Available online at http://scik.org

Commun. Math. Biol. Neurosci. 2025, 2025:32

https://doi.org/10.28919/cmbn/8872

ISSN: 2052-2541

URBAN AREA ROAD SCENE SEGMENTATION USING INTERNIMAGE-ADAPTER MODEL

HENDRI SANTOSA*, GEDE PUTRA KUSUMA

Computer Science Department, BINUS Graduate Program – Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia, 11480

Abstract. With the increasing development of autonomous driving, there is a need for an accurate deep learning model to detect and segment important objects such as other vehicles, traffic lights, road signs, road segments, pedestrians, and drivers accurately. Urban road scene segmentation presents the greatest challenge due to the presence of numerous obstacles, including pedestrians, roadside vegetation, buildings, and various other elements. The proposed model is a combination InternImage as the backbone and taking the Adapter of ViT-Adapter applied in the first block of the backbone model. The outputs of the output of InternImage last block and adapter output will be combined by element-wise addition then fed to segmentor. The model evaluated by public dataset Cityscapes with mIoU as the measuring metric. The result achieved is 81.93 mIoU on test data. The addition of adapter on the first block to InternImage does improve the performance of standalone InternImage.

Keywords: semantic segmentation; convolution neural network; visual transformer; urban road scene; deep learning.

2020 AMS Subject Classification: 68T07.

1. Introduction

The automotive industry is currently making a lot of progress in automating automotive systems. This development is also triggered by the development of electric-powered automobiles.

*Corresponding author

E-mail address: hendri008@binus.ac.id

Received August 26, 2024

This automation is already predicted to be a trend in the automotive industry which will improve traffic safety and efficiency [1]. Automation in automotive can improve traffic safety. To realize safe automotive automation, a deep learning model is needed that is able to detect and segment important objects such as other vehicles, traffic lights, road signs, road segments, pedestrians, and drivers. This segmentation is done with a Deep Learning model, which is the method found to be most capable of performing Computer Vision tasks, which in this context is segmentation.

The approach to segmentation tends to be transfer learning from the image classification model as the backbone and then adding a modification layer for image detection. This model is tasked with labeling each pixel in the image. These labels are used to segment the image (the image is segmented according to the pixel labels). The most difficult road scene segmentation is in urban areas where many objects will hinder / interfere with the road segmentation such as many pedestrians, road plant objects, buildings and others, compared to highway or toll road scenes where most objects that appear are only vehicles, traffic signs, and road sections. Therefore, urban area scene datasets are popular to be used as a benchmark for the success of the model to segment roads accurately. A popular urban area scene dataset used for segmentation is Cityscapes [2].

The two leading models that achieve the best performance and are state-of-the-art on the Cityscapes dataset are InternImage and ViT-Adapter. InternImage mimics how Vision Transformer (ViT) [3] work by using dynamic sparse kernel in place of multi head self-attention in Vision Transformer. In ViT-Adapter [4], it is proposed an adapter mechanism that captures local spatial information that ViT lacks before. This addition of adapter increase the performance of ViT significantly. The adapter used in ViT-Adapter applyable to InternImage who mimics the achitecture of ViT, hence this paper proposed the implementation of adapter on InternImage that will act as the backbone of the model. Therefore, this experiment will combined both InternImage and ViT-adapter to try to improve the performance. The results will be evaluated by mIoU metric to measure the proposed combination performance in doing semantic segementation.

2. RELATED WORKS

Currently, computer vision approaches are dominated by the Vision Transformer (ViT) development architecture [3], especially in semantic segmentation tasks. The Vision Transformer architecture generally dethroned the Convolution Neural Network architecture which was previously the main approach in performing tasks in the field of Computer Vision, where the ViT model managed to produce higher performance in accuracy than CNN-based architectures [5]. However, ViT has a disadvantage in that this architecture has a computational cost and requires very high memory and requires massive data to produce high performance. Developments of ViT architectures are mostly looking for ways to mitigate this shortcoming of ViT. On the other hand, CNN-based architectures are more lightweight than ViT-based architectures and tend to require less data to achieve acceptable accuracy. However, the CNN approach has the disadvantage that it is difficult to scale where its effectiveness decreases when scaled (the increase in performance is not worth the additional computational cost at large architectural scales), compared to ViT where models can be scaled without a significant decrease in model effectiveness.

One of the developments made on CNN architecture is the application of Global Aggregation then Local Distribution (GALD) on Fully Convolutional Network proposed by [6]. This GALD is a combination of GA and LD modules. Global Aggregation (GA) is a module that calculates the feature vector at each position on the feature. The GA module takes a large feature vector even as large as the feature map. Since the GA module calculates the statistics of the features in a large window, it tends to be biased towards large patterns and oversmoothing/ignoring small patterns. To overcome this, the GA module combined with the Local Distribution (LD) module recalculates the small patterns that the GA module tends to ignore. The resulting feature map produced by the GA and LD modules is then concatenated into a new feature map. In the architecture, GALD is added at the end of FCN which is used as the final predictor of the model. This model on the Cityscapes test dataset (fine and coarse data) can achieve 83.3% mIoU (with ResNet101 backbone).

Another CNN development is DSNet proposed by [7]. There are 2 models proposed, namely DSNet Accurate version and DSNet Fast version for segmenting road scenes. Both are CNN architecture-based models where the difference between the two is in the encoder layer. The fast version is designed with smaller layers so that it is more compact and low computation. In the fast version, an encoder consisting of one initial black, 4 non-bottleneck blocks (2 3x3 convolution layers with concatination input to the final feature map of the block), 26 bottleneck blocks (1 1x1 convolution layer + non-bottleneck block) and a convolution with stride 2 is used to shrink the feature maps (this feature map shrinking lightens the computation on the network). The number of channels in the Initial block output is set to 32 channels and compressed with a ratio of 0.5 before the pooling operation to further reduce the computational burden. 1x1 convolution layer in the bottleneck block aims to reduce the number of channels. The decoders of both versions consist of 4 convolutions. The resulting feature maps are then up-sampled to 128 resolution and then concatenated. The fast version of DSNet was able to achieve mIoU: 68.6% on the CamVid dataset and mIoU: 69.1% on the Cityscapes test (fine) dataset, while the accurate version achieved mIoU: 72.6% on the dataset.

Another CNN development is also optimization with Dense Connected Search Space DCNAS proposed by [8]. The proposed Neural Architecture Search (NAS) focuses on the efficiency of its algorithm to produce a fast, computationally cheap, and accurate NAS. The proposed NAS was found to be much more efficient than other NAS algorithms with relatively small training time and computational load and even achieved the highest mIoU (compared to DPC [9], Auto-DeepLab [10], CAS [11], GAS [12], FasterSeg [13], Fast-NAS [14], Sparse Mask [15]. This approach achieved 84.3% mIoU on the Cityscapes test dataset.

The CNN development by [16], Panoptic-DeepLab, is a CNN-based model with a backbone ImageNet-pretrained neural network coupled with an ASPP module as well as a decoder built from one convolution layer at each upsampling stage. This method achieved an mIoU of 84.2% on the Cityscapes test dataset. Then there is another CNN development by [17], namely EfficientPS (Efficient Panoptic Segmentation) which consists of a semantics head that outputs semantics prediction, class, and an instance head that outputs bounding box, mask prediction. Then all the outputs of these instances are fused with a fusion module that produces the final panoptic segmentation output. This approach successfully achieved an mIoU score of 84.21% on the Cityscapes test dataset. Another CNN architecture development is the proposed ResNeSt [18] where a Split Attention mechanism is applied that captures global contextual information in the image. Split Attention is the attention mechanism used in ResNeSt. This approach was able to achieve a score of 83.3% mIoU on the Cityscapes test dataset.

The development of ViT is HRNetV2 + OCR [19] where the use of Object-Contextual Representations (OCR) is carried out for Semantic Segmentation. For the backbone used is HRNet [20] and the OCR module is applied. OCR module, the context will be set at each pixel of the object, compared to ASPP which spreads the context to sparse pixels. This approach was able to achieve a score of 84.5% mIoU on the Cityscapes test dataset. Then the proposed development of the ViT model is LawinTransformer [21]. LawinTransformer is a ViT-based model where modifications are made to the decoder using LawinASPP replacing atrous spatial pyramid pooling (ASPP). The difference between Lawin and ASPP is the size of the window used. In Lawin, the window used is large (Large Window) to capture multi-scale contextual information. This lawin acts as the attention mechanism used. Due to the large window used, the computation will increase with the size of the context patch, so pooling of large context patches captured to the spatial dimension in the query patch is done, then a multi-head mechanism is applied to the large window attention and sets the head size to a squared downsample ratio. Then the MLP-Mixer concept is also applied to strengthen the spatial representation. This approach was able to achieve 84.4% mIoU on the Cityscapes test dataset.

InternImage is a CNN base model with modifications to mimic the architecture of the Visual Transformer which is the state-of-the-art of visual deep learning tasks [22]. This is achieved by applying a dynamic sparse kernel instead of multi-head seft-attention. The convolution layer that uses this dynamic sparse kernel is called Deformable convolution (DCN). InternImage can also efficiently scale to a large number of parameters up to 1 billion parameters and 400 million training images. InternImage was also found to be less data hungry than other popular models. The data hungry test was conducted by training the model 300 times on the ImageNet-1K dataset at 1%, 10%, 50%, and 100% data and then comparing the accuracy achieved by each model. As a result, InternImage beat ResNet, ConvNeXt-T, and Swin-T where InternImage achieved the highest accuracy at 1%, 10%, and 100% data (5.9%, 56%, 83.5%). On the Cityscapes dataset [2] which is the dataset for semantic segmentation task, InternImage equipped with Mask2Former as the segmentation framework, InternImage can achieve mIoU of 86.1% (Cityscapes Test).

ViT-Adapter is a ViT [3] model at the end of which an adapter is added that overcomes the shortcomings of the Visual Transformer to capture local spatial information [4]. This adapter is an additional layer that does not require pre-training that can make ViT efficiently adapt to perform task prediction. [4] designed 3 modules, namely: spatial prior module to capture local semantics (spatial prior) from input, spatial feature injector allows ViT to utilize spatial prior, and multi-scale feature extractor to reconstruct multi-scale features needed for prediction tasks. The proposed model consists of 2 parts, namely, plain ViT which consists of patch embedding and layer L transformer encoder, and ViT Adapter which consists of 3 modules previously mentioned. For the attention mechanism used in the adapter is Deformable Attention because it is considered lighter and performs better than other attention mechanisms (attention compared to Global Attention [23], CSwin Attention [24], Pale Attention [25], Deformable Attention [26]. This approach was able to achieve 85.2% mIoU on the Cityscapes dataset.

In the study of ViT-Adapter, integration of Adapter has demonstrated significant improvement when applied to ViT based model. Given InternImage is a CNN based model that built mimicking the structure of ViT, there is a potential that an addition of adapter from ViT-Adapter [4] would improve InternImage model performance [22].

3. PROPOSED METHOD

The proposed model is InternImage-Adapter where InternImage [22] is combined as the backbone and Adapter from ViT-Adapter [4]. The InternImage architecture will be the same until the third. The adapter is added to the first block/step this addition of adapter will enrich produced feature map by InternImage that will be fed to the segmentor.

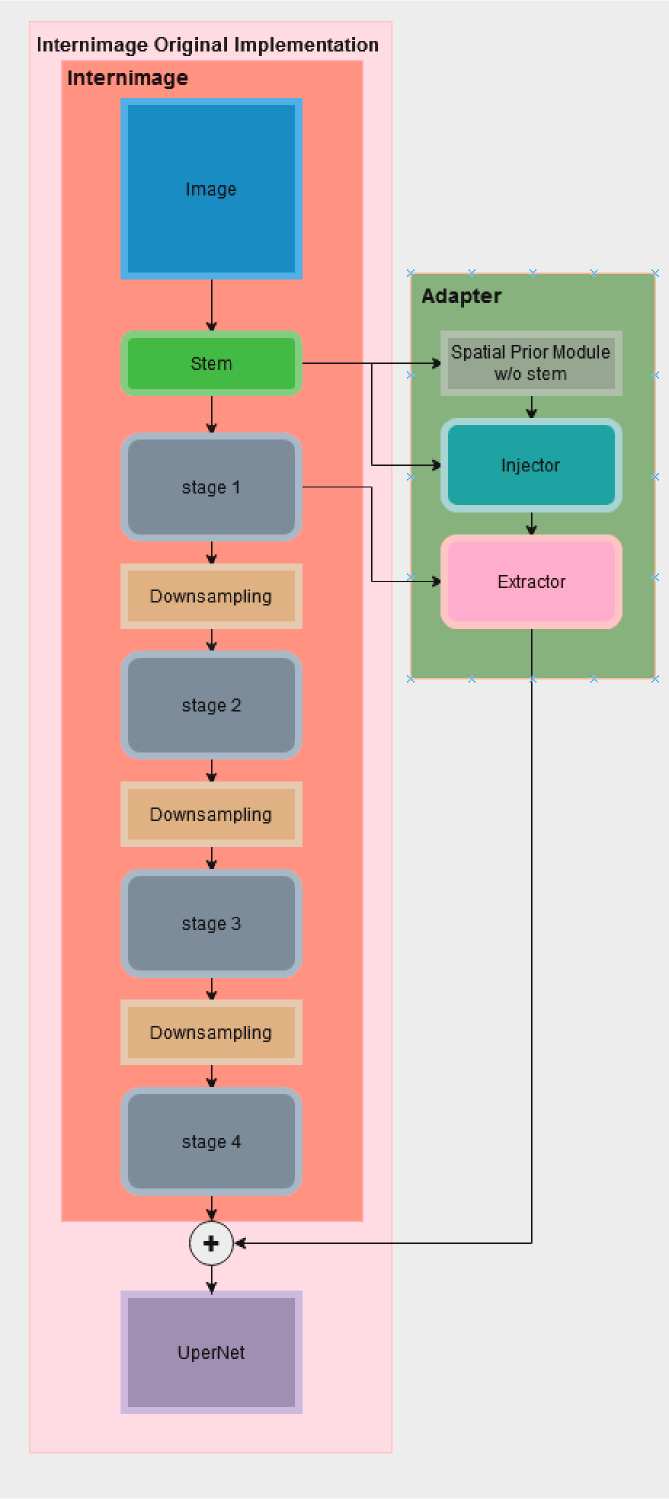

Figure 1. Proposed model (InternImage-Adapter) Architecture

The backbone InternImage implementation has no structural modifications from original InternImage implementation [22]. For the adapter, the only changes is on spatial prior module (SPM) where there is no stem in the module since it will take outuput straight from InternImage stem to remove redundancy. Injector and Extractor module remain the same with original implementation [4]. The Adapter then applied on first block where SPM take output straight from InternImage stem. Injector module take inputs from spm and InternImage stem. Then output is taken with stage 1 of InternImage before downsampling for Extractor module. The output of Adapter will be added (element-wise addition) with the output of stage 4 of InternImage before being fed to the segmentor (UperNet [27]).

4. EXPERIMENTS

4.1. Dataset

The dataset used is Cityscapes fine annotated image [2]. The Cityscapes dataset is data that focuses on understanding urban streets scenes. This dataset contains 30 classes labels grouped into 8 groups (Table 1). The data total of Cityscapes is 25000 data, with 5000 fine annotated images, 20000 coarse annotated images. The dataset that is used is Fine annotated images excluding the test data. This dataset is collected from scenes taken from 50 cities, in different seasons, during daytime, with good to moderate weather (no bad weather). This dataset is popularly used because the urban area scenes in one scene have many instances that must be segmented, resulting in a challenging dataset for the model to segment. This means that the dataset is rich in information making the model trained on this dataset capable of segmenting a scene well.

Table 1. Cityscapes dataset group and class label [2].

Group	Classes
flat	road, sidewalk, parking, rail track
human	person, rider
vehicle	car, truck, bus, on rails, motorcycle, bicycle, caravan, trailer
construction	building, wall, fence, guard rail, bridge, tunnel
object	pole, pole group, traffic sign, traffic light
nature	vegetation, terrain
sky	sky
void	ground, dynamic, static

4.2. Experimental Design

4.3.1 Device and Environtment Configuration.

The implementation of the proposed model is built using python (version 3.7.12) with anaconda for managing the libraries environment. The experiment runs Ubuntu OS that is run on Windows machines through WSL2. The model accelerated by a single NVIDIA RTX 3080 Ti 12GB GDDR6X GPU.

Table 2. Device specification used for the experiment.

Parts	Specification
CPU	Ryzen 5 7500F
GPU	NVIDIA RTX 3080 Ti 12GB GDDR6X
RAM	32 GB DDR5 6000MHz (dual channel configuration 2*16 GB)
OS	Windows 11 Pro (64 bit) (Version: 22H2; Build: 22621.2715) (**Code runs on WSL2 Ubuntu**)

4.3.2 Data splitting and preprocessing

The data used Cityscapes fine annotated images which the training data is splitted for training and validation and the validation data used for testing. The data totaled 2475 training data, 500 validation data, 500 testing data. The data preprocessing follows default preprocessing implementation applied InternImage implementation on Cityscapes dataset [22].

Table 3. Data train preprocessing done.

Preprocessing	Configuration

Resize and Random scaling	img_scale=(2048, 1024),
	ratio_range=(0.5, 2.0)

Random Crop	crop_size= (512, 1024), category_max_ratio=0.75
Random Flip	probability=0.5
Photo Metric Distortion	-

Normalize	mean=[123.675, 116.28, 103.53],
	std=[58.395, 57.12, 57.375],
	to_rgb=True

Padding	size=(512, 1024), pad_val=0,
	seg_pad_val=255

4.3.3 Configuration of Hyperparameter

The training process will be performed as in the InternImage paper [22] training InternImages on the Cityscapes dataset [2] and accelerated with GPU. The InternImage used is InternImage-T which is the smallest model that is proposed in the paper (Li et al,. 2022). The proposed model backbone applies transfer learning from pre-trained checkpoint provided by open-source InternImage repository [22]. Then the model is fine-tuned on the dataset of the experiment. The model will be trained 160800 iteration (134 Epoch, 1200 iteration each), and using AdamW as the optimizer with learning rate of 0.00006 and weight decay of 0.05. The first epoch is warmup epoch where the learning rate will increase from 0 to 0.00006 at the end of the warmup epoch. The dataset fed to the model in 2 batches.

4.3.4 Performance Metric

The mIoU score will be the benchmark for evaluating the InternImage-Adapter model on the Cityscapes dataset. mIoU (mean Intersection of Union) is the average of IoU. IoU is a measure that calculates the amount of overlap in the predicted segmentation mask against the ground truth mask for a specific label (one label). It is calculated as the ratio of the intersected area between the predicted and ground truth masks to the union area of the predicted and ground truth masks. IoU can be calculated with the following formula (1).

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad (1)$$

4.3. Experimental Result

4.3.1 Validation Result

Table 5. Validation result comparison: IoU per classes and the mIoU of InternImage-T versus InternImage-T Adapter.

Class	InternImage-T (val IoU)	InternImage-T Adapter (val IoU)
road	99.04	99.11
sidewalk	92.42	93.17
building	95.37	95.52
wall	78.81	82.39
fence	81.69	84.12
pole	70.58	70.70
traffic light	74.77	74.55
traffic sign	85.18	85.06
vegetation	94.09	94.42
terrain	81.02	84.14
sky	95.96	95.98
person	87.26	87.36
rider	74.75	75.78
car	96.68	96.72
truck	94.38	94.65
bus	95.34	95.31
train	95.62	95.74
motorcycle	79.57	81.24
bicycle	82.85	82.96
mIoU	87.13	**87.84**

On per class IoU, the most pronounced increase is on class wall, fence, terrain, and motorcycle (wall by 3.58 IoU, fence by 2.43 IoU, terrain by 3.12, and motorcycle by 1.67). Most of other class only increased by little. Overall result of the validation during the training shows the proposed model InternImage-T-Adapter mIoU is greater than mIoU achieved by InternImage-T standalone[22] by 0.71 mIoU.

The validation results show particularly on segmenting wall, fence, terrain, and motorcycle categories experienced noticeable gains in performance. While the other categories also show gains of performance, but the improvement are less pronounced. This indicates that addition of adapter on InternImage first block does improve noticibly at the performance of most categories where InternImage initially achieved mid performance (75-85 mIoU), while in categories where InternImage already achieved high mIoU (more than 85 mIoU), the improvement is less pronounced.

4.3.2 Testing Result

Table 6. Testing result comparison: IoU per classes and the mIoU of InternImage-T versus InternImage-T Adapter.

Class	InternImage-T (testing IoU)	InternImage-T Adapter (testing IoU)
road	98.52	98.62
sidewalk	87.24	88.11
building	93.55	93.52
wall	57.72	59.11
fence	67.44	66.34
pole	71.04	70.42
traffic light	74.09	73.38
traffic sign	83.26	82.99
vegetation	93.31	93.28
terrain	67.87	67.05
sky	95.49	95.56
person	84.94	84.8
rider	65.89	66.35
car	96.07	96.06
truck	85.87	86.52
bus	92.31	93.68
train	82.67	88.06
motorcycle	71.42	72.0
bicycle	81.15	80.82
mIoU	81.57	**81.93**

In the testing results, the changes in performance on per class IoU more varies than evalution and training results, where the noticeable increase in mIoU is on categories sidewalk by 0.87, wall by 1.39, bus by 1.37, train by 5.39. In the other categories, the performance changes varies from slightly increased to slightly decreased. The overall result of testing, the proposed model InternImage-T Adapter achieved 0.36 mIoU increase from what standalone InternImage-T achieved.

Overall the testing results show more spreadout noticeable gains throughout all the categories than train and validation results, where there also slight decrease in some categories indicating on specific categories the proposed model InternImage-T Adapter performs more or less the same with standalone InternImage-T [22].

5. CONCLUSION AND FUTURE WORK

In this work we have applied adapter on the first block of InternImage-T [22] model. The model then evaluated by public dataset Cityscapes dataset [2] and the results shows that the addition of Adapter on InternImage-T does achieve better mIoU than the standalone InternImage-T, improving the mIoU by 0.36 (from 81.57 to 81.93) in testing results. The addition of adapter has the potential to improve models' performance in semantic segmentation task.

The increase in performance is not significant. But this result achieved by implementing adapter block only in the first block of InternImage-T [22]. For future work, this research can be a reference for the researcher to develop InternImage-Adapter on all the InternImage block or bigger InternImage model to find out if the addition of adapter on models is consistently improving results across all scales of InternImage model (the backbone) or applying adapter on better performing backbone models.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

REFERENCES

[1] Sang Choi, W.J. Eakins, T.A. Fuhlbrigge, Trends and Opportunities for Robotic Automation of Trim & Final Assembly in the Automotive Industry, in: 2010 IEEE International Conference on Automation Science and Engineering, IEEE, Toronto, ON, 2010: p. 5584524. https://doi.org/10.1109/COASE.2010.5584524.

[2] M. Cordts, M. Omran, S. Ramos, et al. The Cityscapes Dataset for Semantic Urban Scene Understanding, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, 2016: pp. 3213–3223. https://doi.org/10.1109/CVPR.2016.350.

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, et al. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale, arXiv:2010.11929 [cs.CV]. https://doi.org/10.48550/arXiv.2010.11929.

[4] Z. Chen, Y. Duan, W. Wang, et al. Vision Transformer Adapter for Dense Predictions, arXiv:2205.08534 [cs.CV] (2022). https://doi.org/10.48550/arXiv.2205.08534.

[5] A. Krizhevsky, I. Sutskever, G.E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks, in: Advances in Neural Information Processing Systems 25 (NIPS'2012), 2012.

[6] X. Li, L. Zhang, A. You, M. Yang, K. Yang, Y. Tong, Global Aggregation Then Local Distribution in Fully Convolutional Networks, arXiv:1909.07229 [cs.CV] (2019). https://doi.org/10.48550/arXiv.1909.07229.

[7] P.R. Chen, H.M. Hang, S.W. Chan, J.J. Lin, DSNet: An Efficient CNN for Road Scene Segmentation, APSIPA Trans. Signal Inf. Process. 9 (2020), e27. https://doi.org/10.1017/ATSIP.2020.25.

[8] X. Zhang, H. Xu, H. Mo, et al. DCNAS: Densely Connected Neural Architecture Search for Semantic Image Segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 13956-13967.

[9] L.C. Chen, M. Collins, Y. Zhu, et al. Searching for Efficient Multi-Scale Architectures for Dense Image Prediction, in: S. Bengio, H. Wallach, H. Larochelle, et al. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc., 2018.

[10] C. Liu, L.C. Chen, F. Schroff, et al. Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Long Beach, CA, USA, 2019: pp. 82–92. https://doi.org/10.1109/CVPR.2019.00017.

[11] Y. Zhang, Z. Qiu, J. Liu, et al. Customizable Architecture Search for Semantic Segmentation, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Long Beach, CA, USA, 2019: pp. 11633–11642. https://doi.org/10.1109/CVPR.2019.01191.

[12] P. Lin, P. Sun, G. Cheng, et al. Graph-Guided Architecture Search for Real-Time Semantic Segmentation, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Seattle, WA, USA, 2020: pp. 4202–4211. https://doi.org/10.1109/CVPR42600.2020.00426.

[13] W. Chen, X. Gong, X. Liu, et al. FasterSeg: Searching for Faster Real-Time Semantic Segmentation,

arXiv:1912.10917 [cs.CV], (2020). https://doi.org/10.48550/arXiv.1912.10917.

[14] V. Nekrasov, H. Chen, C. Shen, 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), IEEE, Long Beach, CA, USA, 2019: pp. 9118–9127. https://doi.org/10.1109/CVPR.2019.00934.

[15] H. Wu, J. Zhang, K. Huang, Sparsemask: Differentiable Connectivity Learning for Dense Image Prediction, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6768–6777.

[16] B. Cheng, M.D. Collins, Y. Zhu, et al. Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Seattle, WA, USA, 2020: pp. 12472–12482. https://doi.org/10.1109/CVPR42600.2020.01249.

[17] R. Mohan, A. Valada, EfficientPS: Efficient Panoptic Segmentation, Int. J. Comput. Vis. 129 (2021), 1551–1579. https://doi.org/10.1007/s11263-021-01445-z.

[18] H. Zhang, C. Wu, Z. Zhang, et al. ResNeSt: Split-Attention Networks, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, New Orleans, LA, USA, 2022: pp. 2735–2745. https://doi.org/10.1109/CVPRW56347.2022.00309.

[19] Y. Yuan, X. Chen, X. Chen, et al. Segmentation Transformer: Object-Contextual Representations for Semantic Segmentation, arXiv:1909.11065 [cs.CV], (2021). https://doi.org/10.48550/arXiv.1909.11065.

[20] K. Sun, Y. Zhao, B. Jiang, et al. High-Resolution Representations for Labeling Pixels and Regions, arXiv:1904.04514 [cs.CV], (2019). https://doi.org/10.48550/arXiv.1904.04514.

[21] H. Yan, C. Zhang, M. Wu, Lawin Transformer: Improving Semantic Segmentation Transformer with Multi-Scale Representations via Large Window Attention, arXiv:2201.01615 [cs.CV] (2023). https://doi.org/10.48550/arXiv.2201.01615.

[22] W. Wang, J. Dai, Z. Chen, et al. InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Vancouver, BC, Canada, 2023: pp. 14408–14419. https://doi.org/10.1109/CVPR52729.2023.01385.

[23] A. Vaswani, N. Shazeer, N. Parmar, et al. Attention Is All You Need, in: Advances in Neural Information Processing Systems, (2017), pp. 5998-6008.

[24] X. Dong, J. Bao, D. Chen, et al. CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, New Orleans, LA, USA, 2022: pp. 12114–12124. https://doi.org/10.1109/CVPR52688.2022.01181.

[25] S. Wu, T. Wu, H. Tan, G. Guo, Pale Transformer: A General Vision Transformer Backbone with Pale-Shaped Attention, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2022, pp. 2731–2739.

[26] X. Zhu, W. Su, L. Lu, et al. Deformable DETR: Deformable Transformers for End-to-End Object Detection, arXiv:2010.04159 [cs.CV] (2021). https://doi.org/10.48550/arXiv.2010.04159.

[27] T. Xiao, Y. Liu, B. Zhou, et al. Unified Perceptual Parsing for Scene Understanding, in: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.), Computer Vision – ECCV 2018, Springer International Publishing, Cham, 2018: pp. 432–448. https://doi.org/10.1007/978-3-030-01228-1_26.