ISSN 2706-655X Vol.5, Issue 11: 86-93, DOI: 10.25236/IJFET.2023.051113

Road Scene Semantic Segmentation Based on Deep Learning

Zhaoxiang Wang[1,a], Kaiqi Huang[1,b,*]

[1]School of Electrical Engineering and Automation, Jiangxi University of Science and Technology, Ganzhou, Jiangxi, China
[a]wzx111666@163.com, [b]kaiqi.huang@163.com
[*]Corresponding author

Abstract: *This study aims to address the problem of semantic segmentation in complex road scenes, which has significant applications in fields such as autonomous driving, traffic monitoring, and urban planning. The methods investigated in our research primarily include key steps such as data collection, preprocessing, and annotation. We employ CNN models for data augmentation and introduce the DAFormer semantic segmentation algorithm. In the end, this paper proposes an enhanced DAFormer network architecture, incorporating techniques such as rare class sampling, Object Category ImageNet Feature Distance (FD), and learning rate warm-up. The application of these techniques enables DAFormer to better understand image content in complex road scenarios, providing a powerful tool to tackle real-world challenges. We evaluate its performance in this challenging task by comparing it with four traditional algorithms. Experimental results demonstrate a significant performance improvement in the enhanced DAFormer algorithm in complex road environments, achieving an average intersection over union (MIoU) of 0.82, pixel accuracy (PA) of up to 89%, and improved timeliness. Compared to other algorithms, the enhanced DAFormer exhibits superior performance in terms of accuracy, stability, and timeliness.*

Keywords: *Semantic Segmentation, DAFormer Algorithm, Complex Road Scenes, Unsupervised Domain Adaptation*

1. Introduction

Automated driving is a revolutionary transportation technology designed to enable vehicles to autonomously drive without human intervention. The development in this field involves the interdisciplinary application of computer vision, sensor technology, artificial intelligence, and machine learning. Automated driving vehicles gather information about their surrounding environment through sensors, utilizing complex algorithms and models to make driving decisions, including avoiding obstacles, adhering to traffic rules, and planning the optimal route. The application of automated driving technology spans personal vehicles, public transportation, logistics, and urban traffic management, with potential economic and societal impacts. Semantic segmentation is a crucial computer vision task in automated driving. It assigns each pixel in an image to a specific semantic category, such as road, pedestrian, vehicle, or building. This is essential for automated driving vehicles as they need to accurately understand and identify various elements in the road environment to make intelligent driving decisions.

Firstly, Chen et al. [1] introduced "DeepLab," a semantic segmentation method that combines deep convolutional networks with atrous convolution and fully connected conditional random fields (CRF). The innovation of DeepLab lies in improving segmentation accuracy and effectively integrating CRF, providing support for precise segmentation. In [2], Ronneberger et al. proposed "U-Net: Convolutional networks for biomedical image segmentation," introducing the U-Net network structure, an innovative convolutional neural network for biomedical image segmentation. U-Net employs an encoder-decoder structure to efficiently handle segmentation tasks in medical images, becoming a crucial tool in the field of medical image analysis. In [3], Badrinarayanan et al. introduced "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," presenting SegNet, a deep convolutional encoder-decoder architecture for image segmentation. The innovation of SegNet lies in its use of an efficient decoder to reduce network computational complexity, making it suitable for resource-constrained applications like embedded systems. In [4], Zhao et al. proposed "ICNet for

ISSN 2706-655X Vol.5, Issue 11: 86-93, DOI: 10.25236/IJFET.2023.051113

real-time semantic segmentation on high-resolution images," introducing ICNet, a method for real-time semantic segmentation. ICNet combines multiscale information, achieving fast semantic segmentation on high-resolution images, making it valuable for real-time image processing tasks.

This paper aims to address the semantic segmentation problem in complex road scenes. We will first introduce the background and importance of image semantic segmentation tasks. Subsequently, we propose an improved DAFormer algorithm, where labeled images undergo data augmentation using CNN and then semantic segmentation using the DAFormer network architecture. We then perform a comprehensive performance comparison with four other classical algorithms. Our experimental results demonstrate a significant advantage of the improved DAFormer algorithm in complex road scenes, achieving an average intersection over union (MIoU) of 0.82, a pixel accuracy (PA) of up to 89%, and an improvement in timeliness [5-7].

2. Image Semantic Segmentation Models

Image semantic segmentation models are computer vision models designed to classify each pixel in an image, assigning it to a specific semantic category such as a person, car, tree, or building. This task requires the model not only to understand objects and regions in the image but also to accurately capture their contours and boundaries to achieve high-precision segmentation results.

2.1. Data Collection, Preprocessing, and Annotation

The data collection, preprocessing, and annotation operations in the field of image semantic segmentation are of significant importance. Data collection aims to construct an extensive and diverse dataset of images that cover various scenes, lighting conditions, and object categories. This helps in training models with better generalization capabilities. Preprocessing operations typically involve resizing images and adjusting brightness and contrast to ensure that the input data aligns with the model's requirements. The most crucial operation is annotation, where each pixel is assigned the correct semantic category label. This provides the model with information about object boundaries and regions in the image, enabling it to perform segmentation tasks accurately. Specifically, this paper conducts data collection, preprocessing, and annotation through the following steps:

1) Data Collection: The data collection phase involves obtaining a large amount of image data that covers the diversity required for the task. This typically includes using cameras, remote sensing devices, or online image repositories to capture images. In the field of automated driving, data is often collected through sensors such as in-car cameras or LiDAR to obtain real-world data of road scenes. For convenience in subsequent operations, we utilize the ACDC dataset [8-9]. The ACDC dataset comprises 4006 images evenly distributed across four common adverse conditions: fog, nighttime, rain, and snow, as shown in Fig.1.

Figure 1: Four Common Adverse Conditions Schematic Diagram

2) Data Preprocessing: The collected images typically undergo preprocessing to ensure their suitability for training segmentation models. Preprocessing steps may involve resizing images, standardizing color spaces, denoising, and adjusting brightness, among others. This helps reduce noise and inconsistencies in the data, thereby enhancing the stability of the model.

3) Data Annotation: Data annotation is a time-consuming and complex task that involves assigning a semantic category label to each pixel in every image. These labels typically represent objects or regions in the image, such as roads, pedestrians, vehicles, etc, as shown in Fig.2.

ISSN 2706-655X Vol.5, Issue 11: 86-93, DOI: 10.25236/IJFET.2023.051113

Figure 2: Diagram Illustrating Data Annotation Results

4) Data Augmentation: To increase the diversity of training data, data augmentation techniques such as random rotation, flipping, scaling, and brightness adjustment are commonly applied. This helps the model generalize better to different scenes and conditions.

5) Data Splitting: The training set is used for model training, the validation set is used for tuning hyperparameters and monitoring model performance, and the test set is used for the final evaluation of the model's performance. We specify that 70% of the data from the ACDC dataset is used as the training set for training, while the remaining 30% serves as the validation set for validation.

These data collection, preprocessing, and annotation steps are crucial for training effective image semantic segmentation models. They require careful planning and strict quality control to ensure the model can accurately understand and segment semantic information in the images.

2.2. Image Data Preprocessing and Data Augmentation

Image transmission is susceptible to the interference of noise. Noise refers to undesirable random variations in an image and can be caused by various factors such as signal interference during transmission, electromagnetic radiation, and electronic noise from electronic devices. This type of noise can result in random and undesirable changes in pixel values, thereby reducing the quality and readability of the image's information content.

In the ACDC dataset, images are captured by in-car cameras, and the predominant types of noise are Gaussian noise and salt-and-pepper noise. Salt-and-pepper noise manifests as sudden bright or dark pixel points scattered across the image, resembling salt and pepper. This is often caused by abrupt errors during data transmission or storage processes. Gaussian noise, on the other hand, is a uniformly distributed random noise that causes pixel values in the image to fluctuate randomly within a certain range. It is typically introduced by weak random interference from electronic devices or environmental factors. Commonly used image denoising algorithms include mean filtering, Gaussian filtering, and median filtering, with median filtering being effective against salt-and-pepper noise, and Gaussian filtering primarily targeting Gaussian noise [10-11].

After denoising, we need to perform data augmentation. Unsupervised domain adaptation often faces the challenge of mismatched data distributions between the source and target domains, meaning differences in image characteristics and statistical information between the two domains. Data augmentation introduces diversity and richness, making the images in the target domain more similar to the distribution of the source domain, thereby improving the model's performance in the target domain. In this paper, Convolutional Neural Networks (CNNs) are used for image data augmentation, mainly involving the following steps:

1) Data Collection and Preparation: Firstly, the denoised images mentioned above need to be annotated and then serve as the raw image dataset, including images from both the source and target domains.

2) Establish CNN Model: Researcher should choose an appropriate CNN architecture, which can be a pre-trained model (such as a model trained on ImageNet) or a custom model tailored to the specific task, ensure that the model has sufficient depth and complexity to effectively learn and generalize image features.

3) Data Augmentation Layer: The experimenter should be in in the CNN model, add a data augmentation layer for online data augmentation. These layers apply predefined augmentation operations to modify input images in real-time. Common augmentation operations include:

- Random Rotation: Randomly rotate the image by a certain angle.

- Random Flip: Horizontally or vertically flip the image with a certain probability.

ISSN 2706-655X Vol.5, Issue 11: 86-93, DOI: 10.25236/IJFET.2023.051113

- Random Scaling: Randomly scale the image by a certain factor.

- Brightness and Contrast Adjustment: Randomly adjust the brightness and contrast of the image.

- Random Cropping: Randomly crop a portion of the image.

4) Train the Model: Computer trained CNN model using augmented image data. The augmentation operations are applied in each training batch to introduce diversity and richness. This helps improve the model's robustness and generalization capability.

5) Validation and Adjustment: During the training period, validation and evaluation of the model performance are performed regularly. Based on the validation results, the model architecture and hyperparameters are adjusted to achieve better performance.

6) Application to the Target Domain: The relevant person will apply the trained CNN model to the images in the target domain once the training is complete. The model has already acquired adaptability and can better handle data from the target domain.

By using CNN for data augmentation, diversity and richness can be effectively introduced, enhancing the model's generalization capability and making it better suited for different domains of image data.

2.3. Unsupervised Domain Adaptation (UAD)

Unsupervised Domain Adaptation (UDA) is a machine learning technique aimed at addressing the challenge of generalizing models across different data domains without labeled data. Widely applied, especially in tasks involving mismatched data distributions between source and target domains, UDA is highly useful in natural language processing, image processing, and various other domains.

In the context of road image data, significant distribution differences may arise due to variations in geographical locations, climate conditions, seasonal changes, and traffic situations [11]. Unsupervised Domain Adaptation helps the model learn useful features from a known road image domain (source domain) and apply these features to road images in unknown geographic regions or different conditions (target domain), thereby enhancing semantic segmentation performance for road information. By adapting to distribution differences between different road image domains, the model can more accurately identify various elements on the road, such as vehicles, pedestrians, and traffic signs. This is crucial for applications like autonomous driving, traffic monitoring, and urban planning.

In UDA, there are typically two key domains:

1) Source Domain: This is the domain where we have labeled data, typically used for training the model. This is the model's source.

2) Target Domain: This is the domain where we want the model to generalize in real-world applications but often lacks labeled data. The target domain is our area of interest.

The goal of Unsupervised Domain Adaptation is to leverage data from both the source and target domains to enable the model to perform well in the target domain, even when the data distribution in the target domain does not perfectly match that of the source domain. The specific model workflow is illustrated as shown in Fig.3.

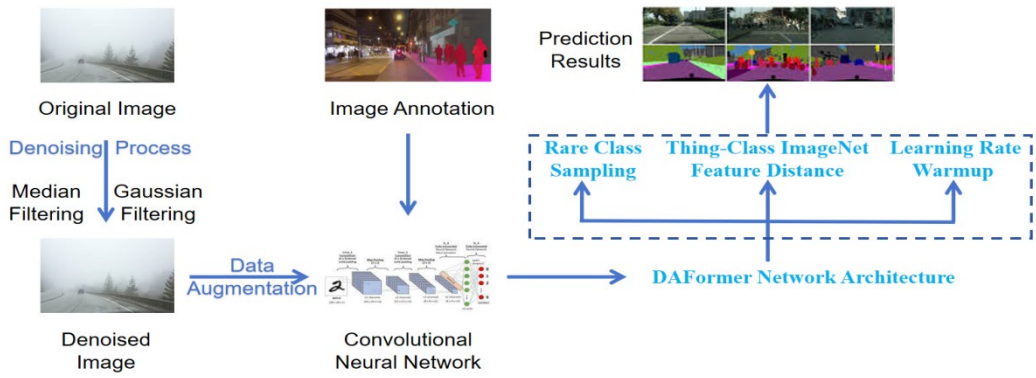

Figure 3: Unsupervised Domain Adaptation Workflow Diagram

ISSN 2706-655X Vol.5, Issue 11: 86-93, DOI: 10.25236/IJFET.2023.051113

3. DAFormer Network Architecture

Past Unsupervised Domain Adaptation (UDA) methods, when evaluated, typically employed earlier model architectures, primarily based on a (simplified) DeepLabV2 network. These methods may have become relatively outdated and might not fully meet the complex demands of domain adaptation at that time. Therefore, to enhance model performance, this paper adopts the DAFormer Network Architecture to satisfy their dual requirements in terms of supervised performance and domain adaptation. Structurally, the DAFormer Network Architecture is an improvement based on DACS, incorporating three training strategies: Rare Class Sampling, Thing-Class ImageNet Feature Distance, and Learning Rate Warmup for UDA [12-13].

3.1. Rare Class Sampling

Rare Class Sampling (RCS) is a technique used to address imbalanced datasets, based on the principle of oversampling samples from rare classes to balance the distribution of samples across different classes in the dataset. In binary classification problems, it typically involves increasing the number of samples for rare classes to bring it closer to the number of samples for common classes. This can be achieved by duplicating samples from rare classes, generating synthetic samples, or employing other methods. RCS helps improve the model's performance on imbalanced datasets, reducing classification errors for rare classes, and thus better adapting to real-world applications [14].

Specifically, for rare classes in the original dataset, the performance of Unsupervised Domain Adaptation (UDA) varies significantly across different executions. To better learn rare classes, Rare Class Sampling (RCS) more frequently selects images from rare classes in the source domain, allowing for earlier modeling of these rare classes. The frequency f_c of each class c in the source dataset can be calculated based on the number of pixels for that class:

$$f_c = \frac{\sum_{i=1}^{N_S} \sum_{j=1}^{H \times W} [y_S^{(i,j,c)}]}{N_S \cdot H \cdot W} \tag{1}$$

The sampling probability $P(c)$ for a particular class c is defined as a function of its frequency $f(c)$:

$$P(c) = \frac{e^{(1-f_c)/T}}{\sum_{c'=1}^{C} e^{(1-f_{c'})/T}} \tag{2}$$

The sampling probability is higher for classes with smaller frequencies. The temperature T controls the smoothness of the distribution.

3.2. Thing-Class ImageNet Feature Distance (FD)

Thing-Class ImageNet Feature Distance (FD) is a method used to measure the feature distance between object categories. It involves training a neural network on the ImageNet dataset and then using the network's activation features to calculate the similarity between different object categories. FD can be employed to assess the similarity between object categories, aiding in the identification and classification of different objects. This method contributes to improving the performance of object classification and recognition tasks in computer vision.

Typically, the initial parameters of the segmentation model we use are pretrained on the ImageNet dataset. However, ImageNet sometimes contains real-world images where certain categories (such as buses and cars) are challenging for unsupervised domain adaptation (UDA) models to distinguish. Therefore, this algorithm proposes a hypothesis that features in ImageNet may contain useful guidance information not covered in regular pretraining.

The ImageNet model is primarily trained on thing-classes (object categories with well-defined shapes) rather than stuff-classes (abstract categories like roads and skies). Therefore, when calculating the FD loss, it is necessary to use \(M_{thing}\) to exclude the influence of specific classes.

$$\mathcal{L}_{FD}^{(i)} = \frac{\sum_{j=1}^{H_F \times W_F} d^{(i,j)} \cdot M_{things}^{(i,j)}}{\sum_j M_{things}^{(i,j)}} \tag{3}$$

Where:

$$M_{things}^{(i,j)} = \sum_{c'=1}^{C} y_{S,small}^{i,j,c'} \cdot [c' \in \mathcal{C}_{things}] \tag{4}$$

ISSN 2706-655X Vol.5, Issue 11: 86-93, DOI: 10.25236/IJFET.2023.051113

This ensures that only bottleneck feature pixels containing dominant thing classes are considered in the feature distance. The overall training loss is then:

$$\mathcal{L} = \mathcal{L}_S + \mathcal{L}_T + \lambda_{FD}\mathcal{L}_{FD} \tag{5}$$

These are the losses on the source domain, losses on the target domain, and Thing-Class ImageNet Feature Distance, respectively.

3.3. Learning Rate Warmup for UDA

Learning Rate Warmup for UDA helps the model better adapt to the data distribution of the target domain by gradually increasing the learning rate in the early stages of training, improving UDA performance. Warming up the learning rate can reduce instability in the early stages of training, assisting the model in converging faster to an appropriate parameter state, thereby enhancing domain adaptation effectiveness. During the warm-up period, the learning rate at iteration *t* is set to:

$$\eta_t = \eta_{base} \cdot t/t_{warm} \tag{6}$$

4. Conclusion

We applied the improved DAFormer algorithm to compute the ACDC dataset images. The performance of the algorithm needs to be evaluated from multiple perspectives, as elucidated by the confusion matrix. In the context of lane line segmentation, TP represents the number of pixels where lane lines are correctly predicted, TN indicates the number of pixels in the background that are correctly predicted, FP denotes the number of background pixels incorrectly predicted as lane lines, and FN signifies the number of lane line pixels incorrectly predicted as background, as illustrated in the table 1 below.

Table 1: Confusion Matrix

Detection Results	Positive instance	Negative instance
Detection result is positive.	TP	FP
Detection result is negative	FN	TN

The performance evaluation of the semantic segmentation model is primarily obtained through the combination of the four cases in the table above, resulting in the following evaluation metrics.

"mIoU" stands for "Mean Intersection over Union" and is commonly used for performance evaluation in image segmentation tasks. It is a common metric for segmentation quality and is calculated as follows:

$$\text{IoU} = \frac{TP}{TP+FP+FN} \tag{7}$$

For each category, calculate its Intersection over Union (IoU). IoU is the intersection of true positive pixels for that category divided by the union of true positive pixels and predicted positive pixels.

Average the IoU values for all categories to obtain mIoU.

$$\text{MIoU} = \frac{\left(\frac{TP}{TP+FP+FN}+\frac{TN}{TN+FN+FP}\right)}{2} \tag{8}$$

mIoU provides a global performance measure, assessing the segmentation quality of the model across different categories. Typically, mIoU values range from 0 to 1, with values closer to 1 indicating more accurate segmentation results. This is one of the widely used evaluation metrics in the field of image segmentation for comparing the performance of different models on segmentation tasks.

Pixel accuracy is the proportion of correctly predicted samples to the total number of samples, calculated using the formula:

$$\text{PA} = \frac{TP+TN}{TP+TN+FP+FN} \tag{9}$$

The category pixel accuracy is the proportion of pixels in each category that are correctly classified, calculated using the formula:

$$\text{CPA} = \frac{TP}{TP+FP} \text{ or } \text{CPA} = \frac{TN}{TN+FN} \tag{10}$$

ISSN 2706-655X Vol.5, Issue 11: 86-93, DOI: 10.25236/IJFET.2023.051113

Recall is the probability of a certain category being predicted correctly, calculated using the formula:

$$\text{Recall} = \frac{TP}{TP+FN} \tag{11}$$

The formula for calculating the F1 score evaluation criterion is as follows:

$$\text{F1} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{12}$$

Among these metrics, average pixel accuracy, category pixel accuracy, average intersection over union (IoU), and category intersection over union are used as the evaluation indicators for this study. For the lane segmentation task, the IoU matching threshold is generally set to 0.5, representing strict matching.

In this paper, FasterRCNN, FCN, DeepLabV2, SegFormer, and the proposed improved DAFormer semantic segmentation models were trained on the ACDC dataset, with a total of 4006 collected images. 80% of the dataset was used as training samples, and 20% was used as validation samples. The initial learning rate was set to 0.001, and during training, the images were batch-normalized to an appropriate resolution. The positive sample IoU threshold was set to >0.5.

Semantic segmentation can be understood as clustering pixels of different categories, distinguishing different categories of vehicles, lane lines, streetlights, pedestrians, and non-motorized vehicles in the road. This experiment verifies the lane data segmentation effect of FasterRCNN, FCN, DeepLabV2, SegFormer, and the proposed improved DAFormer semantic segmentation models in complex road scenes. The evaluation metrics include mean intersection over union (MIoU) and pixel accuracy (PA) to represent segmentation accuracy and precision. A higher value indicates a more accurate segmentation model, and the detection time represents the time taken to segment an image in seconds. The results of the performance comparison for the five semantic segmentation models according to the experimental design are shown in the table 2 below:

Table 2: Performance Comparison of Five Semantic Segmentation Models.

Detection Method	Number of Training Samples (Images)	Number of Test Samples (Images)	MIoU	PA	Detection Time (s)
FastRCNN	3204	801	0.7	84%	0.059
FCN			0.66	84.5%	0.048
DeepLabV2			0.6	84.9%	0.112
SegFormer			0.75	85.6%	0.039
Improved DAFormer			0.82	89%	0.054

The improved DAFormer semantic segmentation algorithm demonstrates superior overall performance in complex road scenes, with an average intersection over union (MIoU) of 0.82, pixel accuracy (PA) of 89%, and improved efficiency. Compared to the other four algorithms, the improved DAFormer segmentation algorithm has advantages in terms of accuracy, stability, and efficiency in complex road scenarios.

References

[1] Chen Liang, et al. "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs." IEEE Transactions on Pattern Analysis and Machine Intelligence. 2018, 40(4), 834-848. https://doi.org/10.1109/TPAMI.2017.2699184

[2] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional networks for biomedical image segmentation." In International Conference on Medical image computing and computer-assisted intervention. 2015, 234-241. https://doi.org/10.1007/978-3-319-24574-4_28

[3] Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla. "SegNet: A deep convolutional encoder-decoder architecture for image segmentation." IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017, 39(12), 2481-2495. https://doi.org/10.1109/TPAMI.2016.2644615

[4] Zhao Hengshuang, et al. "ICNet for real-time semantic segmentation on high-resolution images." In Proceedings of the European conference on computer vision (ECCV), 2018, 20-32. https://doi.org/10.1007/978-3-030-01219-9_25

ISSN 2706-655X Vol.5, Issue 11: 86-93, DOI: 10.25236/IJFET.2023.051113

[5] Milletari, Fausto, Nassir Navab, and Seyed-Ahmad Ahmadi. "V-Net: Fully convolutional neural networks for volumetric medical image segmentation." In 3D Vision (3DV), 2016 Fourth International Conference on. 2016, 565-571 https://doi.org/10.1109/3DV.2016.79

[6] Hoyer L, Dai D, Van Gool L. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, 9924-9935.

[7] Paszke, A., Chaurasia, A., Kim, S., & Culurciello, E. ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(12), 2846-2858. DOI: 10.1109/TPAMI.2017.2760923

[8] Badrinarayanan, V., Kendall, A., & Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, 41(12), 2891-2904. DOI: 10.1109/TPAMI.2018.2818320

[9] Romera, E., Alvarez, J. M., Bergasa, L. M., & Arroyo, R. ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(3), 1039-1050. DOI: 10.1109/TITS.2018.2835378

[10] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, 41(7), 1661-1674. DOI: 10.1109/TPAMI. 2018.2862814

[11] Meletis, P., & Dubbelman, G. Fast, Robust, Continuous Monocular Depth and Normal Estimation. IEEE Robotics and Automation Letters, 2019, 4(2), 2040-2047. DOI: 10.1109/LRA.2019.2899305

[12] Valada, A., Mohan, R., & Burgard, W. Self-Supervised Model Adaptation for Multimodal Semantic Segmentation. International Journal of Computer Vision, 2020, 128(4), 970-992. DOI: 10.1007/s11263-019-01253-9

[13] Mazzini, D. Guided Upsampling Network for Real-Time Semantic Segmentation. IEEE Transactions on Intelligent Transportation Systems, 2020, 21(12), 5160-5170. DOI: 10.1109/TITS. 2019. 2959256

[14] Xiaofeng. Li, Jing Wei and Hongshuang Jiao. Real-time Tracking Algorithm for Aerial Vehicles using Improved Convolutional Neural Network and Transfer Learning. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(3):2296-2305.