IEICE TRANSACTIONS

on Fundamentals of Electronics, Communications and Computer Sciences

DOI:10.1587/transfun.2024EAP1023

Publicized:2024/08/21

A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY

The Institute of Electronics, Information and Communication Engineers

Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

Shape-aware Convolution with Convolutional Kernel Attention for RGB-D Image Semantic Segmentation

Kun ZHOU[†], **Zejun ZHANG**[†], **Xu TANG**[†], **Wen XU**[†], **Jianxiao XIE**[†], ***and*** **Changbing TANG**[†], ***Nonmembers***

SUMMARY RGB-D semantic segmentation has attracted increasing attention over the past few years. The depth feature encodes both the shape of a local geometry as well as the base (whereabout) of it in a larger context. RGB and depth images can be concatenated into one and inputted into a network model, reducing additional computation but resulting in some distractive information as they are multimodal. For the problem, we propose a Shape-aware Convolutional layer with Convolutional Kernel Attention(CKA-ShapeConv) for reducing the distractive information by leveraging each unique input feature to rectify the kernels. Instead of using a single convolution kernel, we aggregate N parallel convolution kernels based on input-dependent attention. Specifically, four sets of attention weights are firstly calculated from each input feature map, next N parallel convolution kernels are weighted and aggregated along different dimensions, which ensure that the generated convolution kernel is more capable of catching semantic information from the input feature map, reducing interference between RGB and depth features. Then the aggregated convolution kernel is decomposed into two components: base and shape, two new learnable weights are introduced to cooperate with them independently, and finally a convolution is applied on the re-weighted combination of these two components. These two components can capture semantic and shape information of regions effectively, respectively. Meanwhile, our CKA-ShapeConv layer can be easily integrated into most existing backbone models with only a small amount of additional computation. Our experiments on NYUDv2 and SUN RGB-D datasets show that the proposed CKA-ShapeConv layer can improve the performance of backbone models effectively.

key words: *RGB-D semantic segmentation, input-dependent attention, single-stream network, dynamic convolution*

1. Introduction

Image semantic segmentation, a fundamental task in computer vision[1]–[5], is an ideal perception solution to transform an image input into its underlying semantically meaningful regions, providing pixel-wise dense scene understanding for Intelligent Transportation Systems (ITS)[6], [7]. Despite this, the RGB-based segmentation approaches might largely degenerate when applied to complex scenarios. To address this challenge, the depth image, which provides geometric information to the RGB image, has been used for achieving RGB+Depth (RGB-D)[8]–[14] semantic segmentation, demonstrating promising performance.

Convolutional neural networks (CNNs) have been widely applied in RGB image segmentation[15]–[19], whose architecture consists of encoder and decoder. The encoder is used to extract features from RGB images, with popular models such as ResNet[20], ResNeXt[21] and Mix Transformer(MiT)[24], which are pretrained on the ImageNet[22] dataset. The goal of the decoder is to restore image resolution and assign semantic class labels to each input pixel. Methods in this stage include Upsample[15], PPM[23], ASPP[17], [18], MLP-decoder[24] and so on. Semantic image segmentation for indoor scenes often faces significant challenges due to uneven lighting, severe occlusion, diverse object categories, and high similarity in surface color and texture. One of methods to overcome the problem is to increase the size of convolutional layers(kernel size, input channels, output channels), which have huge computational requirements, so it is not possible to blindly increase the size of a model. With the widespread use of depth images capture devices, it has become easy to obtain RGB images and corresponding depth images of scenes, which has greatly facilitated the development of applying depth information to semantic segmentation. Existing RGB-D semantic segmentation methods can be divided into two categories: (1) The first one employs a single network to extract features from RGB and depth modality, which are fused in the input stage. Xing et al.[25] introduce a new operator called malleable 2.5D convolution for considering depth information. Cao et al.[8] design a dedicated convolutional layer for depth data. Chen et al.[9] design a S-Conv to infer the sampling offset of convolution kernel guided by the 3D spatial information. Zheng et al.[10] utilize pre-segmentation labels from traditional image segmentation and concatenate them with RGB-D features to provide more accurate guidance for semantic segmentation. In this kind of single-stream network, RGB and depth images are concatenated into a dedicated convolutional layer, which reduces the computational cost in the encoding stage, but maybe introduce interference information and result in suboptimal performance because of the domain gap between the RGB and depth modalities. (2) The second one deploys two backbones to perform feature extraction from RGB and depth modality separately followed by a feature fusion based decoder for semantic prediction. Seichter et al.[11] design a series of modules to fuse RGB and depth feature. Zhou et al.[12] propose a feature reconstruction network and a multi-scale fusion strategy. Zhou et al.[13] introduce a progressive guided fusion strategy and a depth enhancement network to progressively fuse features and improve the quality of depth images. Zhang et al.[14] utilize the transformer as the backbone to extract features, and then fuse the different modality information to achieve RGB-D semantic segmentation. This kind of double-stream network requires to retain two heavy encoders after training, increasing the deployment

[†]The authors are with the College of Physics and Electronic Information Engineering, Zhejiang Normal University, Jinhua, China.

burden of RGB-D semantic segmentation. And it is hard to decide when and where the RGB and depth features need to be fused. To tackle the aforementioned challenges, we construct a single-stream network and introduce a series of rectification modules to rectify our convolution kernel.

The core idea of attention mechanism is to dynamically adjust the weights of features, imitating the selective perception mechanism of the human visual system, and to focus attention on more important parts of the feature while suppressing irrelevant parts. Vaswani et al.[26] first propose to use self-attention mechanism to compute global dependencies in the inputs and applied it to machine translation. At the same time, attention modules have been widely used in the field of computer vision. Hu et al.[27] propose a channel attention module called "squeeze-and-excitation (SE)" to weight each channel of the feature map based on the mutual dependencies between convolutional feature channels. Zhou et al.[28] propose a hybrid cross-fusion co-attention module to improve RGB and depth semantic information. Li et al.[29] calculate channel attention and present channel split to highlight important channels and refine features. Most previous researches focused on rectifying the output convolutional feature with attention modules, which incurred a significant computational cost. Therefore, we consider applying the attention mechanism to the rectification of convolution kernels. The same convolution kernel is applied to each input feature when designing convolutional layers. However, each input feature is unique and should be consumed using its corresponding rectified convolution kernel. The dynamic mechanism of using input-dependent convolution kernel allows model to adapt to the input feature and has the potential to improve model performance and generalization. Jia et al.[30] generate small input-dependent convolution kernel filters for transforming images for the next-frame and the stereo prediction. Yang et al.[31] introduce CondConv, which increases the attention weights determined by the input of the convolutional layer and combines them to obtain the convolution kernel weights. Chen et al.[32] propose dynamic convolution, which has a similar idea to CondConv but replaces the *Sigmoid* function with the *Softmax* function to calculate attention weights and introduces a temperature annealing strategy. This optimizes the calculation and training process of attention weights. Hou et al.[33] propose a hybrid gradient convolution to capture edge information more effectively by dynamically adjusting the weights of the gradient convolution kernel. These works only compute a set of attention weight to rectify the convolution kernels along a particular dimension, which may result in ineffective corrections for other dimensions. Therefore, we consider using input-dependent attention mechanism to weight and aggregate N convolution kernels across four dimensions: spatial, input channels, output channels, and the number of kernels.

The purpose of our study is to rectify the convolution kernels using both RGB and depth features, while enhancing the capacity and performance of the model without increasing its depth and width. The ShapeConv[8] method rectifies the convolution kernel by introducing two parameters to balance the shape and base components. Building upon this, we incorporate the idea of dynamic convolution[32], relying on each specific RGB-D input, to achieve a more comprehensive rectification of each dimension of the convolution kernels meanwhile extracting more accurate 3D shape information from the input. Specifically, we propose CKA-ShapeConv, a shape-aware convolutional layer with convolutional kernel attention for RGB-D image semantic segmentation. Instead of increasing the depth and width of the model, we enhance the model's capacity by combining N parallel convolution kernels into one for each convolutional layer. In detail, four sets of attention weights are firstly calculated from each input feature map. Then, N parallel convolution kernels are weighted and aggregated along different dimensions, ensuring that the resulting convolution kernel is more effective in capturing semantic information from the input feature map. Subsequently, the aggregated convolution kernel is decomposed into two components: base and shape, two new learnable weights are introduced to independently cooperate with them. Finally, a convolution operation is applied on the re-weighted combination of these two components. These two components can effectively capture semantic and shape information of regions, respectively. The input-dependent attention strategy reduces the interference between RGB and depth features, improving model capacity with a small increase in computational cost. The rectified convolution kernels can extract more accurate feature information from higher layer input, thereby improving the model's performance and achieving better results.

To validate the effectiveness of CKA-ShapeConv, we conduct experiments on the NYUDv2[34] and SUN RGB-D[35] datasets. We apply CKA-ShapeConv to various backbone networks and several representative semantic segmentation architectures. In all cases, we observe corresponding performance improvements, which demonstrate the effectiveness of our approach.

In short, out main contributions are as follows:

- We propose a shape-aware convolutional layer with convolutional kernel attention (CKA-ShapeConv) to rectify the kernels by utilizing each unique input feature, reducing the interference caused by concatenating RGB and depth features.
- We do not increase the depth and width of the model, but instead enhance its capacity by combining N convolution kernels into one for each convolutional layer.
- We conduct extensive experiments on two indoor RGB-D semantic segmentation benchmarks and show that the proposed CKA-ShapeConv layer can improve the performance of backbone models effectively.
- We analyze the limitations of our proposed method and provide suggestions for future improvement work.

2. Method

In this section, we first introduce the principle and formula of vanilla dynamic convolution and Shape-aware convolution,

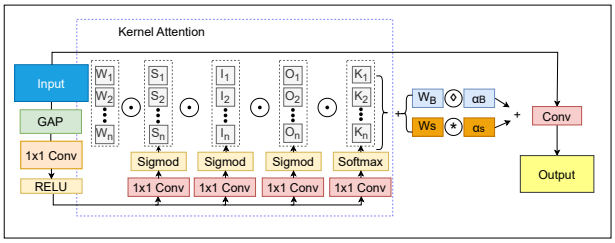

Fig. 1 A CKA-ShapeConv layer.

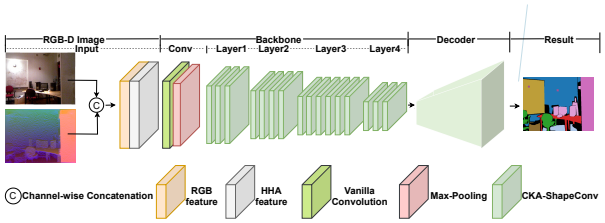

Fig. 2 The overall semantic segmentation network architecture.

as well as analyze their limitation in image segmentation field. Finally, we introduce our CKA-ShapeConv to overcome the shortcomings of these two models and provide details on its specific implementation.

2.1 Vanilla Dynamic Convolution and Shape-aware Convolution

2.1.1 Vanilla Dynamic Convolution

For the vanilla convolution layer, it consists of single static convolution kernel shared for all inputs. In contrast, the dynamic convolutional layer uses a linear combination of N convolution kernels and weights them with input-dependent attention. Specifically, given an input feature map $X_{in} \in R^{C_{in}\times H\times W}$, H and W are the spatial dimensions of the input feature map, C_{in} represents the channel numbers in the input feature map, the output feature map is obtained by

$$Y = Conv\left(\left(\sum_{i=1}^{N} K_i W_i\right), X_{in}\right) \tag{1}$$

where $Y \in R^{C_{out}\times H\times W}$ denotes the output feature map, C_{out} represents the channel numbers in the output feature map; operator $Conv$ denotes the convolution operation; $W_i \in R^{K_h\times K_w\times C_{in}\times C_{out}}$ denotes the i_{th} convolution kernel, K_h and K_w are the spatial dimensions of the kernel; $K_i \in R^1$ is the attention for weighting W_i, which is calculated by Eq.(2). To simplify, the bias term has been omitted:

$$K(X_{in}) = Sigmoid\left(FC\left(GAP\left(X_{in}\right)\right)\right) \tag{2}$$

where GAP is a global average pooling operation, through which an input feature is squeezed to $R^{C_{in}\times 1\times 1}$, and FC is a fully-connected operator, which generates N scalar values. Finally, the attention weights are calculated using the $Sigmoid$ function.

Subsequent work[32] has improved the dynamic convolution with promoting the model learning by using $Softmax$ function instead of the $Sigmoid$ function to limit $\sum_{i=1}^{N} K_i = 1$. However, a single attention scalar is still allocated to the entire convolution kernel, and this N is still huge which results in a massive number of parameters, making it difficult to deploy in practical applications.

2.1.2 Shape-aware Convolution

The core idea of Shape-aware Convolution[8] is to decompose a convolution kernel into two kernels for extracting shape and base components from depth feature, which can be combined by weights to get improved depth feature. This can be calculated as Eq.(3):

$$Y = Conv\left(\left(W_B \diamond \alpha_B + W_S * \alpha_S\right), X_{in}\right) \tag{3}$$

where $\diamond$ and $*$ denote the base-product and shape-product operator, respectively. $X_{in} \in R^{C_{in}\times H\times W}$ and $Y \in R^{C_{out}\times H\times W}$ denote the input feature map and the output feature map, respectively; $W_B \in R^{1\times 1\times C_{in}\times C_{out}}$ is the average value of the convolution kernel $W \in R^{K_h\times K_w\times C_{in}\times C_{out}}$ over the $K_h \times K_w$ dimensions, which extracts base component of input feature map, $W_s = W - W_B$ is to extract the shape component, and $W_s \in R^{K_h\times K_w\times C_{in}\times C_{out}}$; $\alpha_B \in R^1$ and $\alpha_S \in R^{C_{in}\times(K_h\times K_w)\times(K_h\times K_w)}$ denote the corresponding learnable parameters.

In [8], the ShapeConv operator is only used for depth feature. Therefore, when RGB and depth data are concatenated as input, the operator may lead to inaccurate feature extraction. Therefore, we introduce an input-dependent attention mechanism to rectify our convolution kernels, which allows the weight of the shape-aware convolution kernel to match the input data better.

2.2 Shape-aware Convolution with Convolutional Kernel Attention

Inspired by dynamic convolution[32] and ShapeConv[8], we design our shape-aware convolution with convolutional kernel attention (CKA-ShapeConv), which leverages their advantages in using $Softmax$ function instead of the $Sigmoid$ function to limit $\sum_{i=1}^{N} K_i = 1$ and addresses the issue of segmentation errors caused by variations in depth value for objects of the same class. The main idea of our method is to calculate four sets of input-dependent convolution kernel attention weights for rectification and combination of N convolution kernels. A CKA-ShapeConv layer of our model is illustrated in Fig.1 and formulated as:

$$\begin{aligned} W &= \sum_{i=1}^{N} \left(S_i \odot I_i \odot O_i \odot K_i \odot W_i\right) \\ W &= W_B + W_S \\ Y &= Conv\left(\left(W_B \diamond \alpha_B + W_S * \alpha_S\right), X_{in}\right) \end{aligned} \tag{4}$$

where $\odot$ denotes the multiplication operation along different dimensions of the kernel space; $Conv$ denotes

Algorithm 1 Shape-aware Convolution with Convolutional Kernel Attention

Input: The input feature map X_{in}, convolution kernels W_i $(i \in \{1:N\})$
Output: The output feature map Y

1: Calculate $S_i, I_i, O_i, K_i (i \in \{1:N\})$ by Eq.(5)
2: Rectify weight $W \leftarrow \sum_{i=1}^{N} (S_i \odot I_i \odot O_i \odot K_i \odot W_i)$
3: Decompose W into base component W_B and shape component W_S
4: Aggregate weight $W \leftarrow W_B \diamond \alpha_B + W_S * \alpha_S$
5: Learn the output feature map $Y \leftarrow Conv(W, X_{in})$

the convolution operation; $\alpha_B \in R^{C_{in}\times 1\times 1}$ and $\alpha_S \in R^{C_{in}\times(K_h\times K_w)\times(K_h\times K_w)}$ denote the two learnable parameters used to balance the average and residual components. $W_i \in R^{K_h\times K_w\times C_{in}\times C_{out}}$ denotes the i_{th} convolution kernel. $S_i \in R^{(K_h\times K_w)}$ assigns different attention scalars to W_i at $K_h \times K_w$ spatial locations; $I_i \in R^{C_{in}}$ assigns different attention scalars to each input channel of each convolution kernel W_i; $O_i \in R^{C_{out}}$ assigns different attention scalars to each output channel of each convolution kernel W_i; $K_i \in R^1$ assigns one attention scalar to each convolution kernel W_i. In this way, all dimensions of the convolution kernels receive attention weights that are input-dependent, ensuring the extraction of rich and accurate feature information from the upper-layer input. The specific calculation method is formulated as follows:

$$\begin{aligned} Y_{in} &= RELU\left(Conv_{1\times1}\left(GAP\left(X_{in}\right)\right)\right) \\ S_i &= Sigmoid\left(Conv_{1\times1}\left(Y_{in}\right)\right) \\ I_i &= Sigmoid\left(Conv_{1\times1}\left(Y_{in}\right)\right) \\ O_i &= Sigmoid\left(Conv_{1\times1}\left(Y_{in}\right)\right) \\ K_i &= Softmax\left(Conv_{1\times1}\left(Y_{in}\right)\right) \end{aligned} \tag{5}$$

where $Conv_{1\times1}$ denotes the 1×1 convolution layer. The input X_{in} is squeezed into a feature vector with the size $R^{C_{in}\times1\times1}$ by a global average pooling layer, followed by a 1×1 convolution layer and a *RELU*[36] operation. Next the 1×1 convolution layer maps the squeezed feature vector to a lower dimensional space. For four head branches, each has a 1×1 convolution layer with the output size of $R^{(K_h\times K_w)\times1\times1}$, $R^{C_{in}\times1\times1}$, $R^{C_{out}\times1\times1}$, $R^{N\times1\times1}$, and a *Softmax* or *Sigmoid* function to generate the normalized attentions S_i, I_i, O_i, K_i, respectively. We have incorporated temperature annealing strategy[32] into the *Sigmoid* and *Softmax* function, which is represented in Eq.(6).

$$\begin{aligned} Sigmoid(x) &= \frac{1}{1+e^{-(x/\tau)}} \\ Softmax(x_i) &= \frac{e^{(x_i/\tau)}}{\sum_{j=1}^{N} e^{(x_j/\tau)}} \end{aligned} \tag{6}$$

where x and x_i are the input vector. τ is the temperature value. The original functions are a special case($\tau = 1$). The corresponding ablation experimental result are provided in Sect. 3.5.3. The algorithm of proposed CKA-ShapeConv is illustrated as Algorithm 1.

ShapeConv[8] directly decomposes the convolution kernel weight into shape component and base component.

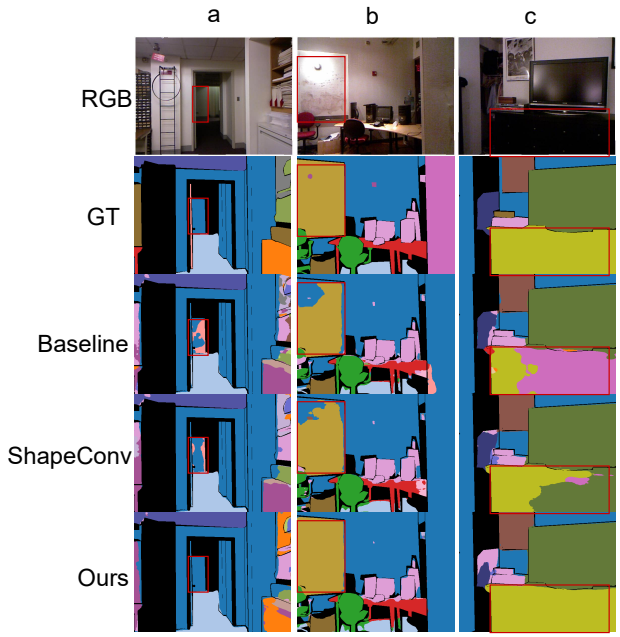

Fig. 3 Visualization results from NYUD-v2 dataset, where each column represents the segmentation results of the same image. The red boxes indicate significant improvements.

Table 1 Performance comparison with the same backbone network on the NYUDv2-13 dataset. Deeplabv3+ is the adopted architecture.

Backbone	Setting	Pixel Acc	Mean Acc	Mean IoU	f.w.IoU
ResNet50	Baseline	79.89	73.03	61.12	67.50
	Baseline*	80.48	73.41	62.08	68.23
	ShapeConv	81.54	75.62	63.24	69.65
	ShapeConv*	81.73	75.89	64.09	70.09
	Ours	**82.14**	**75.89**	**64.12**	**70.43**
	Ours*	**82.51**	**76.12**	**64.84**	**70.90**
ResNet101	Baseline	79.94	73.34	61.34	67.68
	Baseline*	80.79	74.12	62.56	68.31
	ShapeConv	81.72	75.40	63.85	69.90
	ShapeConv*	82.13	76.33	64.49	70.35
	Ours	**82.30**	**76.71**	**64.70**	**70.66**
	Ours*	**82.60**	**77.32**	**65.13**	**71.03**

In contrast, we decompose the weight after using input-dependent attention-rectified convolution kernel to mitigate the impact of decomposition on RGB feature. Unlike dynamic convolution[32], we rectify all dimensions of the convolution kernels and obtain better results with a smaller N value, ensuring that the parameter count remains within a usable range. Figure 2 depicts the overall method architecture. We apply CKA-ShapeConv to all convolutional layers except for the first layer of the CNN architecture.

3. Experiments

3.1 Experimental Setup

To validate the effectiveness of our method, we conduct experiments on two indoor RGB-D benchmarks: NYU-Depth-V2[34] and SUN RGB-D[35]. The NYUDv2 dataset con-

Table 2 Performance comparison with the same backbone network on the NYUDv2-40 dataset. Deeplabv3+ is the adopted architecture.

Backbone	Setting	Pixel Acc	Mean Acc	Mean IoU	f.w.IoU
ResNet50	Baseline	73.75	58.84	46.04	60.13
	Baseline*	74.61	59.29	47.47	60.85
	ShapeConv	74.78	60.74	47.61	61.18
	ShapeConv*	75.35	61.59	49.38	61.71
	Ours	**75.09**	**61.49**	**48.90**	**61.80**
	Ours*	**75.66**	**61.92**	**49.81**	**62.27**
ResNext50_32x4d	Baseline	72.84	57.09	44.87	58.92
	Baseline*	73.48	57.78	45.84	59.39
	ShapeConv	74.46	60.91	47.72	61.06
	ShapeConv*	74.91	61.16	48.44	61.39
	Ours	**75.12**	**61.65**	**48.64**	**61.76**
	Ours*	**75.63**	**61.78**	**49.63**	**62.18**
ResNet101	Baseline	74.49	60.56	47.59	61.14
	Baseline*	75.20	61.05	48.84	61.67
	ShapeConv	75.49	61.30	48.89	62.25
	ShapeConv*	76.05	61.45	49.85	62.64
	Ours	**75.90**	**63.41**	**50.29**	**62.79**
	Ours*	**76.39**	**63.87**	**51.28**	**63.08**
ResNext101_32x8d	Baseline	75.06	61.49	48.50	61.78
	Baseline*	75.55	61.45	49.38	62.06
	ShapeConv	76.05	62.68	50.08	63.01
	ShapeConv*	76.47	**63.26**	50.95	**63.31**
	Ours	**76.17**	**63.21**	**50.53**	**63.13**
	Ours*	**76.49**	63.23	**50.97**	63.22

Table 3 Performance comparison with other methods on NYUDv2-40 dataset.

Method	Backbone	Pixel Acc	Mean Acc	Mean IoU	f.w.IoU
FCN[15]	VGG16	65.4	46.1	34.0	49.5
RAFNet[41]	ResNet50	73.8	60.3	47.5	–
Link-RGBD[38]	ResNet50	**76.8**	59.6	49.5	–
MMANet[40]	ResNet50	–	–	49.6	–
Ours	ResNet50	75.66	**61.92**	**49.81**	**62.27**
NAM[10]	ResNet101	75.0	60.7	47.9	61.5
ShapeConv[8]	ResNet101	75.5	60.7	49.0	61.7
GGED[39]	ResNet101	75.9	62.4	49.4	–
M2.5D[25]	ResNet101	**76.9**	–	50.9	–
SGNet[9]	ResNet101	76.8	63.3	51.1	–
Ours	ResNet101	76.39	**63.87**	**51.28**	**63.08**

sists of 1,449 RGB-D scene images, where 795 images are split for training and 654 images for testing. We adopt two popular settings for this dataset, i.e., 13-class[34] and 40-class[37], where all pixels are labeled with 13 or 40 classes, respectively. SUN RGB-D dataset consists of 10,335 RGB-D indoor images, with 37 class labels assigned to each pixel. Following widely used settings[35], we divide the dataset into a training set with 5,285 images and a test set with 5,050 images. We report results using the same evaluation protocol and metrics as FCN[15], i.e., pixel accuracy (Pixel Acc.), mean accuracy (Mean Acc.), mean region intersection over union (Mean IoU), and frequency weighted intersection over union (f.w.IoU). We employ several popular architectures with different backbones to demonstrate the effectiveness and generalization ability of CKA-ShapeConv. The baseline is generic architectures equipped with vanilla convolutional layers. We rerun the experiments for both the baseline and ShapeConv[8] and obtain new results. For all baseline and ShapeConv, we only replace the vanilla convolutional layers or ShapeConv layers with CKA-ShapeConv layers, while keeping other settings such as batch size, number of training epochs, learning rate, weight decay, and momentum constant. This ensures that the obtained performance improvements are solely due to the application of CKA-ShapeConv, but not other factors. We use ResNet[20] and ResNeXt[21] with pre-trained models on ImageNet[22] as our backbone models in the training phase. All inputs are the concatenation of RGB and HHA images. We adopt single-scale and multi-scale testing strategies during inference. For the latter one, left-right flipped images and six scales are exploited: [0.5, 0.75, 1.0, 1.25, 1.5, 1.75]. The * in tables of this section denotes the multi-scale strategy. We train our model on a single NVIDIA GeForce RTX3090 GPU.

Table 4 Performance comparison with the same backbone network on the SUN RGB-D dataset. Deeplabv3+ is the adopted architecture.

Backbone	Setting	Pixel Acc	Mean Acc	Mean IoU	f.w.IoU
ResNet50	Baseline	80.60	56.24	44.28	69.09
	Baseline*	80.85	56.76	45.44	69.31
	ShapeConv	80.66	56.74	44.34	69.14
	ShapeConv*	81.13	57.24	45.73	69.52
	Ours	**81.02**	**57.09**	**45.24**	**69.78**
	Ours*	**81.34**	**57.68**	**46.32**	**70.13**
ResNet101	Baseline	80.70	56.92	44.58	69.24
	Baseline*	81.18	57.76	45.75	69.74
	ShapeConv	81.14	57.83	45.98	69.89
	ShapeConv*	81.69	58.66	47.23	70.52
	Ours	**81.53**	**58.88**	**46.51**	**70.29**
	Ours*	**81.93**	**59.39**	**47.43**	**70.63**

3.2 Experiments on Different Datasets

NYUDv2 Dataset: We adopt two popular settings for this dataset, i.e., 13-class and 40-class, and show the results of baseline, ShapeConv[8] and our CKA-ShapeConv with different backbones in Table 1 and Table 2, respectively. It can be seen that our CKA-ShapeConv achieve better results than both the baseline and ShapeConv methods. The core idea of ResNeXt is to use a parallel multi-branch structure to enhance the feature extraction capability of the network. It is similar to our approach, but there is some overlap. Therefore, compared to the ResNeXt series networks, our method is more effective on the ResNet series networks and achieves the best results with the ResNet101 backbone. We also compare the performance of our CKA-ShapeConv with several developed methods in Table 3 where our method achieves a competitive performance on NYUDv2-40.

SUN RGB-D Dataset: The baseline, ShapeConv, and our CKA-ShapeConv results with different backbone networks are shown in Table 4. Our CKA-ShapeConv achieves better results than both the baseline and ShapeConv methods. But our method shows more noticeable performance improvement on NYUDv2 dataset compared to SUN RGB-D.

3.3 Experiments on Different Architectures

Our proposed CKA-ShapeConv is a general layer for RGB-D semantic segmentation that can be easily inserted into

Table 5 Performance comparison with different architecture network on the NYUDv2-40 dataset.

Architecture	Backbone	Setting	Pixel Acc	Mean Acc	Mean IoU	f.w.IoU
Deeplabv3+	ResNet50	Baseline	73.75	58.84	46.04	60.13
		ShapeConv	74.78	60.74	47.61	61.18
		Ours	**75.09**	**61.49**	**48.90**	**61.80**
	ResNet101	Baseline	74.49	60.56	47.59	61.14
		ShapeConv	75.49	61.30	48.89	62.25
		Ours	**75.90**	**63.41**	**50.29**	**62.79**
Deeplabv3	ResNet50	Baseline	72.82	57.53	44.74	58.84
		ShapeConv	74.33	59.12	46.90	60.59
		Ours	**74.71**	**60.21**	**48.05**	**61.26**
	ResNet101	Baseline	74.05	59.34	46.36	60.44
		ShapeConv	74.97	61.47	48.48	61.52
		Ours	**75.55**	**62.01**	**49.07**	**62.37**
FPN	ResNet50	Baseline	71.82	55.34	42.55	57.86
		ShapeConv	72.79	56.66	44.44	59.04
		Ours	**73.70**	**59.54**	**46.61**	**60.60**
	ResNet101	Baseline	73.28	57.48	45.11	59.57
		ShapeConv	73.92	58.85	46.08	60.50
		Ours	**74.60**	**60.63**	**47.69**	**61.47**
PSPNet	ResNet50	Baseline	72.08	55.75	43.15	58.21
		ShapeConv	74.14	59.42	46.66	60.37
		Ours	**74.40**	**60.10**	**47.25**	**60.95**
	ResNet101	Baseline	72.34	55.38	43.16	58.43
		ShapeConv	74.76	61.56	48.04	61.25
		Ours	**75.33**	**62.58**	**49.30**	**62.00**

most CNN architectures as a replacement for vanilla convolution. To validate its generalization performance, we evaluate the effectiveness of our method in several representative semantic segmentation architectures: Deeplabv3+[18], Deeplabv3[17], FPN[19], PSPNet[23] with different backbone networks(ResNet-50[20], ResNet-101[20]), and conduct experiments on the NYUDv2-40[37] dataset. The performance is reported in Table 5. We can observe that CKA-ShapeConv brings performance improvements in all settings, demonstrating the generalization ability of our method.

3.4 Visualization

Figure 3 shows the visual comparison results of our CKA-ShapeConv, Baseline, and ShapeConv[8] on the Deeplabv3+[18] architecture with the ResNet101[20] backbone network. Our CKA-ShapeConv extracts more accurate features from the higher-level inputs by leveraging the input-dependent attention. The rectification of multiple convolution kernels in different dimensions effectively enhances the model's capacity and performance, greatly improving the robustness (e.g., in Fig.3(b), the highlighted area in the RGB image would lead a segmentation error in other models, but our model produces the correct result).

3.5 Ablation Study

3.5.1 Convolutional Kernel Number

We conduct ablation experiments to validate the impact of the number of convolution kernels on our method. We train models with different number of convolution kernels and backbone networks. The results are shown in Table 6. From the table, it can be observed that when using *N=2* convolution kernels is the optimal choice. This is because when using *N=1*, the model's capacity does not receive sufficient improvement. On the other hand, the model's gains reach saturation when the number of convolution kernels is too high. Having an excessive number of convolution kernels lead to difficulties in optimization and a significant increase in the parameter count.

Table 6 Performance comparision with different *N* in CKA-ShapeConv on the NYUDv2-40 dataset. The architecture adopted in this table is deeplabv3+ with different backbone.

Backbone	Setting	Pixel Acc	Mean Acc	Mean IoU	f.w.IoU
ResNet50	Baseline	73.75	58.84	46.04	60.13
	ShapeConv	74.78	60.74	47.61	61.18
	N=1	74.73	61.03	47.99	61.30
	N=2	**75.09**	**61.49**	**48.90**	**61.80**
	N=3	75.01	61.39	48.57	61.57
	N=4	74.76	61.34	48.71	61.23
ResNet101	Baseline	74.79	60.56	47.59	61.14
	ShapeConv	75.49	61.30	48.89	62.25
	N=1	75.62	62.53	49.19	62.49
	N=2	**75.90**	**63.41**	**50.29**	**62.79**
	N=3	75.78	62.22	49.47	62.51
	N=4	75.72	62.30	49.96	62.64

Table 7 Investigating the complementarity of four types of attentions in CKA-ShapeConv on the NYUDv2-40 dataset. The architecture adopted in this table is deeplabv3+ with ResNet50 as backbone.

Backbone	S_i	I_i	O_i	K_i	Pixel Acc	Mean Acc	Mean IoU	f.w.IoU
ResNet50	–	–	–	–	73.75	58.84	46.04	60.13
	–	✓	✓	✓	74.91	61.35	48.67	61.56
	✓	–	✓	✓	74.50	60.58	47.71	61.19
	✓	✓	–	✓	74.62	60.33	47.88	61.24
	✓	✓	✓	–	74.73	61.03	47.99	61.30
	✓	✓	✓	✓	**75.09**	**61.49**	**48.90**	**61.80**

Table 8 Performance comparision with different temperature in CKA-ShapeConv on the NYUDv2-40 dataset. The architecture adopted in this table is deeplabv3+ with ResNet50 as backbone.

Backbone	Setting	Pixel Acc	Mean Acc	Mean IoU	f.w.IoU
ResNet50	Baseline	73.75	58.84	46.04	60.13
	ShapeConv	74.78	60.74	47.61	61.18
	τ=10	74.68	61.14	48.13	61.26
	τ=20	**75.09**	**61.49**	**48.90**	**61.80**
	τ=30	74.93	60.68	48.10	61.53

3.5.2 Four sets of Attentions

We conduct ablation experiments to validate the necessity of the four sets of input-dependent convolution kernel attentions proposed in our method, and the results are shown in Table 7. From the table, it can be observed that the inclusion of each attention individually leads to performance improvements. When all four attentions are used together, the best results are achieved.

3.5.3 Temperature value

To validate the impact of temperature value[32] in our

method, we conduct experiments with different temperature value. The results are shown in Table 8, where we obtain the best results with a temperature value of 20. A too small temperature value can lead to significant weight differences in the early stages of training, causing optimization to focus only on a small subset of convolution kernels and resulting in slow convergence. On the other hand, a too large temperature value can lead to overly uniform attention, making it difficult for the convolution kernels to receive effective training.

4. Conclusion

In this paper, we propose CKA-ShapeConv that is a plug-and-play convolutional layer. Instead of using a single convolution kernel, we aggregate *N* parallel convolution kernels based on input-dependent attention, which ensure that the aggregated convolution kernel is more capable of catching semantic information from the input feature map, reducing interference between RGB and depth features. Then the aggregated convolution kernel is decomposed into two components: base and shape, two new learnable weights are introduced to cooperate with them independently, and finally a convolution is applied on the re-weighted combination of these two components. These two components can capture semantic and shape information of regions effectively, respectively. CKA-ShapeConv increases model capacity and reduces distractive information, while addressing the issue of segmentation errors caused by the positions of different classes in the depth map. CKA-ShapeConv only requires a small amount of computation to calculate four sets of attention weights and perform one combination of the *N* kernels. We conduct extensive experiments on two indoor RGB-D semantic segmentation benchmarks and show that the proposed CKA-ShapeConv layer can improve the performance of backbone models effectively. The input-dependent attention effectively reduce the mutual interference between RGB and depth features caused by their concatenation. However, RGB and depth information are inherently different from each other. We recommend incorporating this approach to double-stream networks and design a suitable fusion module to integrate RGB and depth information when pursing performance.

References

[1] N. Mukojima, M. Yasugi, Y. Mizutani, T. Yasui, H. Yamamoto, "Deep-learning-assisted single-pixel imaging for gesture recognition in consideration of privacy," IEICE Trans.Electron, vol.E105-C, pp.79-85, 2022.

[2] S. Sedukhin, Y. Tomioka, and K. Yamamoto, "In search of the performance- and energy-efficient CNN accelerators," IEICE Trans.Electron, vol.E105-C, no.6, pp.209-221, 2022.

[3] D. A. Ando, Y. Kase, T. Nishimura, T. Sato, T. Ohgane, Y. Ogawa, J. Hagiwara, "Deep neural networks based end-to-end DOA estimation system," IEICE Trans. Commun, vol.E106-B, no.12, pp.1350-1362, 2023.

[4] W. Zhou, J. S. Berrio, S. Worrall, and E. Nebot, "Automated evaluation of semantic segmentation robustness for autonomous driving," IEEE TITS, vol.21, no.5, pp.1951–1963, 2020.

[5] K. Yang, X. Hu, Y . Fang, K. Wang, and R. Stiefelhagen, "Omnisupervised omnidirectional semantic segmentation," IEEE TITS, vol.23, no.2, pp.1184–1199, 2022.

[6] L. Sun, K. Yang, X. Hu, W. Hu, and K. Wang, "Real-time fusion network for RGB-D semantic segmentation incorporating unexpected obstacle detection for road-driving images," IEEE RA-L, vol.5, no.4, pp.5558–5565, 2020.

[7] J. Zhang, K. Yang, A. Constantinescu, K. Peng, K. Müller, and R. Stiefelhagen, "Trans4Trans: Efficient transformer for transparent object and semantic scene segmentation in real-world navigation assistance," IEEE TITS, vol.23, no.10, pp.19 173–19186, 2022.

[8] J. Cao, H. Leng, D. Lischinski, D. Cohen-Or, C. Tu and Y. Li, "ShapeConv: shape-aware convolutional layer for indoor RGB-D semantic segmentation," ICCV, pp.7068-7077, 2021.

[9] L.-Z. Chen, Z. Lin, Z. Wang, Y.-L. Yang, and M.-M. Cheng, "Spatial information guided convolution for real-time RGBD semantic segmentation," IEEE TIP, vol.30, pp.2313–2324, 2021.

[10] Y. Zheng, Y. Xu, S. Qiu, W. Li, G. Zhong and M. Sarem, "A novel semantic segmentation algorithm for RGB-D images based on non-symmetry and anti-packing pattern representation model," IEEE Access, vol.11, pp.36290-36299, 2023.

[11] D. Seichter, M. Köhler, B. Lewandowski, T. Wengefeld and H. -M. Gross, "Efficient RGB-D semantic segmentation for indoor scene analysis," ICRA, pp.13525-13531, 2021.

[12] W. Zhou, E. Yang, J. Lei, and L. Yu, "Frnet: Feature reconstruction network for rgb-d indoor scene parsing," IEEE JSTSP, vol.16, no.4, pp.677-687, 2022.

[13] W. Zhou, E. Yang, J. Lei, J. Wan, and L. Yu, "Pgdenet: Progressive guided fusion and depth enhancement network for rgb-d indoor scene parsing," IEEE TMM, vol.25, pp.3483-3494, 2023.

[14] J. Zhang, H. Liu, K. Yang, X. Hu, R. Liu, and R. Stiefelhagen, "Cmx: Cross-modal fusion for rgb-x semantic segmentation with transformers," IEEE TITS, vol.24, pp.14679-14694, 2023.

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," CVPR, pp.3431–3440, 2015.

[16] L.-C Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," IEEE TPAMI, vol.40, pp.834–848, 2017.

[17] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," arXiv preprint arXiv:1706.05587, 2017.

[18] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-Decoder with atrous separable convolution for semantic image segmentation," ECCV, pp.801–818, 2018.

[19] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," CVPR, pp.2117–2125, 2017.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," CVPR, pp.770–778, 2016.

[21] S. Xie, R. Girshick, P. Dollár, Z. Tu,and K. He, "Aggregated residual transformations for deep neural networks," CVPR, pp.1492–1500, 2017.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, "ImageNet large scale visual recognition challenge," IJCV, vol.115, pp.211–252, 2015.

[23] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," CVPR, pp.2881–2890, 2017.

[24] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," NeurIPS, 2021.

[25] Y. Xing, J. Wang, and G. Zeng, "Malleable 2.5D convolution: learning receptive fields along the depth-axis for RGB-D scene parsing," ArXiv abs/2007.09365, 2020.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention is all you need,"

NIPS, 2017.
[27] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation networks," CVPR, pp.7132-7141, 2018.
[28] H. Zhou, L. Qi, H. Huang, X. Yang, Z. Wan, X. Wen, "CANet: Co-attention network for RGB-D semantic segmentation," Pattern Recognition, vol.124, 2021.
[29] G. Li and Q. Fang and L. Zha, X. Gao, N. Zhen, "HAM: Hybrid attention module in deep convolutional neural networks for image classification," Pattern Recognition, vol.129, 2022.
[30] X. Jia, B. D. Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," NIPS, pp.667–675, 2016.
[31] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, "Condconv: conditionally parameterized convolutions for efficient inference," NeurIPS, 2019.
[32] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan and Z. Liu, "Dynamic convolution: attention over convolution kernels," CVPR, pp.11027-11036, 2020.
[33] J. Hou, Z. Guo, Y. Wu, W. Diao, T. Xu, "BSNet: Dynamic hybrid gradient convolution based boundary-sensitive network for remote sensing image segmentation," IEEE TGRS, vol.60, pp.1-22, 2022.
[34] N. Silberman, D. Hoiem, P. Kohli,and R. Fergus, "Indoor segmentation and support inference from RGBD images," ECCV, 2012.
[35] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: a RGB-D scene understanding benchmark suite," CVPR, pp.567–576, 2015.
[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," NIPS, pp.84-90, 2012.
[37] S. Gupta, P. Arbelaez, and J. Malik, "Perceptual organization and recognition of indoor scenes from RGB-D images," CVPR, pp.564–571, 2013.
[38] P. Wu, R. Guo, X. Tong, S. Su, Z. Zuo, B. Sun, J. Wei, "Link-RGBD: Cross-guided feature fusion network for RGBD semantic segmentation," IEEE Sensor Journal, vol.22, pp.24161-24175, 2022.
[39] W. Zou, Y. Peng, Z. Zhang, S. Tian, X. Li, "RGB-D gate-guided edge distillation for indoor semantic segmentation," MTA, vol.81, pp.35815-35830, 2022.
[40] W. Shicai, Y. Luo and C. Luo, "MMANet: margin-aware distillation and modality-aware regularization for incomplete multimodal learning," CVPR, pp.20039-20049, 2023.
[41] X. Yan, S. Hou, K. Awudu, W. Jia, "RAFNet: RGB-D attention feature fusion network for indoor semantic segmentation," Displays, vol.70, pp.102082, 2021.

Kun Zhou received the B.S. degree in electronic information engineering from Zhejiang Normal University, Jinhua, China, in 2022. He is currently pursuing the M.S. degree at Zhejiang Normal University. His research interests include computer vision and pattern recognition.

Zejun Zhang received the B.S. and M.S. degrees in computer science from Guizhou University, Guiyang, China, in 2007 and 2010, respectively, and the Ph.D. degree in electronic engineering from Xidian University, Xi'an, China, in 2014. He is currently a Lecturer with the College of Physics and Electronic Information Engineering, Zhejiang Normal University, Jinhua, China. His current research interests include computer vision and pattern recognition.

Xu Tang received the B.S. degree from Dezhou University, Dezhou, China, in 2021. He is currently pursuing the M.S. degree at Zhejiang Normal University. His research interests include computer vision and pattern recognition.

Wen Xu received the B.S. degree from Xihua University, Chengdu, China, in 2022. She is currently pursuing the M.S. degree at Zhejiang Normal University. Her research interest is computer vision.

Jianxiao Xie received the B.S. and M.S. degree in electronic information engineering and Detection Technology and Automation from the Nanchang Hangkong University, Nanchang, China in 2013, and 2016, respectively, and the Ph.D. degrees in Information and Communication Engineering from Beijing University of Posts and Telecommunications, Beijing, China in 2020 He is now a lecturer in the College of Physics and Electronic Information Engineering, Zhejiang Normal University. His research interests include wireless mobile networks and wireless communications.

Changbing Tang received his Ph.D. from the Department of Electronic Engineering, Fudan University, Shanghai, China in2014. He received his B.S. and M.S. degrees in mathematics and applied mathematics from Zhejiang Normal University, Jinhua, China, in 2004 and 2007, respectively. Dr. Tang was the recipient of the Academic New Artist Doctoral Post Graduate from the Ministry of Education of China in 2012 and the recipient of the Academician Pairing Training Program for Young Talents of Zhejiang Province in 2019. He is currently an Associate Professor with the College of Physics and Electronic Information Engineering, Zhejiang Normal University, Jinhua, China. His research interests include game theory and its applications, intelligent optimization and decision-making.