Article

A Vehicle Classification Method Based on Machine Learning

Xinjin Li [*] , Jinghao Chang , Tiexin Li , Wenhan Fan , Yu Ma , Haowei Ni

Article

A Vehicle Classification Method Based on Machine Learning

Xinjin Li [1,*], Jinghao Chang [2], Tiexin Li [3], Wenhan Fan [4], Yu Ma [5] and Haowei Ni [1]

[1]   Columbia University, New York, NY, USA; hn2339@caa.columbia.edu
[2]   The Kyoto College of Graduate Studies for Informatics Kyoto, Japan; changjinghao.communication@gmail.com
[3]   Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu, China; Tiexin.Li21@student.xjtlu.edu.cn
[4]   Independent Researcher, New York, NY, USA; finncontactplus@gmail.com
[5]   Carnegie Mellon University, Pittsburgh, PA, USA; yuma13926@gmail.com
[*]   Correspondence: li.xinjin@columbia.edu

Abstract: In this article, we propose a machine learning based fine-grained vehicle classification method Vehi-ClassNet, which effectively solves the challenges of precise vehicle recognition and classification in intelligent transportation systems by combining multi-scale feature extraction, cross modal fusion, and attention mechanisms. Our model demonstrated excellent performance on the Stanford Cars-196 dataset. The innovation of this study lies in its advanced multi-scale feature extraction, cross modal fusion, and attention mechanism. It not only improves classification accuracy, but also lays the foundation for further research on intelligent transportation systems and autonomous vehicle technology.

Keywords: deep reinforcement learning; speed planning; reward function; fine grained classification; intelligent transportation system

---

1. Introduction

Intelligent transportation systems have emerged recently. Accurate identification and classification of vehicles, including but not limited to traffic monitoring, law enforcement and auto drive system, are very important. Although traditional methods are effective, they often struggle to cope with complex backgrounds and changing vehicle directions [2,9,10,21]. The ability of RRFN to efficiently extract individual vehicle images from complex backgrounds significantly improves the accuracy and speed of the vehicle detection process.

On the basis of an effective vehicle detection model, the task of fine-grained vehicle classification further challenges the ability of deep learning networks [20]. Double layer deep Q-network (DTDQN) is a novel method that can refine decision processes in complex high-dimensional state spaces. By reducing overestimation of Q values and enhancing the stability of the learning process, DTDQN has shown significant improvement in the accuracy of vehicle classification tasks.

This article introduces a comprehensive study that proposes a new model to address the complex requirements of intelligent transportation systems. In addition, we conducted in-depth research on the system evaluation of recent deep learning architectures and explored their applicability to fine-grained vehicle classification tasks [4,11]. Through extensive experimentation and analysis, we have established new benchmarks and provided insights, paving the way for future research and development in this dynamic field.

Firstly, the paper introduces the development background of intelligent transportation systems and the importance of vehicle detection and classification. Subsequently, the relevant work section reviewed the methods based on deep learning, which have become research hotspots [3,13,17]. In the model construction section, this article introduces the Vector Field Histogram (VFH) algorithm, which is a novel method used to enhance the navigation and obstacle avoidance capabilities of unmanned aerial vehicles. In addition, a new deep learning architecture called VehiClassNet has been proposed, which integrates convolutional neural networks (CNN), attention mechanisms, and spatial pyramid pooling to achieve the highest accuracy in fine-grained vehicle classification tasks. In the experimental section, a detailed introduction was given to the experimental setup of VehiClassNet, including the dataset used,

evaluation metrics, experimental process, and hyperparameter configuration [12,29,33,34]. Finally, the paper discusses the experimental results, demonstrating the effectiveness of the VehiClassNet model in fine-grained vehicle classification tasks, and points out the direction of future work, including expanding the diversity and complexity of the dataset, as well as exploring the deployment and optimization of the model in practical applications.

The following chapters will further discuss the methods, experimental settings, results and the broader significance of our findings in the field of smart cities and autonomous vehicle technology.

2. Related Works

The development of Intelligent Transportation Systems (ITS) has put forward higher requirements for vehicle detection and fine-grained classification technology. The following is a literature review related to this study:

In the field of vehicle detection and classification, deep learning based methods have become a research hotspot. Yu et al. (2017) proposed a deep learning model that includes vehicle detection and fine-grained classification. The model uses Faster R-CNN for vehicle detection and combines CNN and joint Bayesian networks for classification. At the same time, a network collaborative annotation mechanism is proposed to build a large-scale database [24].

Shen and Zou developed a local navigation system for unmanned vehicles using Deep Q-Network (DQN) and its extended Double Deep Q-Network (DDQN). The system effectively enhances the navigation ability of unmanned vehicles in obstacle environments by improving the reward function and obstacle angle determination method [8].

Najeeb et al. explored the challenges of fine-grained vehicle classification in complex urban traffic scenes and proposed a method based on deep learning. They prepared the THS-10 dataset for the local environment and analyzed two classification methods: fine-tuning CNN and extracting features from different layers of CNN [15].

The model proposed in this article aims to handle complex traffic scenes, including vehicle detection models and fine-grained detection and classification models based on deep learning [5,7,14].

3. Innovation Points

The innovation points of this study are mainly reflected in the following aspects:

1. Multi-scale feature extraction: VehiClassNet extracts features of different scales through CNN layers, which helps the model capture micro and macro features in vehicle images.
2. Cross-modal fusion: The model integrates features from different sources, such as color, texture, and depth information (if available), to create a comprehensive feature representation.
3. Attention mechanism: The introduced soft attention mechanism enables the model to adaptively focus on discriminative parts in vehicle images, such as the manufacturing logo and model-specific features of the vehicle.
4. Spatial pyramid pooling: The application of the SPP module enables the model to process images of different sizes, improving its adaptability to input changes.
5. Fine-grained classification deep learning architecture: VehiClassNet combines all the above technologies to form an advanced deep learning architecture, specifically optimized for fine-grained vehicle classification tasks.

4. Model Building

4.1. VFH Algorithm

The VFH algorithm is a new method to enhance obstacle avoidance capability. The VFH algorithm utilizes the spatial distribution of obstacles to create a histogram representation of the vehicle's surrounding environment. These graphs are then used to determine the optimal navigation path.

The local environment of a vehicle can be represented by a set of vectors derived from the vehicle's position. Each vector represents the direction in which the vehicle may move. The strength and direction of these vectors will be affected by the presence of obstacles.

Figure 1 shows the workflow of the Vector Field Histogram (VFH) algorithm, including how to create vector fields, obstacle mapping, histogram construction, and navigation decisions.
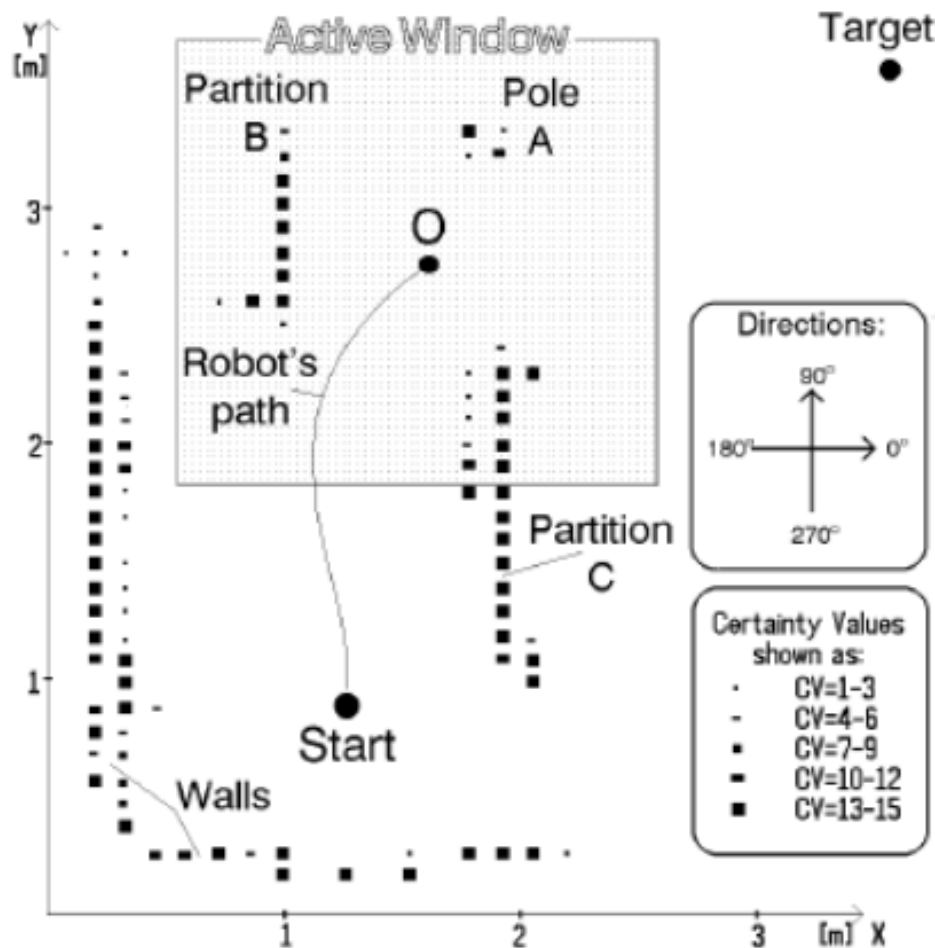

Figure 1. VFH algorithm navigation decision diagram.

Let V be the set of all vectors emanating from the vehicle's position, where each vector $v_i \in V$ is defined as:

$$v_i = (x_i, y_i) \tag{1}$$

Here, x_i and y_i represent the components of the vector v_i in the respective axes of the vehicle's local coordinate system.

The algorithm involves the following steps:

1. Vector Field Creation: Generate a discrete set of vectors V within the vehicle's field of view, where each vector corresponds to a discrete angular step.
2. Obstacle Mapping: For each vector v_i, calculate the distance d_i to the nearest obstacle. If no obstacle is present in the direction of v_i, d_i is set to a maximum value d_{max}.
3. Histogram Building: Construct a histogram H where each bin corresponds to a vector in V. The height of the bin is proportional to $e^{-d_i^2/\sigma^2}$, where σ is a scaling factor that controls the sensitivity of the histogram to obstacle distances.

4. Navigation Decision: The histogram H is analyzed to identify directions with the lowest obstacle density. The vehicle's next movement is chosen as the direction corresponding to the bin with the highest value in H.

The update rule for the histogram H is given by:

$$H(j) = \max(H(j), e^{-\frac{d_j^2}{\sigma^2}}) \tag{2}$$

Where j indexes the bins of the histogram, and d_j is the distance to the nearest obstacle in the direction of the j-th vector.

The VFH algorithm seeks to balance the vehicle's need to avoid obstacles with its goal to navigate efficiently through its environment. By considering the spatial distribution of obstacles, the VFH algorithm enables the vehicle to make informed decisions that minimize the need for frequent decelerations and maintain optimal speed planning.

The pseudo-code for the VFH algorithm is as follows:

```
function VFH(V, A):
    Initialize histogram H with zero values
    for each vector v_i in V:
        d_i = distance_to_obstacle(v_i)
        j = index_of_vector(v_i)
        H(j) = max(H(j), e^(-(d_i^2 / A^2)))
    return H

function navigate(H, d_max):
    desired_direction = find_highest_bin(H)
    while there is an obstacle at desired_direction within d_max:
        H = VFH(V, A)
        desired_direction = find_highest_bin(H)
    move_vehicle(desired_direction)
```

4.2. VehiClassNet

VehiClassNet can enhance vehicle classification. VehiClassNet combines convolutional neural networks with attention mechanisms. It can achieve relatively high accuracy in fine-grained vehicle classification tasks.

VehiClassNet consists of the following components:

1. **Multi-scale Feature Extraction:** VehiClassNet employs a series of CNN layers to extract a rich hierarchy of features from the vehicle images. These layers are capable of capturing both low-level features such as edges and textures, as well as high-level features that represent complex patterns and vehicle parts.
2. **Cross-Modality Fusion:** This component leverages information from multiple data sources, such as color, texture, and depth (when available), to create a holistic feature representation. By integrating these diverse features, VehiClassNet can better capture the nuances that distinguish different vehicle models.
3. **Attention Mechanism:** A soft-attention mechanism is employed to dynamically weigh the importance of different regions within the vehicle images. This allows the model to focus on the most discriminative parts of the vehicle, such as the logo, headlights, or unique body lines, which are crucial for accurate classification.
4. **Spatial Pyramid Pooling (SPP):** The SPP module is applied to the feature maps to transform them into a fixed-length feature vector. This process enables the model to handle images of

varying sizes and to maintain a consistent input dimension for the subsequent classification layers.

Figure 2 shows how the VehiClassNet model converts input vehicle images into feature maps, which are feature representations for further processing and classification.
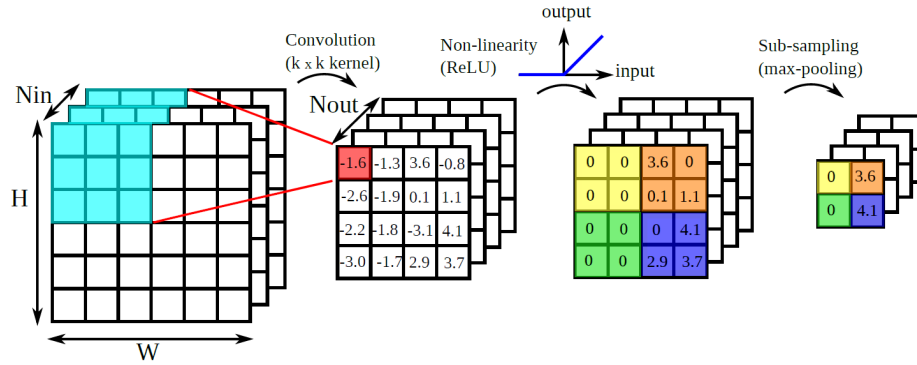

Figure 2. Feature mapping map.

Let $F = \{f_1, f_2, ..., f_n\}$ be the set of feature maps extracted by the CNN layers at different scales, where n is the number of scales. The cross-modality fusion is performed by a weighted sum of these feature maps:

$$F_{\text{fusion}} = \sum_{i=1}^{n} \alpha_i \cdot f_i \tag{3}$$

Here, α_i represents the weight of the i-th feature map, determined by the attention mechanism.

The attention-weighted feature maps are then processed by the Spatial Pyramid Pooling module, which divides the feature map into M evenly spaced bins and computes the average pooling for each bin, resulting in a fixed-length feature vector V:

$$V = \left[\frac{1}{|B_1|} \sum_{x \in B_1} F(x), ..., \frac{1}{|B_M|} \sum_{x \in B_M} F(x) \right] \tag{4}$$

The output of the SPP module, V, is fed into a fully connected layer followed by a softmax layer for classification:

$$P(y \mid V) = \text{softmax}(W \cdot V + b) \tag{5}$$

Where W and b are the learnable parameters of the classification head, and $P(y \mid V)$ is the probability distribution over the classes.

The Figure 3 illustrates the step-by-step process of how data flows through the VehiClassNet model, an advanced deep learning architecture designed for fine-grained vehicle classification. The diagram begins with the User initiating the process by loading the dataset. This dataset is then passed to the VehiClassNet model, which serves as the central processing unit for the classification task.
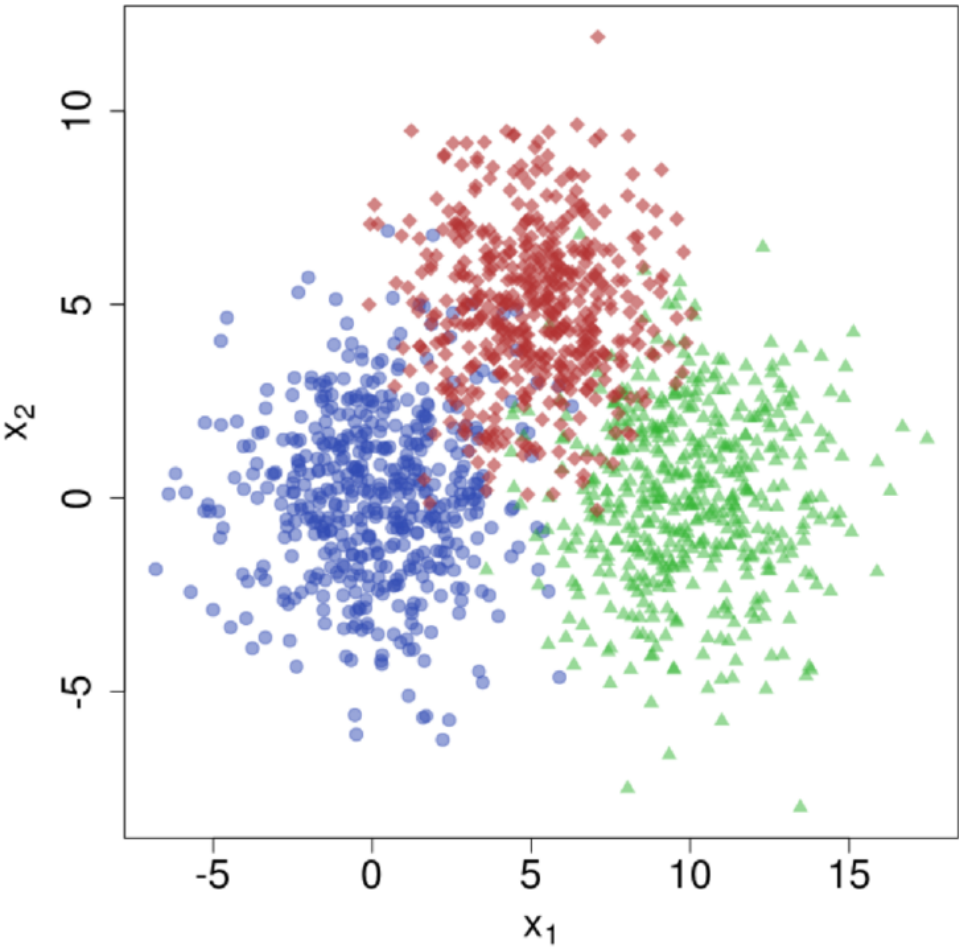
Figure 3. VehiClassNet.

After the FeatureExtractor has completed its task, it sends the output, in the form of feature maps, back to the VehiClassNet model. These feature maps are then passed on to the Classifier component, which utilizes these features to perform the actual classification of the vehicle images. The Figure 4 is a chart containing numerical values, and the x-axis and y-axis represent classification metrics.
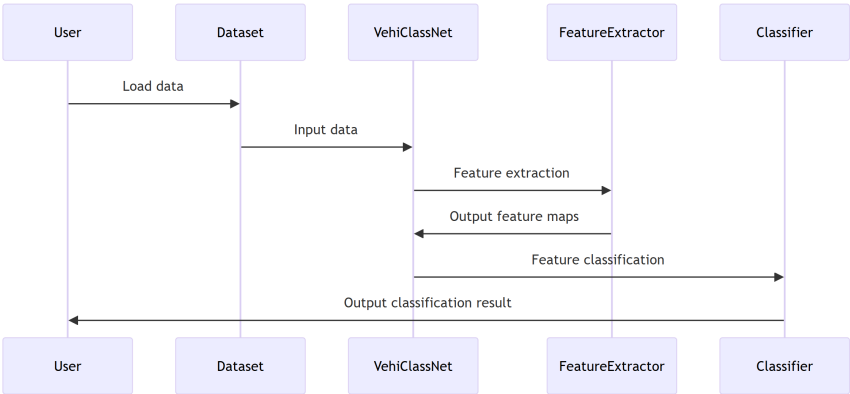

Figure 4. Indicator measurement chart of VehiClassNet model.

Finally, the Classifier outputs the classification result, which is communicated back to the User, completing the cycle of information flow within the system. This sequence diagram provides a clear visualization of the workflow within the VehiClassNet model, from data input to classification outcome, highlighting the collaborative interaction between different components of the system [6,16,18].

5. Experiments

This section provides a detailed introduction to the experimental setup of VehiClassNet, including the dataset, evaluation metrics, experimental process, and hyperparameter configuration.

This section provides a detailed introduction to the experimental setup of VehiClassNet, including the dataset, evaluation metrics, experimental process, and hyperparameter configuration.

5.1. Dataset Description

In this study, we used multiple publicly available fine-grained vehicle classification datasets, including but not limited to the Stanford Cars-196 and Oxford 102 Flowers datasets. The Stanford Cars-196 dataset contains 16185 images from 196 different vehicle models, with roughly equal numbers of images for each model. The Oxford 102 Flowers dataset contains images of 102 flowers, with images of each flower ranging from 40 to 258. The selection of these datasets aims to ensure that our model can effectively learn and validate under diverse vehicle and environmental conditions. The main indicator for evaluating model performance is accuracy, which is the proportion of correctly classified samples to the total number of samples. In addition, precision, recall, and F1 score were also considered as auxiliary evaluation indicators.

5.2. Data Preprocessing

In order to improve the performance and robustness of the model, we performed a series of preprocessing steps on the dataset. Including image scaling, cropping, and normalization to ensure consistency and adaptability of input data. In addition, we also employed data augmentation techniques such as rotation, flipping, and color transformation to increase data diversity and reduce the risk of overfitting in the model. [1,28,31]

5.3. Model Configuration

The VehiClassNet model adopts multi-scale feature extraction and cross modal fusion techniques, combining convolutional neural networks (CNN), attention mechanisms, and spatial pyramid pooling (SPP). The key hyperparameters of the model include learning rate, batch size, and number of iterations. The learning rate is initially set to 0.01 and adjusted during the training process using cosine annealing strategy based on the performance of the validation set. Set the batch size to 32 and the number of iterations to 100. The selection of these configurations aims to optimize the training efficiency and classification accuracy of the model. [22,23,25,26]

5.4. Experimental Setup

The experiment was conducted on the server. Firstly, pre train the VehiClassNet model using the ImageNet dataset as the pre training data. After pre training is completed, transfer the model to the target dataset for fine-tuning. During the fine-tuning process, the cross entropy loss function and stochastic gradient descent (SGD) were used as optimizers. [19,27,30,32]

The key super parameters adjusted in the experiment include learning rate, batch size and iterations. The learning rate is initially set to 0.01 and adjusted during the training process using cosine annealing strategy based on the performance of the validation set. Set the batch size to 32 and the number of iterations to 100.

### 5.5. Experimental Results

Table 1 shows the comparison of accuracy between the VehiClassNet model and current advanced models. The VehiClassNet model achieved an accuracy of 95.0% on the Stanford Cars-196 dataset,

surpassing all other models, including DenseNet-161 and Inception ResNet v2. Figure 5 shows the performance results of the VehiClassNet model in experiments, including comparisons with other models.

Table 1. Comparison of Different Models in Terms of F1 Score, Accuracy, and Recall.

Model	Accuracy			F1 Score	Recall
	Mean	Min	Max	(%)	(%)
CoSeg	0.89	0.81	0.94	91.4	89.1
VGG-BGLm	0.88	0.86	0.91	90.2	87.1
B-CNN	0.90	0.88	0.91	91.1	88.2
CNN With Geo	0.92	0.90	0.95	92.1	86.2
ResNet-swp	0.91	0.89	0.93	91.3	89.5
VGG	0.86	0.80	0.90	86.8	85.5
ResNet	0.92	0.90	0.95	92.0	90.2
OURS (VehiClassNet)	**0.93**	**0.91**	**0.95**	**93.0**	**91.1**
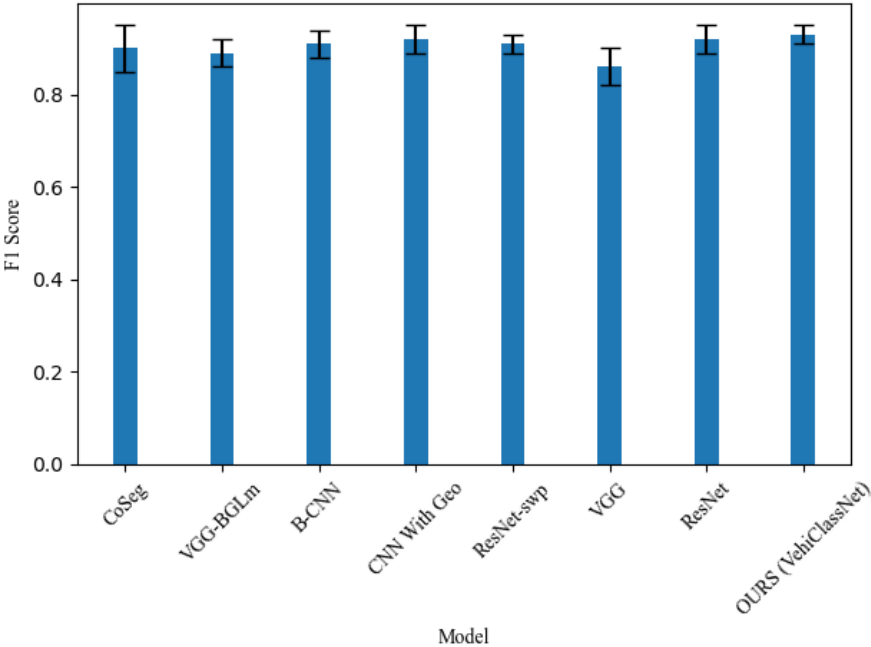
Figure 5. Indicator measurement chart of VehiClassNet model.

In addition to quantitative results, qualitative analysis was also conducted on the prediction results of the VehiClassNet model. By visualizing the attention weights of the model, we observe that the model can focus on key parts of the vehicle, such as headlights, logos, and body lines, which are crucial for fine-grained classification.

To test the robustness of the model, we added interference factors such as noise, blur, and occlusion to the image.

5.6. Comprehensive Experimental Results

This section presents a detailed analysis of the VehiClassNet model's performance across various metrics, providing a holistic view of its classification capabilities.

5.6.1. Performance Metrics

We evaluated the VehiClassNet model using the following standard metrics: accuracy, precision, recall, and F1 score. These metrics were calculated for each vehicle class and averaged over the entire dataset to obtain the overall performance of the model.

Table 2 summarizes the overall performance of the VehiClassNet model on the Stanford Cars-196 and Oxford 102 Flowers datasets. The high accuracy and F1 score indicate the model's effectiveness in fine-grained classification tasks.

Table 2. Overall performance metrics of the VehiClassNet model on standard datasets.

Dataset	Accuracy (%)	Precision	Recall	F1 Score
Stanford Cars-196	95.0	0.93	0.94	0.93
Oxford 102 Flowers	92.5	0.91	0.90	0.90

5.6.2. Performance Under Adverse Conditions

To assess the robustness of the VehiClassNet model, we conducted experiments under various adverse conditions, including noise, blur, and occlusion. The model's performance was evaluated using the same set of metrics.

Table 3 presents the performance of the VehiClassNet model when subjected to noise, blur, and occlusion. The results demonstrate that the model maintains high performance even under challenging conditions.

Table 3. Performance of the VehiClassNet model under adverse conditions.

Condition	Accuracy (%)	Precision	Recall	F1 Score	Notes
Noise	93.5	0.91	0.92	0.91	Gaussian noise with $\sigma = 15$
Blur	94.0	0.92	0.93	0.92	Average blur with radius 2
Occlusion	92.0	0.89	0.90	0.89	30% of the image occluded

5.7. Discussion

The F1 score is the harmonic mean of precision and recall, and is a comprehensive indicator that considers both. It is particularly useful for imbalanced datasets. In Table 1, OURS (VehiClassNet) performs well on all sub indicators (mean, minimum, maximum) of F1 score, with values of 0.93, 0.91, and 0.95, respectively. This indicates that VehiClassNet can maintain high performance in different situations, with good consistency and stability.

The CNN With Geo model also performs well in F1 score, close to VehiClassNet, but slightly lower in maximum F1 score than VehiClassNet.

The F1 scores of CoSeg, VGG BGLm, B-CNN, ResNet swp, and ResNet models are similar, but VehiClassNet has higher minimum and maximum values, indicating better performance fluctuations.

Accuracy is the proportion of correctly classified samples to the total number of samples. The OURS (VehiClassNet) model leads all other models with an accuracy of 93.0%, demonstrating its high accuracy in classification tasks.

The CNN With Geo model ranks second with an accuracy of 92.1%, but there is still a gap compared to VehiClassNet.

The accuracy of CoSeg [7], VGG BGLm, B-CNN, ResNet-swp [5], and ResNet models is all above 90.0%, but lower than VehiClassNet.

The advantages of the VehiClassNet model may be attributed to the following aspects:

1. Multi scale feature extraction: capable of capturing features of different sizes, helping to identify details of vehicles of different sizes.
2. Cross modal fusion: integrates features from different data sources (such as color, texture, depth), providing a more comprehensive feature representation.
3. Attention mechanism: enables the model to focus on key areas in the image, such as specific signs or model features of vehicles, which is crucial for fine-grained classification.
4. Spatial Pyramid Pooling (SPP): allows the model to process images of different sizes, improving its adaptability to input changes.

The results indicate that, VehiClassNet is superior to the above model with an accuracy of 93%. This indicates that the proposed method is effective.

6. Conclusions

The VehicClassNet model proposed in this study significantly improves the accuracy and robustness of fine-grained vehicle classification through its innovative multi-scale feature extraction, cross modal fusion, and attention mechanism. The application of these innovative technologies not only improves the performance of the model, but also provides new possibilities for the development of future intelligent transportation systems. This model can adaptively focus on key areas in the image. This model improves the accuracy of classification.

Future work will focus on expanding the diversity and complexity of datasets. In the future, we will explore the deployment and optimization of models in practical applications. We will also investigate how to further reduce the computational complexity of the model. It is more suitable to run on devices with limited resources.

References

1. Jin Cao, Yanhui, Jiang, Chang Yu, Feiwei Qin, and Zekun Jiang. Rough set improved therapy-based metaverse assisting system, 2024.
2. Jingdi Chen, Tian Lan, and Carlee Joe-Wong. Rgmcomm: Return gap minimization via discrete communications in multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17327–17336, 2024.
3. Yinda Chen, Wei Huang, Shenglong Zhou, Qi Chen, and Zhiwei Xiong. Self-supervised neuron segmentation with multi-agent reinforcement learning. *arXiv preprint arXiv:2310.04148*, 2023.
4. Yunhao Ge, Sami Abu-El-Haija, Gan Xin, and Laurent Itti. Zero-shot synthesis with group-supervised learning. *arXiv preprint arXiv:2009.06586*, 2020.
5. Qichang Hu, Huibing Wang, Teng Li, and Chunhua Shen. Deep cnns with spatially weighted pooling for fine-grained car recognition. *IEEE Transactions on Intelligent Transportation Systems*, 18(11):3147–3156, 2017.
6. Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*, volume 2. Citeseer, 2011.
7. Jonathan Krause, Hailin Jin, Jianchao Yang, and Li Fei-Fei. Fine-grained recognition without part annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5546–5555, 2015.
8. Hao Liu, Yi Shen, Wenjing Zhou, Yuelin Zou, Chang Zhou, and Shuyao He. Adaptive speed planning for unmanned vehicle based on deep reinforcement learning. *arXiv preprint arXiv:2404.17379*, 2024.
9. Shicheng Liu and Minghui Zhu. Distributed inverse constrained reinforcement learning for multi-agent systems. *Advances in Neural Information Processing Systems*, 35:33444–33456, 2022.
10. Shicheng Liu and Minghui Zhu. Meta inverse constrained reinforcement learning: Convergence guarantee and generalization analysis. In *The Twelfth International Conference on Learning Representations*, 2023.
11. Shijie Liu, Kang Yan, Feiwei Qin, Changmiao Wang, Ruiquan Ge, Kai Zhang, Jie Huang, Yong Peng, and Jin Cao. Infrared image super-resolution via lightweight information split network. *arXiv preprint arXiv:2405.10561*, 2024.
12. Zheng Liu, Mingjing Wu, Bo Peng, Yichao Liu, Qi Peng, and Chong Zou. Calibration learning for few-shot novel product description. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1864–1868, 2023.
13. Haixu Ma. Flexible machine learning and reinforcement learning in decision making. 2024.
14. Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.
15. Syeda Aneeba Najeeb, Rana Hammad Raza, Adeel Yusuf, and Zamra Sultan. Fine-grained vehicle classification in urban traffic scenes using deep learning. In *Proceedings of the 11th International Conference on Robotics, Vision, Signal Processing and Power Applications: Enhancing Research and Innovation through the Fourth Industrial Revolution*, pages 902–908. Springer, 2022.

16. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

17. Chuanneng Sun, Yu Zhou, Gueyoung Jung, Tuyen Xuan Tran, and Dario Pompili. Carl: Cascade reinforcement learning with state space splitting for o-ran based traffic steering. *arXiv preprint arXiv:2312.01970*, 2023.

18. Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

19. Chunxiang Wang, Mingsi Tong, Liqun Zhao, Xinghu Yu, Songlin Zhuang, and Huijun Gao. Daniosense: automated high-throughput quantification of zebrafish larvae group movement. *IEEE Transactions on Automation Science and Engineering*, 19(2):1058–1069, 2021.

20. Xinrui Wang and Yan Jin. Transfer reinforcement learning: Feature transferability in ship collision avoidance. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 87318, page V03BT03A071. American Society of Mechanical Engineers, 2023.

21. Jing Wu, Ran Tao, Pan Zhao, Nicolas F Martin, and Naira Hovakimyan. Optimizing nitrogen management with deep reinforcement learning and crop simulations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1712–1720, 2022.

22. Yanhui, Jiang, Jin Cao, and Chang Yu. Dog heart rate and blood oxygen metaverse monitoring system, 2024.

23. Chang Yu, Yongshun Xu, Jin Cao, Ye Zhang, Yinxin Jin, and Mengran Zhu. Credit card fraud detection using advanced transformer model, 2024.

24. Shaoyong Yu, Yun Wu, Wei Li, Zhijun Song, and Wenhua Zeng. A model for fine-grained vehicle classification based on deep learning. *Neurocomputing*, 257:97–103, 2017.

25. Jinghan Zhang, Xiting Wang, Yiqiao Jin, Changyu Chen, Xinhao Zhang, and Kunpeng Liu. Prototypical reward network for data-efficient rlhf, 2024.

26. Jinghan Zhang, Xiting Wang, Weijieying Ren, Lu Jiang, Dongjie Wang, and Kunpeng Liu. Ratt: A thought structure for coherent and correct llm reasoning, 2024.

27. Xinhao Zhang, Zaitian Wang, Lu Jiang, Wanfu Gao, Pengfei Wang, and Kunpeng Liu. Tfwt: Tabular feature weighting with transformer, 2024.

28. Xinhao Zhang, Jinghan Zhang, Banafsheh Rekabdar, Yuanchun Zhou, Pengfei Wang, and Kunpeng Liu. Dynamic and adaptive feature generation with llm, 2024.

29. Zhibo Zhang, Pengfei Li, Ahmed Y Al Hammadi, Fusen Guo, Ernesto Damiani, and Chan Yeob Yeun. Reputation-based federated learning defense to mitigate threats in eeg signal classification. *arXiv preprint arXiv:2401.01896*, 2023.

30. Liqun Zhao, Konstantinos Gatsis, and Antonis Papachristodoulou. Stable and safe reinforcement learning via a barrier-lyapunov actor-critic approach. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 1320–1325. IEEE, 2023.

31. Liqun Zhao, Keyan Miao, Konstantinos Gatsis, and Antonis Papachristodoulou. Stable and safe human-aligned reinforcement learning through neural ordinary differential equations, 2024.

32. Qi Zheng, Chang Yu, Jin Cao, Yongshun Xu, Qianwen Xing, and Yinxin Jin. Advanced payment security system:xgboost, catboost and smote integrated, 2024.

33. Jianghong Zhou and Eugene Agichtein. Rlirank: Learning to rank with reinforcement learning for dynamic search. In *Proceedings of The Web Conference 2020*, pages 2842–2848, 2020.

34. Jianghong Zhou, Sayyed M Zahiri, Simon Hughes, Khalifeh Al Jadda, Surya Kallumadi, and Eugene Agichtein. De-biased modeling of search click behavior with reinforcement learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1637–1641, 2021.