VOL. 24, NO. 3, 2025, 276-283
elektrika.utm.my
ISSN 0128-4428

ELEKTRIKA
Journal of Electrical Engineering

Fine-Tuning Faster R-CNN with ResNet50 for Infrared-Based Pedestrian Detection in Autonomous Vehicles: Performance and Comparative Analysis

Muhammad Habibullah Abdulfattah[1], **Usman Ullah Sheikh**[1*], **Mohd Afzan Othman**[1], **Nurulaqilla Khamis**[1] and **Fatima Aliyu Shugaba**[1]

[1]Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310, Johor Bahru, Malaysia.

*Corresponding author: usman@fke.utm.my

Abstract: Hyperparameter tuning plays a critical role in optimizing deep learning models for pedestrian detection, particularly in challenging scenarios such as low-light and occluded environments. This study investigates the effect of fine-tuning key hyperparameters in Faster R-CNN with a ResNet50 backbone, focusing on learning rate, optimizer choice, batch size, weight decay, and scheduling. Two models were compared: a baseline Faster R-CNN and a fine-tuned version with optimized training strategies. The fine-tuned model incorporated a reduced learning rate (0.0001), AdamW optimizer with weight decay (0.0005), and a warm-up strategy to improve training stability. Trained for 50 epochs, the fine-tuned model demonstrated superior mean Average Precision (mAP@0.5) of 0.8505 compared to 0.816 in the baseline, with reduced fluctuations and improved convergence. These findings underscore the importance of hyperparameter optimization in enhancing detection accuracy and generalization, particularly for pedestrian detection.

Keywords: Autonomous Vehicles, Faster R-CNN, Hyperparameter Tuning, Infrared Imagery, Pedestrian Detection

Article History: received 26 March 2025; accepted 14 May 2025; published 22 December 2025
Digital Object Identifier 10.11113/elektrika.v24n3.714

1. INTRODUCTION

Pedestrian detection has been a major focus in computer vision research because of its critical importance in diverse applications such as surveillance systems, robotics, and most notably, autonomous driving. Modern pedestrian detection techniques increasingly rely on deep learning-based object detection models like Faster R-CNN [1], Single Shot Detector (SSD) [2], and You Only Look Once (YOLO) [3, 4] to identify and classify pedestrians. Among these, Faster R-CNN is particularly effective in handling occlusions due to its region proposal network (RPN), which enhances localization accuracy even in densely populated or obstructed scenes. Unlike SSD [2] and YOLO [3, 4], which may struggle with small or partially visible objects due to their predefined anchor sizes, Faster R-CNN dynamically refines its region proposals, allowing it to perform better in challenging conditions such as low contrast, varying object scales, and occlusion. Although its two-stage detection process prioritizes precision over speed, this trade-off makes it an ideal choice for applications where detection accuracy is critical, such as autonomous vehicle perception systems.

Several studies have demonstrated the potential of Faster R-CNN for object detection across various challenging scenarios. For instance, Asad Ullah et al. [5] explored pedestrian detection using infrared images and proposed two modifications to Fast R-CNN to enhance detection accuracy and speed. Their work showed that using a single-channel input significantly improved speed, while adding an extra convolutional layer increased detection accuracy.

Similarly, Gao et al. [6] assessed the performance of Faster R-CNN on the Caltech Pedestrian dataset, reporting an average precision (AP) of 51.9%. While this result reflects a respectable level of detection accuracy and an impressive inference speed of 0.07 seconds per image, the authors acknowledged that the performance could be limited by the inherent challenges posed by the dataset. The Caltech Pedestrian dataset comprises a highly diverse collection of pedestrian images exhibiting significant variations in pose, scale, and occlusion levels. Such diversity makes it difficult for standard detection models to achieve high accuracy consistently across all scenarios. These challenges highlight the need for more refined optimization techniques to enhance model robustness and accuracy. The authors mentioned that to address this, a more rigorous fine-tuning process focusing on hyperparameter optimization, potentially involving strategies such as exhaustive grid search or even more sophisticated methods like Bayesian optimization, could be implemented. Adjusting parameters such as learning rate, batch size, anchor scales, and optimization algorithms could considerably improve the model's ability to generalize across varying pedestrian appearances. Moreover, incorporating additional techniques like data augmentation and multi-scale feature extraction may further contribute to improving detection performance in complex scenarios.

Additionally, Akshatha et al. [7] conducted a study on human detection in aerial thermal images using Faster R-CNN and SSD algorithms, where they aimed to enhance detection performance by fine-tuning hyperparameters such as learning rate and batch size. By carefully adjusting these parameters during the training process and monitoring the loss function, they ensured the model learned effectively without overfitting despite the limitations of available resources. Their approach resulted in the Faster R-CNN model with a ResNet50 backbone achieving an impressive mAP of 100% on the OSU thermal dataset and 55.7% on the AAU PD T dataset. Moreover, optimizing the anchor parameters contributed to a notable 10% improvement in mAP, demonstrating the effectiveness of their tuning process.

Furthermore, Gonzales-Martínez et al. [8] also investigated the impact of hyperparameter tuning on Faster R-CNN for persistent object detection in radar images, highlighting the importance of initializing weights and selecting the appropriate optimizer to improve recall from 0.7576 to 0.9394. These findings underscore the effectiveness of enhancing Faster R-CNN through network adjustments, hyperparameter tuning, and customized modifications, resulting in improved accuracy and robustness in various detection tasks.

While Faster R-CNN has demonstrated impressive capabilities in pedestrian detection, the performance is highly dependent on hyperparameter settings. This study evaluates the effect of parameter tuning on Faster R-CNN with ResNet50 backbone by modifying key training parameters and comparing results against a baseline model.

We adopted infrared (IR) imagery in this study because we compared the pedestrian detection performance of IR and RGB image modalities within autonomous vehicle scenarios. The findings revealed that the IR model consistently surpassed the RGB model, achieving approximately 3% higher mAP. This improved performance of the IR model is largely due to its effectiveness in detecting pedestrians under challenging conditions such as low-light environments and partial occlusions. By capturing thermal signatures, the IR model can identify human silhouettes that might not be visible in RGB images. Unlike identification systems that rely on details like color, facial features, or clothing, pedestrian detection for autonomous driving focuses solely on detecting the presence of pedestrians to ensure safe navigation.

In addition to manual tuning approaches, recent advancements in hyperparameter optimization have introduced more efficient and automated strategies. Techniques such as Bayesian Optimization [9], Tree-structured Parzen Estimator (TPE) [10], and Hyperband [11] are increasingly being employed in deep learning pipelines to optimize learning rates, weight decay, and other critical training parameters. These methods reduce the need for exhaustive manual experimentation by intelligently exploring the hyperparameter space. While this study relies on empirical tuning to assess specific parameter effects, these automated approaches offer promising directions for future work aimed at further improving training efficiency and model generalization.

1.1 Dataset

For this research, we made use of the FLIR ADAS dataset [12], which provides both visible light (RGB) and infrared (IR) thermal images. This dataset is particularly valuable for detecting objects under various lighting conditions, including both daytime and nighttime environments. It comprises a total of 10,228 images, with approximately 60% (6,136 images) captured during the day and 40% (4,092 images) taken at night. The images are annotated with bounding boxes for object detection, covering four categories: cars, pedestrians, bicycles, and dogs. The dataset is divided into two parts: a training set containing 8,862 images and a validation set with 1,366 images. All images are standardized to a resolution of 640×512 pixels and were captured using the FLIR Tau2 Camera. This study specifically focuses on pedestrian detection, which is critical for real-world applications like autonomous driving, where accurate detection under varying lighting conditions is essential.

2. METHODOLOGY

The methodology adopted in this study involves training and evaluating two versions of the Faster R-CNN model with ResNet50 as the backbone. The first version is the Baseline Faster R-CNN, which employs the default hyperparameter settings, while the second version is the Fine-Tuned Faster R-CNN, where various hyperparameters were carefully adjusted to enhance performance. The primary goal is to identify how these modifications impact the model's accuracy and robustness when applied to pedestrian detection tasks. Both models were trained using the same dataset under identical conditions to ensure a fair comparison. Key differences between the two models are highlighted and discussed in detail. Figure 1 shows the architecture of the proposed system.

2.1 Dataset and Preprocessing

The FLIR dataset, which offers annotated infrared (IR) images, forms the basis for this research focused on detecting pedestrians in occluded scenarios. Its unique composition makes it ideal for pedestrian detection in difficult environments such as low-light conditions, nighttime settings, and situations involving occlusions, making it highly relevant for autonomous vehicle systems. However, earlier studies [13-16] have highlighted several shortcomings of the FLIR dataset, including inconsistent annotations, limited diversity in environmental conditions, and varying object appearances. These issues present challenges when attempting to build robust and generalizable models based exclusively on this dataset. To address these limitations and enhance the dataset's suitability for detecting occluded pedestrians, several

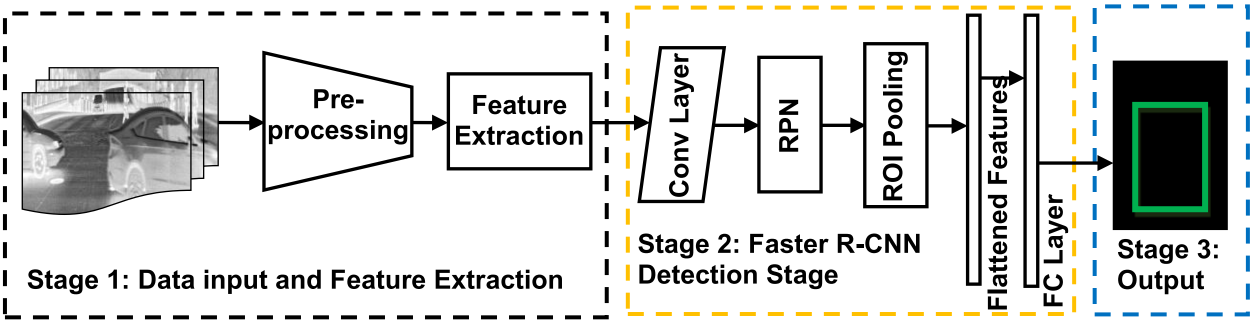


Figure 1. Presents the framework of the proposed detection system, comprising three key stages: Data Input and Pre-processing, Detection using Faster R-CNN, and Output Generation.

refinement steps were implemented:

i. **Selecting Relevant Samples:** Images were meticulously chosen to include only those featuring at least one annotated "person" instance. This approach maintains the dataset's focus on pedestrian detection by eliminating irrelevant or confusing samples.

ii. **Managing Occlusions:** Both visible and partially obscured pedestrian instances were preserved, allowing the model to effectively learn pedestrian detection even when individuals are not fully visible.

During the refinement process, a comprehensive dataset of 5,838 infrared (IR) images was prepared specifically for this study. The dataset was then split into three groups: 4,088 images were used for training, 1,167 for validation, and 583 for testing. This distribution was designed to support effective model training while ensuring unbiased performance evaluation. To enhance image quality and accurately extract important features, several preprocessing steps were applied:

a. **Image Resizing:** The infrared images in the dataset were initially in a resolution of 640×512. To ensure consistency across the dataset and simplify batch processing during training, all images were resized to a uniform resolution of 640×512 pixels. This standardization aimed to maintain coherence in image dimensions throughout the training process.

b. **Pixel Normalization:** To enhance training stability and promote faster convergence, the pixel values were scaled to a range between 0 and 1.

$$I_{norm} = \frac{I - \mu}{\sigma} \tag{1}$$

Where:
I represents the original image,
μ denotes the average pixel value, and
σ indicates the standard deviation of the pixel values.

c. **Noise Reduction:** To mitigate sensor noise, several data augmentation techniques were applied, such as MedianBlur, MotionBlur, and general Blur operations. These methods help in smoothing out image details, effectively reducing high-frequency noise while preserving essential features required for accurate pedestrian detection.

d. **Enhancing Contrast:** Given that pedestrian features in infrared images can be dim due to varying thermal intensities, adaptive histogram equalization (AHE) was utilized to enhance local contrast, thereby improving the visibility of objects.

2.2 Model Architecture

The Faster R-CNN [1] architecture with ResNet50 was selected for this study because of its superior accuracy in detecting objects. This approach follows a two-stage process, where the initial stage involves generating region proposals through a Region Proposal Network (RPN). In the second stage, these proposals are classified and refined to enhance detection precision [1]. Both models utilized Faster R-CNN with a ResNet50 feature extractor. The detection pipeline remained consistent across both configurations to ensure a fair comparison.

2.3 Experimental Setup

The training configurations for both models are as follows:

- **Baseline Faster R-CNN**: Default hyperparameters.
- **Fine-Tuned Faster R-CNN**: Adjusted hyperparameters including learning rate, optimizer, batch size, and scheduling.

2.4 Parameter Adjustments

The primary modifications in the fine-tuned model were:

i. **Learning Rate (LR) Adjustment:** The learning rate was reduced from 0.001 to 0.0001. The choice of learning rate significantly influences model convergence. A high learning rate can cause the model to diverge, whereas a very low learning rate can result in slow learning. By lowering the learning rate, the model updates weights in smaller steps, leading to smoother convergence and better generalization. Additionally, a Cosine Annealing Learning Rate Scheduler was applied, which gradually decreases the learning rate over epochs. This helps prevent premature convergence to a suboptimal solution and improves long-term learning stability.

ii. **Batch Size**: The batch size was maintained at 8 to balance computational efficiency and stability. A larger batch size can stabilize training and improve parallelism but requires higher memory. Conversely, a smaller batch size introduces more noise into gradient updates but allows for more frequent weight updates, leading to better generalization. The chosen batch size provided an optimal trade-off, preventing overfitting while ensuring smooth convergence.

iii. **Optimizer Selection**: The optimizer was changed from Stochastic Gradient Descent (SGD) to AdamW with a weight decay of 0.0005. While SGD is effective for large-scale learning, it requires fine-tuned momentum and decay parameters. AdamW is an adaptive optimizer that dynamically adjusts learning rates for each parameter, improving convergence speed. The incorporation of weight decay helps in regularization, reducing overfitting by penalizing large weights. Similar to Adam, AdamW employs adaptive learning rates and incorporates bias correction. However, what sets it apart is its ability to apply L2 regularization separately, which enhances its capacity for generalization. Recent research has shown that AdamW often outperforms Adam, especially in deep learning applications where effective weight regularization is critical for achieving reliable performance [17, 18]. This characteristic makes AdamW particularly suitable for pedestrian detection tasks, where fine-tuning weights is essential for attaining high detection accuracy.

Update Rule in Adam:

$$\theta_t = \theta_{t-1} - \eta\left(\frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon} + \lambda\theta_{t-1}\right) \qquad (2)$$

Where:

θ = represents the model's parameters, such as weights or biases that are being adjusted.

t = indicates the current iteration or time step during the training process.

η = denotes the learning rate, which determines the step size at each iteration while moving toward the minimum of the loss function.

$\widehat{m}_t$ = refers to the corrected estimate of the first moment (mean of gradients) to reduce bias.

$\hat{v}_t$ = indicates the corrected estimate of the second moment (uncentered variance of gradients) for better stability.

ϵ = is a tiny value added to prevent division by zero during the calculation.

λ = signifies the rate of weight decay, which acts as a regularization term to prevent overfitting.

In the Adam optimizer, weight decay is applied directly within the gradient update step, which can sometimes reduce its effectiveness as a regularization technique.

Update Rule in AdamW:

$$\theta_t = \theta_{t-1} - \eta\left(\frac{\widehat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}\right) \qquad (3)$$

$$\theta_t \leftarrow \theta_t - \eta\lambda\theta_{t-1} \qquad (4)$$

The AdamW optimizer improves regularization by applying weight decay separately from the gradient update, which frequently leads to enhanced model performance.

iv. **Warm-up Strategy**: A learning rate warm-up was introduced to stabilize initial training. Warm-up prevents abrupt weight updates at the start of training when gradients are unstable. This was implemented by initially setting the learning rate to a small value and gradually increasing it over a few iterations before transitioning to the scheduled learning rate. This approach prevents the optimizer from making erratic updates at the beginning of training and ensures a more stable learning trajectory.

2.4.1 Why These Adjustments Were Necessary:

These modifications were crucial for enhancing the training stability, convergence speed, and generalization capability of the model. The learning rate adjustment and Cosine Annealing Scheduler helped the model refine its feature representations without abrupt changes. The choice of AdamW provided better adaptation during weight updates, reducing the risk of overfitting. The batch size ensured a smooth gradient update process without overwhelming GPU memory. Finally, the warm-up strategy prevented instability during early training, leading to a more robust and effective model. Together, these adjustments significantly improved Faster R-CNN's detection accuracy while ensuring a balanced trade-off between precision and computational efficiency.

2.5 Evaluation Metrics

The mAP used in this study, including mAP and mAP@0.5, serves as a key evaluation metric for assessing model performance, particularly its accuracy and precision in object detection. mAP measures the average precision across various recall levels, while mAP@0.5 evaluates precision at a specific Intersection over Union (IoU) threshold of 0.5, where a detection is considered accurate if the predicted bounding box covers at least 50% of the actual object. In addition to mAP-based accuracy measures, we also considered inference speed, reported in frames per second (FPS), as an important metric for assessing real-time suitability. Details on FPS are presented under Results and Discussion section in Quantitative Analysis. During training, various loss functions are applied to enhance model prediction capabilities, including training loss, bounding box regression loss, objectness loss, and RPN loss. These losses are combined to form the total loss function, calculated as:

$$L = L_{cls} + L_{reg} + L_{obj} + L_{rpn} \qquad (5)$$

Classification loss L_{cls} is computed using cross-entropy loss to accurately predict class labels. Bounding box regression loss L_{reg} is evaluated using Smooth L1 loss to balance accuracy and stability. Objectness loss L_{obj} ensures effective differentiation between objects and background, while RPN loss L_{rpn} focuses on refining bounding box proposals.

3. RESULTS AND DISCUSSION

The experimental results were analyzed to assess the impact of hyperparameter tuning on the performance of Faster R-CNN with ResNet50 backbone. The evaluation was performed using metrics such as mean Average Precision (mAP) and various training loss components, including classification loss, bounding box regression loss, objectness loss, and region proposal network (RPN) loss. A comparison between the Vanilla and Fine-Tuned models highlights the improvements gained through careful parameter tuning.

3.1 Quantitative Analysis

The evaluation primarily focused on comparing the highest mAP@0.5, final training loss, and testing mAP@0.5 between the Base Model and the Fine-Tuned Model, as shown in Table 1. The fine-tuned model achieved a superior highest mAP@0.5 of 0.8505 compared to 0.8161 in the base model. Additionally, the final training loss of the fine-tuned model was reduced to 0.1255, showing an improvement in training efficiency compared to the base model's loss of 0.1461. During testing, the fine-tuned model achieved a higher mAP@0.5 of 0.8237 compared to 0.7833 of the base model. These results highlight the effectiveness of hyperparameter tuning in enhancing model performance and robustness. In addition to accuracy-based metrics, inference speed was evaluated on the fine-tuned model in terms of frames per second (FPS), which reflects how many images the model can process per second during inference. The fine-tuned model achieved an inference speed of 10.42 FPS, measured using the best-performing checkpoint on test samples. FPS is a critical metric for real-time applications such as autonomous driving, where systems must detect and respond to objects promptly. The required FPS can vary depending on the vehicle's speed and operational context. For instance, in urban environments, where vehicles typically move at 10-30 km/h, a detection rate of around 10-15 FPS is generally sufficient to ensure timely response [19], [12]. In contrast, highway speeds (e.g., 50-100 km/h) require faster detection rates typically in the range of 20-30 FPS to maintain safety margins and avoid collisions [16]. Thus, the achieved speed satisfies the real-time constraints of urban autonomous driving. It is important to note that FPS performance is influenced by the characteristics of the deployment platform, including GPU capability, memory bandwidth, and hardware configuration. The reported FPS in this study was obtained on a system equipped with an NVIDIA GeForce RTX 3050 GPU (8 GB VRAM), Intel(R) Core(TM) i9-10900F CPU @ 2.80 GHz, and 16 GB RAM. This finding, together with the improved detection accuracy, confirms the model's potential for deployment in real-world autonomous driving systems.

3.2 Performance Comparison and Loss Analysis

The performance comparison and loss analysis reveal the significant benefits of hyperparameter tuning in enhancing the Faster R-CNN model's performance. The fine-tuned model achieved a higher overall mAP@0.5 of 0.8505 compared to the base model's 0.8161, indicating improved localization and classification performance. This improvement is largely attributed to the optimized learning rate, AdamW optimizer, and effective scheduling strategy employed during training. Additionally, the smoother training graph of the fine-tuned model, as shown in Figure 2(b), reflects better convergence stability compared to the fluctuating pattern seen in the base model Figure 2(a).

The use of the AdamW optimizer with weight decay, combined with a well-designed warm-up strategy, contributed to more efficient training and better generalization. The consistent reduction in training loss further highlights the model's learning efficiency. Despite slightly higher initial losses, the fine-tuned model achieved a higher testing mAP@0.5 of 0.8237, outperforming the base model's 0.7833. These findings confirm the importance of properly tuned hyperparameters for achieving robust performance and generalization in object detection tasks.

Figure 3(a) illustrates a scenario where the Baseline Model struggled to detect a pedestrian (indicated by a red circle) within a crowded scene featuring several partially occluded individuals. The overlapping pedestrians likely caused this misdetection. In contrast, Figure 3(b) shows that the Fine-Tuned Model successfully addressed the occlusion challenge, accurately detecting all pedestrians present. And Figure 3(c) highlights a case where the Baseline Model incorrectly identified pedestrians (marked with a red circle) in a scene while the scene has only one pedestrian which was accurately detected by the Fine-Tuned Model in Figure 3(d). Zoomed-in views of missed detections are shown in Figure 3 (e and f), with missed and false positives highlighted in red circles for clarity. These examples highlight the Baseline Model's limitations in dealing with complex conditions like occlusions, while demonstrating the Fine-Tuned Model's enhanced capability to overcome these challenges.

Table 1. Summary Of Training and Testing Performance

Modality	Base Model	Fine-Tuned Model
Epochs	50	50
Batch Size	8	8
Highest mAP@0.5	0.8161	0.8505
Final Training Loss	0.1461	0.1255
Testing mAP@0.5	0.7833	0.8237

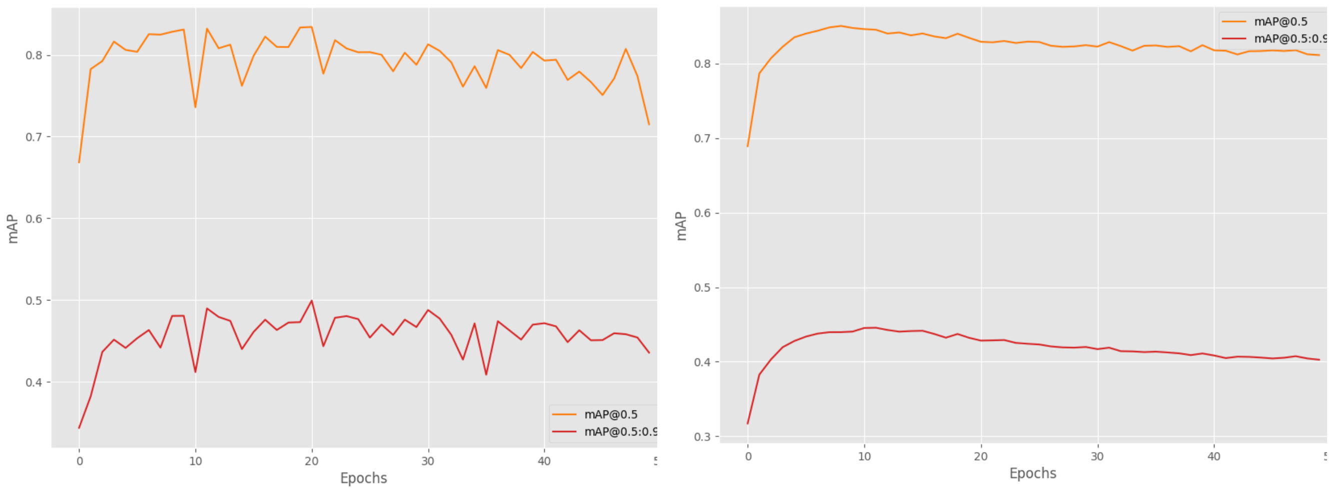
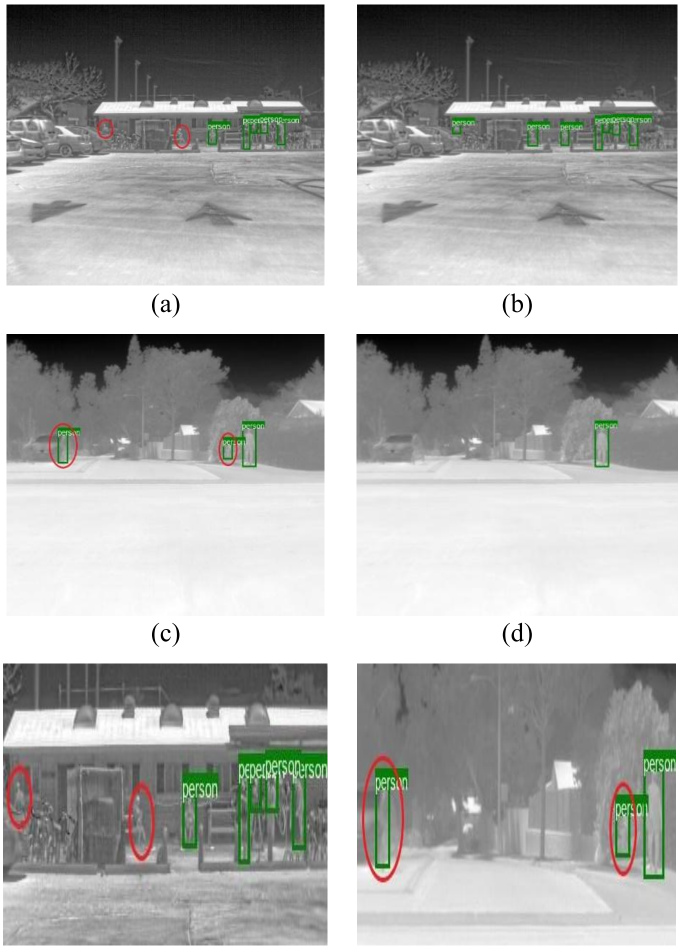


Figure 2. mAP for 50 epochs (a) Baseline Model and (b) Fine-Tuned Model.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3. Shows a comparison between the Baseline Model (a and c) and the Fine-Tuned Model (b and d) in detecting pedestrians. While (e and f), shows the zoomed-in views of missed detections and false positives highlighted in red circles.

3.3 Ablation Study on Hyperparameter Settings

To systematically assess the impact of individual hyperparameter modifications, an ablation study was conducted. Table 2 summarizes the experimental settings and corresponding mAP@0.5 scores. Individually, switching the optimizer from SGD to AdamW improved the mAP@0.5 by 2.57%, while reducing the learning rate yielded a 1.24% gain compared to the baseline. Introducing a warm-up strategy also provided smoother early training dynamics, leading to a moderate improvement of 0.87%. The final fine-tuned model, incorporating all three modifications (AdamW, reduced learning rate, and warm-up), achieved the highest mAP@0.5 of 0.8505, gaining 3.44% against the baseline, confirming the cumulative benefits of these tuning choices.

4. DISCUSSION AND CONCLUSION

The results of this study demonstrate the importance of effective hyperparameter tuning in enhancing the performance of Faster R-CNN with ResNet50 backbone for object detection. The comparison between the base model and the fine-tuned model revealed several key improvements resulting from optimized parameter configurations. Notably, the fine-tuned model achieved higher mAP@0.5 values and exhibited smoother convergence patterns during training, which highlights its enhanced stability and generalization capabilities.

The improved performance of the fine-tuned model can be attributed to several factors, including the use of the AdamW optimizer with weight decay, a well-designed warm-up strategy, and the use of a reduced learning rate to stabilize training. These adjustments contributed to more effective feature extraction and better convergence during the training process. The consistent reduction in training loss and increased mAP values further validate the significance of these tuning choices. Beyond accuracy, the fine-tuned model maintained a real-time inference speed of 10.42 FPS, which meets the operational demands of urban autonomous driving systems.

Comparing the performance of the base model and the fine-tuned model underscores the necessity of careful parameter selection when training deep learning models for object detection. While the base model provided reasonable results, the fine-tuned model clearly demonstrated the potential of tuning strategies to enhance model performance.

While this study adopts manual tuning techniques to

Table 2. Ablation Study and Testing Performance

Experiment	Description	Main Hyperparameter Changes	mAP@0.5 (%)	Remarks
Exp 1	Original model (no fine-tuning)	Default SGD optimizer, default learning rate	82.5	Initial reference performance
Exp 1	Optimizer only (AdamW)	Changed optimizer from SGD to AdamW	84.3	Improved weight decay control
Exp 3	Learning rate only	Reduced LR by a factor of 10	85.7	Enhanced convergence stability
Exp 4	Warm-up strategy only	Added gradual warm-up at start of training	84.9	Smoother initial training
Exp 5 (Final)	Combined (AdamW + LR + Warm-up) (Final fine-tuned model)	AdamW, reduced LR, warm-up strategy	**87.4**	Combined effect, best performance

demonstrate the impact of specific configurations, it is worth noting that automated hyperparameter optimization strategies such as Bayesian Optimization, TPE, and Hyperband have been shown to significantly improve the training efficiency and generalization of deep learning models. These techniques intelligently navigate the hyperparameter search space, reducing the reliance on exhaustive grid searches. Future work can explore the integration of these automated search methods, alongside additional augmentation techniques, to further enhance the robustness and accuracy of Faster R-CNN models.

ACKNOWLEDGMENT

The authors extend their appreciation to Universiti Teknologi Malaysia (UTM) for providing the essential support and resources that made this research possible.

REFERENCES

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 39, 2017.

[2] W. Liu *et al.*, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, 2016: Springer, pp. 21-37.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779-788.

[4] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263-7271.

[5] A. Ullah, H. Xie, M. O. Farooq, and Z. Sun, "Pedestrian detection in infrared images using fast RCNN," in *2018 Eighth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 2018: IEEE, pp. 1-6.

[6] S. Gao, "Exploration and evaluation of faster R-CNN-based pedestrian detection techniques," *Applied and Computational Engineering,* vol. 32, pp. 185-190, 2024.

[7] K. Akshatha, A. K. Karunakar, S. B. Shenoy, A. K. Pai, N. H. Nagaraj, and S. S. Rohatgi, "Human detection in aerial thermal images using faster R-CNN and SSD algorithms," *Electronics,* vol. 11, no. 7, p. 1151, 2022.

[8] 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 Latin America Transactions,* vol. 20, no. 4, pp. 677-685, 2022.

[9] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems,* vol. 25, 2012.

[10] J. Bergstra, D. Yamins, and D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *International conference on machine learning*, 2013: PMLR, pp. 115-123.

[11] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *Journal of Machine Learning Research,* vol. 18, no. 185, pp. 1-52, 2018.

[12] F. Munir, S. Azam, M. A. Rafique, A. M. Sheri, M. Jeon, and W. Pedrycz, "Exploring thermal images for object detection in underexposure regions for autonomous driving," *Applied Soft Computing,* vol. 121, p. 108793, 2022.

[13] H. Zhang, E. Fromont, S. Lefèvre, and B. Avignon, "Guided attentive feature fusion for multispectral pedestrian detection," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 72-80.

[14] R. Yadav, A. Samir, H. Rashed, S. Yogamani, and R. Dahyot, "Cnn based color and thermal image fusion for object detection in automated driving," *Irish Machine Vision and Image Processing,* vol. 2, 2020.

[15] H. Zhang, E. Fromont, S. Lefevre, and B. Avignon, "Multispectral fusion for object detection with cyclic fuse-and-refine blocks," in *2020 IEEE International conference on image processing (ICIP)*, 2020: IEEE, pp. 276-280.

[16] S. Vadidar, A. Kariminezhad, C. Mayr, L. Kloeker, and L. Eckstein, "Robust environment perception for automated driving: A unified learning pipeline for

visual-infrared object detection," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022: IEEE, pp. 367-374.

[17] I. Loshchilov, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101,* 2017.

[18] D. Kinga and J. B. Adam, "A method for stochastic optimization," in *International conference on learning representations (ICLR)*, 2015, vol. 5: San Diego, California;, p. 6.

[19] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213-3223.