Proceedings of the 3rd International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/2025.24721

The Application of Deep Learning Algorithms in Traffic Sign Recognition

Chubin Mai

University of Macau, Macau, China
MC46438@um.edu.mo

Abstract. Traffic sign recognition is a critical component in the field of autonomous driving. In practice, recognising a wide range of different symbol classes with very high accuracy, robust performance, and rapid processing speed is essential. Traffic signs are designed for human readability, however, for computer systems, classifying traffic signs remains a complex pattern recognition problem. Image processing and machine learning algorithms are continually improving to improve this capability. Among them, deep-reinforcement learning (DRL) has become a cutting-edge technology that excels in feature extraction and offers practical solutions for various object recognition challenges. This article has four major contributions: firstly, we integrate CSO and AGA in traffic sign recognition, achieving highly favourable results. Secondly, we refine parameters, reducing training time while improving accuracy from 96.44% to 97.97%. Thirdly, we utilise multiple publicly available traffic sign datasets from various countries and regions to train our model, which achieves high accuracy and demonstrates strong robustness. Finally, we combines CSO for data filtering and training with AGA algorithm to improve both accuracy and efficiency. Overall, our analysis underscores the effectiveness of our proposed technology, highlights its potential in achieving precise traffic sign recognition, and positions it as a viable solution for real-time applications in autonomous systems.

Keywords: DRL, AGA, CSO, Traffic sign recognition, Object Classification.

1. Introduction

In recent years, the rapid development of autonomous driving technology has increased the demand for powerful Traffic Sign Recognition systems. As a core component of Advanced Driver Assistance Systems (ADAS), traffic sign recognition ensures road safety by providing real-time guidance for both drivers and autonomous vehicles. This paper addresses these demands from three critical dimensions: accuracy, robustness, and computational efficiency.

Firstly, we focus on accuracy, which determines whether we can accurately identify various types of signs and obtain the information contained therein. This is crucial for reducing misjudgements and preventing traffic accidents. In the process of feature selection and model training, excessive computational resources are often easily consumed, leading to a significant increase in computational costs associated with achieving high accuracy [1-4].

Furthermore, we focus on robustness, as ensuring reliable recognition capability is essential for traffic sign recognition in various complex real-world environments. Traditional computer vision methods, such as template matching and support vector machines, rely on manual feature extraction and have limited adaptability to complex scenes, often resulting in poor robustness and lower accuracy in complex environments [5-8].

Additionally, we also emphasise computational efficiency. While Deep Reinforcement Learning (DRL) performs well in object classification tasks, it may encounter gradient vanishing problems in deep networks, impacting training effectiveness. Furthermore, DRL may suffer from suboptimal training performance due to insufficient dataset diversity.

In the past, traditional DRL algorithms have conducted extensive research on similar problems and achieved certain results. However, they are also face various challenges. To address these challenges, we explored multiple approaches and concluded that CSO and AGA form an effective combination.

CSO is a group intelligence based optimisation algorithm that updates the position and velocity of particles through a competition mechanism. Its advantages include powerful global search capabilities and high convergence speeds, effectively avoiding local optimal solutions. Compared with conventional particle swarm optimisation (PSO), CSO is well-suited for feature selection and optimisation in high dimensional data, demonstrating excellent performance in large-scale optimisation problems.

AGA dynamically adjusts crossover and mutation rates to enhance algorithm adaptability and search efficiency. Its strong global search function and high convergence accuracy enable the discovery of optimal solutions in complex search spaces. Compared with traditional genetic algorithms, AGA excels in pattern recognition and optimisation design, achieving impressive performance in multimodal optimisation problems.

Conventional DRL, compared to CSO and AGA, offers great advantages in processing high-dimensional data and complex optimisation problems. However, DRL algorithms usually require large amounts of training data and computational resources. However, CSO and AGA leverage group intelligence and evolutionary mechanisms to explore solution spaces more efficiently and identify optimal global solutions [9-11].

AGA and CSO combine their separate advantages, enabling improvements in the performance of the optimisation algorithm. AGA provides robust global search capabilities and high convergence accuracy, while CSO improves search efficiency through conflict mechanisms, effectively avoiding local optimality. This synergy makes them highly suitable for feature selection, pattern recognition, and automatic operation, demonstrating exceptional performance in complex optimisation problems.

We summarise our contributions as follows:

1. To the best of our knowledge, we are the first to combine CSO and AGA on traffic sign recognition. We applied CSO for feature filtering first, followed by AGA to train the model.

2. By adjusting training parameters such as epochs and batch sizes (based on previous methodologies for signage recognition), we achieved an accuracy rate of 97.973%.

3. We leveraged the advantages of residual networks, ensuring reliable performance by replacing the multivariate training set, which demonstrated strong robustness.

4. We reduced computational time by approximately 20% through optimising training parameters such as filters.

2. Related work

There have been numerous studies on traffic signs using traditional DRL algorithms, which have achieved significant results. However, they still face various problems such as robustness, accuracy, and computational efficiency [6-9,12].

1. CNN-based traffic sign recognition systems exhibit limited robustness under varying lighting conditions, resulting in reduced accuracy [6].

2. Research has explored the use of Support Vector Machines (SVM) and image segmentation for traffic sign classification, yet it does not address the computational inefficiencies and high resource demands of CNNs, which pose a significant drawback in real-time applications [7].

3. Some studies highlight that the accuracy in traffic sign recognition within complex environments is affected by insufficient dataset and the low adaptability to variations [8].

4. Hierarchical research on CNN-based traffic sign detection is highly focused. However, it fails to mitigate the vanishing gradient problem, preventing the development of deeper networks [9].

5. Additionally, conventional algorithms such as SVM and KNN have demonstrated limitations in processing large datasets, and they do not fully address the computational complexity and gradient vanishing problems that DRL still encounters [12].

In some areas, AGA outperforms CNN. For instance, AGA has shown superior performance in malware detection, especially for new malware, compared to CNNs. AGA also performs better than CNNs in multimodal optimisation problems. Moreover, the combination of AGA and CSO leverages the strengths of both techniques to enhance optimisation algorithm performance. The automatic feature selection framework combining CSO and AGA significantly improves the efficiency and accuracy of feature selection. This combination achieves superior performance in complex optimisation problems, making it well-suited for feature selection, pattern recognition, and autonomous driving [1-4,10,11,13-18].

1. A study proposed an automatic feature selection framework integrating CSO and AGA, significantly enhancing the efficiency and accuracy of feature selection [2].

2.A study proposed a new hybrid Ant Colony Optimisation (ACO) and genetic algorithms (GA) method for feature selection, demonstrating the advantages of combining AGA and CSO [3].

3. A study conducted a comprehensive survey of deep learning techniques for autonomous driving, highlighting the advantages and limitations of various methods (including many algorithms) in terms of efficiency and real-time performance [8].

4. One study found that AGA outperformed CNNs in malware detection, particularly when identifying new malware [10].

5. Some studies discuss the advantages and disadvantages of GA, noting that AGA is superior to CNNs in multimodal optimisation problems [11].

6. A study reviewed evolutionary computation methods for feature selection, demonstrating their ability to significantly improve efficiency and accuracy in high-dimensional data processing [15].

7. A study examined feature selection methods for high-dimensional data and demonstrated that combining different algorithms can enhance optimisation algorithm performance [16].

8. A study explored deep learning-based traffic sign detection and recognition for autonomous vehicles, showing AGA and CSO can improve the accuracy and efficiency of these systems [17].

9. A study of traffic sign recognition highlighted the advantages of AGA and CSO [18].

3. Methodology

3.1. Overview

Our method involves using CSO for feature screening to select the most suitable features, followed by training with the AGA based on these selections. Figure 1 illustrates our framework, outlining the main steps of our approach. Briefly, we pre-processed the dataset, applied CSO for feature selection, and then trained the model using AGA to obtain the final result.

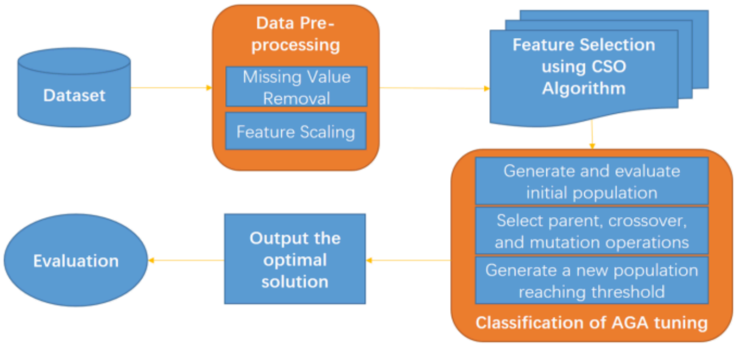


Figure 1: The framework of CSO and AGA

3.2. Implementation details of CSO

In the third step of the framework, we used CSO for feature selection. CSO is a swarm intelligence-based optimisation algorithm widely used in feature selection tasks. In the field of signboard recognition, the CSO algorithm iteratively optimises the feature set to determine the optimal feature subset. Specifically, CSO first receives all extracted features from the signboard image as input and evaluates the classification performance of the feature subset using an objective function. Through competitive strategies and swarm intelligence, the algorithm continuously refines the feature selection process, ultimately outputting a subset of features that performs best according to the objective function.

PSO is a population-based optimisation algorithm that guides particle searches to obtain the global optimal solution by simulating interactions between birds and fish in nature. The PSO algorithm begins with a set of randomly generated particles (algebra), and updates them to converge on the best solution. Each particle is associated with a position and velocity vector, corresponding to the dimensions of the search space. In each iteration, every particle updates two optimal values: one is the best fitness value within a group, and the other is the maximum fitness solution [10].

The algorithm expression for PSO is:

$$v_{itd}+1 \;=\; v_{ttd}+c_1\cdot rand(0,\ 1)\cdot(p_{ttd}-x_{ttd})+c_2\cdot rand(0,\ 1)\cdot(p_{tgd}-x_{ttd})$$

$$x_{idt}+1 \;=\; x_{tid} \;+\; v_{idt}+1$$

Here, each variable represents the position and velocity of the particle, the index of the particle, the number of iterations, the dimension of the search space, the speed that controls the flight distance, and a random value from 0 to 1.

DOI: 10.54254/2755-2721/2025.24721

CSO and PSO have been developed to enhance global search functionality and improve convergence speed. CSO updates the position and speed of particles through a competition mechanism, preventing them from becoming trapped in a local optimum state. The primary objective is to improve the efficiency and accuracy of feature selection by identifying the best solution via the competitive mechanism. Compared with CSO, PSO offers advantages in processing high-dimensional data and tackling complex optimisation problems.

The mathematical formula is as follows: Let the particle position be x_i , the velocity be v_i , and the fitness value be x_i . In each iteration, particles update their positions and velocities according to the competition mechanism, as follows:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (p_i - x_i(t)) + c_2 \cdot r_2 \cdot (g_i - x_i(t)) \qquad x_i(t+1) = x_i(t) + v_i(t+1)$$

Where:

w is the inertia weight; c_1 and c_2 are acceleration coefficients; r_1 and r_2 are random numbers; p_i is the individual optimal position of the particle; g_i is the overall optimal position.

3.3. Implementation details of AGA

The fourth step of the framework involves training the data. AGA is an improved genetic algorithm that dynamically adjusts crossover and mutation probabilities to improve search efficiency. When training a character recognition model, the AGA algorithm first receives an annotated character image along with its corresponding category label as input and generates an initial population of randomly generated multiple model parameter individuals. Fitness functions are then used to evaluate the classification performance of each individual. Through iterative optimisation, AGA continuously refines crossover and mutation probabilities, ultimately identifying the optimal model parameters. Finally, the algorithm outputs a model parameters that achieve the highest fitness function performance, adjusting fitness values throughout the training process to refine the optimal individual parameters.

Before introducing AGA, it is useful to first provide an overview of its foundational model: Genetic Algorithm (GA). GA is an efficient optimisation tool that uses computational models based on natural selection principles to solve problems. It finds optimal solutions through crossover and mutation processes, mimicking genetic recombination and mutation in nature to create new solutions distinct from existing ones, allowing them to evolve within an environment. Over time, GA-driven responses converge towards optimal solutions. One of its key advantages is its ability to select the best combination and explore large search spaces efficiently.

AGA improves upon GA by weighting crossover and mutation probabilities based on fitness values to yield superior results. The AGA process includes encoding, fitness calculation, selection, reproduction, crossover, mutation, and decoding. To enhance the effectiveness of GA, AGA introduces more broadly applicable adaptive crossover and mutation probabilities [10,11].

The specific dynamic adjustment formulas for crossover rate and mutation rate are as follows: Let the individual’s gene representation be Gi and fitness value be Gi . In each iteration, crossover and mutation operations are dynamically adjusted based on fitness values.

$$Gi(t+1) = crossover(Gi(t), Gj(t)) \qquad Gi(t+1) = mutation(Gi(t))$$

where crossover and mutation rates are dynamically adjusted based on fitness values, with the formulas:

DOI: 10.54254/2755-2721/2025.24721

$$Pc = Pc0 \cdot \left(1 - \frac{f(Gi)}{f_{max}}\right) \qquad Pm = Pm0 \cdot \left(1 - \frac{f(Gi)}{f_{max}}\right)$$

where $Pc0$ and $Pm0$ are initial crossover and mutation rates, and $fmax$ is the maximum fitness value.

AGA optimises crossover and mutation processes through an adaptive mechanism, significantly enhancing search efficiency and convergence accuracy. AGA demonstrates excellent adaptability and robustness in complex optimisation problems, including feature selection, pattern recognition, and automated operations. By dynamically adjusting the rates of intersection and mutation, the solution space can be searched more effectively, facilitating the identification of the global optimal solution, and improving the overall performance of the algorithm.

3.4. Benefits and advantages of combining CSO and AGA

By combining the advantages of CSO and AGA, optimisation algorithm performance can be significantly enhanced. CSO selects the optimal feature subset through a competition mechanism, improving the efficiency and accuracy of feature selection. Meanwhile, AGA refines crossover and mutation operations through adaptive mechanisms, boosting search efficiency and convergence accuracy, ultimately improving the model's accuracy and robustness.

This approach is particularly well-suited for feature selection, pattern recognition, and automatic operation, as it delivers exceptional performance in complex optimisation problems. Compared with conventional image recognition algorithms such as Vector Space Model (VSM), Conventional Neural Networks (CNNs) and traditional Deep Reinforcement Learning (DRL), CSO and AGA offer distinct advantages in analysing high-dimensional data and solving complex optimisation problems. Their ability to efficiently explore the solution space and identify global optimal solutions also contributes to improvements in accuracy, robustness, and computational efficiency.

4. Experimental study

4.1. Experimental setup

We used Python language to complete this task. We ran the task on Google Colab with a T4 GPU and an A100 GPU with 40 or 80 memories [19].

4.1.1. Dataset

The dataset used as the basis for this article is the GTSRB dataset [20][21]. This dataset has been recognised for its strong performance in previous studies and consists of 51,839 images. It is divided into 75% training data and 25% test data, all of which are real-world images.

4.1.2. Algorithm

We implemented the CSO-AGA algorithm in our experiment.

4.1.3. Factors

We varied the three parameters: (1) epochs, (2) batch sizes and (3) filters to optimise the baseline.

Epochs refer to the number of times the entire training dataset is processed. More epochs usually mean that the model has additional opportunities to learn and adjust its parameters.

DOI: 10.54254/2755-2721/2025.24721

Batch sizes determine the number of training samples used in each iteration. Larger batch sizes can accelerate training, but require more memory.

Filters represent the number of convolutional filters in a neural network, which are used to extract features from images. A higher number of filters can capture more details, but this also increases computational complexity.

Measurement

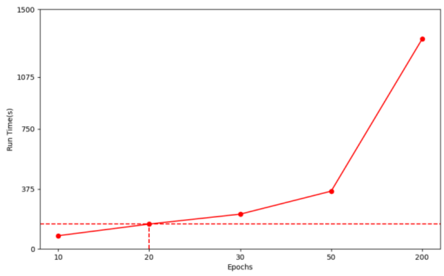
We use (1) accuracy and (2) runtime as evaluation criteria. Accuracy is measured in percentages, and time is measured in seconds.

The accuracy of the model is given by the formula:

$$\text{Accuracy} = \frac{\textit{Number of Correct Predictions}}{\textit{Total Number of Predictions}}$$

4.2. Experimental results

Epochs: The number of epochs we tried was {10, 20, 30, 50, 200}, with the bold value representing our initial setting. Accuracy improved to some extent following both reductions and increases in epochs. The most significant improvement occurred at 30 epochs, achieving an accuracy of 97.74%, which marks a 1.31% increase compared to the baseline. In accordance with common sense, the running time increases with the number of epochs.



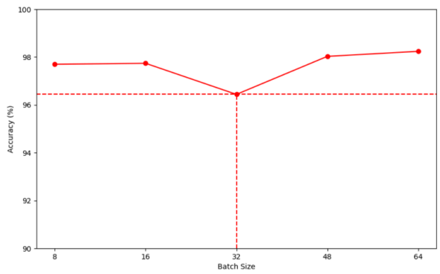
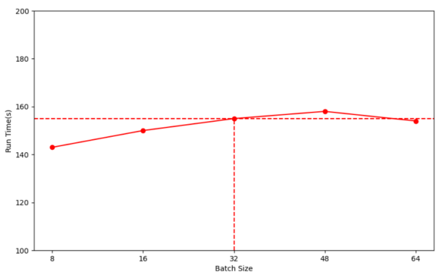


Figure 2: Accuracy of epochs Figure 3: Accuracy of epochs

Batch size: After reducing and increasing the number of batches {8, 16, 32, 48, 64}, the initial basic accuracy with 32 batches was 96.43% and the accuracy improved to a certain extent. The most significant improvement occurred at 64 batches, achieving an accuracy of 98.24%, which increased by 1.81% compared to the baseline. The running time increases slightly with the increase of batch size, but declines at 64 batches. The correlation is not significant.



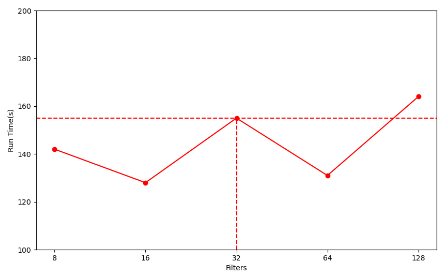


Figure 4: Accuracy of batch sizes Figure 5: Run time of batch sizes

DOI: 10.54254/2755-2721/2025.24721

Filters: The initial baseline accuracy with 32 filters was 96.43%. After testing different filter configurations, {8, 16, 64, 128}, the accuracy improved to some extent. The most significant improvement occurred at 64 filters, achieving 97.59% accuracy – an increase of 1.16%. The running time fluctuates with the change of filters, and the correlation is not significant.

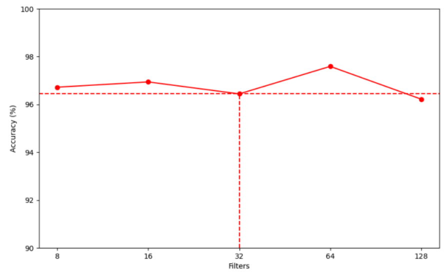


Figure 6: Accuracy of filters

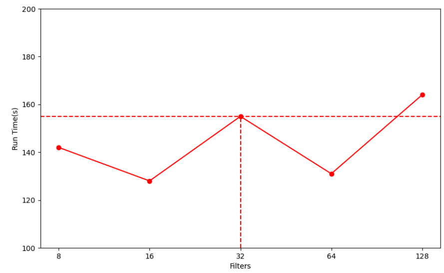


Figure 7: Run time of filters

Replacing Dataset: We replaced some traffic sign images from different countries into the original dataset. Following this replacement, the model maintained strong accuracy, demonstrating excellent robustness [21].

By adjusting training parameters such as epochs, batch sizes, and filters, we achieved a high accuracy rate of 97.973% and reduced computation time by approximately 20%.

Figure 8: Example

5. Conclusion

In this study, we have demonstrated the effectiveness of combining CSO and AGA to optimise traffic sign recognition systems. By adjusting training parameters such as epochs, batch sizes, and filters, we achieved a high accuracy rate ranging from 96.215% to 97.973%, while reducing computation time by approximately 20%. Our approach ensures robust performance under various environmental conditions and efficiently handles high-dimensional data. The integration of CSO and AGA provides a powerful solution for feature selection and model training, making it well-suited for advanced driver assistance systems and autonomous driving applications.

In future work, we aim to employ improved algorithms to enhance the three main evaluation indicators: efficiency, robustness, and computational efficiency. Further advancements in these areas will provide greater benefits for traffic sign recognition. Additionally, we plan to extend this technology to other image recognition domains, such as infrared visible light fusion and medical image recognition.

References

[1] Grigorescu, S., Trasnea, B., Cocias, T., & Macesanu, G. (2020). A survey of deep learning techniques for autonomous driving. Journal of Field Robotics, 37(3), 362-386.
[2] Wang, Y., Zhao, X., Xu, T., & Wu, X. (2022). Autofield: Automating feature selection in deep recommender systems. Proceedings of the ACM Web Conference 2022, 1977-1986.
[3] Kabir, M., Shahjahan, M., & Murase, K. (2012). A new hybrid ant colony optimization algorithm for feature selection. Expert Systems with Applications, 39(3), 3747-3763.
[4] Lee, J., Abdulhai, B., Shalaby, A., & Chung, E. H. (2005). Real-Time Optimization for Adaptive Traffic Signal Control Using Genetic Algorithms. Journal of Intelligent Transportation Systems, 9(3), 111–122.
[5] Bachute, M. R., & Subhedar, J. M. (2021). Autonomous driving architectures: insights of machine learning and deep learning algorithms. Machine Learning with Applications, 6(8), 100164.
[6] Triki, N., Karray, M., & Ksantini, M. (2024). A Comprehensive Survey and Analysis of Traffic Sign Recognition Systems With Hardware Implementation. IEEE Access, 12, 144069-144081.
[7] Chanklan, R., Chaiyakhan, K., Hirunyawanakul, A., Kerdprasop, K., & Kerdprasop, N. (2015). Fingerprint Recognition with Edge Detection and Dimensionality Reduction Techniques. Proceedings of the 3rd International Conference on Industrial Application Engineering, Japan.
[8] Rao, A., Motwani, R., Sarguroh, N., Kingrani, P., & Khandaskar, S. (2021). Real-time traffic sign recognition using convolutional neural network. Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2020, 2, 223-234.
[9] Mao, X., Hijazi, S.L., Casas, R.A., Kaul, P., Kumar, R., & Rowen, C. (2016). Hierarchical CNN for traffic sign recognition. 2016 IEEE Intelligent Vehicles Symposium (IV), 130-135.
[10] Hammood, L., Doğru, İ. A., & Kılıç, K. (2023). Machine Learning-Based Adaptive Genetic Algorithm for Android Malware Detection in Auto-Driving Vehicles. Applied Sciences, 13(9), 5403.
[11] Vié, A., Kleinnijenhuis, A. M., & Farmer, D. J. (2020). Qualities, Challenges and Future of Genetic Algorithms: A Literature Review. arXiv preprint arXiv: 2011.05277.
[12] Chen, Y., Xie, Y., & Wang, Y. (2013). Detection and Recognition of Traffic Signs Based on HSV Vision Model and Shape features. Journal of Computers, 8(5), 1366-1370.
[13] Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2012). Man vs. computer: benchmarking machine learning algorithms for traffic sign recognition. Neural networks : the official journal of the International Neural Network Society, 32, 323–332.
[14] Durlik, I., Miller, T., Kostecka, E., Zwierzewicz, Z., & Łobodzińska, A. (2024). Cybersecurity in Autonomous Vehicles—Are We Ready for the Challenge? Electronics, 13(13), 2654.
[15] Xue, B., Zhang, M., Browne, W., & Yao, X. (2015). A Survey on Evolutionary Computation Approaches to Feature Selection. IEEE Transactions on Evolutionary Computation, 20(4), 606-626.
[16] Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso- Betanzos, A. (2016). Feature selection for high-dimensional data. Progress in Artificial Intelligence, 5, 65-75.
[17] Mani, M. K., Rajagopal, S., Kavitha, D., & Ramachandran, S. (2024). Deep Learning-Based Traffic Sign Detection and Recognition for Autonomous Vehicles. Digital Twin and Blockchain for Smart Cities, 407-428.
[18] Qin, Z., & Yan, W.Q. (2021). Traffic-Sign Recognition Using Deep Learning. Geometry and Vision: First International Symposium, ISGV 2021, Auckland, New Zealand, January 28-29, 2021, Revised Selected Papers 1, 13-25.
[19] Google. (2025). Google Colab. https: //colab.research.google.com. Accessed: 2025-04-17.
[20] Meow Meow Meow Meow Meow. (2025). GTSRB German Traffic Sign Dataset. https: //www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign, Accessed: 2025-04-17.
[21] Dreamstime. (2025). Traffic signs. https: //www.dreamstime.com. Accessed: 2025-04-17.