Applied Information System and Management (AISM)
Volume 8, (1) 2025, p. 169–174
P-ISSN: 2621-2536; E-ISSN: 2621-2544; DOI: 10.15408/aism.v8i1.46226

Deep Learning Model for Automated Tire Crack Detection Using Convolutional Neural Networks

Shofa Shofiah Hilabi[1*], Ahmad Fauzi[2], Savina[3]

***Abstract*—Tire cracks pose a significant safety risk, as undetected defects can lead to severe accidents. Traditional inspection methods rely on manual visual assessments, which are prone to human error. This study proposes an automated tire crack detection system using Convolutional Neural Networks (CNN), leveraging transfer learning techniques to improve accuracy and generalization. A dataset of 600 tire images was collected and preprocessed, including augmentation techniques such as rotation, flipping, and brightness adjustments. The CNN model was trained with different optimizers, including Adam and Stochastic Gradient Descent (SGD), to compare their performance. Experimental results indicate that Adam achieved the highest test accuracy of 78.3% with the lowest test loss of 53%, while SGD required more epochs to reach optimal performance. This study demonstrates the feasibility of deep learning-based automated tire inspection, providing a scalable alternative to traditional methods. Future research should focus on optimizing model architectures, expanding datasets, and integrating real-time detection for industrial applications.**

***Index Terms*—Tire crack detection, CNN, deep learning, image processing, automated inspection.**

I. Introduction

Tires are essential for vehicle safety, ensuring stability, traction, and shock absorption [1], [2], [3], [4]. As seen in Fig. 1, there are some vehicle type of vehicle tire, cracks can develop due to aging, excessive load, or extreme conditions, which increase the risk of failure and road accidents. These cracks, if left undetected, represent a critical safety hazard as they can propagate and lead to catastrophic tire failure, which is a significant contributor to severe road accidents. The conventional approach to tire safety relies on manual visual inspection, a method fraught with limitations [5], [6], [7]. Manual checks are not only time-consuming and labor-intensive, making them inefficient for large-scale operations like fleet management, but they are also fundamentally subjective and inconsistent. The accuracy of such inspections heavily depends on the inspector's experience, lighting conditions, and level of fatigue, leading to a high probability of human error where fine or early-stage cracks are often overlooked.

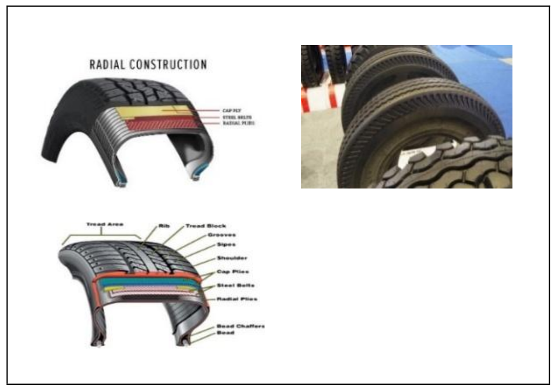

Fig. 1. Vehicle type of vehicle tires [8].

This unreliability poses a direct risk to road safety and highlights an urgent need for an automated, objective, and scalable detection system [9], [10], [11]. To address these shortcomings, this research proposes an automated tire crack detection system leveraging Deep Learning (DL). Recent advances in CNNs have demonstrated remarkable capabilities in image classification and defect detection tasks, offering a path to revolutionize tire inspection [12], [13], [14]. CNNs are particularly effective at recognizing complex and subtle patterns that are often invisible to the human eye, making them ideally suited for identifying tire cracks with high accuracy and consistency. This study aims to develop a robust CNN-based model capable of classifying tire conditions as 'cracked' or 'normal,' thereby providing a more efficient, reliable, and scalable alternative to traditional inspection methods. The significance of this research lies in its potential to enhance vehicle safety proactively, reduce labor costs, and improve maintenance efficiency in the automotive industry. Through transfer learning and data augmentation, this study seeks to enhance model performance, providing an effective tool for automated tire maintenance and safety monitoring. If there are too many cracks in the tires, of course, this will result in traffic accidents, therefore proper handling is needed such as conducting routine and periodic checks to reduce the possibility of accidents [11]. Checking cracks on tires is currently still

Received: 29 April 2025; Revised: 20 May 2025; Accepted: 31 May 2025.
*Corresponding author

[1*]Shofa Shofiah Hilabi, Dept. of Information System, Universitas Buana Perjuangan Karawang, Indonesia (e-mail: shofa.hilabi@ubpkarawang.ac.id).
[2]Ahmad Fauzi, Dept. of Informatics Engineering, Universitas Buana Perjuangan Karawang, Indonesia (e-mail: a.fauzi@ubpkarawang.ac.id).
[3]Savina, Dept. of Information System, Universitas Buana Perjuangan Karawang, Indonesia (e-mail: vickyargubi@gmail.com).

carried out conventionally, where vehicle users look directly to determine whether the tires are cracked or not [15].

The problem in this study is how good is the accuracy and performance of the CNN model in identifying cracks compared to conventional detection methods or other machine learning algorithms and How does the CNN model perform under varying lighting conditions, tire positions, and crack shapes? Does the model remain robust in real environments?

Figure 2 presents a typical example of the fine cracks this research aims to detect. These subtle, web-like fissures on the tire's surface are early indicators of material degradation and represent the precise type of defect that an automated system is designed to identify.

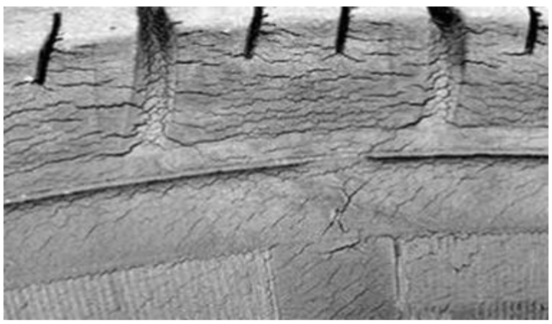

Fig. 2. Fine tire cracks [16]

Meanwhile, Fig. 3 displays a sample of images representing the 'normal' class from the dataset. These images show tires that are free of visible cracks and serve as the negative class for training the CNN model, enabling it to distinguish healthy tires from defective ones.

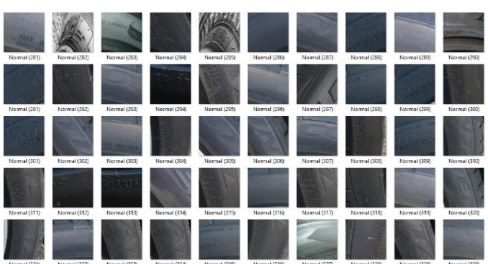

Fig. 3. Normal tires [16]

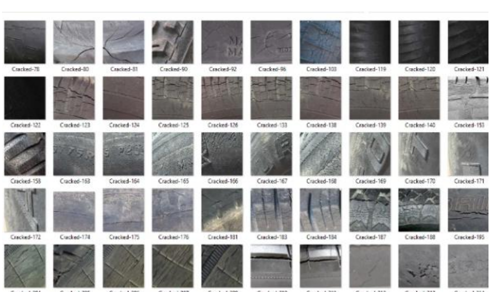

Fig. 4. Cracked tires [16]

Figure 4 showcases a sample of images from the 'cracked' category in the dataset. These images provide the model with diverse visual examples of tire defects and serve as the positive class for the binary classification task, teaching the model what to identify as a defect. Next, Table 1 describe dataset of this study.

Table.1
Dataset Kaggle

Type Data	Data Source	Link Data Source	Training Data 80%	Testing Data 20%
Normal tires	Kaggle	https://www.kaggle.com/code/zizoudinosaur/tire-status-classification/input	240	60
Craked tires	Kaggle	https://www.kaggle.com/code/zizoudinosaur/tire-status-classification/input	240	60
		Total	480	120
	Total overall image		600	

II. Related Work

Automated tire crack detection has been an emerging field, driven by the limitations of traditional manual inspection methods. These methods, often subjective and prone to human error, highlight the need for more accurate and efficient systems. This research emphasized the inadequacy of manual inspections and the importance of automation for enhancing road safety and vehicle maintenance [17].

Recent advancements in integrating CNNs with traditional machine learning classifiers have shown significant promise in various domains, including landslide susceptibility mapping [18].

This study develops an automated inspection system based on CNN to detect cracks and other defects on tire surfaces. The dataset consists of industrial tire images collected directly from the production line. The CNN model successfully achieved 97.3% accuracy. The advantage of this method is that high accuracy on real industrial images can be applied in real-time [19].

This research demonstrated the effectiveness of CNNs in detecting complex patterns in images, such as defects. Their work showed that CNNs can extract hierarchical features from raw data, making them ideal for applications in automotive safety, including tire crack detection. This laid the foundation for the use of CNNs in automating tire inspections [20].

In recent years, CNNs have been widely utilized in various applications, including image classification tasks such as tire crack detection and classification for motorsport analysis, as demonstrated by [21].

This research focused on the application of transfer learning, which allows deep learning models to be trained more efficiently by using pre-trained networks. Their research highlighted how transfer learning could improve model performance with smaller datasets, a technique highly relevant for tire crack detection, where labeled data is often limited [15].

Other research explored tire crack detection using CNN with ResNet-34 architecture, achieving high accuracy in distinguishing between cracked and normal tires. His research highlighted the effectiveness of deep learning techniques in automating tire inspection processes [8].

A novel tire defect detection method based on Faster

P-ISSN: 2621-2536; E-ISSN: 2621-2544; DOI: 10.15408/aism.v8i1.46226

R-CNN has been proposed to improve the accuracy of rehabilitation robots. The method uses Laplace operator and homomorphic filter to enhance the data set, enhancing detection accuracy. Data expansion increases the number of images and improves robustness. The method uses convolution features from the third and fifth layers of the ZF network to extract deep characteristics. The method can accurately classify and locate tire X-ray image defects, with an average recognition rate of 95.4% [22].

Previous study presented a deep learning tire defect detection method, improving traditional ShuffleNet. The method outperforms five other methods, with a detection rate of 94.7%. This robustness and effectiveness save labor costs and reduce detection time, ensuring safety and efficiency in vehicle maintenance [23].

A study explored the use of Generative Adversarial Networks (GAN) for rubber tyre surface inspection, addressing the limitations of traditional inspection methods. By using artificial defect images generated using the Pix2Pix method, the expanded datasets can improve the accuracy of CNN and SVM models. The study also revealed that the artificial data ratio affects classification performance [24]. Also, a proposed a visual inspection framework using a lightweight Transformer for tire defects. The framework uses a dual-path-Transformer feature encoder, a multi-scale fusion Transformer, and a spatial cross Transformer. Tested on a tire radiographic image dataset, the method achieves detection accuracy of 98.57% and mIoU of 85.56%, achieving a balance between accuracy and efficiency [25].

Deep learning, a new field in machine learning, aims to solve object classification in images using Convolutional Neural Network (CNN) methods. CNN requires an iterative training process, requiring heavy computing and a long time. To speed up the training process, GPU performance is needed. Test results show a 96.67% accuracy on a MotoGP racer image with a 473-second training time, demonstrating the effectiveness of CNN in classifying images. GPU performance can also speed up computational processes up to 11 times [26].

To address the common challenge of limited datasets, research has focused on the application of transfer learning, which allows models to be trained more efficiently and can improve performance on smaller datasets [21]. Specific implementations, such as the use of the ResNet-34 architecture, have successfully achieved high accuracy in distinguishing between cracked and normal tires [8]. Furthermore, CNN-based models have also demonstrated their ability to identify subtle defects that might be missed by manual inspection, thus holding great potential for enhancing tire safety and maintenance [22].

It can be concluded that there is still minimal research focusing on the interpretability of CNN models in the context of tire crack detection, which is important for user trust and system debugging.

III. Research Method

A. Research Workflow

This section outlines the systematic methodology employed to develop and evaluate the automated tire crack detection system. The research workflow used in this study is shown in Fig. 5.

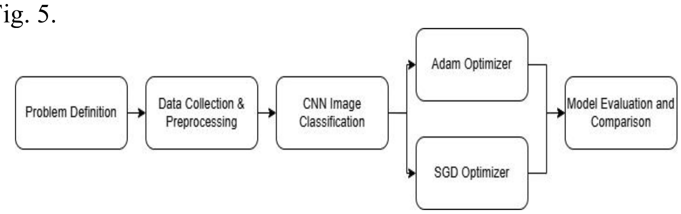

Fig. 5. Research workflow.

B. Dataset and Preprocessing

A dataset comprising 600 tire images was sourced from Kaggle, which includes cracked and normal tire conditions. The dataset was split into 480 training images (80%) and 120 testing images (20%) to ensure a balanced evaluation. The preprocessing steps applied were:

1) *Resizing:* All images were resized to 400×400 pixels.
2) *Normalization:* Pixel values were normalized between 0 and 1.
3) *Data Augmentation:* Techniques such as rotation, flipping, and brightness adjustments were applied to enhance model generalization.

C. Model Architecture

The CNN model utilized in this research follows a standard deep learning pipeline with the following layers:

1) *Input Layer:* Accepts images of size 400×400×3.
2) *Convolutional Layers:* Uses multiple 3×3 filters to extract important features from images.
3) *Activation Function:* RELU (Rectified Linear Unit) applied to introduce non-linearity
4) *Pooling Layers:* Max pooling applied to reduce spatial dimensions while preserving important features
5) *Fully Connected Layers:* Flattened output is passed through dense layers to learn complex patterns
6) *Output Layer:* Uses a Softmax activation function for binary classification (cracked or normal) [18].
7) *Training Process and Hyperparameter Tuning.*

The model training process consists of:

1) *Loss Function:* Categorical cross entropy to measure classification error.
2) *Optimization Algorithm:* Adam and SGD optimizers for training stability and convergence.
3) *Training Steps:*

- Input image augmentation to increase dataset variability.
- Feed-forward propagation through CNN layers.
- Loss calculation and backpropagation for weight adjustments.
- Model evaluation on validation dataset.

Hyperparameters:

- Epochs: 30–100
- Batch size: 32
- Learning rate: 0.001 (Adam), 0.01 (SGD)

The training process was conducted using TensorFlow and Keras frameworks, leveraging GPU acceleration for faster computations. The model was trained using a stratified k-fold cross-validation approach to mitigate overfitting and ensure robust performance across different subsets of data.

CNN architecture was chosen as the main model in this study because of its proven effectiveness in image classification tasks. CNNs can automatically learn and extract hierarchical features, ranging from simple edges to complex crack patterns on tires, making them highly suitable for this detection task [12], [27]. To address the challenge of limited datasets, this study adopts a transfer learning approach. This technique enables the model to leverage knowledge learned from large-scale datasets, thereby improving model performance and generalization capabilities even with limited training data. CNN architecture shown in Fig. 6.

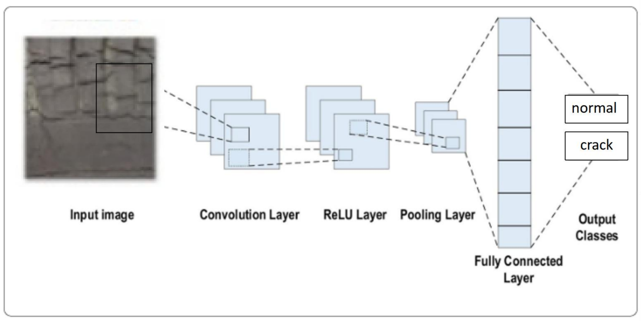

Fig. 6. CNN architecture.

IV. Result

A. Evaluation Model

Table 2 presents the model's performance across different optimizers and epochs. The evaluation focuses on test accuracy and test loss, which are critical indicators of model effectiveness in distinguishing between cracked and normal tires.

Table 2.
The Adam Optimizer

Optimizer	Epochs	Test Accuracy (%)	Test Loss (%)
Adam	30	75.8	57
Adam	70	77.5	54
Adam	100	78.3	53
SGD	30	74.2	70
SGD	70	75.0	60
SGD	100	75.8	55

The Adam optimizer consistently outperforms SGD, demonstrating superior test accuracy and lower test loss across all epoch configurations. The highest test accuracy of 78.3% was achieved at 100 epochs with Adam, whereas the SGD optimizer peaked at 75.8% accuracy under the same epoch setting but with a higher test loss of 55%.

Adam combines the advantages of two other optimizers: Momentum and RMSProp. This optimizer can adjust the learning rate for each parameter adaptively, its advantages are Fast convergence is effective on large & complex datasets is robust to noise (e.g. on blurred prohibition images).

Further performance breakdown in Table 3 includes validation loss and validation accuracy, which provide deeper insights into generalization capabilities.

Table 3.
Detailed Performance Evaluation with Adam and SGD Optimizer

Epoch	Test Loss	Test Accuracy	Val Loss	Val Accuracy	Optimizer
30	0.570	0.758	0.6095	0.7200	Adam 1
70	0.540	0.775	0.5800	0.7400	Adam 2
100	0.532	0.783	0.6349	0.7200	Adam3
30	0.700	0.742	0.7200	0.7000	SGD 4
70	0.600	0.750	0.6800	0.7100	SGD 5
100	0.553	0.758	0.5700	0.8000	SGD 6

B. Impact of Epochs on Model Performance

The model shows a steady increase in accuracy from 30 to 70 epochs for both Adam and SGD optimizer, confirming that training over more epochs enhances feature extraction and pattern learning. However, after 70 epochs, validation accuracy begins to fluctuate, with a slight decline in Adam's validation accuracy at 100 epochs (from 74.0% to 72.0%) despite test accuracy improving. This is a strong indicator of overfitting, where the model starts memorizing training data rather than generalizing well to unseen data. For SGD, validation accuracy improves at 100 epochs (80.0%), suggesting that while SGD converges slower than Adam, it may retain better generalization at later stages of training.

C. Optimizer Efficiency: Adam vs. SGD

Adam converges faster and stabilizes at a lower test loss (53%) compared to SGD (55%), proving its effectiveness in optimizing complex CNN architectures. The SGD optimizer shows higher variance in loss values, requiring more epochs to reach stability. This aligns with previous studies in deep learning, where SGD tends to struggle in complex landscapes with high-dimensional features, whereas Adam adapts learning rates dynamically, making it more efficient for this task. The fluctuations in SGD’s accuracy trend indicate a potential sensitivity to hyperparameter tuning (such as learning rate adjustments), which could be an area for further exploration in future work.

D. Loss Trends and Overfitting

The validation loss for Adam increases at 100 epochs, signaling potential overfitting, where the model learns noise in training data instead of meaningful patterns. The SGD optimizer exhibits a gradual reduction in validation loss at 100 epochs, suggesting that it may require more training iterations to reach optimal performance. This overfitting pattern supports the common deep learning trade-off: higher accuracy comes at the cost of generalization. It also highlights that, while Adam achieves faster convergence, prolonged training without proper regularization (e.g., dropout, weight decay) may degrade

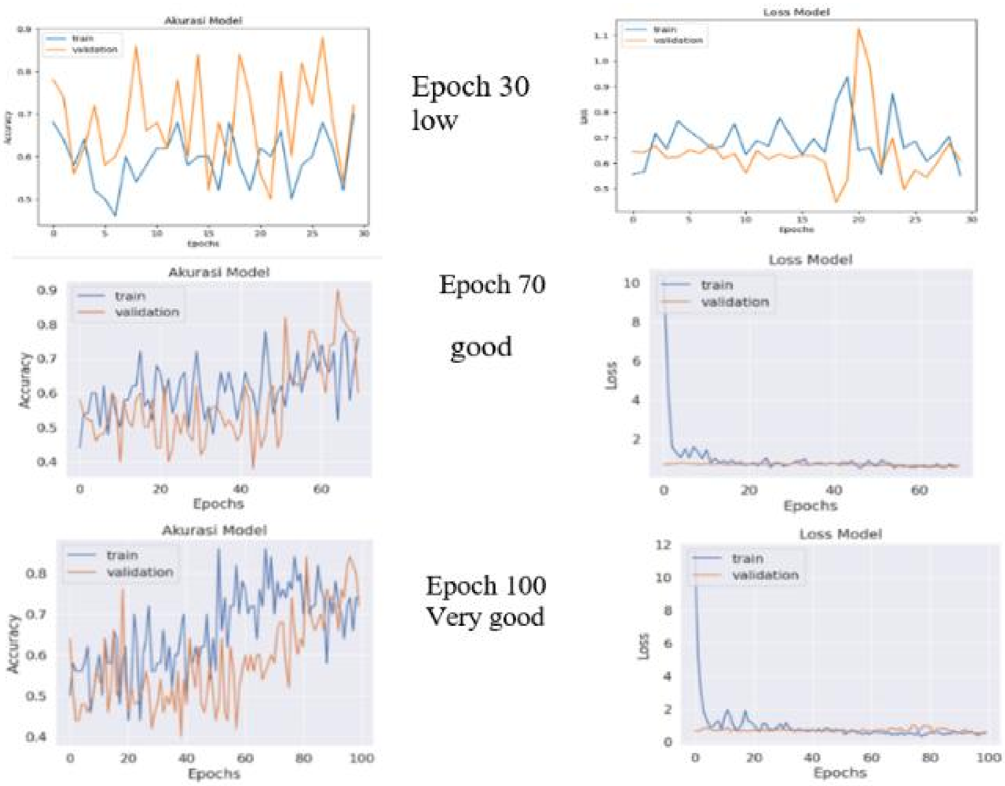

Fig. 7. Graphs with ADAM & SGD optimizer.

generalization ability. *E. Visualizing Model Trends*

Figure 7 illustrates the accuracy and loss trends for different epoch settings. The results confirm that increasing the number of epochs improves accuracy up to a certain point. However, after 70 epochs, the model starts to show signs of overfitting, particularly in the validation loss, which begins to rise slightly. The Adam optimizer shows faster convergence and better stability than SGD, particularly in reducing test loss.

On the other hand, SGD requires more epochs to reach comparable performance and exhibits higher variability in accuracy and loss trends. The colors in the test accuracy and loss charts distinguish between the accuracy curve and the loss curve, following conventional data visualization practices. The classification performance is evaluated using categorical cross-entropy as the loss function, which is effective for 2-class image classification with one-hot encoding. This method offers good numerical stability and ease of implementation. Additionally, the softmax function at the CNN model's output layer generates class probabilities, providing an interpretable probability distribution for tire classification.

F. Discussion

This study examines the trade-off between convergence speed and generalization capability in tire crack detection using Adam and SGD optimizers. Adam provides faster convergence and a peak test accuracy of 78.3% but shows signs of overfitting after 70 epochs, reducing its ability to generalize. In contrast, SGD, though slower, achieves better generalization, reaching a validation accuracy of 80% at 100 epochs. The findings highlight that optimizer choice significantly affects model behavior, complementing previous studies that emphasize architectural complexity. Adam's rapid learning is useful for prototype development, but SGD is preferable for real-world deployment due to its stability and generalization. The study has limitations, including a small dataset (600 images from Kaggle) and the use of a single CNN architecture. Future research should incorporate stronger regularization, explore advanced architectures like ResNet, and expand the dataset for multi-class classification. The ultimate goal is integrating the model into edge computing for real-time industrial detection.

V. CONCLUSION

This study highlights the advantages and challenges of using Adam and SGD optimizers in tire crack detection. Adam's adaptive learning rate and momentum mechanisms enable faster convergence, making it effective for complex deep learning models like CNN. However, after 70 epochs, Adam exhibits signs of overfitting, as indicated by increasing validation loss. SGD, although slower, demonstrates superior generalization, achieving a validation accuracy of 80% at 100 epochs. This suggests that while Adam is beneficial for rapid learning, SGD offers a more stable and robust solution for long-term training, especially in imbalanced datasets. Transfer learning and data augmentation significantly enhance model performance, enabling better feature extraction and mitigating overfitting. Despite these improvements, Adam still requires stronger regularization techniques, such as Dropout and L2 regularization. Future research should focus on advanced architectures like ResNet and EfficientNet, integrate real-time detection via edge computing, and refine optimization strategies to improve model efficiency and stability for practical applications.

Acknowledgment

Thank you to Mr. Dr. Ahmad Fauzi., M.Kom as the Dean of the Faculty of Computer Science, Buana Perjuangan University, Karawang, which has helped in terms of creative and innovative thoughts and ideas.

References

[1] J. Y. Wong, *Theory of Ground Vehicles*. John Wiley & Sons, 2022.

[2] D. Yang, J. Li, C. Huang, K. Li, G. Lu, and K. Guo, "A review of research on tire burst and vehicle stability control," *Science Progress*, vol. 107, no. 3, 2024, doi:10.1177/00368504241272478.

[3] A. Zuska and J. Jackowski, "Influence of changes in stiffness and damping of tyre wheels on the outcome of the condition assessment of motor vehicle shock absorbers," *Energies*, vol. 16, no. 9, Art. no. 3876, 2023, doi: 10.3390/en16093876.

[4] S. Ali, S. A. Shah, M. Ahmad, S. M. Zain, and A. R. Khan, "Design, analysis and comparison of hybrid and Non-Pneumatic tires," Research Square (Research Square), Jun. 2024, doi: 10.21203/rs.3.rs-4418526/v1.

[5] E. E.-D. Hemdan and M. E. Al-Atroush, "A review study of intelligent road crack detection: Algorithms and systems," *International Journal of Pavement Research and Technology*, pp. 1–31, 2025, doi: 10.1007/s42947-025-00556-x.

[6] M. J. Hasan *et al*., "GroundingCarDD: Text-guided multimodal phrase grounding for car damage detection," *IEEE Access*, vol. 12, pp. 179464-179477, 2024, doi: 10.1109/ACCESS.2024.3506563.

[7] P. Ghag, "Automatic tire inflation system," *International Research Journal of Engineering and Technology (IRJET)*, vol. 9, no. 2, pp. 107–111, 2024.

[8] H. C. Mayana and D. Leni, "Deteksi kerusakan ban menggunakan CNN dengan arsitektur resnet-34," *Jurnal Surya Teknika*, vol. 10, no. 2, pp. 842–851, 2023, doi: 10.37859/jst.v10i2.6336.

[9] N. U. A. Tahir, Z. Zhang, M. Asim, J. Chen, and M. ELAffendi, "Object detection in autonomous vehicles under adverse weather: A review of traditional and deep learning approaches," *Algorithms*, vol. 17, no. 3, Art. no. 103, 2024, doi: 10.3390/a17030103.

[10] M. Sadaf *et al*., "Connected and automated vehicles: Infrastructure, applications, security, critical challenges, and future aspects," *Technologies*, vol. 11, no. 5, Art. no. 117, 2023, doi: 10.3390/technologies11050117.

[11] A. Giannaros *et al*., "Autonomous vehicles: Sophisticated attacks, safety issues, challenges, open topics, blockchain, and future directions," *Journal of Cybersecurity and Privacy*, vol. 3, no. 3, pp. 493–543, 2023, doi: 10.3390/jcp3030025.

[12] R. A. A. Saleh and H. M. Ertunç, "Explainable attention-based fused convolutional neural network (XAFCNN) for tire defect detection: An industrial case study," *Eng. Res. Express*, vol. 6, no. 1, Art. no. 015090, 2024, doi: 10.1088/2631-8695/ad23c8.

[13] R. A. A. Saleh and H. M. Ertunç, "Attention-based deep learning for tire defect detection: Fusing local and global features in an industrial case study," *Expert System With Application*, vol. 269, Art. no. 126473, 2025, doi: 10.1016/j.eswa.2025.126473.

[14] R. A. A. Saleh, M. Z. Konyar, K. Kaplan, and H. M. Ertunç, "End-to-end tire defect detection model based on transfer learning techniques," *Neural Computing and Applications*, vol. 36, no. 20, pp. 12483–12503, Apr. 2024, doi: 10.1007/s00521-024-09664-4.

[15] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural Networks, vol. 61, pp. 85–117, Oct. 2014, doi: 10.1016/j.neunet.2014.09.003.

[16] Zizoudinosaur, "Tire_Status_Classification," Kaggle, May 23, 2022. https://www.kaggle.com/code/zizoudinosaur/tire-status-classification/input

[17] F. Demasi, G. Loprencipe, and L. Moretti, "Road safety analysis of urban roads: case study of an Italian municipality," *Safety*, vol. 4, no. 4, Art. no. 58, Dec. 2018, doi: 10.3390/safety4040058.

[18] Z. Fang, Y. Wang, L. Peng, and H. Hong, "Integration of convolutional neural network and conventional machine learning classifiers for landslide susceptibility mapping," *Computers & Geosciences*, vol. 139, Art. no. 104470, Mar. 2020, doi: 10.1016/j.cageo.2020.104470.

[19] R. Liu, A. Kothuru, and S. Zhang, "Calibration-based tool condition monitoring for repetitive machining operations," *Journal of Manufacturing Systems*, vol. 54, pp. 285–293, Jan. 2020, doi: 10.1016/j.jmsy.2020.01.005.

[20] N. Kruger et al., "Deep hierarchies in the primate visual cortex: What can we learn for computer vision?," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1847–1871, Jun. 2013, doi: 10.1109/tpami.2012.272.

[21] J. P. Rodríguez *et al*., "Big data analyses reveal patterns and drivers of the movements of southern elephant seals," *Scientific Reports*, vol. 7, no. 1, Mar. 2017, doi: 10.1038/s41598-017-00165-0.

[22] Z. Wu, C. Jiao, J. Sun, and L. Chen, "Tire defect detection based on faster R-CNN," in *Communications in Computer and Information Science*, 2020, pp. 203–218. doi: 10.1007/978-981-33-4932-2_14.

[23] S.-L. Lin, "Research on tire crack detection using image deep learning method," *Sci. Rep.,* vol. 13, Art. no. 8027, 2023, doi: 10.1038/s41598-023-35227-z.

[24] Y. Wang and W. Wang, "Generative Adversarial Network-Based Data Augmentation for Tyre Surface Defect Detection," *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, Auckland, New Zealand, 2023, pp. 1–6, doi: 10.1109/CASE56687.2023.10260675.

[25] Y. Sun *et al*., "Automatic pixel-level detection of tire defects based on a lightweight Transformer architecture," *Measurement Science and Technology*, vol. 34, no. 8, Art. no. 085405, May 2023, doi: 10.1088/1361-6501/acd5f2.

[26] A. N. A. Thohari and G. B. Hertantyo, "Implementasi Convolutional Neural Network untuk Klasifikasi Pembalap MotoGP Berbasis GPU," *CENTIVE*, vol. 1, no. 1, pp. 50–55, Apr. 2019.

[27] I. Kuric, J. Klarák, M. Sága, M. Císar, A. Hajdučík, and D. Wiecek, "Analysis of the possibilities of Tire-Defect inspection based on unsupervised learning and deep learning," *Sensors*, vol. 21, no. 21, Art. no. 7073, Oct. 2021, doi: 10.3390/s21217073.