ISSN 2442-6571

An automated learning method of semantic segmentation for train autonomous driving environment understanding

Yang Wang [a,1,*], Yihao Chen [a,2], Hao Yuan [a,3], Cheng Wu [a,4]

[a] Soochow University, Suzhou, 215000, China
[1] 20214246032@stu.suda.edu.cn; [2] 20204246013@stu.suda.edu.cn; [3] 20205246017@stu.suda.edu.cn; [4] cwu@suda.edu.cn
* corresponding author

ARTICLE INFO

Article history
Received August 2, 2023
Revised November 14, 2023
Accepted January 14, 2023
Available online February 29, 2024

Selected paper from The 2023 6th International Symposium on Advanced Intelligent Informatics (SAIN'23), Yogyakarta (Virtually), September 21, 2023, http://sain.ijain.org/2023/. Peer-reviewed by SAIN'23 Scientific Committee and Editorial Team of IJAIN journal.

Keywords
Automated machine learning
Semantic segmentation
Reinforcement learning
Autonomous driving

ABSTRACT

One of the major reasons for the explosion of autonomous driving in recent years is the great development of computer vision. As one of the most fundamental and challenging problems in autonomous driving, environment understanding has been widely studied. It determines whether the entire in-vehicle system can effectively identify vehicles' surrounding objects and correctly plan paths. Semantic segmentation is the most important means of environment understanding among the many image recognition algorithms used in autonomous driving. However, the success of semantic segmentation models is highly dependent on human expertise in data preparation and hyperparameter optimization, and the tedious training process is repeated over and over for each new scene. Automated machine learning (AutoML) is a research area for this problem that aims to automate the development of end-to-end ML models. In this paper, we propose an automatic learning method for semantic segmentation based on reinforcement learning (RL), which can realize the automatic selection of training data and guide automatic training of semantic segmentation. The results show that our scheme converges faster and has higher accuracy than researchers manually training semantic segmentation models while requiring no human involvement.

1. Introduction

Autonomous driving is one of the internationally recognized future development directions and focuses of attention. At present, many countries have taken autonomous driving technology as a key development direction in the transportation field and have made strategic arrangements. High-level autonomous driving has gradually evolved from the technology research stage to the product implementation stage. However, with the occurrence of numerous accidents, public concerns about safety have been increasing. This is also one of the difficulties that impede the development of autonomous driving [1]–[3]. A comprehensive analysis of various accident cases shows that misjudgment of the surrounding environment is the main cause of the disaster. Due to the uncertainty of the driving scene, there may be a large deviation in the judgment of the image recognition algorithm for the unknown external environment. Among the many image recognition algorithms used in autonomous driving, semantic segmentation is the most important means of environmental understanding. It enables pixel-level image classification and separates different objects so that machines can understand their semantics. It is especially important for autonomous driving systems to accurately grasp information about the surrounding environment [4], [5]. The improvement of the semantic segmentation model detection effect is irreplaceable for reducing hidden dangers and ensuring the safety of autonomous driving.

In the field of autonomous driving, autonomous driving in rail transit is currently the most used. Compared with road traffic, rail transit trains have independent tracks and the scene is relatively simple. However, unpredictable abandoned objects, people, trains, etc. may also appear [6], [7]. Once an accident occurs, the social impact is extremely severe. This requires the train to be able to realize perception and understanding of the environment in the track ahead, which also needs to rely on semantic segmentation and other image recognition algorithms.

The mainstream semantic segmentation process roughly includes the following steps [8]–[11]: first, data collection and labeling are performed for the scene to be segmented, then the segmentation network is trained by labeled data, and after a certain period of training, the network parameters are updated to make it converge. Finally, the converged model is used for inference. It is worth noting that the accuracy of the model after convergence may not always makes people satisfied, especially when it is migrated to a completely new scene. Therefore, the model training process in most cases also requires researchers to continuously analyze the model training effect and adjust the model parameters based on experience until the model meets the requirements.

Based on the above process, academia and industry generally agree that there are two major problems in the current semantic segmentation technology: (1) Collection and cleaning of training data. Data determines the upper limit of machine learning, and high-quality training data is crucial for the training of semantic segmentation networks. However, the collection of data is often a time-consuming and laborious work. Meanwhile, the collected data inevitably have duplication, broken and errors. How to efficiently deal with these problematic data is one of the difficulties in improving data quality. (2) Automated training of the model. The current model evaluation, parameter adjustment, and model re-training all rely on the expertise of professionals. There is a lack of an objective and convenient automated training method that enables those with little relevant knowledge to train high-quality models.

In response to the above problems, we have conduct multifaceted researches and found that this problem is similar to those studied by Automated Machine Learning (AutoML). AutoML is the intersection of the two disciplines of automation and machine learning. It generally refers to an implementation that automates one or more stages of the process of machine learning without manual participation [12], [13]. In order to use machine learning techniques and achieve good performance, researchers usually need to be deeply involved in the entire process of model building. However, with the widespread application of deep learning and increasingly complex neural networks being proposed, even experts require significant resources and time to create well-performing models. The purpose of AutoML is to free people from these machine learning applications, get rid of the above-mentioned tedious model design and optimization process, and achieve true machine learning. The most critical and time-consuming step of machine learning is model training, which usually involves optimization methods.

Optimization methods focus on optimizing the hyperparameters used for training. Popular methods include Grid Search (GS) [14], Random Search (RS) [15] and Bayesian Optimization (BO) [16], [17]. GS divides the search space into regular intervals and selects the best execution point after evaluating all points, while RS selects the best point from a set of randomly selected points, and BO builds a probabilistic model mapping from hyperparameters to the validation probabilistic model mapping from hyperparameters to evaluation metrics on the set, which well balances exploration and exploitation.In addition, Gradient-based Optimization (GO) [18]–[20] uses gradient information to optimize hyperparameters and significantly improve the efficiency of HPO. Maclaurin [21] et al. proposed the reversible-dynamics memory-tape method, which efficiently handles thousands of hyperparameters through gradient information. To further improve efficiency, Pedregosa [22] used approximate gradient information instead of real gradients to optimize continuous hyperparameters. Chandra [23] proposed a final gradient-based optimizer that not only optimizes regular hyperparameters (such as learning rate) but also optimizes hyperparameters of the optimizer (such as moment coefficients of the Adam optimizer [24]).

Over the past few years, a large number of algorithms and systems have emerged and verified the feasibility of using automated machine learning methods. Thus we hope to adopt the idea of automated machine learning to solve the challenges faced by the environment understanding.

Automated learning for semantic segmentation is necessary and has achieved many results. Zhang [25] pointed out that manually designing and tuning parameters of semantic segmentation networks requires a lot of expert work, and it is difficult to find a balance between speed and performance for some real-time applications such as autonomous driving. Therefore, he proposed a customizable architecture search method to automatically generate lightweight networks with specific constraints. This is the first attempt in the direction of automatic network architecture generation for semantic segmentation. Nekrasov [26] pointed out that since manually designing networks is tedious and difficult to handle, automated design of neural network architectures for specific tasks is a very promising route. He used an RNN controller to cyclically output network structure and operations of each layer for semantic segmentation, with specialized modifications for compact semantic segmentation and the inclusion of auxiliary units to speed up search and training. Liu [27] proposed a network-level search space containing many popular designs and developed a formulation allowing gradient-based architectural search. Kim [28] applied NASNet, an AutoML reinforcement learning algorithm, to Deep U-Net network to improve image semantic segmentation performance. Chen [29] proposed a decoupled, fine-grained delay regularization method to address the problem of crashing semantic segmentation models designed automatically using NAS and better achieves a balance between high accuracy and low delay. Yang et al. [30] introduced automated semantic segmentation to the medical field by proposing a composite structure for dense labeling in which a custom 3D fully convolutional network explores spatial intensity concurrency of the initial labeling, and RNN encodes spatial orderliness to counteract boundary ambiguity, resulting in significant refinement. It allows simultaneous segmentation of multiple anatomical structures with clinical significance, such as fetus. It can be seen that automated learning of semantic segmentation is becoming a very important and practical research direction.

In this paper, our contribution is to propose an automated learning method for semantic segmentation for understanding autonomous driving environments. It innovatively uses RL to automate data selection, which avoids the huge cost of data collection and cleaning by researchers. Meanwhile, this method can be combined with generic semantic segmentation models. The agent judges training degree of the model and adjusts training set accordingly to guide automatic training of the model.

The focus of this study is to design AutoML methods for semantic segmentation of autonomous driving, aiming to solve difficulties in acquiring training data and automating model training. The structure is as follows: Section 2 we present our proposed automated learning method for semantic segmentation in detail; Section 3 presents experiments and results analysis; Section 4 concludes our paper.

2. Method

2.1. Method Framework

The main modules include automatic data collection and identification on the vehicle side, server-side model training, model evaluation, training set adjustment and retraining, and final model deployment and update. The framework supports end-to-end AutoML that maps vehicle driving scenes to semantic segmentation results. Reinforcement learning algorithm is used as key classifiers for data collection and dataset adjustment throughout the pipeline. The reinforcement learning agent is able to select training data based on the level of model training, eliminate worthless data, focus on increasing proportion of incorrectly segmented data, and guide the model to perform automatic training to achieve higher accuracy.

In this scenario, we define external environment as the image set which has been segmented using trained model. The agent interacts with environment (this segmented dataset) and selects images that are valuable for re-training. In each round of semantic segmentation model training, since parameters of the model are not the same, the filtered valuable images should be different. The agent has the ability to explore the dynamic filtering criteria self-learningly and adaptively.

2.2. Model Composition

The interaction of an agent with the environment is modeled using a markov decision process. This model consists of a quintet M=(S,A,p,r,γ). S is the state space and $s_t \in S$ denotes the state of the agent at moment t; A is the action space and $a_t \in A$ denotes the action taken by the agent at moment t; $p(s_{t+1}|s_t,a_t) \in (0,1)$is the state transfer probability of the agent taking action a_t in state s_t to state s_{t+1}; the reward function r_{t+1} denotes the reward obtained by taking action a_t in state s_t; $\gamma \in [0,1]$ is the discount factor. When in any state $s_t \in S$, performing an action $a_t \in A$ will cause the environment to enter a new state s_{t+1} with transfer probability $p(s_{t+1}|s_t,a_t)$ and give a reward r_{t+1}.

Considering complex state space of data selection, the reinforcement learning model used in this paper is DQN [31], [32]. DQN uses a neural network to replace Q-Table, where the network inputs states and each possible action has a separate output unit giving its predicted value. All feasible action values under the state can be given through a forward pass calculation. It avoids the dimension explosion brought by using Q-Table.

For simplicity, we make a streaming assumption that unlabeled data arrives as a stream. As each piece of data arrives, the agent must decide the action to take, i.e. whether that data is retained for retraining. The state s_t includes the candidate data being considered for retention or deletion and the data left after processing at time steps $1,\cdots,t-1$. The vector space S is defined as an N-dimensional space, and the vector $\phi(s_t)$ is used to represent the state at moment t. Each dimension stores the data x retained after selection by the agent, where the $1,\cdots,N-1$ dimensions represent the data retained after previous processing and the Nth dimension represents the candidate data being processed. In particular, considering that reinforcement learning receives information from segmented images, as shown in Fig. 1, each image contains a small amount of information, but the storage cost of preserving the images is high. Therefore, we add a preprocessing module to quantify value of segmented images. For each segmented image, the mechanism first performs an initial scoring with a score scale of 1 to 100. The worse the segmentation, the higher the score, representing that the image is more valuable for model training. It converts stored images into more intuitive scores, which will make it easier for machines to perform calculations.

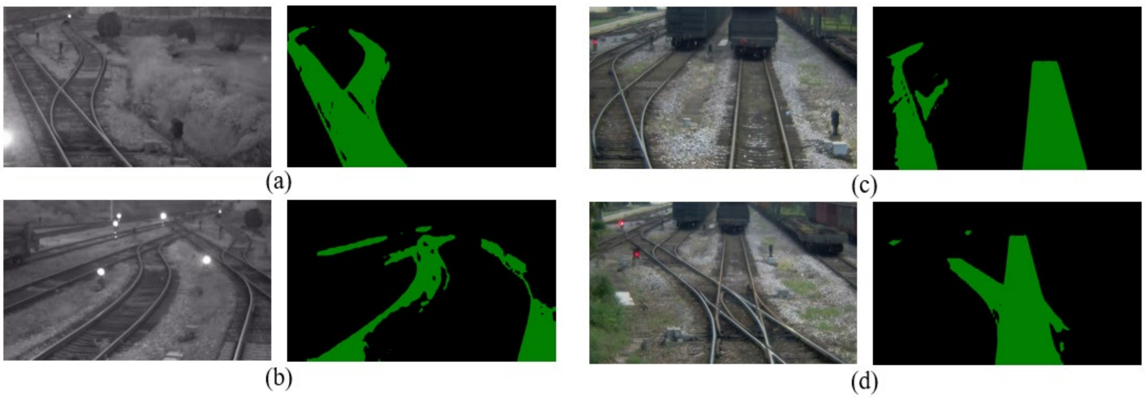


Fig. 1. Semantic segmentation of images

Correspondingly, the action space A is a set consisting of N kinds of actions, that is, $A = < a_1, a_2, \cdots, a_N >$, which correspond to the replacement of 1 to $N-1$ dimensional data with candidate data and whether to discard it.

In state iteration, each time the agent performs an action, the environment needs to return corresponding reward value to evaluate execution of action. It is clear that incorrectly segmented images are more valuable for subsequent training than correctly segmented images, so the model should be encouraged to retain more of the poorly segmented data. The reward is defined as:

$$r = \frac{score_N - score_i}{\sqrt{min(score_N, score_i)}} \tag{1}$$

where $score_N$ refers to the N^{th} dimension score of s_t , which is the score of the candidate data being processed, and score$_i$ represents the score of the original data replaced by current candidate data.

In this method, we further design two neural networks (current Q network and target Q network) containing three convolutional layers and a fully connected layer to approximate value function. The input layer of this neural network has N nodes corresponding to the N-dimensional data scores in the state space, which represent the information of N images obtained from the environment. Three convolutional layers are hidden layers with forty nodes, and the output layer is connected behind them to output the N-dimensional images that are retained in the final state. These are selected images with poor segmentation, and the overall constitutes a reinforcement learning controller network. We design neural network $L2$ loss function based on Td error:

$$L(\theta) = E_{s,a,r}\left[\left(E_{s'}[y|s,a] - Q(s,a;\theta)\right)^2\right] \tag{2}$$

$$L(\theta) = E_{s,a,r,s'}\left[(y - Q(s,a;\theta))^2\right] - E_{s,a,r}[v_{s'}[y]] \tag{3}$$

where θ refers to the set of parameters included in the neural network, $y = r + \gamma \max_{a'} Q(s',a';\theta)$, $E_{s,a,r}[v_{s'}[y]]$ is the expectation of variance of y.

Meanwhile, to make the network training more stable, we introduce an independent target network parameterized by θ^-. The structure, input and output of this network are exactly the same as the original network to obtain a stable TD target. Its network parameter θ^- is updated to the parameter θ of the DQN at intervals c. Thus the loss function is expressed as:

$$L(\theta) = E_{s,a,r,s'}\left[\left(r + \gamma \max_{a'} Q(s',a';\theta^-) - Q(s,a;\theta)\right)^2\right] \tag{4}$$

Its gradient is:

$$\nabla_\theta L(\theta) = E_{s,a,r,s'}\left[(r + \gamma \max_{a'} Q(s',a';\theta^-) - Q(s,a;\theta))\nabla_{\theta_i} Q(s,a;\theta)\right] \tag{5}$$

2.3. Training of Reinforcement Learning Models

In our proposed reinforcement learning model, the value function is approximated by a neural network, so what we have to do is to train the neural network.

In each round of training, the state s_0, the experience replay pool D, the current Q-network and the target Q-network are first initialized. At each step, the agent uses the ε-Greedy method to make current replacement action a_t according to the current state. The ε-Greedy method selects the action with the greatest value based on the prediction of the current Q network with probability $1-\varepsilon$, i.e. $a_t = arg \max_{a \in A} Q(s,a;\theta)$, while randomly selecting an action a_t from other actions as the current action with probability ε.

After executing the action a_t, the instant reward r_t is obtained, and the agent transfers to the state s_{t+1}. The agent past interaction experience $e_t = (s_t, a_t, r_t, s_{t+1})$ will be stored in the experience replay pool

D_t={e_1,··e_t}. At the same time, a batch of experiences {e_1,··e_j} is randomly selected from D_t and the target Q value y_j is calculated (when the MDP is not terminated) as follows:

$$y_j = r_j + \gamma \max_{a'} \hat{Q}(s', a'; \theta^-) \quad (6)$$

Gradient descent is used to update the Q network parameters, and the loss function is as described above. The current Q network copies its own network parameters θ to the target Q network θ^- after each c rounds of training. During the process of training, with the iteration and update of network, it gradually approaches the real value function.

2.4. Enable Data Selection and Guide Model Training

On the basis of the completed training model, the model is used to perform data selection. In each round of selection, first, the state is initialized by segmented images and the N-dimensional state vector is obtained. Then the images to be selected are continuously input, and the model gives rewards for different replacement strategies according to the Q-network in this state, selects the behavior with the highest reward as the current decision, and updates state vector after decision. The MDP transfers to the next state, and so on until the MDP transfers to the termination state.We get the final remaining images with higher value. This constitutes a complete MDP process. After going through all data selection process, the system aggregates all segmented images retained by the MDP, and retrieves original training data for the next targeted training of the semantic segmentation model.

3. Results and Discussion

In this section, training of reinforcement learning models, evaluation of reinforcement learning model and performance evaluation of automated learning methods for semantic segmentation are performed.

3.1. Experiment Preparation

This paper uses the camera data provided in the MRSI dataset [33] as a test scene to simulate scenarios that may occur in autonomous driving. The MRSI dataset uses various sensing devices mounted on the vehicle to record track scenes under different lighting and weather conditions, including straight lines, curves, and turnouts during day, dusk, night, and rain. After data cleaning, MRSI has a total of 5046 images for semantic segmentation.

There is no restriction on the choice of semantic segmentation network in this method, and this paper uses BiSeNet [34], [35] as an example, which is a lightweight real-time semantic segmentation model with a high level of comprehensive accuracy and speed.

3.2. Reinforcement Learning Model Evaluation

3.2.1. Training Effect

The training data is the segmented images output by BiSeNet. In the training, one round is taken from the initial state of MDP to the termination state. The reinforcement learning model is evaluated every 20 epochs of training. Five randomly selected data selection tasks are tested in each evaluation, and the total reward of their outputs is calculated as the result of this model evaluation. Obviously, a higher total reward indicates that more valuable training data are selected. The experimental results obtained using 30,000 rounds of data selection training based on reinforcement learning are given in Fig. 2, where the horizontal axis indicate the number of training rounds and the vertical axis indicate the cumulative reward value of each round. This result demonstrates that reinforcement learning-based data selection methods can effectively learn how to handle image selection tasks. At the beginning, the model effect improves rapidly, and then gradually becomes flat. This indicates that the selection strategy is continuously optimized during the training process and the model performance gradually becomes better.

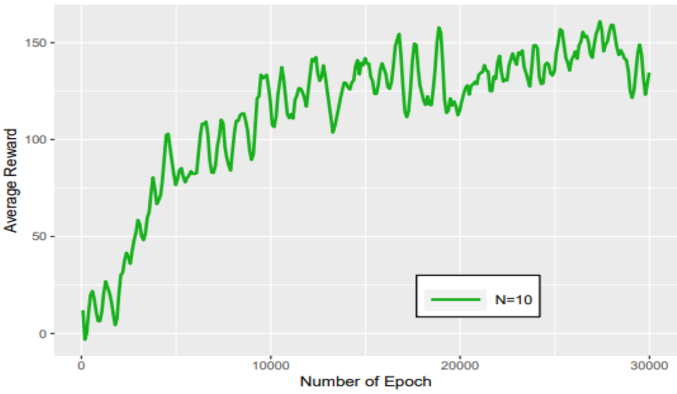


Fig. 2. Iterative process of reward for reinforcement learning model

3.2.2. Display of Data Selection

We use the trained reinforcement learning model for data selection. Fig. 3 shows a complete MDP process. After 8 Steps, the poorer images are finally selected for subsequent training.From Fig. 3, we can see that the final images filtered by agent from numbers 1 to 18 are 6, 7, 10, 12, 13, 14, 15, 16, 17, which are obviously the worst segmented images.

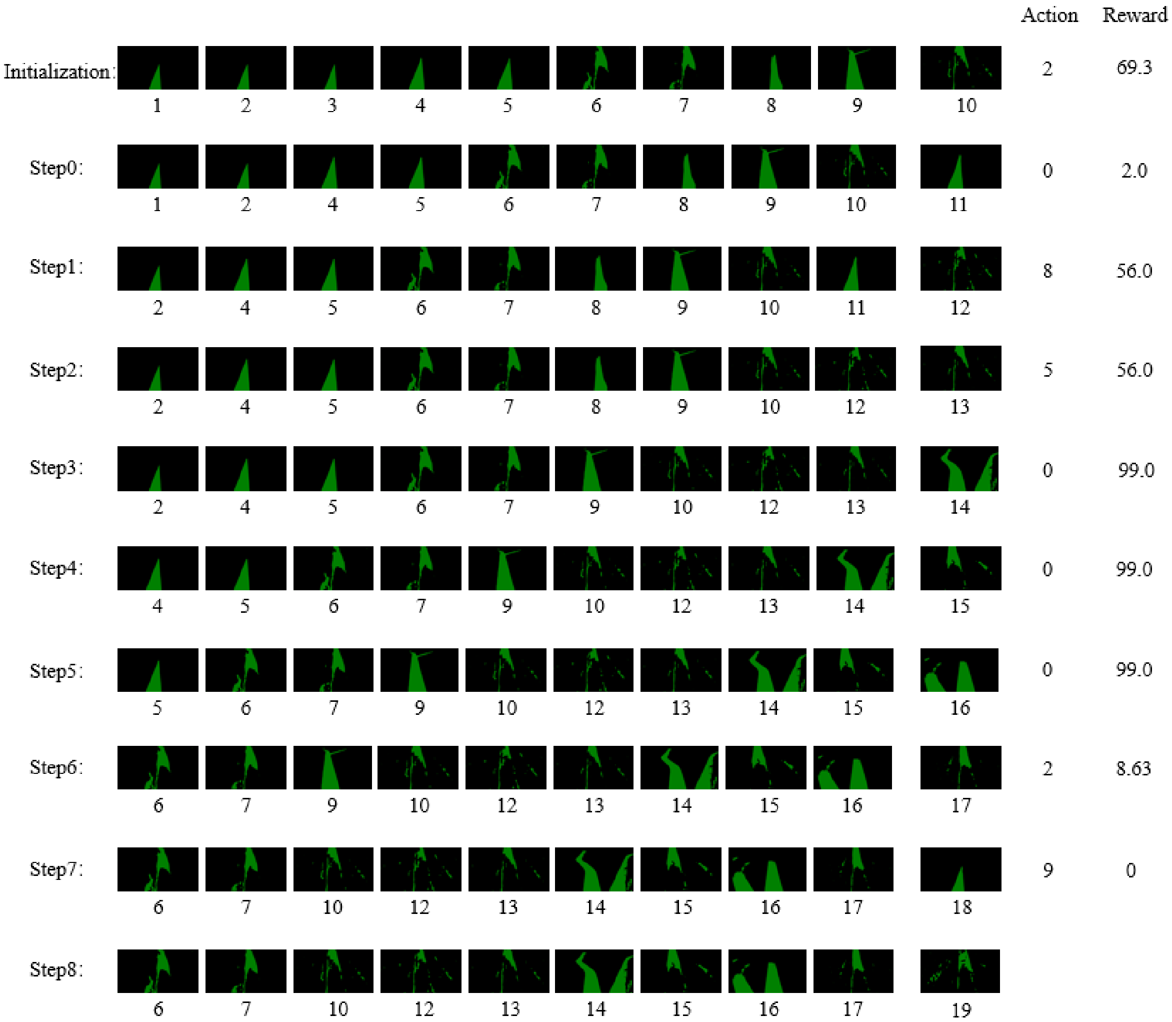
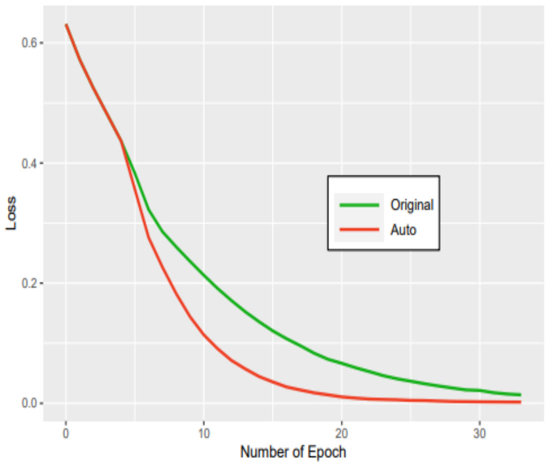


Fig. 3. An example of data selection based on reinforcement learning

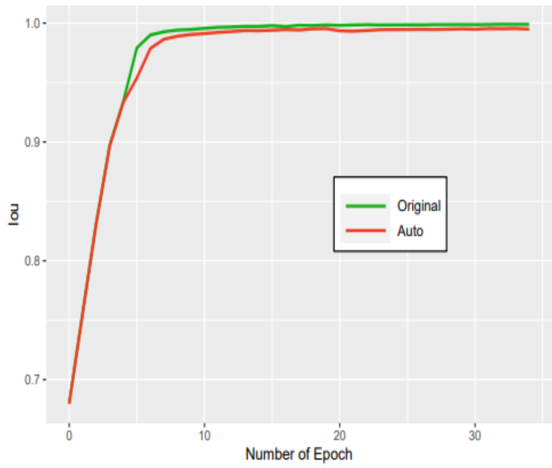
3.3. Automated Semantic Segmentation Learning Model

In addition to the above data selection model training process, our proposed automated systems adaptively controling the training of semantic segmentation model is more important. In order to visualize the effect of the automatic control system clearly, our model is compared with the original BiSeNet model in terms of loss, iou and validation set accuracy.

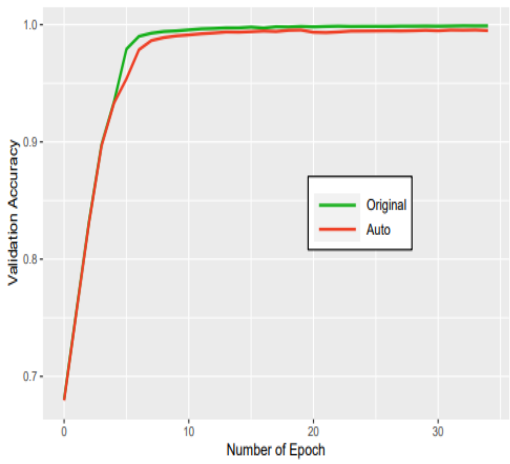
In order to ensure that the initial training parameters are consistent, the semantic segmentation network is first partially trained in the comparative experiments (corresponding to the overlapping part of two lines in Fig. 4(a), and then both continue training on this basis separately. As shown in Fig. 4(a), it can be seen that the proposed model in this paper has a significant improvement in convergence speed compared with the original BiSeNet, and the model reaches convergence state in only 15 epoches from the beginning of their respective training, while the original model is far from convergence at this time. In contrast, the rate of convergence is increased by an average of 46\%. Meanwhile, as shown in Fig. 4(b) and Fig. 4(c), our model outperforms the original model in terms of iou and validation set accuracy as well(0.91% and 0.83%). This shows that our model can greatly improve model training speed while taking into account the accuracy, and can free researchers from the tedious training. The method proposed in this article can realize automated training of autonomous driving semantic segmentation models, effectively improve the model convergence speed, and have higher accuracy.



(a) Loss



(b) Iou



(c) Validation accuracy

Fig. 4. Model effect comparison

4. Conclusion

In order to solve problems of difficulty in acquiring training data for semantic segmentation and manual model training, this paper proposes an automated learning method for semantic segmentation for autonomous driving environment understanding, and validates it in experiments. Our experimental results demonstrate that: (1) The reinforcement learning mechanisms to achieve data selection has good performance. The agent can continuously optimize selection strategy and finally achieve accurate data selection; (2) We realize data selection and guide automatic training of semantic segmentation by training the agent. Compared with the traditional method, using this method to train the semantic segmentation model has a faster convergence speed and higher model accuracy. At the same time, we noticed that although our model can effectively learn from training data, the effect may not be excellent when generalized to unseen data, which may lead to a decrease in accuracy when encountering unknown scenarios. In the future, We will work to improve this.

Declarations

Author contribution. The experimental ideas and method design were completed by the first author Wang Yang, Chen Yihao completed the data collection and processing, and Yuan Hao was responsible for writing the first draft of the paper.
Funding statement. There are no funding institutions
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

Data and Software Availability Statements

The [MRSI] data used to support the findings of this study are available at https://zenodo.org/record/5732905.

References

[1] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, "Autonomous driving in urban environments: approaches, lessons and challenges," Philos. Trans. R. Soc. A Math. Phys. Eng. Sci., vol. 368, no. 1928, pp. 4649–4672, Oct. 2010, doi: 10.1098/rsta.2010.0110.

[2] L. Mora, X. Wu, and A. Panori, "Mind the gap: Developments in autonomous driving research and the sustainability challenge," J. Clean. Prod., vol. 275, p. 124087, Dec. 2020, doi: 10.1016/j.jclepro.2020.124087.

[3] S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, and W. Shi, "Edge Computing for Autonomous Driving: Opportunities and Challenges," Proc. IEEE, vol. 107, no. 8, pp. 1697–1716, Aug. 2019, doi: 10.1109/JPROC.2019.2915983.

[4] D. Feng et al., "Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges," IEEE Trans. Intell. Transp. Syst., vol. 22, no. 3, pp. 1341–1360, Mar. 2021, doi: 10.1109/TITS.2020.2972974.

[5] K. Zheng and H. A. H. Naji, "Road Scene Segmentation Based on Deep Learning," IEEE Access, vol. 8, pp. 140964–140971, 2020, doi: 10.1109/ACCESS.2020.3009782.

[6] J. Zhao, J. Liu, L. Yang, B. Ai, and S. Ni, "Future 5G-oriented system for urban rail transit: Opportunities and challenges," China Commun., vol. 18, no. 2, pp. 1–12, Feb. 2021, doi: 10.23919/JCC.2021.02.001.

[7] P. Singh, M. A. Dulebenets, J. Pasha, E. D. R. S. Gonzalez, Y.-Y. Lau, and R. Kampmann, "Deployment of Autonomous Trains in Rail Transportation: Current Trends and Existing Challenges," IEEE Access, vol. 9, pp. 91427–91461, 2021, doi: 10.1109/ACCESS.2021.3091550.

[8] M. Siam, S. Elkerdawy, M. Jagersand, and S. Yogamani, "Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges," in 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Oct. 2017, vol. 2018-March, pp. 1–8, doi: 10.1109/ITSC.2017.8317714.

[9] S. Hao, Y. Zhou, and Y. Guo, "A Brief Survey on Semantic Segmentation with Deep Learning," Neurocomputing, vol. 406, pp. 302–321, Sep. 2020, doi: 10.1016/j.neucom.2019.11.118.

[10] F. Lateef and Y. Ruichek, "Survey on semantic segmentation using deep learning techniques," Neurocomputing, vol. 338, pp. 321–348, Apr. 2019, doi: 10.1016/j.neucom.2019.02.003.

[11] S. Chennupati, G. Sistu, S. Yogamani, and S. Rawashdeh, "AuxNet: Auxiliary Tasks Enhanced Semantic Segmentation for Automated Driving," in Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 2019, vol. 5, pp. 645–652, doi: 10.5220/0007684106450652.

[12] Q. Yao et al., "Taking Human out of Learning Applications: A Survey on Automated Machine Learning," in arxiv Artificial Intelligence, p.20, Oct. 2018. [Online]. Available at: https://arxiv.org/abs/1810.13306v4.

[13] X. He, K. Zhao, and X. Chu, "AutoML: A survey of the state-of-the-art," Knowledge-Based Syst., vol. 212, p. 106622, Jan. 2021, doi: 10.1016/j.knosys.2020.106622.

[14] J. Y. Hesterman, L. Caucci, M. A. Kupinski, H. H. Barrett, and L. R. Furenlid, "Maximum-Likelihood Estimation With a Contracting-Grid Search Algorithm," IEEE Trans. Nucl. Sci., vol. 57, no. 3, pp. 1077–1084, Jun. 2010, doi: 10.1109/TNS.2010.2045898.

[15] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," J. Mach. Learn. Res., vol. 13, pp. 281–305, 2012, [Online]. Available at: https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf.

[16] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the Human Out of the Loop: A Review of Bayesian Optimization," Proc. IEEE, vol. 104, no. 1, pp. 148–175, Jan. 2016, doi: 10.1109/JPROC.2015.2494218.

[17] T. Yu and H. Zhu, "Hyper-Parameter Optimization: A Review of Algorithms and Applications," in arxiv Machine Learning, p. 56, Mar. 2020. [Online]. Available: https://arxiv.org/abs/2003.05689v1.

[18] Y. Bengio, "Gradient-Based Optimization of Hyperparameters," Neural Comput., vol. 12, no. 8, pp. 1889–1900, Aug. 2000, doi: 10.1162/089976600300015187.

[19] J. Domke, "Generic methods for optimization-based modeling," J. Mach. Learn. Res., vol. 22, pp. 318–326, 2012, [Online]. Available at: https://proceedings.mlr.press/v22/domke12/domke12.pdf.

[20] L. Franceschi, M. Donini, P. Frasconi, and M. Pontil, "Forward and Reverse Gradient-Based Hyperparameter Optimization." PMLR, pp. 1165–1173, Jul. 17, 2017. [Online]. Available at: https://proceedings.mlr.press/v70/franceschi17a.html.

[21] D. Maclaurin, D. Duvenaud, and R. P. Adams, "Gradient-based Hyperparameter Optimization through Reversible Learning," 2015, vol. 37, p. 10, [Online]. Available at: http://arxiv.org/abs/1502.03492.

[22] F. Pedregosa, "Hyperparameter optimization with approximate gradient." PMLR, pp. 737–746, Jun. 11, 2016. [Online]. Available at: https://proceedings.mlr.press/v48/pedregosa16.html.

[23] K. Chandra, A. Xie, J. Ragan-Kelley, and E. Meijer, "Gradient Descent: The Ultimate Optimizer," Adv. Neural Inf. Process. Syst., vol. 35, p. 12, Sep. 2019. Available at: https://arxiv.org/abs/1909.13371v2.

[24] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. International Conference on Learning Representations, ICLR, pp. 1–15, Dec. 22, 2014. [Online]. Available at: https://arxiv.org/abs/1412.6980.

[25] Y. Zhang, Z. Qiu, J. Liu, T. Yao, D. Liu, and T. Mei, "Customizable Architecture Search for Semantic Segmentation," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2019, vol. 2019-June, pp. 11633–11642, doi: 10.1109/CVPR.2019.01191.

[26] V. Nekrasov, H. Chen, C. Shen, and I. Reid, "Fast Neural Architecture Search of Compact Semantic Segmentation Models via Auxiliary Cells," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2019, vol. 2019-June, pp. 9118–9127, doi: 10.1109/CVPR.2019.00934.

[27] C. Liu et al., "Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2019, vol. 2019-June, pp. 82–92, doi: 10.1109/CVPR.2019.00017.

[28] H. S. Kim, K. Y. Yoo, and L. H. Kim, "Improved performance of image semantic segmentation using NASNet," Korean Chem. Eng. Res., vol. 57, no. 2, pp. 274–282, 2019, [Online]. Available at: https://www.cheric.org/research/tech/periodicals/view.php?seq=1716510.

[29] W. Chen, X. Gong, X. Liu, Q. Zhang, Y. Li, and Z. Wang, "FasterSeg: Searching for Faster Real-time Semantic Segmentation," 8th Int. Conf. Learn. Represent. ICLR 2020, Dec. 2019. [Online]. Available at: https://arxiv.org/abs/1912.10917v2.

[30] X. Yang et al., "Towards Automated Semantic Segmentation in Prenatal Volumetric Ultrasound," IEEE Trans. Med. Imaging, vol. 38, no. 1, pp. 180–193, Jan. 2019, doi: 10.1109/TMI.2018.2858779.

[31] V. Mnih et al., "Playing Atari with Deep Reinforcement Learning," arxiv Mach. Learn., pp. 1–9, Dec. 2013, Accessed: Nov. 14, 2023. [Online]. Available at: https://arxiv.org/abs/1312.5602v1.

[32] V. Mnih et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, Feb. 2015, doi: 10.1038/nature14236.

[33] Y. Chen, N. Zhu, Q. Wu, C. Wu, W. Niu, and Y. Wang, "MRSI: A multimodal proximity remote sensing data set for environment perception in rail transit," Int. J. Intell. Syst., vol. 37, no. 9, pp. 5530–5556, Sep. 2022, doi: 10.1002/int.22801.

[34] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "BiSeNet: Bilateral Segmentation Network for Real-Time Semantic Segmentation," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11217 LNCS, Springer Verlag, 2018, pp. 334–349, doi: 10.1007/978-3-030-01261-8_20.

[35] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "BiSeNet V2: Bilateral Network with Guided Aggregation for Real-Time Semantic Segmentation," Int. J. Comput. Vis., vol. 129, no. 11, pp. 3051–3068, Nov. 2021, doi: 10.1007/s11263-021-01515-2.