024	Accepted : May, 27th 2025	Published : May, 31st 2025

I. INTRODUCTION

Trains are one of the modes of land public transport that offer several advantages not found in other forms of land transportation. Among these advantages are the clear schedules for departure and arrival, the capacity to accommodate many passengers, and the ability to avoid traffic congestion [1]. Seeing the advantages of this train, it is not surprising that the government began to revitalize several railway lines in Indonesia, reaching 1,731 km/sp in 55 locations throughout Indonesia, with the aim of many people starting to use public transportation modes as an effort to overcome congestion [2]. The main components of the railway include railway gates, railway signs, and railway tracks. The railway has several components, including railway wires, bearings, and bolts.

Widely used connecting elements in wind turbine towers, building structures, and railways are bolts [3]. Bolts are very important components. The condition of bolts must be monitored, such as checking the safety of bolts and cables on shallow roads [4]. Bolts are an important component of the railway because they provide stability and support for the rail line, where the task of the bolt is to fasten the rail to the rail pad and keep it in place [5]. Given the importance of bolt components on the railway, checks are carried out every day to maintain the condition of railway bolts in good condition. Officers will be assigned to walk along the railway tracks in the morning. The process of walking along the railway tracks is done by walking along the railway tracks or by using a vehicle that has been modified to be able to walk on the railway tracks. The purpose of this railway walk is to check the condition of the railway track, one of which is to check the condition of the railway bolts.

Checking the condition of railway bolts that still use manual methods is less effective due to differences in perception when defining bolt conditions. In addition, records related to the condition of bolts on railway tracks have not been made. Therefore, research is needed to minimize this. There has been a lot of research on railways, including research on a train maintenance prediction system for the metro railway system, which uses a clustering algorithm to identify the probability of intervention levels by categorizing maintenance interventions into three probability levels, namely low, medium, and high [6]. There is also research to improve safe traffic control technology for digital railway crossings, which in this research presents ideas about train control technology that is connected to the environment and components around the train, such as door bars, lights, and others [7]. In addition to research on trains, there is also research on operator fatigue detection on railway tracks [8].

DOI: http://dx.doi.org/10.25139/ijair.v7i1.9351

Related to research on railway tracks, researchers detect defects in railway tracks using CNN [9], and some use the Edge Detection method [10]. Rail cracks are also researched using Deep Learning with the Gabor transformation method [11]. Previous studies show that a lot of research has been done on trains or railways that aim to improve safety on trains or railways. The difference between the research and previous research is that the research focused on bolt detection research on railway tracks using the SSD Resnet V1 method. Some research on bolt detection includes detecting loose bolts using optical sensors [12] and detecting missing bolts located at the bottom of fast trains using YOLO V5 [13]. Other research on bolt detection is the detection of missing nuts and bolts on the rail plate [14]. This research uses SSD Resnet V1 because, in previous research, a classification was carried out regarding tomato leaf diseases based on the severity of the disease in tomato leaves, which resulted in the Average Precision (mAP) value reaching more than 50%, namely at a value of 61.58% [15].

Since bolts on railway tracks are important in railway parts, this research aims to detect railway bolts using Deep Learning, where the method used is SSD Resnet 50 V1. This research is one of the bases for developing railway bolt conditions, whether there is looseness of bolts or not, and whether it is still feasible.

II. METHOD

The research conducted is qualitative research, where the output of this research is to determine the precision value of using the SSD Resnet 50 V1 method to recognize bolts on railroad tracks. The research section can be seen in the block diagram in Fig.1.

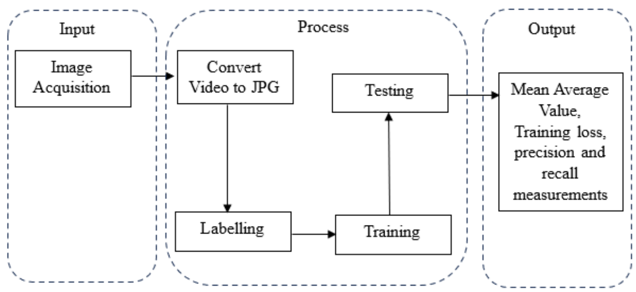


Fig.1. Research Block Diagram

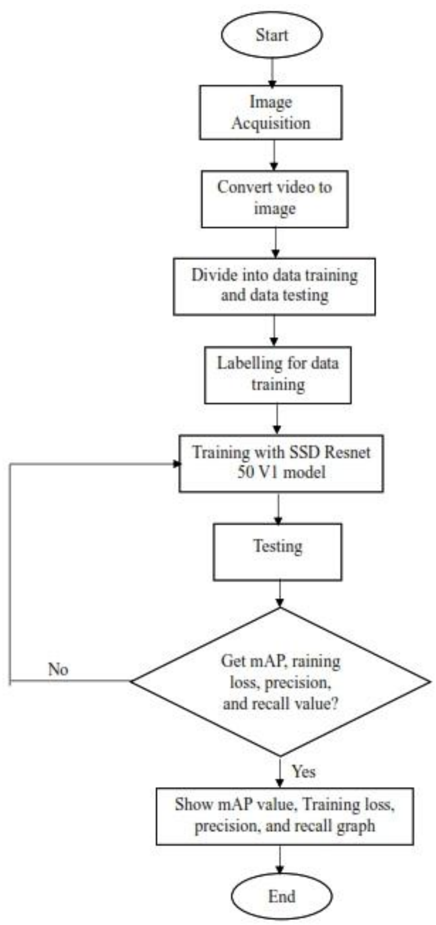


Fig.2. Research Flowchart

DOI: http://dx.doi.org/10.25139/ijair.v7i1.9351

Fig. 1 is an image of a research block diagram consisting of input, process, and output. The input part is image acquisition or image capture at the railway location. Image capture is done using a mobile phone camera in the form of video. In the process part, the first thing to do is to convert from video into jpg images. In the first experiment, using 200 datasets, approximately 90% of the dataset will be used as training data, totalling 182 frames, and approximately 10% of the data will be used for testing data, which is approximately 18 frames. The proportion of training data usage greater than testing data is expected to increase the accuracy of bolt object recognition on railroad tracks. The second experiment used 300 datasets divided into 282 training and 18 testing datasets. The next step was labelling for 183 images in the first experiment and 283 in the second. The training dataset was then trained using the Resnet 50 V1 SSD model to recognize bolt objects. The training process is an inseparable part of the Deep Learning process.

This research uses Deep Learning because deep learning is effective for systems that continuously learn. Some research on deep learning includes the classification of grape leaves [16], unexpected accident detection for poor CCTV conditions [17], and predicting student performance in programming courses [18]. After the training process is complete, the testing process will be conducted using 18 previously prepared testing datasets. The Resnet 50 V1 SSD model is often used in other image fields such as [19]. The Resnet 50 V1 SSD can perform classification tasks, such as research that uses the Resnet 50 V1 SSD model for memory efficiency classification. [20]. The output of this research is the (mean average) mAP value using the Resnet 50 V1 model from 200 datasets and 300 previously prepared datasets. The flowchart of the research conducted can be seen in Fig.2.

III. RESULT AND DISCUSSION

A. *Training using 182 datasets*

The training process utilized 182 datasets, with the classification loss performance depicted in Fig.3. Classification loss is the objective function for guiding the model in accurately identifying and categorizing target objects. The vertical axis represents the classification loss values, while the horizontal axis denotes the number of training steps. The black line illustrates the recorded loss values throughout the training process, whereas the grey line represents a smooth trend of these values. The classification loss steadily decreases and converges to approximately 0.2 after 2,000 training steps, indicating an improved over-time model performance.

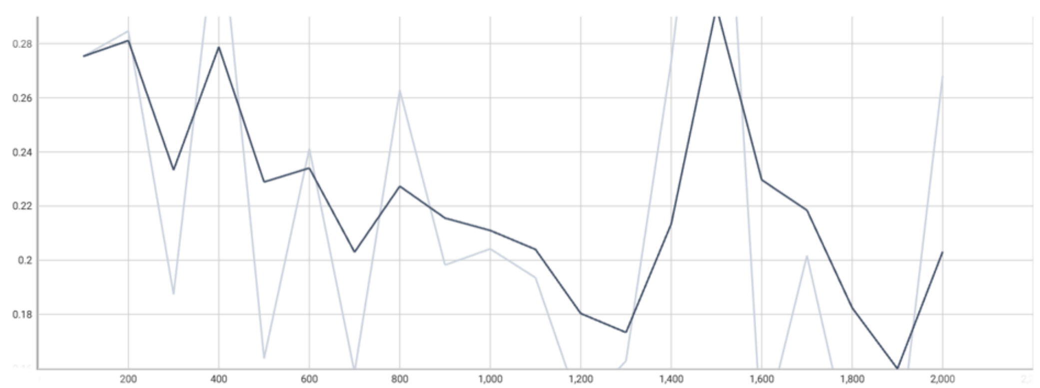


Fig.3. Classification Loss Training

Fig.4 illustrates the localization loss observed during the training process using 182 datasets. Localization loss is a function that optimizes the alignment between the predicted bounding boxes and the corresponding ground truth values. The vertical axis represents the localization loss values, while the horizontal axis indicates the number of training steps. The black line depicts the recorded loss values, whereas the grey line represents the smoothed trend. The localization loss progressively decreases and reaches approximately 0.07 after 2,000 training steps, demonstrating improved bounding box prediction accuracy throughout the training.

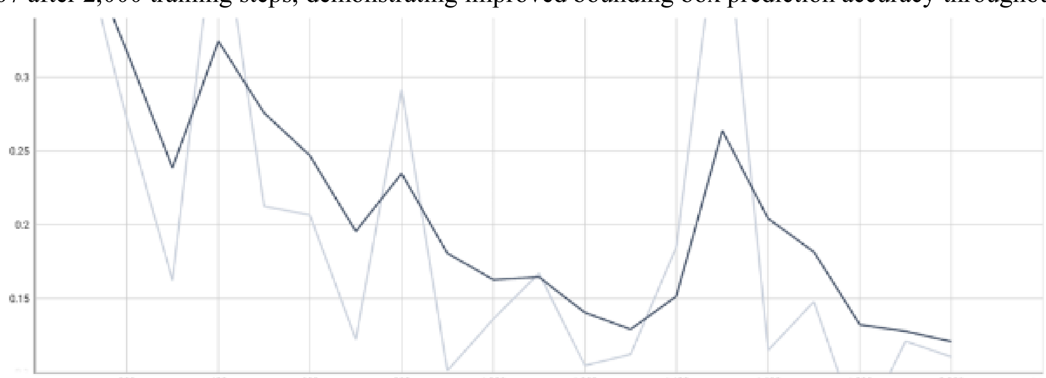


Fig. 4. Localization Loss

DOI: http://dx.doi.org/10.25139/ijair.v7i1.9351

E - ISSN : 2686-6269

Fig.5 illustrates the regularization loss observed during the training process using 182 datasets. Regularization loss is a function used to prevent overfitting and enhance the generalization capability of neural networks. The vertical axis represents the regularization loss values, while the horizontal axis indicates the number of training steps. The grey line shows the smoothed trend of the regularization loss, whereas the black line displays the actual recorded values during Training. The regularization loss continues to decrease steadily as it approaches 2,000 training steps. This decreasing trend is consistent with the behaviors observed in the validation loss, which also shows a similar downward pattern, indicating improved model generalization over time.

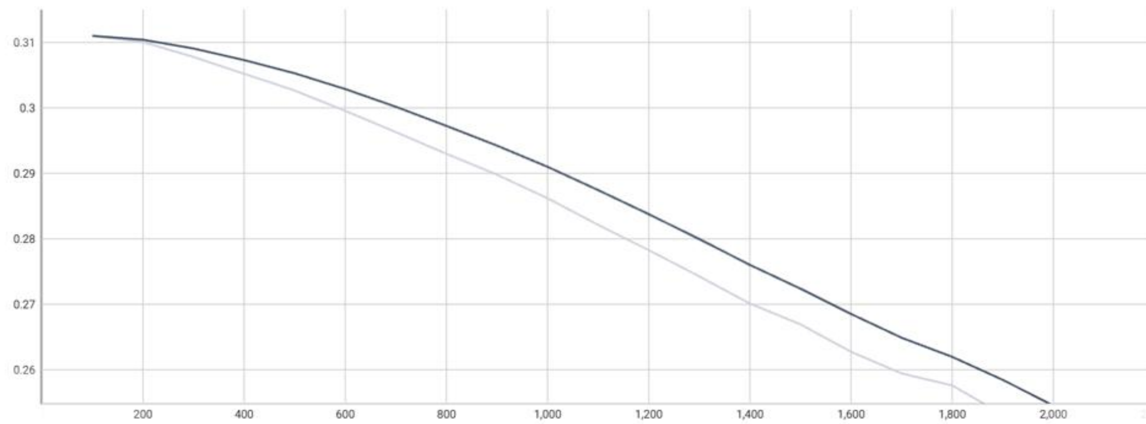


Fig.5. Regularization Loss

Fig.6 presents the total loss observed during the training process. The total loss is calculated as the average of three components: classification loss, localization loss, and regularization loss. The vertical axis represents the total loss values, while the horizontal axis indicates the number of training steps. The grey line illustrates the smoothed trend of the total loss, whereas the black line shows the actual recorded values during Training using 182 training datasets. The model reaches a total loss of approximately 0.4 after 2,000 training steps, indicating progressive improvement throughout the training process.

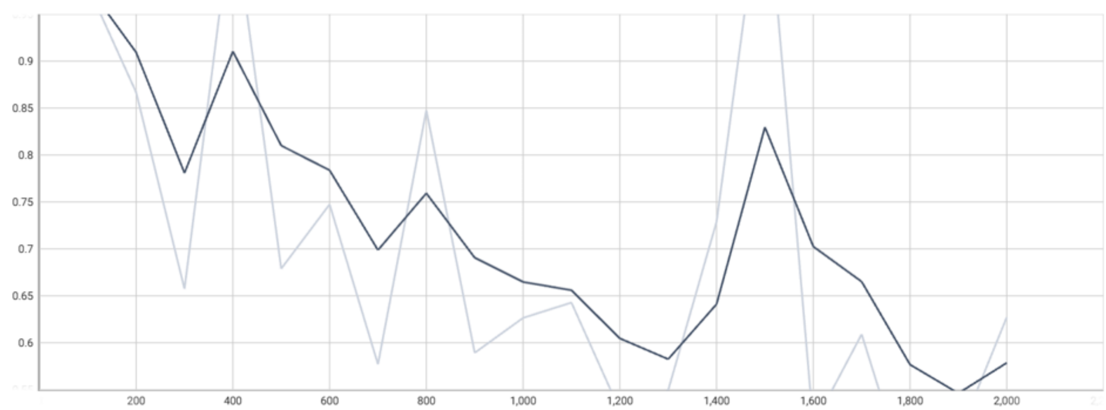


Fig.6. Total Loss

Fig.7. Learning Rate

DOI: http://dx.doi.org/10.25139/ijair.v7i1.9351

 E - ISSN : 2686-6269

Fig.7 illustrates the learning rate progression throughout the training process. The learning rate is a critical hyperparameter that determines the extent to which model weights are updated during each training step, thereby influencing the speed and stability of convergence. In Fig. 7, the vertical axis represents the learning rate values, while the horizontal axis corresponds to the number of training steps. The plot includes two lines: a grey line indicating the smoothed trend of the learning rate and a black line showing the actual values recorded during Training. It is observed that with a training set of 182 images, the learning rate stabilizes at approximately 0.065 around the 1,800th training step, within a total of 2,000 steps. This stabilization indicates that the model maintains a consistent update rate as it approaches the later stages of Training, supporting effective learning and convergence.

B. Training using 282 datasets

Based on the research results using 282 training data sets, the data for classification loss training can be seen in Fig. 8. Classification loss is a function that trains the classification to determine the target object type. Fig.8 presents the classification loss observed during the training process using 282 training images. The vertical axis represents the classification loss values, while the horizontal axis denotes the number of training steps. The grey line indicates the smoothed trend of the classification loss, and the black line illustrates the actual values recorded during Training. The classification loss progressively decreases and reaches approximately 0.18 after 2,000 training steps, indicating improved model performance in correctly identifying target classes.

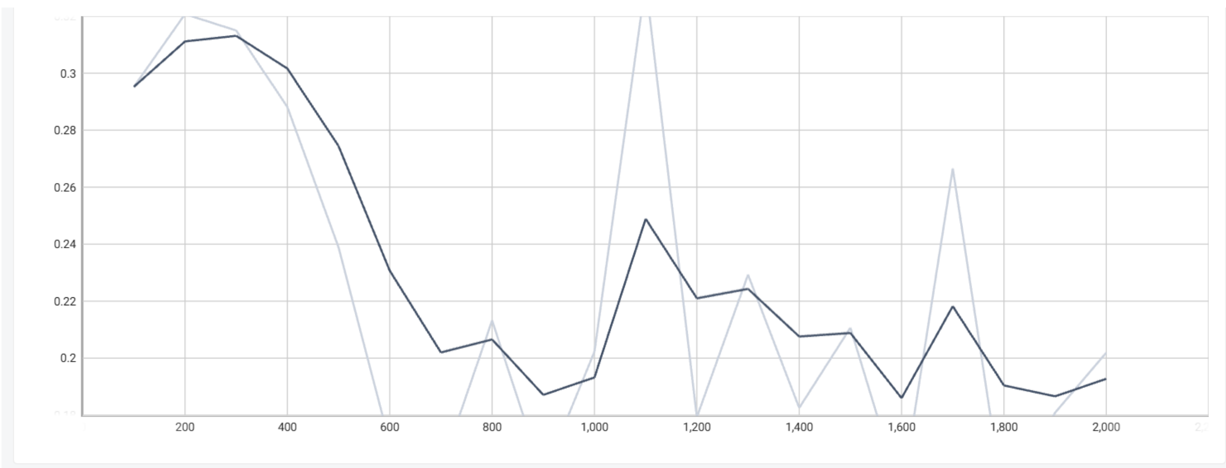


Fig.8. Classification Loss Training

Meanwhile, Fig.9 illustrates the localization loss observed throughout the training process. This loss function measures the discrepancy between the predicted bounding boxes and the ground truth annotations, enabling the model to learn precise object localization. It plays a critical role in improving the spatial accuracy of object detection outcomes. The vertical axis represents the localization loss values, while the horizontal axis denotes the number of training steps. The grey line reflects the smoothed trend of the loss, and the black line indicates the actual recorded values. Based on Fig.9, using 282 training datasets, the localization loss decreases over time and reaches approximately 0.13 after 2,000 training steps, indicating a strong alignment between predicted and actual object positions.

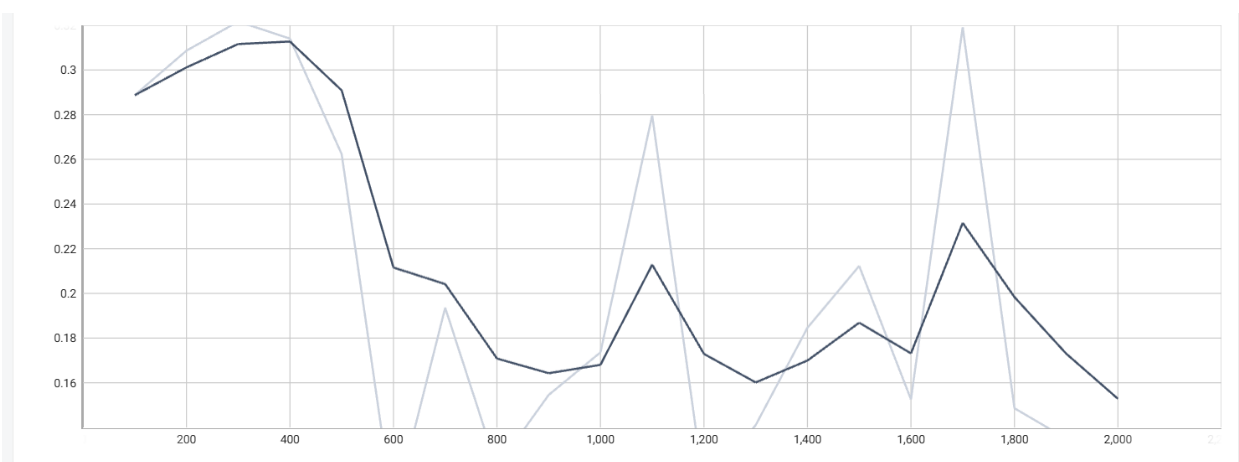


Fig.9. Localization Loss

DOI: http://dx.doi.org/10.25139/ijair.v7i1.9351

 E - ISSN : 2686-6269

Fig.10 presents the regularization loss observed during the training process using 282 datasets. Regularization loss is employed to minimize overfitting and enhance the generalization capability of neural networks by penalizing complex model parameters. The vertical axis represents the regularization loss values, while the horizontal axis corresponds to the number of training steps. The grey line shows the smoothed progression of the loss, whereas the black line depicts the actual recorded values throughout Training. As illustrated, the regularization loss exhibits a steady downward trend as the model approaches 2,000 training steps, reflecting improved generalization. However, an anomaly is observed during the validation process, where the regularization loss briefly increases around step 500, followed by a subsequent decrease at step 600, indicating a momentary fluctuation before the model resumes stable learning behaviors.

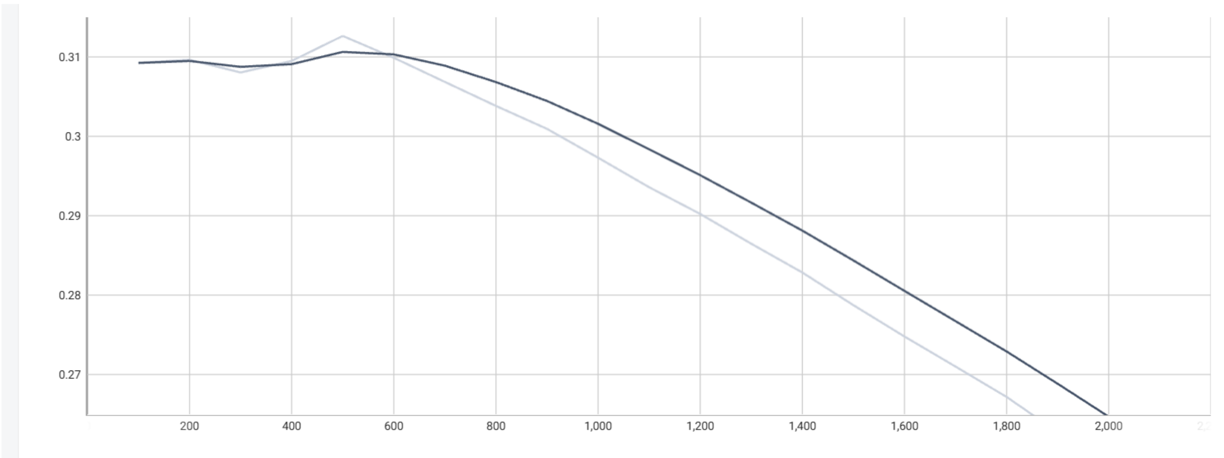


Fig.10. Regularization Loss

Fig.11 depicts the total loss throughout the training process using 282 training datasets. The total loss represents the combined measure of all individual loss components, classification loss, localization loss, and regularization loss—providing a comprehensive indication of the model's overall learning performance. The vertical axis indicates the total loss values, while the horizontal axis represents the number of training steps. The grey line illustrates the smoothed trend of the total loss, whereas the black line shows the actual recorded values at each step. The total loss consistently declines as training progresses and reaches approximately 0.2 after 2,000 training steps, demonstrating effective convergence and stable model optimization.

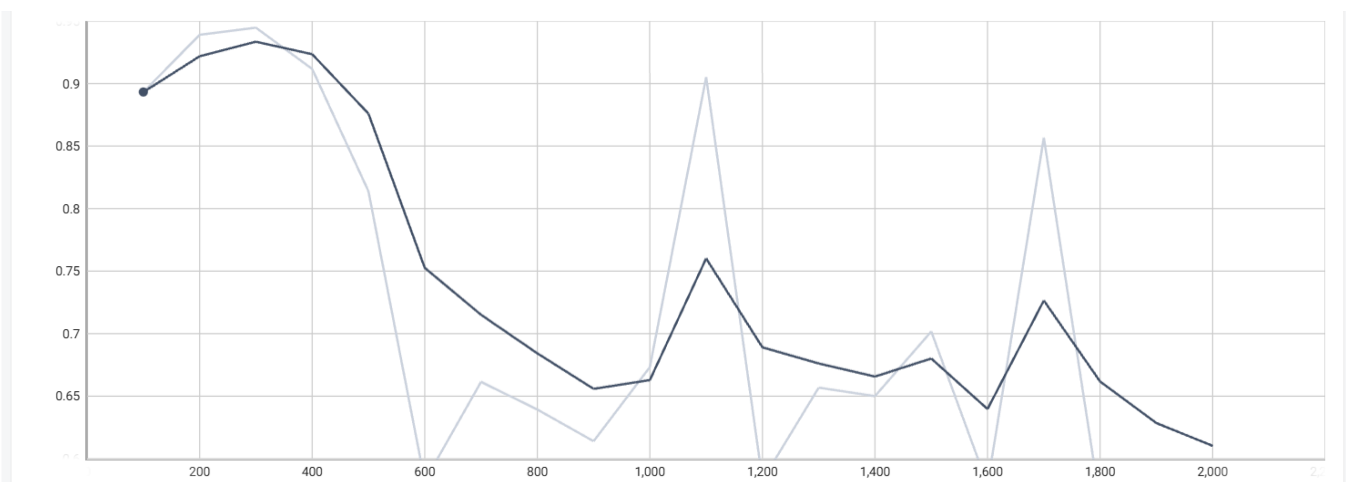


Fig.11. Total Loss

The learning rate progression during the training process, as shown in Fig.12, reflects how this hyperparameter regulates the magnitude of parameter updates throughout model optimization. It plays a vital role in determining convergence speed and Training stability. The vertical axis represents learning rate values, while the horizontal axis corresponds to the number of training steps. The grey line indicates the smoothed trend, and the black line displays the actual recorded values. Using 282 training datasets, the learning rate gradually stabilizes and reaches approximately 0.028 at around 1,800 steps within 2,000 training iterations—indicating that the learning schedule is well-adjusted to support efficient model convergence.

DOI: http://dx.doi.org/10.25139/ijair.v7i1.9351

 E - ISSN : 2686-6269

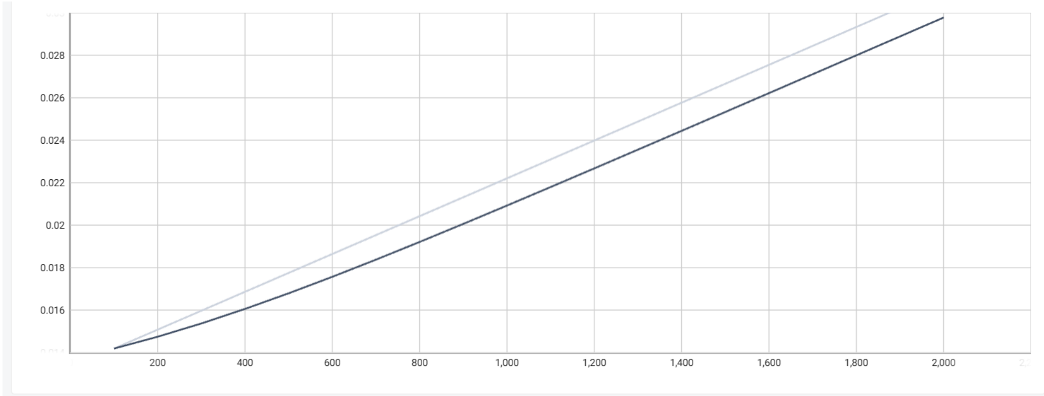
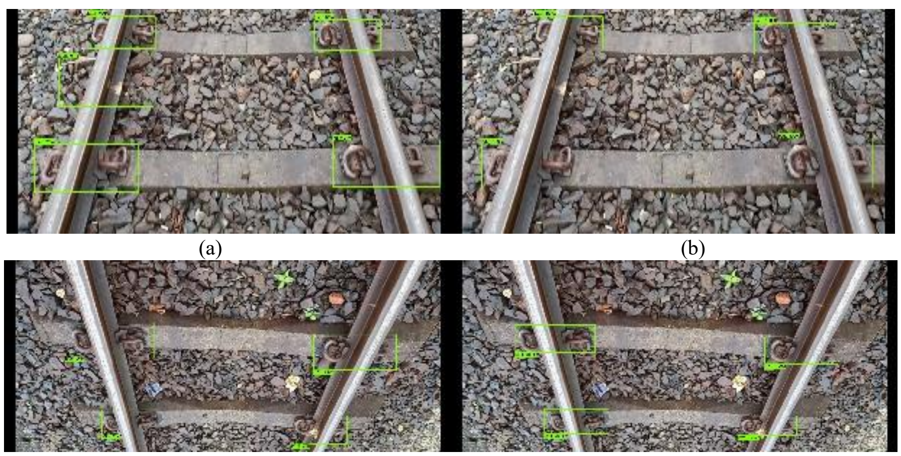


Fig. 12. Learning Rate

C. Testing results using 18 datasets

The object detection results are illustrated in Fig.13. During the testing phase, the system successfully identified bolt positions on the railway tracks. However, initial tests revealed misclassifications, where the system falsely detected bolts in locations where none were present. In subsequent testing, the model improved performance by accurately recognizing the correct bolt positions. Several external factors can influence model performance in image processing, with lighting conditions being one of the most critical. To further enhance precision in object recognition, it is recommended that the diversity of the training dataset be increased by including images captured under various conditions.


(a) (b)
(c) (d)
Fig.13. Image of bolt object detection result

Additionally, improvements in model architecture—such as modifying specific layers, may contribute to higher detection accuracy. Previous studies have reported an accuracy rate of approximately 79% [21], indicating that preprocessing strategies and dataset expansion enhancements are essential to improve detection performance. The precision results obtained from the trained and tested images are presented in Fig.14, where the vertical axis represents the precision values, and the horizontal axis corresponds to the training steps.

Fig.14. Bolt Object Detection Precision Results

DOI: http://dx.doi.org/10.25139/ijair.v7i1.9351

 E - ISSN : 2686-6269

The recall results of the trained and tested images are presented in Fig.15, where the vertical axis represents the recall values, and the horizontal axis corresponds to the number of training steps. Based on the evaluation conducted using 2,000 training steps, the system achieved a precision score of 92.64%. Precision is the ratio of correctly predicted positive instances (true positives) to the total number of predicted positives. Meanwhile, the recall score reached 64.87%, representing the ratio of true positive predictions to the total number of actual positive instances in the dataset. These results indicate that while the model demonstrates high precision, it exhibits lower recall. This suggests that the system is highly accurate when it predicts the presence of bolts but may miss some actual bolts during detection. Improving the recall performance may require adjustments in model sensitivity, data augmentation, or the inclusion of more diverse training samples.

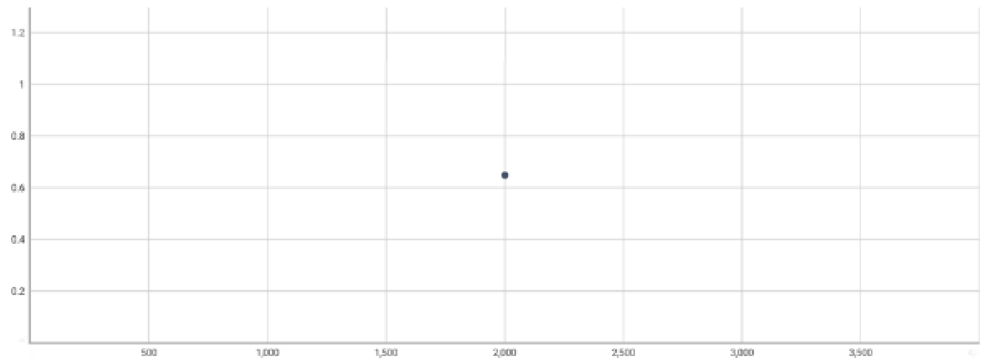


Fig.15. The Recall Result

The mAP value obtained from the object detection process is illustrated in Fig.16. The results of this study show that the mAP score for bolt detection is 0.3475. As mAP is a metric ranging from 0 to 1, with higher values, indicating better overall detection accuracy, this result reflects a relatively low performance in terms of average precision across all tested objects. Despite this, the application of the SSD with ResNet-50 V1 backbone yielded a high precision score exceeding 90%, indicating that when the model detects a bolt, it is highly likely to be correct. Furthermore, the model successfully detected bolts on railway tracks, demonstrating its ability to localize and classify the target object effectively. Although the overall mAP value is modest, the combination of high precision and successful detection illustrates the potential of the ResNet-50 V1 SSD architecture for this application. These findings support the conclusion that the proposed method is effective and can serve as a basis for further refinement and future research in railway infrastructure inspection using deep learning techniques.

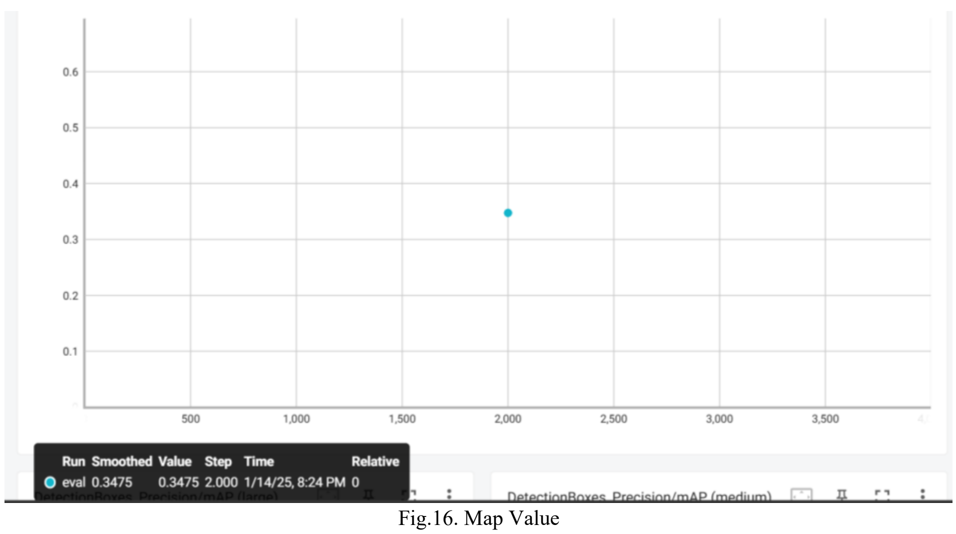


Fig.16. Map Value

IV. CONCLUSION

Several performance improvements were observed based on the experiments conducted using 182 and 282 training datasets. The classification loss decreased from 0.20 to 0.18, and the total loss dropped significantly from 0.40 to 0.20. Additionally, the learning rate at convergence declined from 0.065 to 0.028 when using 282 training samples, indicating a more stable training process. However, there was an increase in localization loss, rising from 0.07 to 0.13, suggesting that additional data alone did not improve the model's ability to localize objects precisely. The regularisation loss also revealed a noticeable difference between the

DOI: http://dx.doi.org/10.25139/ijair.v7i1.9351

 E - ISSN : 2686-6269

training and validation curves, particularly in the experiment using 282 datasets, indicating potential overfitting or sensitivity to data variability. Further evaluation of the detection performance on bolt objects using the ResNet-50 V1 SSD model yielded a precision score of 92.64%, a recall score of 64.87%, and mAP of 0.3475. These values were obtained using a total of 2,000 training steps. The results demonstrate that while the model is highly precise in its predictions, there is room for improvement in recall and overall detection consistency, particularly regarding mAP.

REFERENCES

[1] D. A. Lestari and A. A. Mustafidah, “Strategi Perusahaan PT Kereta Api Indonesia (PERSERO) Dalam Meningkatkan Pelayanan Jasa Transportasi Kereta,” *Gudang J. Multidisiplin Ilmu*, vol. 2, pp. 152–157, 2024.

[2] P. Ardianto, “1.731 Km Jalur Kereta Api Dibangun dan Direvitalisasi di Era Jokowi,” 2024. https://investor.id/business/374987/1731-km-jalur-kereta-api-dibangun-dan-direvitalisasi-di-era-jokowi.

[3] S. Li, "Bolt anomaly detection based on yolov8 and point cloud," in *2024 IEEE 7th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2024, vol. 7, pp. 1918–1922, doi: 10.1109/IAEAC59436.2024.10503788.

[4] Z. Ming, N. Jianguo, L. Xuesheng, and L. Ming, "The safety study of bolt and cable support in shallow roadway," in *2011 International Conference on Multimedia Technology*, 2011, pp. 5651–5655, doi: 10.1109/ICMT.2011.6002822.

[5] AGICO GROUP, "What Is A Track Bolt?," 2023.

[6] P. Prabhakaran, S. Anandakumar, and E. B. Priyanka, "Railroad Maintenance Predictor System for Metro Railroad Systems," in *2022 International Conference on Emerging Smart Computing and Informatics (ESCI)*, 2022, pp. 1–4, doi: 10.1109/ESCI53509.2022.9758349.

[7] D. V. Efanov, G. V. Osadchii, and V. V. Khoroshev, "New Stage in Safety Traffic Control Technologies Development: Digital Railroad Crossing," 2019, doi: 10.1109/RUSAUTOCON.2019.8867700.

[8] Y. Xiao, Z. Yang, J. Xie, and M. Li, "Interface Interaction Design of Railroad Dispatcher Fatigue State Analysis System," in *2023 4th International Conference on Intelligent Design（ICID）*, 2023, pp. 96–100, doi: 10.1109/ICID60307.2023.10396803.

[9] T. P. Kausalya Nandan, T. Rahul, G. Ganga Bhavani, and K. Narsimha, "Railroad Track Defect Detection using Convolutional Neural Networks," 2023, doi: 10.1109/INCET57972.2023.10170140.

[10] S. Saha, S. Karmakar, and D. Manna, "Analysis of Railroad Track Crack Detection using Computer Vision," in *2022 Interdisciplinary Research in Technology and Management (IRTM)*, 2022, pp. 1–4, doi: 10.1109/IRTM54583.2022.9791808.

[11] R. Thendral and A. Ranjeeth, "Computer Vision System for Railway Track Crack Detection using Deep Learning Neural Network," in *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*, 2021, pp. 193–196, doi: 10.1109/ICSPC51351.2021.9451771.

[12] T. Iwasaki, O. Tokumasu, and J. Mitsugi, "Wireless Loose Bolt Detection with Multiple Battery-free Backscatter Sensors," in *2022 IEEE Sensors*, 2022, pp. 1–4, doi: 10.1109/SENSORS52175.2022.9967294.

[13] Y. Wei and X. Lu, "Missing Small Bolt Detection on High-speed Train Using Improved Yolov5," in *2022 8th International Conference on Systems and Informatics (ICSAI)*, 2022, pp. 1–6, doi: 10.1109/ICSAI57119.2022.10005485.

[14] Z. A. Jaffery, D. Sharma, and N. Ahmad, "Detection of missing nuts & bolts on rail fishplate," in *2017 International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT)*, 2017, pp. 36–40, doi: 10.1109/MSPCT.2017.8363969.

[15] I. Mubarokah, P. Laksono, Cepy, R. Safitri, and I. Idris, “Detection of Begomovirus Disease for Identification of Disease Severity Level in Tomato Leaves Using Convolutional Neural Network (CNN),” in *2022 International Symposium on Electronics and Smart Devices (ISESD)*, 2022, pp. 1–6, doi: 10.1109/ISESD56103.2022.9980675.

[16] Q. Lv, "Classification of Grapevine Leaf Images with Deep Learning Ensemble Models," in *2023 4th International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, 2023, pp. 191–194, doi: 10.1109/CVIDL58838.2023.10165757.

[17] K. B. Lee and H. S. Shin, "An Application of a Deep Learning Algorithm for Automatic Detection of Unexpected Accidents under Bad CCTV Monitoring Conditions in Tunnels," *Proc. - 2019 Int. Conf. Deep Learn. Mach. Learn. Emerg. Appl. Deep. 2019*, pp. 7–11, 2019, doi: 10.1109/Deep-ML.2019.00010.

[18] Y.-H. V. Chiang, Y.-R. Lin, and N.-S. Chen, "Using deep learning models to predict student performance in introductory computer programming courses," in *2022 International Conference on Advanced Learning Technologies (ICALT)*, 2022, pp. 180–182, doi: 10.1109/ICALT55010.2022.00060.

[19] D. Patel, F. Patel, S. Patel, N. Patel, D. Shah, and V. Patel, "Garbage Detection using Advanced Object Detection Techniques," 2021, doi: 10.1109/ICAIS50930.2021.9395916.

[20] G. Li, Z. Liu, F. Li, and J. Cheng, "Block Convolution: Toward Memory-Efficient Inference of Large-Scale CNNs on FPGA," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 41, no. 5, pp. 1436–1447, 2022, doi: 10.1109/TCAD.2021.3082868.

[21] *et al.*, “Deteksi Kecacatan Permukaan Rel Menggunakan Metode Deep Learning Neural Network,” *Infotek J. Inform. dan Teknol.*, vol. 6, no. 1, pp. 92–103, 2023, doi: 10.29408/jit.v6i1.7415.

DOI: http://dx.doi.org/10.25139/ijair.v7i1.9351