DCNet: diffusion convolutional networks for semantic image segmentation

Lan Yang

School of Software,
Quanzhou University of Information Engineering,
Quanzhou, Fujian, China
Email: 502968883@qq.com

Zhixiong Jiang*

School of Business,
Shanghai Dianji University,
Shanghai, China
Email: jiangzx@sdju.edu.cn
*Corresponding author

Hongbo Zhou and Jun Guo

School of Software,
Quanzhou University of Information Engineering,
Quanzhou, Fujian, China
Email: zhb591111@163.com
Email: guojun20121203@163.com

Abstract: Semantic image segmentation makes a pixel-level classification play an essential role in scene understanding. Recently, most approaches exploit deep learning neural networks, especially convolutional neural networks (CNNs), to tackle the image segmentation challenge. Common issues of these CNN-based methods are the loss of spatial features during learning representations and the limited capacity for capturing contextual information in a large receptive field. This paper proposes a diffusion convolutional network (DCNet) to combine the CNN and graph convolutional neural network (GCNN) for semantic image segmentation. In the proposed model, diffusion convolution is formulated as a graph convolutional layer to aggregate structural and contextual information without losing spatial features. The final segmentation results on the PASCAL VOC 2012 and Cityscapes datasets show better performance than baseline approaches and can be competitive with state-of-the-art methods.

Keywords: deep learning; semantic image segmentation; graph convolutional neural networks; GCNNs.

Reference to this paper should be made as follows: Yang, L., Jiang, Z., Zhou, H. and Guo, J. (2021) 'DCNet: diffusion convolutional networks for semantic image segmentation', *Int. J. Embedded Systems*, Vol. 14, No. 3, pp.300–311.

Biographical notes: Lan Yang received his MS in Computer Science and Technology from Huaqiao University, China, in 2014. He is currently a Lecturer in the School of Software, Quanzhou University of Information Engineering, China. His research interests include data mining, big data technologies, and software engineering applications.

Zhixiong Jiang received his PhD in Sociology from Wuhan University in 2014. He is currently a Lecturer in the School of Business at the Shanghai DianJi University.

Hongbo Zhou received his MS in Computer Science and Technology from Beijing Institute of Technology in 1993. He is a Professor in the School of Software, Quanzhou University of Information Engineering, China. His research interests include dependable systems/networks, network security, hardware security, and IP protection.

Jun Guo received her MS from Xiamen University Software School in 2011. She is currently a Lecturer in the School of Software, Quanzhou University of Information Engineering, China. Her research interests include software engineering, blockchain, and information system.

1 Introduction

Semantic image segmentation is a fundamental task in many visual understanding works (Liang et al., 2020; Jiang et al., 2020). It tries to divide images into multiple semantic objects, which can be applied in a broad range of applications such as healthcare (Huang et al., 2009), autonomous driving (Grigorescu et al., 2020), and video surveillance (Ammar et al., 2020). Semantic image segmentation is also a classification task of pixels with semantic labels that performs a pixel-level classification for all pixels with a set of predefined labels such as human, car, bus, bird, and dog, etc. This dense classification involves the local features and contextual information to assign correct labels to pixels in the receptive field and the spatial information to determine the clear boundaries.

Over the last few years, AI (Liang et al., 2019), IoT (Liang et al., 2011; Li et al., 2018b, 2020b), big data (Liang et al., 2016a; Li et al., 2020a) technology, and deep learning have been widely used in various fields of computer vision. Deep learning models, especially convolutional neural networks (CNNs), have yielded a new generation of semantic image segmentation. Besides, CNN has demonstrated its powerful ability to learning feature representations for making the classification. However, CNN has its intrinsic problems when applied in semantic image segmentation.

Although CNNs are good at extracting features of objects for image classification, CNNs lose spatial features such as dimensions, locations, structures, and boundaries that are important for semantic segmentation, which leads to misclassification and blurred boundaries. The reason for this problem is that CNNs apply multiple pooling operators to introduce invariance, expand the receptive field, and reduce the number of parameters. This operator is good for classification that is required to be invariant to various transformations. But for the localisation, semantic segmentation approaches should be sensitive to transformations to precisely locate pixels for all categories. There are many studies trying to solve this issue. Fully convolutional networks (FCNs) (Long et al., 2015) employ the deconvolution layer to increase the resolution of feature maps and make a dense prediction. SegNet (Badrinarayanan et al., 2017) reuses pooling positions that are recorded in max-pooling layers to upsample feature maps. Some studies tend to construct an effective architecture for extracting features and obtaining spatial information such as DenseAspp (Yang et al., 2018), DeconvNet (Noh et al., 2015) and U-Net (Ronneberger et al., 2015). But these methods always focus on utilising hierarchical feature maps efficiently and cannot extract the long-range contextual features flexibly.

Another problem of CNNs for semantic image segmentation is that the traditional CNNs cannot efficiently learn multi-scale feature maps for different objects of various sizes. Due to the fixed size of the receptive field of CNNs, which is determined by its grid-like local connections in the kernel and pooling operators, irregular dependencies among pixels and global contextual information is lost. This limitation may cause that CNNs learn an imprecise structural representation and focus on local features, which results in misclassification when few pixels change.

To obtain global contextual information, DeepLabv1 (Chen et al., 2014) uses a fully connected conditional random field (CRF) to learn the overall relationships between pixels, which refine the segmentation result by spatial dependencies. Zheng et al. (2015) integrated CRFs with the deep CNNs to automatically learn spatial relationships among all pixels in the input image. There are other approaches that apply the attention mechanism to model global dependencies. Hu et al. (2020) proposed fast spatial attention to obtain spatial context. Attention complementary network (ACNet) (Hu et al., 2019) uses multiple attention complementary modules to extract features from RGB and depth branches.

In this paper, the diffusion convolutional network (DCNet) is proposed, which combines CNNs and graph convolutional neural networks (GCNNs). The DCNet employs an encoder-decoder structure. In the encoder, hierarchical feature maps are extracted by inputting a 2D image through stacked convolutional layers and multiple pooling layers. These feature maps are then fed into the decoder to restore to the original resolution of the input image and output segmentation results. To represent the input image in a new coordinate space, a GCNN is then inserted into the network, which builds the graph from the input image. Nodes in the graph indicate pixels in the image, while edges in the graph show relationships between pixels. Through this graph, DCNet embeds structure information into the learning of feature representation.

The diffusion convolution is applied in the graph neural network to extract features. The selected low-level feature maps from the encoder and semantic feature maps from the decoder are input into a GCNN. The features of every node are updated by gathering information from nodes in its neighbourhood, which maintains the spatial information and capture structure information. With the information exchanges along edges, contextual features can be extracted in a large receptive field, which is expanded as this process runs. Therefore, DCNet obtains a flexible receptive field by combining CNNs with graph neural networks.

Moreover, the final feature maps containing rich structure and contextual information are used to make the segmentation results. The experiments show outstanding performance on the PASCAL VOC 2012 (Everingham et al., 2015) and Cityscapes (Cordts et al., 2016) datasets. The contribution of this work is summarised as follows:

- A novel model, DCNet, is proposed for semantic image segmentation, which combines CNNs and GCNNs to capture local and global contextual features.
- The proposed DCNet introduces diffusion convolution as graph convolution in semantic image segmentation, which aggregates global contextual information and enlarges the size of the receptive field without missing spatial features.

Figure 1 The architecture of the proposed model (see online version for colours)

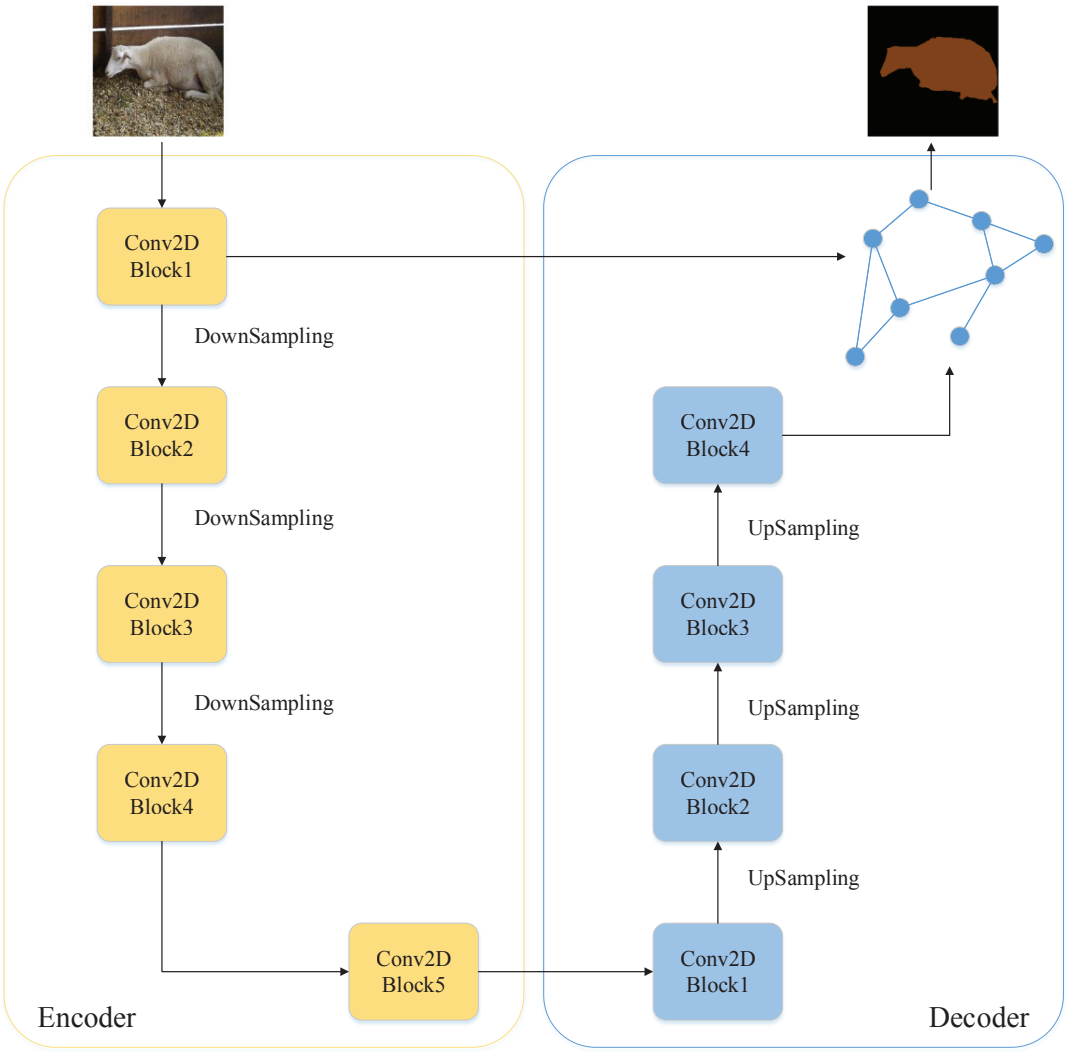
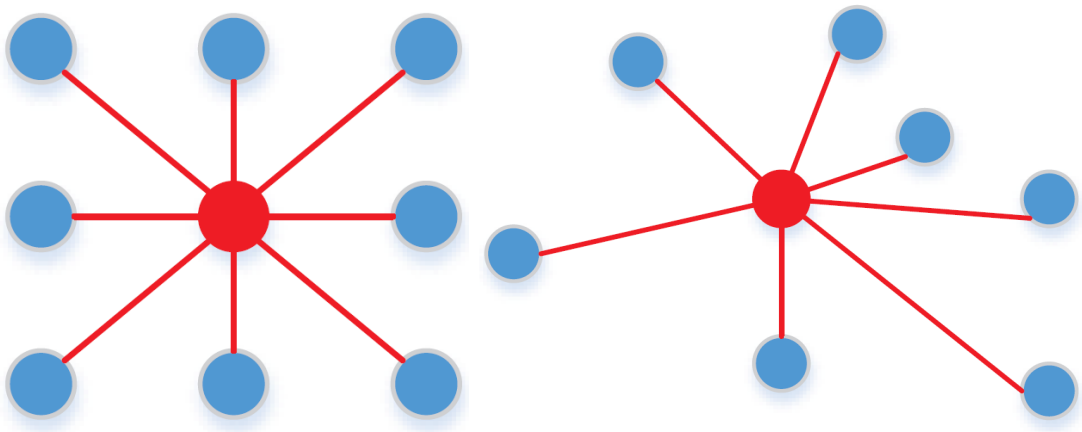


Figure 2 2D convolution and graph convolution, (a) 2D convolution (b) graph convolution (see online version for colours)



- Experiments are conducted on the PASCAL VOC 2012 and Cityscapes datasets. The results have demonstrated the proposed model yields outstanding segmentation performance.

This paper is organised as follows. Section 1 introduces the semantic image segmentation, the problems of CNNs, and the contributions of the paper. Section 2 presents the related works about semantic image segmentation and graph neural networks. Section 3 demonstrates the architecture and methods used in DCNet. Section 4 shows the details about experiments. Section 5 provides a conclusion of the paper.

2 Related work

2.1 Semantic segmentation based on deep learning

With the success of deep learning networks, FCNs (Long et al., 2015) have become the breakthrough work of applying CNNs in semantic image segmentation. Since then, many deep learning models have been applied to semantic segmentation. However, to recognise and classify deformed objects, deep CNNs stack multiple convolutional and pooling layers to expand the receptive field and reduce the resolution of feature maps, which results in losing spatial and contextual information.

To obtain a large receptive field and capture multi-scale contextual information, the DeepLab families (Chen et al., 2014, 2017, 2018) have employed atrous convolution (also called dilated convolution) and Atrous Spatial Pyramid Pooling (ASPP) to increase the size of the receptive field and capture contextual features without losing spatial information. ASPP employs multiple dilated convolutions with different dilation rates to extract multi-scale features in various sizes of the receptive field. To get a global view of feature maps, a fully connected CRF is applied in DeepLabv1 (Chen et al., 2014) to model the long-range spatial dependencies between pixels in the input image. To reuse low-level features, DeepLabv3 (Chen et al., 2017) introduces image-level features by using global average pooling on the feature maps, while DeepLabv3+ (Chen et al., 2018) adopts an encoder-decoder structure. Without using dilated convolution, PSPNet (Zhao et al., 2017) uses multiple parallel pooling layers that have different filter sizes to aggregate multi-scale contextual features. These contextual features are then upsampled to the size of the input image and cascaded to make the final pixel-level classification. Peng et al. (2017) proposed a global convolutional network that has large kernel sizes to solve the problem of classification and localisation for the semantic image segmentation. The method also applies a residual-based boundary refinement to get a sharp object boundary. Contextual aggregating network (CAN) (Cong et al., 2019) gathers all middle-level features by applying a convolutional layer with various kernel sizes to extract contextual information for final image segmentation. APCNet (He et al., 2019) proposes an adaptive context module (ACM) to extract context features by learning the local affinity coefficients for sub-region of various sizes. CDN (Fu et al., 2020) designs a channel contextual module for obtaining image-level semantic information and a spatial contextual module to capture patch-level semantic context.

The local connections of the convolution operator can represent the relationships of pixels in the receptive field and disregard wider spatial dependencies. To capture long-range spatial dependencies among pixels, some methods utilise CRFs. For example, Zheng et al. (2015) run a fully connected CRF as a recurrent neural network, and all parameters are trained in an end-to-end way. Colovic et al. (2017) combined CNNs with a local connected CRF to encourage consistency within the segmentation results and clarify the boundaries. Some works use a recurrent neural network to extract spatial dependencies. Visin et al. (2016) employed CNNs to extract local generic features and stacked multiple ReNet layers to learn spatial dependencies all over the feature map by sweeping the feature map horizontally and vertically. Byeon et al. (2015) designed a 2D LSTM network to extract local and global contextual information and learned spatial dependencies of labels. Other approaches exploit attention mechanism to embed global contextual information into local features. Li et al. (2018a) proposed a feature pyramid attention module to combine global information with high-level spatial features and a global attention upsample module to reuse low-level features by the guidance from high-level features. Chen et al. (2016) input multiple images with different sizes into a shared CNN and used attention mechanism to weight these multi-scale features.

In recent years, graph neural networks have shown their ability to exploit structural information and have been applied in drug discovery, disease prediction, and citation networks. For semantic image segmentation, graph LSTM (Liang et al., 2016b) constructs the graph over a set of arbitrary-shaped superpixels and connects the spatial neighbours with undirected edges. Then it uses the proposed confidence-driven scheme to update the hidden and memory states of the nodes. Zhang et al. (2019) proposed a dual graph convolutional network to model the global context of the input feature by using two orthogonal graphs. Liu et al. (2020) presented a self-constructing graph module to automatically learn a long-range dependency graph from the image and extracted contextual features to improve the segmentation performance. Lu et al. (2019) extended the grid feature maps to graph structure data and reformulated the image segmentation task as the node classification task by applying a graph convolutional network. Qi et al. (2017) proposed a 3D graph neural network to exchange and update feature representations of each node by a graph recurrent neural network. Wolterink et al. (2019) aggregated local image features of each node and features of its neighbours to determine the spatial location of vertices for segmenting the coronary artery lumen. Soberanis et al. (2020) used graph convolutional network and uncertainty analysis to refine segmentation results by transferring the segmentation refinement to a semi-supervised node classification task that is solved by graph convolutional network.

Figure 3 Examples that semantic segmentation results on PASCAL VOC 2012 (see online version for colours)

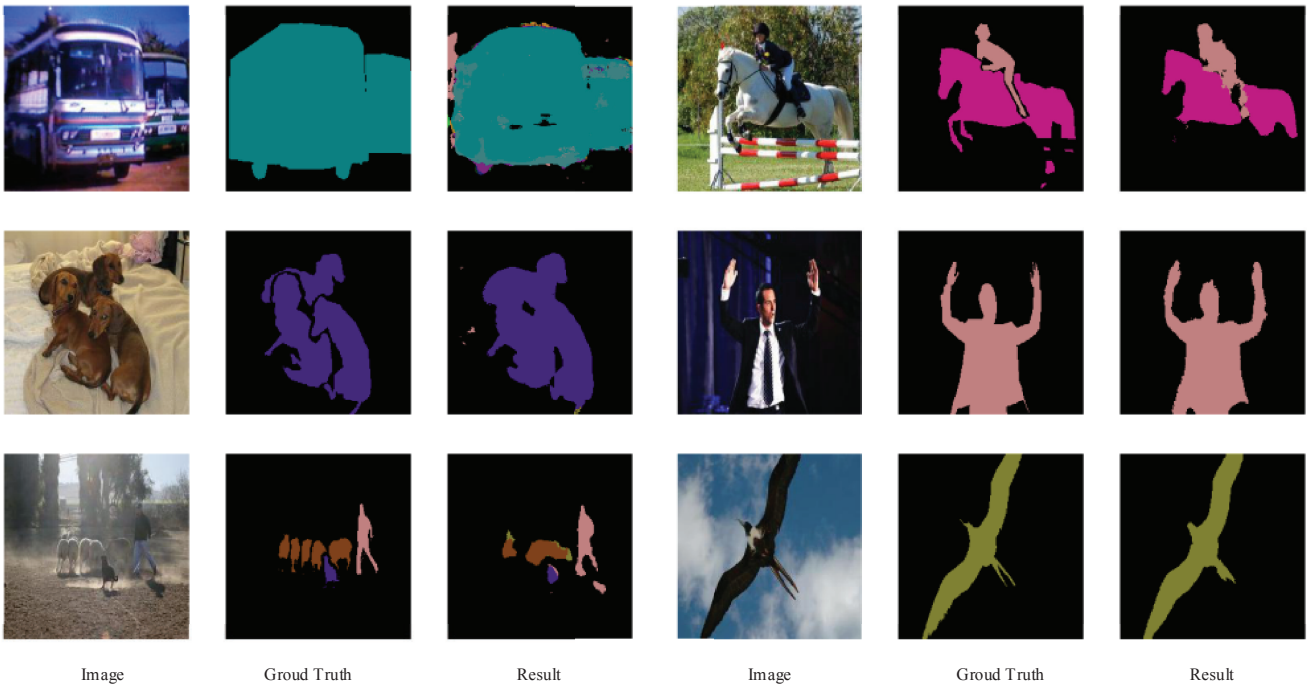
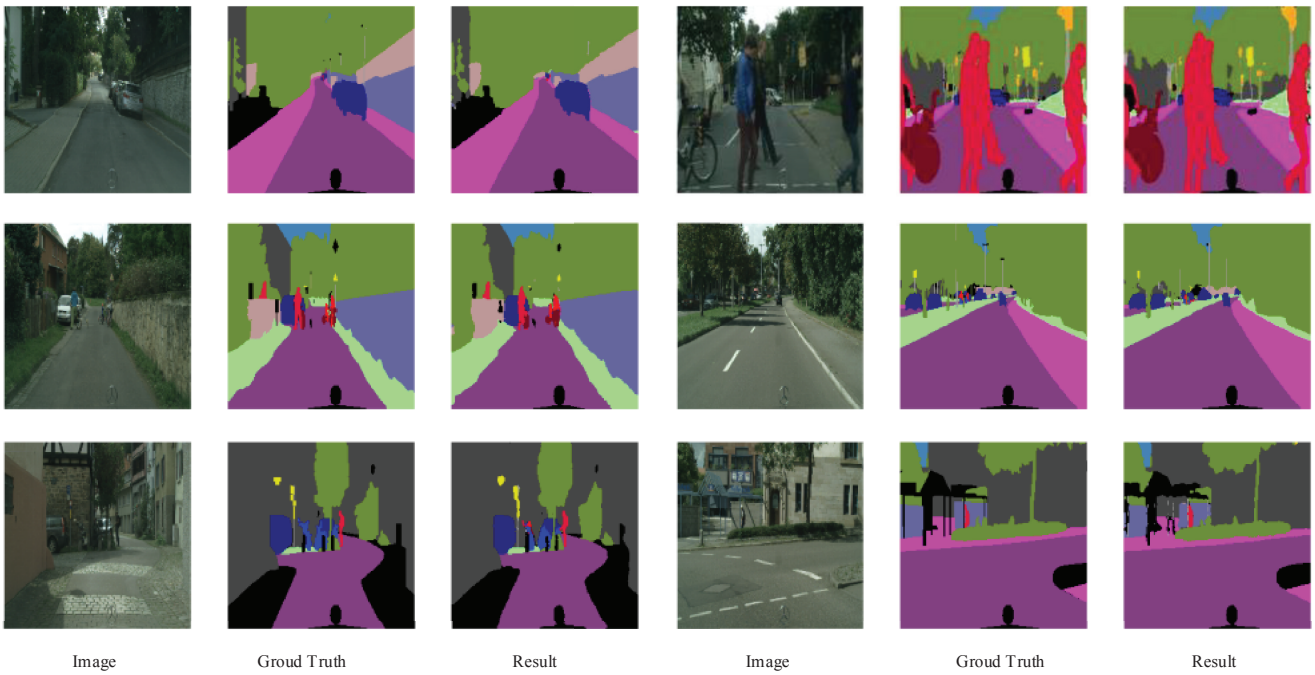


Figure 4 Examples of semantic segmentation results on the Cityscapes dataset (see online version for colours)



Different from these methods above, we propose a novel network, DCNet, which combines CNNs and GCNNs to capture local and structure information by introducing the diffusion convolution into semantic image segmentation. By mapping a 2D image to a 5D coordinate space, a directed graph is constructed by connecting K nearest neighbours of each node. Then the feature maps selected from CNNs are input into a graph convolutional network to aggregate contextual information in a broader receptive field and structural information. These features can produce better representations for the following pixel-level classification.

3 Diffusion convolutional network

In this section, the overall network is introduced. The work process of the proposed graph neural network is also demonstrated, including the construction of the graph and the diffusion convolution layer.

3.1 Architecture

As shown in Figure 1, an encoder-decoder structure is applied in the proposed model. The encoder contains five convolution blocks, which have three convolutional layers in each block with a 3 × 3 filter size. Multi-level feature maps are extracted in the encoder, and the dimensions of the feature maps are decreased gradually by using max-pooling operators. The last two max-pooling layers are set to keep the sizes of the feature maps and capture more contextual information. During this process, the receptive field of the network is also expanded. Correspondingly, the decoder has four convolution blocks, which also have three convolutional layers with 3 × 3, 1 × 1, and 3 × 3 filter sizes. Unpooling operators are utilised in the decoder for upsampling to reconstruct the spatial features and increase the dimensions, which use pooling indexes computed in the max-pooling step of the corresponding layers in the encoder to apply nonlinear upsampling. A softmax layer is stacked at the end of the decoder to produce a probability result. In the decoder, a graph neural network is employed to capture structural and contextual features. And the final label for each pixel is the label with the maximum probability. Every convolutional layer in the network is equipped with a ReLU activation layer and batch normalisation layer.

The input image passes through convolutional blocks of the encoder to produce hierarchical feature maps, including low-level features that contain contextual and detailed information and high-level features that have global and coarse information. However, spatial and contextual features are lost, which is important for segmentation. Then the extracted feature maps are input into the decoder to recover the original resolution and generate a dense prediction. During this process, unpooling layers put values in the feature maps to the maximum location recorded by the corresponding max-pooling layers, which precisely reconstruct some spatial features and output sparse feature maps. These sparse feature maps are input into the following convolution block to make dense feature maps. To exploit the structural and spatial dependencies in the feature maps, a graph neural network is inserted in the decoder, which aggregates information for each node by using the graph constructed from the original input image and the feature maps extracted from the encoder. The final feature maps of the decoder are input into the softmax layer to generate probability maps that contain probabilities for every label of each pixel.

3.2 Spatial-based graph convolution

As is used by most CNNs, the 2D convolution takes an ordered grid-like image as input. Each pixel in the input is treated as a node that is connected to its neighbours in the fixed receptive field determined by the convolution filter. To compute a new value, the 2D convolution obtains a weighted average of features of the node and its neighbours. However, the graph convolution takes hidden representations of unordered nodes as input. A graph is built by utilising relationships between nodes. To update the hidden representation of each node, the graph convolution aggregates information of the node and its neighbours in various shapes of the receptive field.

The spatial-based graph shares some properties with the standard convolution operator that performs convolution on a node and its neighbours defined by spatial relationships. An input image can be regarded as a graph with isolated nodes representing all pixels. As shown in Figure 2(a), the red node is connected to its surrounding nodes in a 3 × 3 kernel. In the receptive field, the neighbours of the red node are eight pixels around it. The locations of these eight nodes imply an ordering of neighbours of the red node. To perform the convolution, a 3 × 3 filter is applied on the graph to compute the weighted average of the red node and its neighbours over each channel. Therefore, the trainable wights can be shared by each node in the same channel for the fixed shape and order of its neighbours.

Correspondingly, the spatial-based graph convolution updates each node's hidden representation by aggregating the representation of the red node and its neighbours, as illustrated in Figure 1(b). Different from the standard convolution, the neighbours of each node in spatial-based graph convolution are unordered and variable in size.

3.3 Graph construction

The graph is constructed on pixels that are obtained through a downsampled image by using a pooling operator. Note that the input feature maps have been downsampled. So, in order to use the original image for graph construction in the proposed graph neural network, the input image need to be downsampled into the size of the feature map.

The graph $\mathcal{G}$ for the input image is constructed by connecting the neighbour nodes $ne[v]$ of each node v via the edges $\mathcal{E}$. The graph node v_i denotes a pixel and the graph edge $\mathcal{E}_{ij}$ expresses the connection between node v_i and v_j. The features of node v_i are represented by $f_i \in R^d$ with d dimensions. In this paper, the graph is constructed by using K nearest neighbours algorithm to select the top K nearest node as the neighbours of each node. To calculate the distance between two nodes, a feature vector with five dimensions is designed for each node. This feature vector contains (r, g, b) values of each node that represents the appearance of the node and (x, y) coordinates of the node in the input image. So, the feature f_i can be expressed as (r, g, b, x, y). By using the K nearest neighbours algorithm, the directed graph $\mathcal{G}$ is constructed and an weighted adjacency matrix W is generated.

Figure 5 Comparison results on the PASCAL VOC 2012 dataset (see online version for colours)

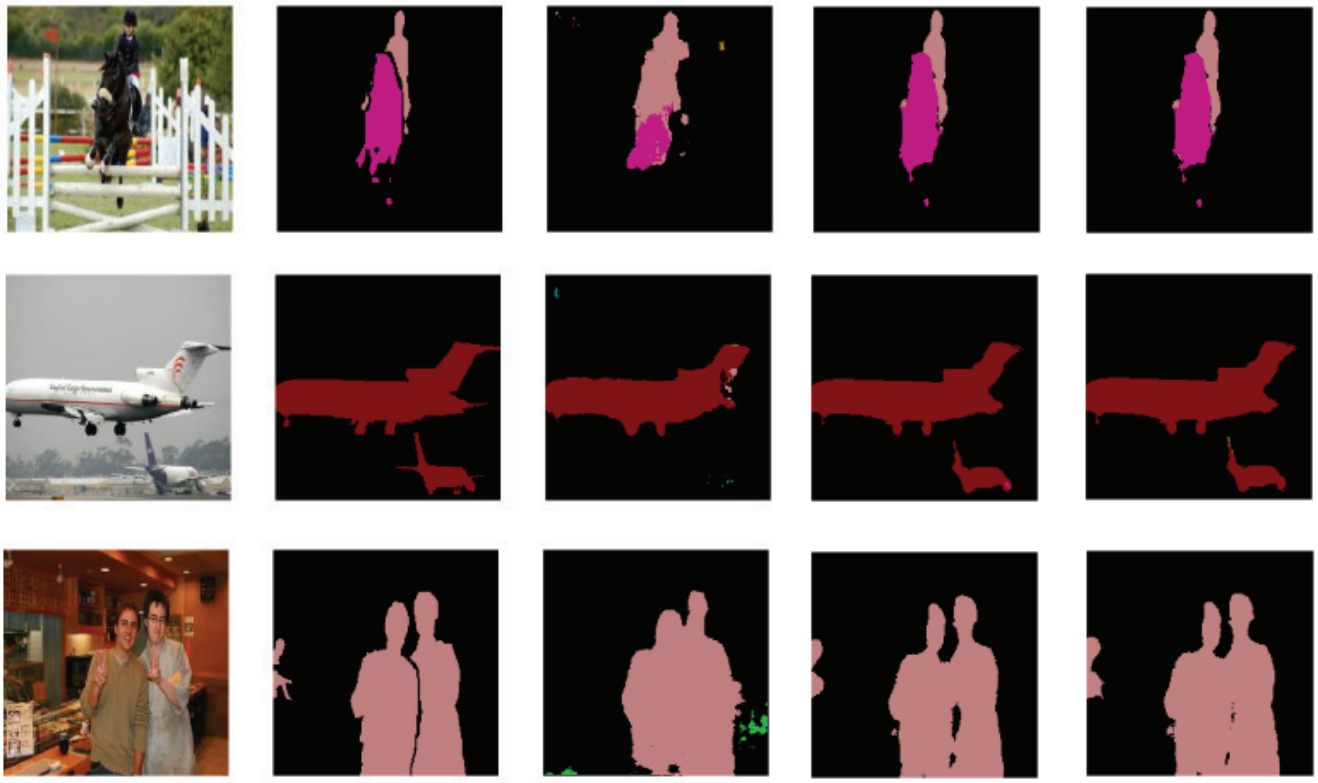


Table 1 Detailed configuration of the proposed encoder network

Blocks	*Layers*	*Output size*	*Kernel*	*Stride*	*Pad*
Block 1	Conv2D	64 × 224 × 224	3 × 3	1	1
	Conv2D	64 × 224 × 224	3 × 3	1	1
	Max-pooling	64 × 112 × 112	2 × 2	2	0
Block 2	Conv2D	128 × 112 × 112	3 × 3	1	1
	Conv2D	128 × 112 × 112	3 × 3	1	1
	Max-pooling	128 × 56 × 56	2 × 2	2	0
Block 3	Conv2D	256 × 56 × 56	3 × 3	1	1
	Conv2D	256 × 56 × 56	3 × 3	1	1
	Conv2D	256 × 56 × 56	3 × 3	1	1
	Max-pooling	256 × 28 × 28	2 × 2	2	0
Block 4	Conv2D	512 × 28 × 28	3 × 3	1	1
	Conv2D	512 × 28 × 28	3 × 3	1	1
	Conv2D	512 × 28 × 28	3 × 3	1	1
	Max-pooling	512 × 28 × 28	3 × 3	1	1
Block 5	Conv2D	512 × 28 × 28	3 × 3	1	1
	Conv2D	512 × 28 × 28	3 × 3	1	1
	Conv2D	512 × 28 × 28	3 × 3	1	1
	Max-pooling	512 × 28 × 28	3 × 3	1	1

3.4 Diffusion convolution layer

The diffusion convolution is selected as graph convolution to capture the spatial dependencies among the features. The directed graph is expressed as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$, where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of edges and $W \in R^{N\times N}$ is the adjacency matrix. Let $X \in R^{N\times P}$ be the initial features of the nodes in the graph, where P is the dimension of each feature vector.

The diffusion process models the spatial dependencies as a random walk on the constructed graph $\mathcal{G}$ with a restart probability $\alpha \in [0, 1]$ and a transition matrix $D_O^{-1}W$. $D_O = diag(W1)$ is the out-degree diagonal matrix, and 1 is a all one vector. When the diffusion process runs enough time steps, this Markov process converges to a stationary distribution $P \in R^{N\times N}$, as illustrated as follows:

$$P = \sum_{m=0}^{\infty} \alpha(1-\alpha)^m (D_O^{-1}W)^m \tag{1}$$

where m denotes the time steps. In the experiment, a finite M time step is used to simulate the diffusion process. In this paper, the diffusion process is defined as:

$$X_{:,p} f(\theta) = \sum_{m=0}^{M-1} (\theta_{k,1}(D_O^{-1}W)^m + \theta_{k,2}(D_I^{-1}W^T)^m) X_{:,p} \tag{2}$$

where $p \in \{1, ..., P\}$ are the dimensions of the feature, $\theta \in R^{M\times 2}$ are the trainable weights, D_I^{-1} is the in-degree matrix, and $D_O^{-1}W$, $D_I^{-1}W^T$ denote the transition matrix and the reverse transition matrix, respectively. Therefore, the diffusion convolution layer utilised in this paper can be expressed as:

$$Z_{:,q} = \sigma\left(\sum_{p=1}^{P} X_{:,p} f(\Theta_{q,p,:,:})\right) \tag{3}$$

where $Z \in R^{N\times Q}$ are the output features, $\Theta \in R^{Q\times P\times M\times 2}$ are the parameter vectors and Q is the number of dimensions of the output. The function σ can be an activation function.

In this paper, the graph $\mathcal{G}$ is constructed from the downsampled image. Then, X is initialised by the reshaped feature maps that are extracted in the encoder. After K time steps, the diffusion convolution layer output the feature maps Z. Finally, these feature maps are fed into a convolutional layer with 3 × 3 filter size.

4 Experiment

In this section, the implementation details of the proposed model are presented, as well as its training process. Then, the results of experiments conducted on the PASCAL VOC 2012 and Cityscapes dataset are provided. Finally, the analysis of experimental results is demonstrated.

4.1 Implementation

The network includes the encoder, decoder, and diffusion convolution layer, as is shown in Figure 1, which is implemented by the Pytorch framework. In practice, the encoder is utilised a VGG16 (Simonyan and Zisserman, 2014) network which is pretrained on the ImageNet dataset for image classification, and the last max-pooling layer in the VGG16 are modified to keep the size of feature maps unchanged. The detailed configurations of the encoder are demonstrated in Table 1. The proposed decoder is initialised by zero-mean Gaussian, and its detailed configurations are shown in Table 2. The decoder has four blocks that each block contains three convolutional layers with filter sizes are 3 × 3, 1 × 1, and 3 × 3. There are three unpooling layers in the decoder to recover the resolution of feature maps. The feature maps from the encoder and decoder are finally inputted into a convolution layer with a 1 × 1 filter size for feature fusion. Then, the final fused features are input into a diffusion convolutional layer to embed structural and contextual information. The K value in the diffusion operator is set to 3. A softmax layer is stacked at last to make the final pixel-wise classification. The ReLU activation and batch normalisation layer are applied after each convolutional layer. The proposed model is evaluated on two semantic image segmentation datasets.

PASCAL VOC 2012 is a widely used dataset in computer vision, which can be a benchmark dataset for image classification, image segmentation, object detection, action recognition, and person layout. For the semantic image segmentation task, the dataset contains 21 classes of object labels such as airplane, train, car, bus, horse, person, etc. This dataset has 1,464 images for training and 1,449 images for validation. Ten percent of the image dataset images are used for testing. The dataset is also augmented for training by flipping the image horizontally and vertically.

Cityscapes is an image dataset for semantic segmentation of urban street scenes. For the semantic image segmentation, the dataset has 19 semantic categories such as human, sky, vehicle, wall, sidewalk, bicycle, traffic light, etc. The images are divided into 2975 for training, 500 for validation, and 1526 for testing. In practice, the dataset is augmented for training by flipping the image horizontally and vertically.

Table 2 Detailed configuration of the proposed decoder network

Blocks	*Layers*	*Output size*	*Kernel*	*Stride*	*Pad*
Block 1	Conv2D	1,024 × 28 × 28	3 × 3	1	1
	Conv2D	512 × 28 × 28	1 × 1	1	0
	Conv2D	256 × 28 × 28	3 × 3	1	1
	Unpooling	256 × 56 × 56	2 × 2	2	0
Block 2	Conv2D	512 × 56 × 56	3 × 3	1	1
	Conv2D	256 × 56 × 56	1 × 1	1	0
	Conv2D	128 × 56 × 56	3 × 3	1	1
	Unpooling	128 × 112 × 112	2 × 2	2	0
Block 3	Conv2D	256 × 112 × 112	3 × 3	1	1
	Conv2D	128 × 112 × 112	1 × 1	1	0
	Conv2D	64 × 112 × 112	3 × 3	1	1
	Unpooling	64 × 224 × 224	2 × 2	2	0
Block 4	Conv2D	128 × 224 × 224	3 × 3	1	1
	Conv2D	64 × 224 × 224	1 × 1	1	0
	Conv2D	21 × 224 × 224	3 × 3	1	1

Notes: 'Conv2D' denotes the convolutional layer. ReLU layers and batch normalisation layers are omitted from the table for brevity.

For training, we pre-process the images which are resized to 250 × 250 and flipped horizontally and vertically. During training, the input image in randomly cropped to 224 × 224. The loss function is defined as:

$$Loss = \frac{1}{N}\sum_i L_i = \frac{1}{N}\sum_i -\log\left(\frac{e^{p_i}}{\sum_j e^{p_j}}\right) \tag{4}$$

where p is the output probability generated by the network. The parameters of the encoder, decoder and diffusion convolution layer are optimised in a end-to-end fashion by using Adam optimiser. Let θ_t be parameters in the network at time step t. The loss function can be denoted as $L(\theta_t)$.

$$g_t = \nabla_\theta L_t(\theta_{t-1}) \tag{5}$$
$$m_t = \beta_1 \cdot m_{t-1} + (1-\beta_1)\cdot g_t \tag{6}$$
$$v_t = \beta_2 \cdot v_{t-1} + (1-\beta_2)\cdot g_t^2 \tag{7}$$
$$\widehat{m_t} = m_t/(1-\beta_1^t) \tag{8}$$
$$\widehat{v_t} = v_t/(1-\beta_2^t) \tag{9}$$
$$\theta_t = \theta_{t-1} - \alpha\cdot\widehat{m_t}/(\sqrt{\widehat{v_t}}+\varepsilon) \tag{10}$$

α is set to an fixed step size of 10^{--13} and momentums β_1, β_2 set to 0.99 and 0.999 respectively. β_1^t denotes β_1 power to t. β_2^t denotes β_2 power to t. ε is set to 0.0005. m_0 and v_0 are initialised to 0. The max epoch is set to 200. The proposed model is trained and tested on a single NVIDIA Titan X GPU with 12 G memory.

Figure 6 Comparison results on the Cityscapes dataset (see online version for colours)

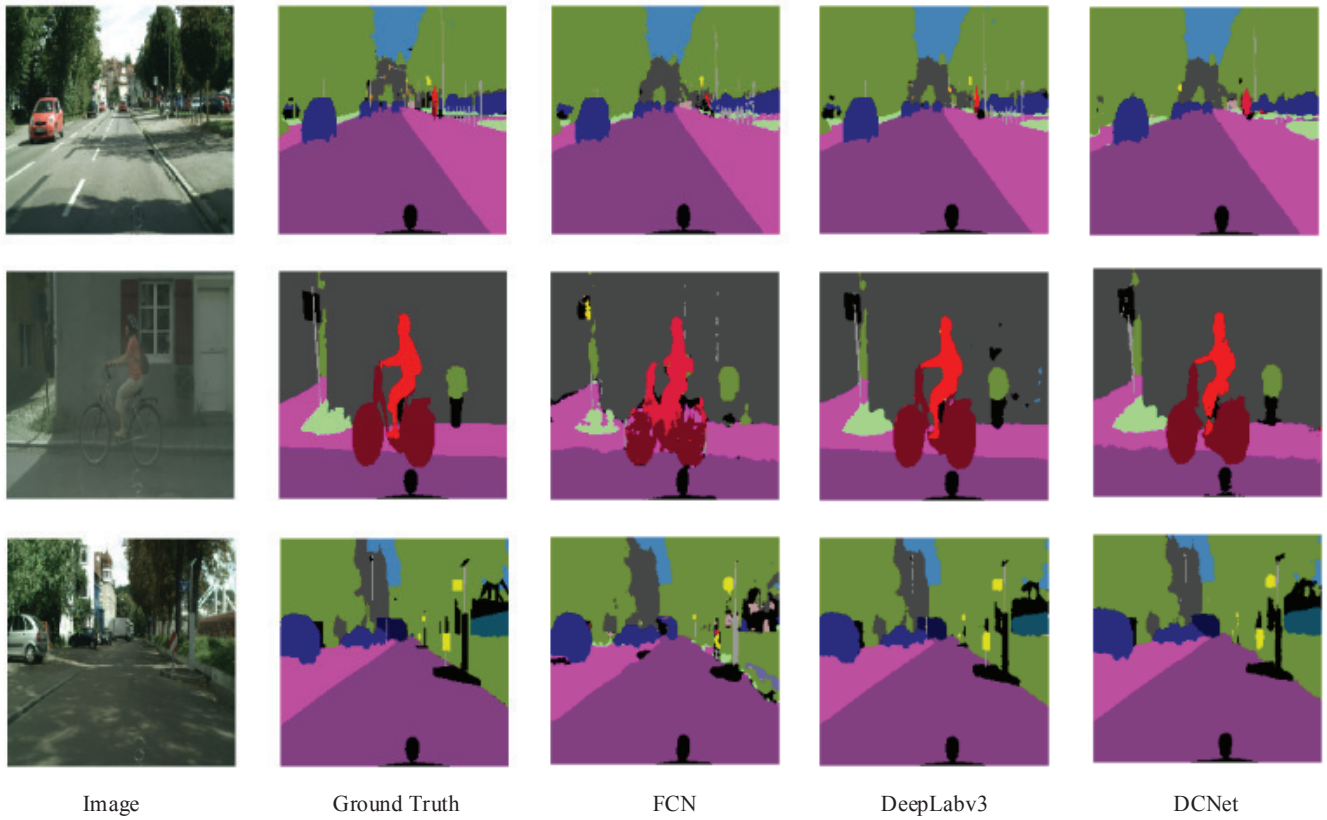
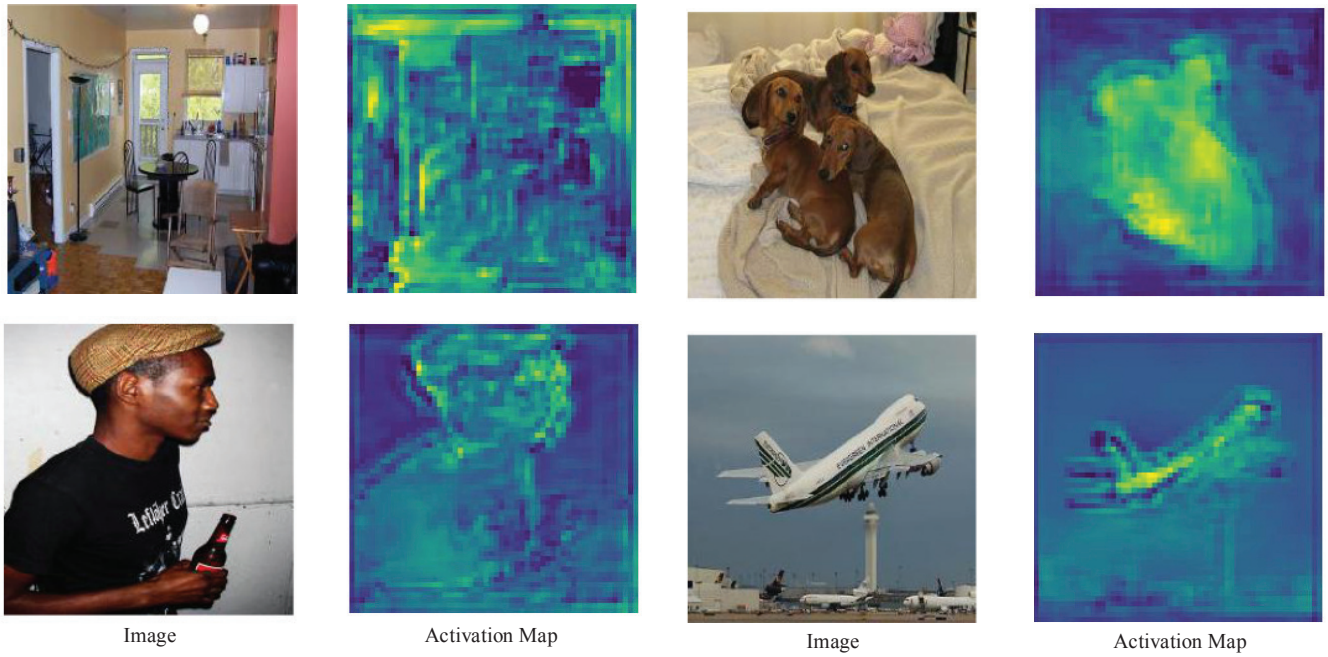


Figure 7 Visualisation of activation map in the diffusion convolution layer (see online version for colours)



The performance of the proposed model is measured in terms of mean of class-wise intersection over union (mIoU), which is defined as follows:

$$mIoU = \frac{1}{n_{cl}} \sum_{i} \frac{n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}} \tag{11}$$

where n_{ij} is the number of pixels of class i predicted to belong to class j, n_{cl} is the number of classes, and $t_i = \sum_j n_{ij}$ is the total number of pixels of class i.

4.2 Results on the PASCAL VOC 2012 dataset

The proposed DCNet is compared with five baseline models, FCN-8s, U-Net, DeconvNet, DeepLabv1, DeepLabv3, to verify the segmentation performance of DCNet. FCN, U-Net, and DeconvNet are baseline methods. DeepLabv1 and DeepLabv3 are two state-of-the-art methods in semantic image segmentation. VGG16 is used as the backbone network of FCN-8s, DeepLabv1, DeepLabv3, and the encoder network of DeconvNet and U-Net. These five approaches are also trained on our preprocessed image datasets.

Table 3 Segmentation results on PASCAL VOC 2012

Method	*mIoU*
FCN-8s	61.9%
U-Net	65.2%
DeconvNet	71.2%
DeepLabv1	70.3%
DeepLabv3	75.6%
DCNet	72.6%

As shown in Table 3, the proposed DCNet obtain 72.6% of mIoU outperforms FCN-8s' 61.9%, U-Net's 65.2%, DeconvNet's 71.2%, and DeepLabv1's 70.3%. The backbone networks of U-Net, DeconvNet, and DCNet have similar structures. The differences between U-Net and DCNet are the unpooling layers and diffusion convolution that are applied in DCNet. The only difference between DCNet and DeconvNet is that the diffusion convolutional layer is introduced in the DCNet. Therefore, the ability of the diffusion convolution in improving semantic image segmentation performance is validated by comparing it with U-Net and DeconvNet.

For qualitative analysis, some segmentation examples of DCNet, FCN-8s, and DeepLabv3 are shown in Figure 5. These comparison results reflect the category recognition and localisation capability of the models. As is seen from the example in Figure 5, the DCNet achieves a better segmentation performance than FCN-8s and can be competitive to DeepLabv3. In the first row, FCN-8s roughly locates and outlines the semantic area of horse and human in the image. But most of the horse area is classified as human, and the shape of the horse is totally distorted. Moreover, the legs of the horse are missing. The segmentation results of DeepLabv3 and DCNet are more accurate. The shape and structure of horses and people are complete. The boundaries of objects are clear. However, the legs of the horse are also not segmented as a different part of the whole structure. In the second row, FCN-8s has segmented the big plane with a complete shape. But it totally ignores the small plane in the lower right corner of the image. In addition, a part of the aircraft's tail is misclassified by FCN-8s. DeepLabv3 and DCNet have segmented two planes in the image with complete shape and smooth boundaries. In the third row, FCN-8s still disregards the small semantic area of the human and incorrectly classifies some background areas. The areas of the two people are connected together. But DCNet and DeepLabv3 has segmented all three semantic areas of human and separated the two people in the centre of the image to some extent.

These segmentation examples demonstrate that the DCNet can correct the classification errors due to local feature and spatial dependencies. In addition, the results in Figure 5 illustrate that the DCNet can effectively deal with small objects in the image due to obtaining a large receptive field by using the diffusion convolution. Besides, the complete shapes and accurate classification in the segmentation results of DCNet also verify its ability to extract and exploit multi-scale context information to make inferences, which maintains the consistency in the segmentation results.

4.3 Results on the Cityscapes dataset

The proposed model is also evaluated on the Cityscapes dataset as well as the baseline methods. As shown in Table 4, DCNet obtains 73.2% of mIoU outperforms FCN-8s' 59.6%, U-Net's 62.8%, DeconvNet's 69.1%, and DeepLabv1's 68.9%. These results also prove that DCNet's diffusion convolutional layer can improve segmentation performance by learning structure and contextual features.

Table 4 Segmentation results on Cityscapes

Method	*mIoU*
FCN-8s	59.6%
U-Net	62.8%
DeconvNet	69.1%
DeepLabv1	68.9%
DeepLabv3	74.4%
DCNet	72.2%

Some segmentation examples produced by DCNet, FCN-8s, and DeepLabv3 on the Cityscapes dataset are shown in Figure 6. In the first row of Figure 6, the FCN-8s can not segment the areas of lawn and human in the image with a complete shape. However, the DCNet recognises the human and the lawn with smooth boundaries. In the second row, the semantic areas of human and bicycle are twisted. Some parts of the bicycle are classified as human, road, or lawn. The DCNet has segmented the bicycle and human with complete structure. Especially, the leg of the human is recognised. In the third row, some isolated areas are incorrectly classified by the FCN-8s. But the DCNet can segment the objects in the image with consistent labels.

These results in Figure 6 also demonstrate that the DCNet can precisely locate and recognise semantic objects in the image, including small ones. Moreover, the DCNet can keep the consistency in the segmentation result for its graph-based structure learning.

4.4 Discussion

To further analyse the performance of the DCNet, more segmentation results produced by the DCNet on the

PASCAL VOC 2012 and Cityscapes datasets are shown in Figures 3 and 4. The DCNet can make an accurate localisation and classification such as the human, bird, and dog in Figure 3. In addition, the proposed model can effectively deal with small objects such as humans, traffic lights, and cars in Figure 4.

To show what the diffusion convolutional layer is learned, some activation maps of the diffusion operator are visualised in Figure 7. As is seen in Figure 7, the graph-based module has located and outlined some semantic areas in each input image, such as the furniture, dogs, human, and plane. Moreover, the model obtains a broad receptive field as the bright area covers all semantic areas.

Therefore, the proposed model has shown its ability to reconstruct spatial details such as locations, boundaries, and structures and keep consistency among labels. However, there are some misclassification and deformation for small objects or small parts of the objects. The reasons for this phenomenon may be that there are not enough pixels for the model to capture the characteristics of the small objects, and the learned representations are still not good enough for reconstructing spatial information completely. To solve this problem, the model can be trained with high-resolution images.

In practice, the graph convolution neural network effectively utilises the dependencies among features for capturing structural information and contextual information. However, the computation of the diffusion process consumes a lot of computing resources, especially memory. This limitation may result in paying more attention to local dependencies and features. To remedy this issue, the model can be trained in a distributed system with more GPUs and memory.

5 Conclusions

In this paper, a novel model, DCNet, combines CNNs and GNNs for the fundamental semantic image segmentation, which combines the powerful abilities of CNNs for representation learning and GNNs for structure learning. The graph for each image is constructed by connecting K nearest neighbours in a higher feature space. The diffusion process is applied to aggregate information from neighbour nodes for capturing structural and contextual features in the receptive field with various sizes. Experiments conducted on PASCAL VOC 2012 and Cityscapes datasets have validated that the diffusion convolutional layer can significantly improve the segmentation performance by equipping the deep convolutional networks with structure learning.

Acknowledgements

This work is supported by Fujian Province Educational Research Projects of Young and Middle-aged Teachers, under Grant JAT200818 and 2018J01570.

References

Ammar, S., Bouwmans, T., Zaghden, N. and Neji, M. (2020) 'Deep detector classifier (DeepDC) for moving objects segmentation and classification in video surveillance', *IET Image Processing*, Vol. 14, No. 8, pp.1490–1501.

Badrinarayanan, V., Kendall, A. and Cipolla, R. (2017) 'SegNet: a deep convolutional encoder-decoder architecture for image segmentation', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 12, pp.2481–2495.

Byeon, W., Breuel, T.M., Raue, F. and Liwicki, M. (2015) 'Scene labeling with lstm recurrent neural networks', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.3547–3555.

Chen, L-C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A.L. (2014) *Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFS* [online] https://arxiv.org/abs/1412.7062.

Chen, L-C., Papandreou, G., Schroff, F. and Adam, H. (2017) *Rethinking Atrous Convolution for Semantic Image Segmentation* [online] https://arxiv.org/abs/1706.05587.

Chen, L-C., Yang, Y., Wang, J., Xu, W. and Yuille, A.L. (2016) 'Attention to scale: scale-aware semantic image segmentation', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.3640–3649.

Chen, L-C., Zhu, Y., Papandreou, G., Schroff, F. and Adam, H. (2018) 'Encoder-decoder with atrous separable convolution for semantic image segmentation', *Proceedings of the European Conference on Computer Vision (ECCV)*, pp.801–818.

Colovic, A., Knöbelreiter, P., Shekhovtsov, A. and Pock, T. (2017) 'End-to-end training of hybrid CNN-CRF models for semantic segmentation using structured learning', *Computer Vision Winter Workshop*, Vol. 2.

Cong, D., Zhou, Q., Cheng, J., Wu, X., Zhang, S., Ou, W. and Lu, H. (2019) 'CAN: contextual aggregating network for semantic segmentation', *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp.1892–1896.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R. et al. (2016) 'The Cityscapes dataset for semantic urban scene understanding', *29th International Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.3213–3223.

Everingham, M., Van Gool, L., Williams, C.K., Winn, J. and Zisserman, A. (2015) 'The PASCAL visual object classes (VOC) challenge', *International Journal of Computer Vision*, Vol. 88, No. 2, pp.303–338.

Fu, J., Liu, J., Li, Y., Bao, Y., Yan, W., Fang, Z. and Lu, H. (2020) 'Contextual deconvolution network for semantic segmentation', *Pattern Recognition*, Vol. 101, No. 2020, p.107152.

Grigorescu, S., Trasnea, B., Cocias, T. and Macesanu, G. (2020) 'A survey of deep learning techniques for autonomous driving', *Journal of Field Robotics*, Vol. 37, No. 3, pp.362–386.

He, J., Deng, Z., Zhou, L., Wang, Y. and Qiao, Y. (2019) 'Adaptive pyramid context network for semantic segmentation', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.7519–7528.

Hu, X., Yang, K., Fei, L. and Wang, K. (2019) 'ACNet: attention based network to exploit complementary features for RGBD semantic segmentation', *2019 IEEE International Conference on Image Processing (ICIP)*, pp.1440–1444.

Hu, P., Perazzi, F., Heilbron, F.C., Wang, O., Lin, Z., Saenko, K. and Sclaroff, S. (2020) 'Real-time semantic segmentation with fast attention', *IEEE Robotics and Automation Letters*, Vol. 6, No. 1, pp.263–270.

Huang, Y-M., Hsieh, M-Y., Chao, H-C., Hung, S-H. and Park, J.H. (2009) 'Pervasive, secure access to a hierarchical sensor-based healthcare monitoring architecture in wireless heterogeneous networks', *IEEE Journal on Selected Areas in Communications*, Vol. 27, No. 4, pp.400–411.

Jiang, Y., Liang, W., Tang, J., Zhou, H., Li, K-C. and Gaudiot, J-L. (2020) 'A novel data representation framework based on nonnegative manifold regularisation', *Connection Science*, pp.1–17.

Li, H., Xiong, P., An, J. and Wang, L. (2018a) *Pyramid Attention Network for Semantic Segmentation* [online] https://arxiv.org/abs/1805.10180.

Li, X., Liu, S., Wu, F., Kumari, S. and Rodrigues, J.J. (2018b) 'Privacy preserving data aggregation scheme for mobile edge computing assisted IoT applications', *IEEE Internet of Things Journal*, Vol. 6, No. 3, pp.4755–4763.

Li, X., Tan, J., Liu, A., Vijayakumar, P., Kumar, N. and Alazab, M. (2020a) 'A novel UAV-enabled data collection scheme for intelligent transportation system through UAV speed control', *IEEE Transactions on Intelligent Transportation Systems*, Vol. 22, No. 4, pp.2100–2110.

Li, X., Liu, T., Obaidat, M.S., Wu, F., Vijayakumar, P. and Kumar, N. (2020b) 'A lightweight privacy-preserving authentication protocol for VANETs', *IEEE Systems Journal*, Vol. 14, No. 3, pp.3547–3557.

Liang, W., Xingming, S., Zhiqiang, R., Jing, L. and Chengtao, W. (2011) 'A sequential circuit-based IP watermarking algorithm for multiple scan chains in design-for-test', *Radioengineering*, Vol. 20, No. 2, pp.533–539.

Liang, W., Liao, B., Long, J., Jiang, Y. and Peng, L. (2016a) 'Study on PUF based secure protection for IC design', *Microprocessors and Microsystems*, Vol. 45, No. PA, pp.56–66.

Liang, X., Shen, X., Feng, J., Lin, L. and Yan, S. (2016b) 'Semantic object parsing with graph LSTM', *14th International Proceedings of the European Conference on Computer Vision*, pp.125–143.

Liang, W., Li, K-C., Long, J., Kui, X. and Zomaya, A.Y. (2019) 'An industrial network intrusion detection algorithm based on multifeature data clustering optimization model', *IEEE Transactions on Industrial Informatics*, Vol. 16, No. 3, pp.2063–2071.

Liang, W., Long, J., Li, K-C., Xu, J., Ma, N. and Lei, X. (2020) 'A fast defogging image recognition algorithm based on bilateral hybrid filtering', *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, Vol. 17, No. 2, pp.1551–6857.

Liu, Q., Kampffmeyer, M., Jenssen, R. and Salberg, A-B. (2020) *SCG-Net: Self-Constructing Graph Neural Networks for Semantic Segmentation* [online] https://arxiv.org/abs/2009.01599.

Long, J., Shelhamer, E. and Darrell, T. (2015) 'Fully convolutional networks for semantic segmentation', *28th International Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.3431–3440.

Lu, Y., Chen, Y., Zhao, D. and Chen, J. (2019) 'Graph-FCN for image semantic segmentation', *International Symposium on Neural Networks*, pp.97–105.

Noh, H., Hong, S. and Han, B. (2015) 'Learning deconvolution network for semantic segmentation', *14th International Proceedings of the IEEE International Conference on Computer Vision*, pp.1520–1528.

Peng, C., Zhang, X., Yu, G., Luo, G. and Sun, J. (2017) 'Large kernel matters – improve semantic segmentation by global convolutional network', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.4353–4361.

Qi, X., Liao, R., Jia, J., Fidler, S. and Urtasun, R. (2017) '3D graph neural networks for RGBD semantic segmentation', *Proceedings of the IEEE International Conference on Computer Vision*, pp.5199–5208.

Ronneberger, O., Fischer, P. and Brox, T. (2015) 'U-Net: convolutional networks for biomedical image segmentation', *18th International Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp.234–241.

Simonyan, K. and Zisserman, A. (2014) *Very Deep Convolutional Networks for Large-Scale Image Recognition* [online] https://arxiv.org/abs/1409.15567.

Soberanis-Mukul, R.D., Navab, N. and Albarqouni, S. (2020) 'Uncertainty-based graph convolutional networks for organ segmentation refinement', *Medical Imaging with Deep Learning*, pp.755–769.

Visin, F., Ciccone, M., Romero, A., Kastner, K., Cho, K., Bengio, Y. and Courville, A. (2016) 'ReSeg: a recurrent neural network-based model for semantic segmentation', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp.41–48.

Wolterink, J.M., Leiner, T. and Išgum, I. (2019) 'Graph convolutional networks for coronary artery segmentation in cardiac CT angiography', *International Workshop on Graph Learning in Medical Imaging*, pp.62–69.

Yang, M., Yu, K., Zhang, C., Li, Z. and Yang, K. (2018) 'DenseASPP for semantic segmentation in street scenes', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.3684–3692.

Zhang, L., Li, X., Arnab, A., Yang, K., Tong, Y. and Torr, P.H. (2019) *Dual Graph Convolutional Network for Semantic Segmentation* [online] https://arxiv.org/abs/1909.06121.

Zhao, H., Shi, J., Qi, X., Wang, X. and Jia, J. (2017) 'Pyramid scene parsing network' *30th International Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.2881–2890.

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D. and Torr, P.H. (2015) 'Conditional random fields as recurrent neural networks', *Proceedings of the IEEE International Conference on Computer Vision*, pp.1529–1537.