Exploit Visual Dependency Relations for Semantic Segmentation

Mingyuan Liu, Dan Schonfeld and Wei Tang*
University of Illinois at Chicago
{mingyuan, dans, tangw}@uic.edu

Abstract

Dependency relations among visual entities are ubiquity because both objects and scenes are highly structured. They provide prior knowledge about the real world that can help improve the generalization ability of deep learning approaches. Different from contextual reasoning which focuses on feature aggregation in the spatial domain, visual dependency reasoning explicitly models the dependency relations among visual entities. In this paper, we introduce a novel network architecture, termed the dependency network or DependencyNet, for semantic segmentation. It unifies dependency reasoning at three semantic levels. Intra-class reasoning decouples the representations of different object categories and updates them separately based on the internal object structures. Inter-class reasoning then performs spatial and semantic reasoning based on the dependency relations among different object categories. We will have an in-depth investigation on how to discover the dependency graph from the training annotations. Global dependency reasoning further refines the representations of each object category based on the global scene information. Extensive ablative studies with a controlled model size and the same network depth show that each individual dependency reasoning component benefits semantic segmentation and they together significantly improve the base network. Experimental results on two benchmark datasets show the DependencyNet achieves comparable performance to the recent states of the art.

1. Introduction

Semantic segmentation aims at assigning a categorical label to each pixel to partition an image into multiple meaningful segments. It is a fundamental task in computer vision and has many practical applications, such as autonomous driving, image editing, and medical image analysis. In the past decade, convolutional neural networks (CNNs) have become a dominant solution to it [26, 11].

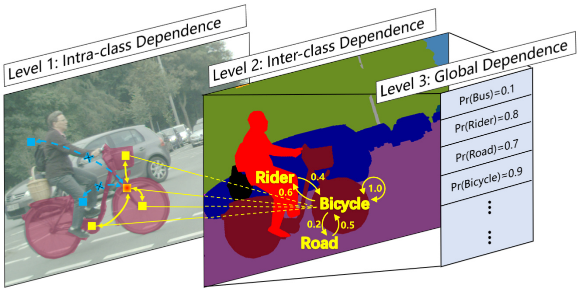
Level 1: Intra-class Dependence
Level 2: Inter-class Dependence
Level 3: Global Dependence
Pr(Bus)=0.1
Pr(Rider)=0.8
Pr(Road)=0.7
Pr(Bicycle)=0.9
⋮
⋮
Rider
Bicycle
Road
0.4
0.6
1.0
0.2
0.5

Figure 1. Visual dependence relations are ubiquity since both objects and scenes are highly structured. They provide prior knowledge about the real world that can be used to improve the generalization ability of a learning model. We consider three levels of visual dependency. Intra-class dependency means parts of an object are related due to the object's internal structural patterns, *e.g.*, a bicycle consists of wheels, pedals and a frame. Inter-class dependency means objects are related as certain objects (*e.g.*, a rider and a bicycle) co-occur more frequently than the others or by chance. Global dependency means the global scene type enforces strong prior on the categories of objects that should appear in it.

Recent works [54, 56, 51, 9, 39] have achieved great improvement by leveraging contextual information in CNNs. The context of a pixel refers to the collection of its surrounding pixels. It provides rich visual cues to resolve ambiguities in pixel classification. For example, the presence of water in the context suggests that a pixel is more likely to belong to a boat than other objects like a car or a bed. Existing approaches like ASPP [1] and PSPNet [57] explore optimal strategies for multi-scale context aggregation. The non-local network [46] and its variants [61, 21, 20] employ the self-attention mechanism [44] to capture the long-range context in an image. These methods effectively leverage the context to enrich the representations of each pixel. However, they focus on feature aggregation in the spatial domain, and the explicit dependence relations among different semantic categories are largely ignored.

Visual dependence relations are ubiquity since both objects and scenes are highly structured, and they occur at various semantic levels. Parts are related as they compose into objects, *e.g.*, a car consists of wheels and a frame. Objects

[*] Corresponding author.

are related as certain objects (*e.g.*, a desk and a chair) co-occur more frequently than the others (*e.g.*, a bed and a car). At the image level, the type of a scene enforces strong prior on the categories of objects that should appear in it. For example, it is unlikely to see a bed in an outdoor scene.

Different from contextual reasoning which focuses on aggregating pixel features from the context, visual dependency reasoning puts more emphasis on exploiting the dependency relations among semantic entities, *e.g.*, parts, objects, and a scene. This makes it possible to inject explicit prior knowledge of the real world into a learning model and thus promotes its generalization ability.

In this paper, we introduce a novel approach termed the dependency network or *DependencyNet* for semantic segmentation. It explicitly models visual dependency relations in a CNN. As shown in Figure 1, we divide visual dependency into three levels, *i.e.*, intra-class, inter-class and global dependency. Accordingly, the DependencyNet performs three levels of dependency reasoning. It first decouples the representations of different object categories so that each representation contains spatial and semantic information of only one category. Intra-class reasoning means to update the representations of each object category based on their respective internal structures, *e.g.*, a person is composed of body parts. Inter-class reasoning performs spatial and semantic reasoning based on the dependency relations among different object categories. We first mine prior knowledge about dependency relations from the training annotations and encode it via a dependency graph. Two objects are strongly related if they co-occur frequently in images. Then, the DependencyNet performs reasoning via group weighted convolutions, wherein category-specific representations interact with each other according to the dependency graph. Unlike the attention mechanisms [37, 19] which compute the feature correlations across the spatial locations or feature channels, our graph does not depend on the input image and acts as prior knowledge. The inter-class reasoning is also different from graph convolution networks (GCNs) [18, 2, 30]. GCNs take as input feature vectors of positioned objects while we perform both spatial and semantic reasoning to localize objects. Global dependency reasoning further refines the representations of each category based on the scene information. Specially, we encode a scene via probabilities of the presence of each category and use them to enrich the object representations. The contributions of this paper are summarized below.

- We introduce a novel DependencyNet to explicitly exploit visual dependency relations for semantic segmentation. It is the first neural architecture to unify three levels of dependency reasoning. The research is important as it bridges CNNs and dependency modeling commonly achieved via graphical models.
- We introduce intra-class, inter-class, and global dependency reasoning modules, which are the core components of the DependencyNet. They effectively utilize the internal object structures, object-object relations, and scene information to perform dependency reasoning. We also have an in-depth investigation of mining prior knowledge of dependency relations from training annotations.
- We perform extensive ablation studies with a controlled model size and the same depth on the three levels of dependency reasoning. Results show that each individual component benefits semantic segmentation and they together lead to significant improvement over the base network. Experimental results on two datasets demonstrate the effectiveness of our approach.

2. Related Work

Semantic Segmentation. Most state-of-the-art models for semantic segmentation are based on CNNs [57, 18, 29, 11]. The context in a CNN is formed and enlarged by a stack of convolutions and pooling operations. In each layer, features of neighboring pixels are aggregated to obtain more powerful representations. Theoretically, the receptive field of a neuron is large enough and can even cover the whole image. However, the effective context is much smaller [58, 33]. In recent years, a series of works try to explore better utilization of context.

Multi-Level Context Aggregation. Dilated convolutions [53, 1, 50, 34] and various pooling operations [17, 33] play an important role in this field. They could gather rich context within a few layers. Representative works include PSP [57] and ASPP [1]. They construct a feature pyramid to aggregate multi-scale context. [28] uses gates to selectively fuse multi-scale features. Some works replace 2D convolutions by a series of 1-D convolutions [40, 51] to enlarge the context. ACFNet [55] and OCR [54] extract class-wise global context by merging coarse segmentation results and feature maps. Different from these approaches, we focus on modeling explicit dependency relations among semantic entities rather than feature aggregation from larger context.

Attention-Based Methods. The non-local neural networks [46, 14, 21, 61] use the self-attention mechanism [44] to capture long-range spatial context. Each output neuron receives information from all input neurons based on their feature correlations. Besides, [11, 37] explore the channel attention. Some works [61, 21] try to lower the high computational cost. Instead of modeling feature correlations, our method explicitly models dependency relations among different semantic entities. Furthermore, our approach is computationally cheaper and more explainable.

Graph-Based Networks. Recently, graph convolutional networks (GCNs) [24] are used for image segmentation [2, 31, 18, 30]. Some approaches [18, 2] use GCNs for reasoning, but there is no clear semantics. BGRNet [47]

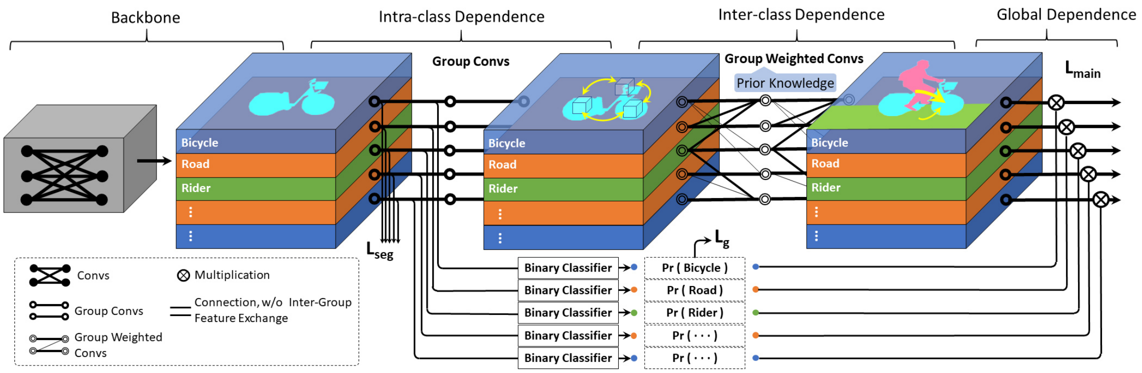

Figure 2. Architecture of the proposed DependencyNet. A backbone network extracts a convolutional feature map from the input image. The intra-class reasoning module, implemented via group convolutions, decouples the feature map into category-specific representations and updates the representations of each object category based on their respective internal structures. The inter-class reasoning module, implemented via group weighted convolutions, then performs spatial and semantic reasoning based on the dependency relations among different object categories. The global reasoning module further refines the category-specific representations via the global scene information. The final representations of each object category are used to predict their presence at each pixel. The network training is supervised by three losses: L_{seg}, L_g and L_{main}.

constructs a graph with each object category as a node. Each category is represented as a feature vector, obtained by pooling the feature map according to a coarse segmentation. Spatial information is lost before their GCN reasoning. By contrast, we perform both spatial and semantic reasoning to update the representations of each object category.

Dependency Modeling in Pre-Deep Era. The relations among parts and an object are conventionally modeled via deformable part-based models (DPMs) [10], compositional models [22, 59], and grammar models [60]. They explicitly model the displacement of each part w.r.t. the object. At the object level, some prior approaches [41, 12, 7, 3] build a CRF to model the co-occurrence statistics of objects and their spatial arrangements in an image. Heitz and Koller [16] divide objects into things and stuff, and explicitly model their spatial relations. At the global level, a few methods [38, 35] predict the presence of each object via global image features and then use it to turn on/off local detectors in a graphical model. Several other works [42, 32, 43, 23, 49] retrieve the best matches of an input image from an annotated image database via global descriptors and transfer their labels via dense pixel or superpixel correspondence. Mottaghi *et al*. [36] extend the DPM with potential functions modeling the presence of objects in the global image and local neighborhood. The work most related to our global reasoning module is [13], which refines the detection score of a window by multiplying it with the probability of object presence in the image.

Our DependencyNet bridges CNNs and dependency modeling commonly achieved via graphical models, inherits their advantages and overcomes their respective limitations. It provides a principled way for a CNN to model explicit visual dependency. Compared with graphical models, whose relational models, *e.g*., Gaussian, can be too simple to capture sophisticated relations among visual entities and whose inference and learning can be painstaking, dependency modeling via neural networks leads to greater modeling capacity, stronger visual discrimination, and better scalability to big data.

3. Method

The dependency network or *DependencyNet* means to exploit explicit visual dependency relations for semantic segmentation. It takes as input an image and outputs a categorical label for each pixel. As illustrated in Figure 2, the DependencyNet consists of a base network and three dependency reasoning modules. The base network extracts a convolutional feature map from the input image. The intra-class reasoning module decouples the feature map into category-specific representations and updates the representations of each object category based on their respective internal structures. The inter-class reasoning module then performs spatial and semantic reasoning based on the dependency relations among different object categories. The global reasoning module further refines the category-specific representations based on the scene information. The final representations of each object category are used to predict their presence at each pixel. We describe each component in detail in the rest of this section.

3.1. Intra-class Dependence Reasoning

After obtaining the feature map from the base network, we use another convolutional layer to get a decoupled representation $\mathbf{X}_k \in \mathbb{R}^{Z\times H\times W}$ for each category k ($k \in$

$\{1, \cdots, K\}$), where H and W respectively denote the height and width, Z is the number of channels, and K is the number of object categories. Let $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$ denote the collection of all category-specific representations: $\mathbf{X} = Concat(\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_K)$, where $C = K \times Z$ and $Concat(\cdot)$ is concatenation on the channel dimension.

We add intermediate supervision so that each decoupled representation $\mathbf{X}_k$ only contains information specific to the corresponding category k. Specifically, we use each representation to predict a segmentation mask and a classification score for their respective categories. They are compared with labels during training via a segmentation loss L_{seg} and a classification loss L_g. Since $\mathbf{X}_k$ only makes prediction for category k, it is guided to encode the spatial and semantic information of this object category in the image.

Finally, we perform intra-class dependency reasoning by applying two convolution layers to each category-specific representation, which can be easily implemented via group convolutions. It enables the network to update the representations of each object category based on their respective internal structures, *e.g.*, relations among object parts, without being affected by the representations of other categories. Note while the group convolution operation is not new, we use it here to process the category-specific representations. By contrast, no explicit semantics is associated to group convolutions in previous works, *e.g.*, [25, 48].

3.2. Inter-class Dependence Reasoning

Inter-class dependence reasoning updates the representations of each object category from those of others based on their dependency relations. We first introduce how we obtain a dependence graph to encode the pairwise dependency relations among object categories and then describe the reasoning module.

Dependence Graph. If one object depends on the other (for example, a car and a road, and a computer and a desk), they are usually spatially close in an image. This also reflects the interactions of forces and spatial arrangements in the physical world. Besides, visual dependence is a kind of prior knowledge or common sense that embodies the relations among visual entities and is invariant to the input image. We design two methods to discover the visual dependency relations from the training annotations and use them for inter-class dependence reasoning.

We encode pairwise dependence relations via a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{v_i | i = 1, \ldots, K\}$ is a set of nodes and $\mathcal{E} = \{e_{i,j} | i = 1, \ldots, K; j = 1, \ldots, K\}$ is a set of edges. Node v_i represents the i^{th} object category. Edge $e_{i,j} \in [0, 1]$ is the degree of category i's dependency on category j. A large value of $e_{i,j}$ indicates the presence of category j is very helpful to identify category i, and 0 means they are irrelevant. Thus, $e_{i,j}$ also determines how much category j will contribute to the inter-class dependency reasoning of category i. The diagonal elements $\{e_{i,i}\}$ are set to 1 and other elements are normalized so that $\sum_{j \neq i} e_{i,j} = 1$.

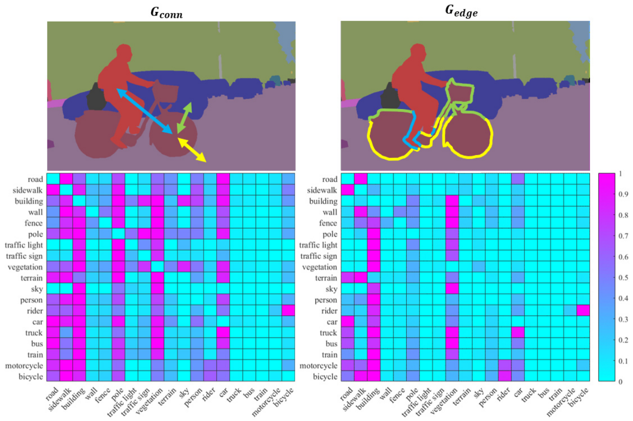

Figure 3. Illustration of two dependency graphs $\mathcal{G}_{conn}$ and $\mathcal{G}_{edge}$ calculated using the Cityscapes training annotations [5]. $\mathcal{G}_{conn}$ uses a binary indicator to count the frequency of contact between two objects. $\mathcal{G}_{edge}$ uses a soft count taking into account the length of the shared boundary between two objects. For better visualization, each row is divided by its maximum value.

The first dependency graph, denoted as $\mathcal{G}_{conn}$, is constructed based on how often objects contact each other. In an image $\mathbf{I}_m$, if the segment of category i contacts that of category j, we define $\mathbb{I}(i, j \mid \mathbf{I}_m)$ to be 1 and otherwise 0. Then, the count of images in which category i and category j contact is:

$$c_{i,j} = \sum_{m=1}^{N} \mathbb{I}(i, j \mid \mathbf{I}_m) \tag{1}$$

where N is the total number of images in the training set. $e_{i,j}$ is obtained by normalizing $c_{i,j}$: $e_{i,j} = c_{i,j} / \sum_{j \neq i} c_{i,j}$.

The second dependency graph, denoted as $\mathcal{G}_{edge}$, treats multiple objects contacting the same object in an image differently. For example, the head of a person riding a bicycle may contact the sky in an image, but obviously, the dependency relation between the rider and the sky should be much weaker than that between the rider and the bicycle. Thus, instead of using a binary contact indicator as in $\mathcal{G}_{conn}$, $\mathcal{G}_{edge}$ takes into account the length of shared boundaries. We use a soft count of images in which category i and category j contact:

$$c_{i,j} = \sum_{m=1}^{N} (L_{i,j}^{m} / L_{i}^{m}) \tag{2}$$

where $L_{i,j}^{m}$ is the length of the boundary shared by category i and category j in the m^{th} training image, and L_i^m is the perimeter of the segment of category i in the m^{th} training image. $e_{i,j}$ is obtained by normalizing $c_{i,j}$ as in $\mathcal{G}_{conn}$.

The affinity matrices obtained by the two methods are demonstrated in Figure 3. They have their respective pros and cons. $\mathcal{G}_{edge}$ could distinguish strong and weak dependence. But sometimes the length of the shared boundary is misleading. For example, a traffic sign is always located on a pole but the length of their shared boundary is short. So although the pole and traffic sign always contact each other, their dependence value will be small in $\mathcal{G}_{edge}$. However, since $\mathcal{G}_{conn}$ only considers whether two objects contact or not, the dependency relation between the pole and traffic sign will be more significant. The effectiveness of these two dependency graphs is compared in the experiments.

Group Weighted Convolution. We propose the *group weighted convolution* to leverage the inter-class dependence relations for spatial and semantic reasoning. Suppose its input is $\mathbf{X} = Concat(\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_K)$, where $\mathbf{X}_k \in \mathbb{R}^{Z \times H \times W}$ is the representation of category k, *e.g.*, obtained from intra-class reasoning. During inter-class dependency reasoning, the representations of each category are updated by the representations of categories that they respectively depend on. We denote the output as $\mathbf{Y} = Concat(\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_K)$, where $\mathbf{Y}_k \in \mathbb{R}^{Z^o \times H \times W}$ is the updated representation of category k, Z^o is the number of channels, $\mathbf{Y} \in \mathbb{R}^{C^o \times H \times W}$, and $C^o = K \times Z^o$. Before presenting the group weighted convolution, we first review the conventional convolution and group convolution.

The conventional convolution could be written as

$$\mathbf{Y} = \mathbf{W} * \mathbf{X} \tag{3}$$

$\mathbf{W} \in \mathbb{R}^{C^o \times C \times S \times S}$ is the convolutional kernels, S is the kernel size, and $*$ is the convolution operation. It treats the representations of each category equally.

The group convolution divides the input feature map into K groups on the channel dimension. Then convolutions are conducted on each group separately. The groups of new feature maps are concatenated to get the output $\mathbf{Y}$:

$$\mathbf{Y} = Concat_{k=1}^{K}(\mathbf{W}_k * \mathbf{X}_k) \tag{4}$$

where $\mathbf{W}_k \in \mathbb{R}^{Z^o \times Z \times S \times S}$ is the convolutional kernels of the k^{th} group. Since the representations of each category are processed separately without interactions from other categories, the group convolution is suitable to perform intra-class reasoning as discussed in Section 3.1.

Our group weighted convolution is designed to perform inter-class dependency reasoning. Each category-specific representation will interact with others under the guidance of the dependency graph. Specifically, it is defined as:

$$\begin{aligned} \mathbf{Y} &= Concat_{k=1}^{K}(\mathbf{Y}_k) && (5) \\ &= Concat_{k=1}^{K}(\sum_{j=1}^{K} e_{k,j} \times (\mathbf{W}_{k,j} * \mathbf{X}_j)) && (6) \end{aligned}$$

where $\mathbf{W}_{k,j} \in \mathbb{R}^{Z^o \times Z \times S \times S}$ is a convolutional kernel modeling the relational pattern between category k and category j, $e_{k,j}$ denotes the dependence of category k on category j. The updated representation of category k, *i.e.*, $\mathbf{Y}_k$, is obtained by aggregating the transformed representations of each category $\{\mathbf{W}_{k,j} * \mathbf{X}_j : \forall j\}$ with dependency weights $\{e_{k,j} : \forall j\}$. A higher dependency weight indicates a more significant contribution. We use different convolutional kernels $\{\mathbf{W}_{k,j} : \forall k, j\}$ to model the diverse relational patterns between different pairs of categories.

In fact, both the conventional convolution and group convolution could be regarded as special cases of the group weighted convolution. When all the edges in the dependency graph are equal to 1, the group weighted convolution becomes the conventional convolution. When edges on the diagonal are 1 and others are 0, it becomes the group convolution. The group weighted convolution could be used to explicitly exploit the dependency relations among different categories. For example, when inferring the bicycle in an image, categories with strong relations to it, *e.g.*, road, sidewalk, and rider, will contribute much more than the irrelevant categories like sky and river.

In addition, the group weighted convolution performs both spatial and semantic reasoning. This makes it different from the graph convolutions [47, 18, 30] in which the spatial information is lost. We experimentally set the number of group weighted convolution layers to 2 in the DependencyNet. Its analysis is in the supplementary material.

3.3. Global Dependence Reasoning

The global scene is modeled as a set of probabilities $\{P(b_i \mid \mathbf{I}_m) : \forall i\}$, $b_i = 1$ if the object of category i exists in the m^{th} image and 0 otherwise. It is a multi-class classification problem, where all existing classes will be labeled as 1. During training, it is supervised by the binary cross entropy loss L_g. We perform global dependency reasoning by multiplying the probability of each class to the corresponding class-specific representation. The intuition is that if category i does not exist but is wrongly identified locally, then it could be rectified by multiplying a very small probability.

Some recent methods [33, 1, 57] also exploit global information for semantic segmentation. They obtain the global context via global pooling without supervision and concatenate it with local features. Different from them, our global scene representation explicitly encodes the probability that each category exists in the image.

3.4. Loss Function

The loss function consists of three parts: L_{main} for the final segmentation output, a classification loss L_g for the global scene representation, a segmentation loss L_{seg} to spatially supervise the category-specific representations. L_{seg} and L_{main} are cross entropy losses and L_g is a binary cross entropy loss. The total loss function L is

$$L = L_{main} + \lambda_1 \times L_g + \lambda_2 \times L_{seg} \tag{7}$$

where the weights λ_1 and λ_2 are empirically set to 0.1 and 0.1, respectively. Detailed study about them is listed in the supplementary material.

4. Experiments

4.1. Datasets and Implementation Details

To verify the effectiveness of the DependencyNet, we conduct experiments on the Cityscapes [5] and BDD100K [52] datasets. We use the mean Intersection over Union (mIoU%) index for quantitative evaluation.

Cityscapes Dataset [5] is collected for urban scene understanding. It contains 5K finely annotated urban scene images and has 2975 / 500 / 1525 images for training, validation, and testing, respectively. The resolution of each image is 2048×1024. It contains 30 classes, and 19 classes are used for evaluation. There are also 20K coarsely annotated images, but they are not used for training in this work.

BDD100K Dataset [52] is the Berkeley Deep Drive dataset. Its segmentation dataset contains 7000 images for training and 1000 images for validation. The resolution of each image is 1280×720. Images in this dataset cover various conditions, such as day, night and, different weather.

When training the network on the Cityscapes dataset, we employ the SGD optimizer with an initial learning rate of 0.01, the momentum 0.9, and the weight decay 0.0005. We use a polynomial learning rate policy: the initial learning rate is multiplied by $(1 - iter/max\_iter)^{power}$ and $power$ equals 0.9. The hyper-parameter Z mentioned in Section 3.2 is set to 64, so that it is a power of 2 and the total number of channels $C = K \times Z = 20 \times 64 = 1280$ is between 1024 and 2048, which are two typical output channels of a backbone. $K = 20$ is the number of categories in Cityscapes including the background. When performing evaluation on the validation set, typical data augmentations methods are used, including random horizontal flipping, random scaling within [0.5, 2], color jittering, and random cropping. The crop size is 768×768. The training will last for 180 epochs. We use the ResNet [15] pre-trained on ImageNet [6] as the backbone. Its output stride is 8. We also adopt an auxiliary cross-entropy loss to supervise the intermediate layers [57, 17, 4] and the weight is set to 0.4 by default. The hyper-parameters for the BDD100K dataset are the same as those of Cityscapes except that the crop size is changed to 608×608, the batch size to 16, and the training iteration to 72K. We make these adjustments because there are more training data in BDD100K.

Backbone	Intra	Inter	Global	mIoU%	#W
ResNet50	-	-	-	71.38	33M
ResNet50	✓	-	-	73.54	33M
ResNet50	-	✓	-	72.55	33M
ResNet50	✓	✓	-	74.74	33M
ResNet50	-	-	✓	74.92	33M
ResNet50	✓	-	✓	75.68	33M
ResNet50	-	✓	✓	75.59	33M
ResNet50	✓	✓	✓	76.32	33M

Table 1. Ablative studies of the intra-class, inter-class and global reasoning modules on the Cityscapes validation dataset. "-" in the columns of "Intra" and "Inter" means the respective reasoning module is replaced by conventional convolutions. "-" in the column of "Global" means the global reasoning module is removed. $\#W$ is the number of trainable weights.

Backbone	L_{seg}	mIoU%	#W
ResNet50	-	76.32	33M
ResNet50	✓	77.66	33M

Table 2. Ablative studies of L_{seg} on the Cityscapes validation dataset. We use the DependencyNet including all three reasoning modules. #W is the number of trainable weights.

4.2. Ablative Study

We perform ablative studies on the Cityscapes dataset. All experiments are conducted under a controlled model size and the same number of layers to ensure the fairness of comparisons. Moreover, to reduce the training time of the experiments, most of the ablative studies are conducted on ResNet50 with a batch size of 2.

Impact of Intra-Class Dependence Reasoning. The effectiveness of the intra-class dependency reasoning is demonstrated in Table 1. There are two group convolutions in this module. We build a baseline by replacing the group convolutions with conventional convolutions. It retains the depth of the network and avoids unfair advantages caused by extra layers. We also adjust the number of channels to make the model sizes in different settings roughly the same. We can see the intra-class dependence reasoning could achieve stable performance gain in all settings. Since the final segmentation masks of each category are predicted by their respective category-specific representations, L_{main} also promotes their decoupling. That explains why there is performance gain without global dependence reasoning.

Impact of Inter-Class Dependence Reasoning. We verify the effectiveness of inter-class dependence reasoning in different settings. The results are shown in Table 1. We use $\mathcal{G}_{conn}$ as the dependency graph. In these experiments, the number of parameters and the number of layers in the network are fixed. The inter-class reasoning could achieve stable performance gain in all different settings.

Backbone	Inter	mIoU%	#W
ResNet50	-	76.14	33M
ResNet50	$\mathcal{G}_{conn}$	77.66	33M
ResNet50	$\mathcal{G}_{edge}$	77.41	33M

Table 3. Ablative studies of the inter-class dependency graph on the Cityscapes validation dataset, where all three reasoning modules are used. "-" means to replace the inter-class reasoning module with convolutions. $\#W$ is the number of trainable weights.

Backbone	Methods	mIoU%
ResNet50	Class Center Mapping [55]	74.61
ResNet50	Ours	77.66

Table 4. Ablative studies of different methods to extract category-specific representations.

Impact of Global Dependence Reasoning. The effectiveness of the global dependence reasoning is shown in Table 1. It is light-weighted and introduces 0.16M trainable weights. The performance gain is expected for two reasons. On the one hand, it helps to supervise the learning of category-specific representations. On the other hand, the probability of the existence of a category could help resolve ambiguities from a scene point of view.

The Impact of Intermediate Supervision $\mathbf{L_{seg}}$. L_{seg} supervises the learning of class-specific representations spatially. In Table 2, under the supervision of L_{seg}, the performance could be further improved. The four experiments above demonstrate intra, inter and global dependency modules are effective. It is worth mentioning that when all of them are hierarchically integrated, they could work harmoniously and achieve further performance gain. In other words, their functions are complementary. They could model dependency relations at different levels and jointly improve semantic segmentation. This demonstrates the effectiveness of our DependencyNet.

The Impact of Dependency Graph. $\mathcal{G}_{conn}$ and $\mathcal{G}_{edge}$ are two different dependency graphs obtained from the training annotations. Their affinity matrices are demonstrated in Figure 3. The conventional convolution is a special case of the group weighted convolution when all the weights $\{e_{i,j}\}$ equal 1. Thus, we construct a baseline by replacing the group weighted convolutions with the conventional convolutions. The results are shown in Table 3. Our superior performance demonstrates the effectiveness of our inter-class dependency reasoning and also indicates that learning dependency relations by conventional convolutions is difficult. We can also observe that $\mathcal{G}_{conn}$ slightly outperforms $\mathcal{G}_{edge}$. As analyzed in Section 3.2, $\mathcal{G}_{conn}$ and $\mathcal{G}_{edge}$ have their respective pros and cons. We also have tried to combine them with arithmetic mean, geometric mean, and quadratic mean, but did not observe obvious improvement.

Backbone	Batch	Model	MS	mIoU%
ResNet101	8	-		76.44
ResNet101	8	Ours		79.15
ResNet101	8	Ours	✓	80.48
ResNet101	8	ASPP+Ours	✓	82.01

Table 5. Other ablative studies on the Cityscapes validation dataset. MS=multi-scale inference.

The Impact of Different Class-wise Feature Extraction Methods. Our DependencyNet exploits intermediate supervision, *i.e.*, L_g and L_{seg}, to facilitate the learning of category-specific representations. An alternative way to achieve this goal is to map the coarse segmentation maps of each category to their respective representations, which has been used to extract class-wise context in [55, 54]. Here, we compare it with the intermediate supervision in our DependencyNet. The results are displayed in Table 4. We can see that our strategy outperforms the mapping strategy, and significantly improves the performance of the DependencyNet.

Other Ablative Studies. We validate some commonly used methods to improve performance, as shown in Table 5. The MS means to include in the inference process horizontal flipping, sliding inference, and multi-scale reasoning with scales = $\{0.5, 1.0, 2.0\}$. It is widely used in [4, 57, 17]. We also show that the ASPP [1], which is designed to enlarge the context, is complementary to our DependencyNet and can further improve its performance. These strategies are retained in subsequent experiments.

Moreover, we take the HRNet-W48 [45] and ResNeXt-101 [48] as advanced backbones. Without tuning any hyper-parameters, the DependencyNet improves the mIoU of the baseline from 79.6% to 81.1% on the HRNet-W48, and from 81.1% to 82.8% on the ResNeXt-101.

4.3. Results on Cityscapes

To compare with the state-of-the-art methods, our model is trained on finely annotated training and validation sets. Because more images are used for training, we increase the training iterations to 108K. The crop size and batch size are 864 and 12 respectively. The results on the testing set are displayed in Table 6. Compared to other methods, our DependencyNet not only achieves the best overall performance, but also produces the best mIoU on most of the classes.

4.4. Results on BDD100K

We show the performance of the DependencyNet on BDD100K in Table 7. It is a new dataset published recently. Here we use the ResNet101 as the baseline, and its performance is 62.3%. Adding our dependency reasoning modules improves the performance to 63.9%. It could further demonstrate the effectiveness of our model.

Method(Year)	mIoU%	road	side-walk	build-ing	wall	fence	pole	traffic-light	traffic-sign	vege	terrain	sky	person	rider	car	truck	bus	train	moto-cycle	bicycle
DenseASPP [50](18)	80.6	98.7	87.1	93.4	**60.7**	62.7	65.6	74.6	78.5	93.6	72.5	95.4	86.2	71.9	96.0	**78.0**	**90.3**	80.7	69.7	76.8
CCNet [21](19)	81.4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BFP [8](19)	81.4	98.7	87.0	93.5	59.8	63.4	68.9	76.8	80.9	93.7	72.8	95.5	87.0	72.1	96.0	77.6	89.0	**86.9**	69.2	77.6
DAN [11](19)	81.5	98.6	86.1	93.5	56.1	63.3	69.7	77.3	81.3	93.9	72.9	95.7	87.3	72.9	96.2	76.8	89.4	**86.5**	**72.2**	78.2
CPNet101 [51](20)	81.3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SpyGR [27](20)	81.6	98.7	86.9	93.6	57.6	62.8	70.3	**78.7**	81.7	93.8	72.4	95.6	88.1	74.5	96.2	73.6	88.8	86.3	72.1	**79.2**
ACFNet[55](19)	81.8	98.7	87.1	**93.9**	60.2	**63.9**	71.1	78.6	81.5	94.0	72.9	**95.9**	88.1	74.1	**96.5**	76.6	89.3	81.5	72.1	**79.2**
OCR [54](20)	81.8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DependencyNet	**81.9**	**98.9**	**88.0**	**93.9**	59.2	63.6	**72.3**	78.6	**82.0**	**94.1**	**73.6**	**95.9**	**88.2**	**75.1**	**96.5**	73.5	89.6	83.3	70.6	78.8

Table 6. Comparison with the state-of-the-art methods on the Cityscapes testing set. Only finely annotated images are used for training. The backbone of all listed methods is ResNet-101.

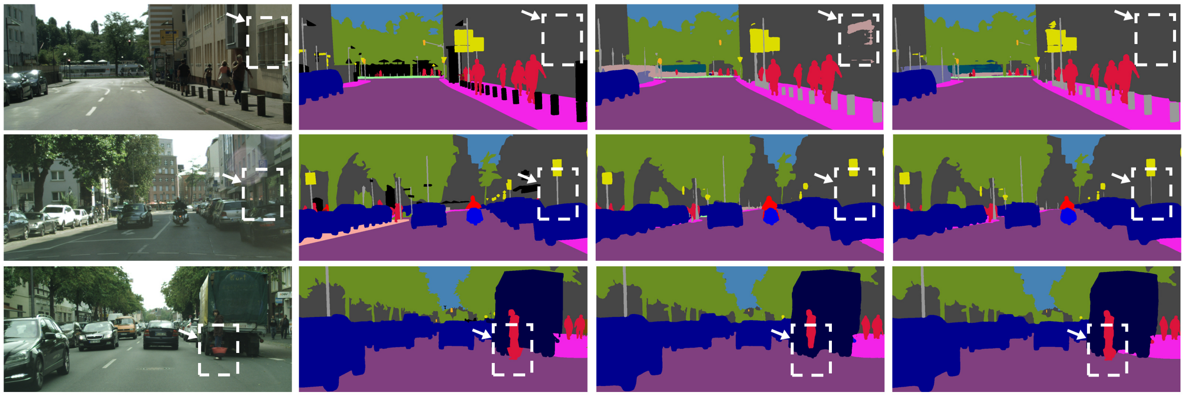

Figure 4. Visualization results of representative examples. From left to right are input images, ground truth, results of the baseline, and results of our DependencyNet. The white boxes together with the white arrow denote important differences.

Methods	mIoU%
ResNet101	62.3
ResNet101+Ours	63.9

Table 7. Results on the validation set of BDD100K.

4.5. Visualization

In Figure 4, we present three representative examples to demonstrate the effectiveness of our model. In the first row, a window, marked by a white box, is misidentified as a fence by the baseline network. It is obviously unreasonable that a fence appears in the middle of a building. This mistake is rectified by incorporating the visual dependence in our design. In the second row, under the undesirable lighting condition, it is difficult to identify the existence of a pole. However, based on the knowledge that there is usually a pole under a traffic sign, the pole is identified by our approach. In the third row, the basket held by the person together with his or her foot is wrongly classified. But our design corrects this mistake. These visualization results could further support our statement that the visual dependence could resolve ambiguities and improve generalization.

5. Conclusion

This paper introduces the DependencyNet for semantic segmentation. Different from contextual reasoning which focuses on aggregating features in the spatial domain, it explicitly takes into account visual dependency relations among semantic entities. By performing dependency reasoning at different levels, the DependencyNet can resolve semantic ambiguity and enjoy better generalization. Both quantitative and qualitative results on the Cityscapes dataset and the BDD100K dataset demonstrate the effectiveness of each component of the DependencyNet and our unified framework.

Acknowledgments: This work was supported in part by Wei Tang's startup funds from the University of Illinois at Chicago, the National Science Foundation (NSF) award CNS-1828265 and Mingyuan Liu's University Fellowship from the graduate college of the University of Illinois at Chicago.

References

[1] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 1, 2, 5, 7

[2] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *CVPR*, pages 433–442, 2019. 2

[3] Myung Jin Choi, Joseph J Lim, Antonio Torralba, and Alan S Willsky. Exploiting hierarchical context on a large database of object categories. In *CVPR*, pages 129–136, 2010. 3

[4] Sungha Choi, Joanne T Kim, and Jaegul Choo. Cars can't fly up in the sky: Improving urban-scene segmentation via height-driven attention networks. In *CVPR*, pages 9373–9383, 2020. 6, 7

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016. 4, 6

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 6

[7] Chaitanya Desai, Deva Ramanan, and Charless Fowlkes. Discriminative models for static human-object interactions. In *CVPR*, pages 9–16, 2010. 3

[8] Henghui Ding, Xudong Jiang, Ai Qun Liu, Nadia Magnenat Thalmann, and Gang Wang. Boundary-aware feature propagation for scene segmentation. In *ICCV*, pages 6819–6829, 2019. 8

[9] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Semantic segmentation with context encoding and multi-path decoding. *TIP*, 29:3520–3533, 2020. 1

[10] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2009. 3

[11] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, pages 3146–3154, 2019. 1, 2, 8

[12] Carolina Galleguillos, Andrew Rabinovich, and Serge Belongie. Object categorization using co-occurrence, location and appearance. In *CVPR*, pages 1–8, 2008. 3

[13] Hedi Harzallah, Frédéric Jurie, and Cordelia Schmid. Combining efficient object localization and image classification. In *ICCV*, pages 237–244, 2009. 3

[14] Junjun He, Zhongying Deng, and Yu Qiao. Dynamic multiscale filters for semantic segmentation. In *ICCV*, pages 3562–3572, 2019. 2

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6

[16] Geremy Heitz and Daphne Koller. Learning spatial context: Using stuff to find things. In *ECCV*, pages 30–43, 2008. 3

[17] Qibin Hou, Li Zhang, Ming-Ming Cheng, and Jiashi Feng. Strip pooling: Rethinking spatial pooling for scene parsing. In *CVPR*, pages 4003–4012, 2020. 2, 6, 7

[18] Hanzhe Hu, Deyi Ji, Weihao Gan, Shuai Bai, Wei Wu, and Junjie Yan. Class-wise dynamic graph convolution for semantic segmentation. In *ECCV*, 2020. 2, 5

[19] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018. 2

[20] Lang Huang, Yuhui Yuan, Jianyuan Guo, Chao Zhang, Xilin Chen, and Jingdong Wang. Interlaced sparse self-attention for semantic segmentation. *arXiv preprint arXiv:1907.12273*, 2019. 1

[21] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, pages 603–612, 2019. 1, 2, 8

[22] Ya Jin and Stuart Geman. Context and hierarchy in a probabilistic image model. In *CVPR*, volume 2, pages 2145–2152, 2006. 3

[23] Jaechul Kim and Kristen Grauman. Shape sharing for object segmentation. In *ECCV*, pages 444–458, 2012. 3

[24] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 4

[26] Xiangtai Li, Xia Li, Li Zhang, Guangliang Cheng, Jianping Shi, Zhouchen Lin, Shaohua Tan, and Yunhai Tong. Improving semantic segmentation via decoupled body and edge supervision. In *ECCV*, 2020. 1

[27] Xia Li, Yibo Yang, Qijie Zhao, Tiancheng Shen, Zhouchen Lin, and Hong Liu. Spatial pyramid based graph reasoning for semantic segmentation. In *CVPR*, pages 8950–8959, 2020. 8

[28] Xiangtai Li, Houlong Zhao, Lei Han, Yunhai Tong, Shaohua Tan, and Kuiyuan Yang. Gated fully fusion for semantic segmentation. In *AAAI*, volume 34, pages 11418–11425, 2020. 2

[29] Xia Li, Zhisheng Zhong, Jianlong Wu, Yibo Yang, Zhouchen Lin, and Hong Liu. Expectation-maximization attention networks for semantic segmentation. In *ICCV*, pages 9167–9176, 2019. 2

[30] Yin Li and Abhinav Gupta. Beyond grids: Learning graph representations for visual recognition. In *NeurIPS*, pages 9225–9235, 2018. 2, 5

[31] Xiaodan Liang, Zhiting Hu, Hao Zhang, Liang Lin, and Eric P Xing. Symbolic graph reasoning meets convolutions. In *NeurIPS*, pages 1853–1863, 2018. 2

[32] Ce Liu, Jenny Yuen, and Antonio Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *CVPR*, pages 1972–1979, 2009. 3

[33] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. In *ICLR workshop*, 2016. 2, 5

[34] Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda Shapiro, and Hannaneh Hajishirzi. Espnet: Efficient spatial

pyramid of dilated convolutions for semantic segmentation. In *ECCV*, pages 552–568, 2018. 2

[35] Davide Modolo, Alexander Vezhnevets, and Vittorio Ferrari. Context forest for object class detection. In *BMVC*, volume 1, page 6, 2015. 3

[36] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, pages 891–898, 2014. 3

[37] Lichao Mou, Yuansheng Hua, and Xiao Xiang Zhu. A relation-augmented fully convolutional network for semantic segmentation in aerial scenes. In *CVPR*, pages 12416–12425, 2019. 2

[38] Kevin P Murphy, Antonio Torralba, and William T Freeman. Graphical model for recognizing scenes and objects. In *NeurIPS*, pages 1499–1506, 2003. 3

[39] Kien Nguyen, Clinton Fookes, and Sridha Sridharan. Context from within: Hierarchical context modeling for semantic segmentation. *Pattern Recognition*, page 107358, 2020. 1

[40] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters–improve semantic segmentation by global convolutional network. In *CVPR*, pages 4353–4361, 2017. 2

[41] Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge Belongie. Objects in context. In *ICCV*, pages 1–8, 2007. 3

[42] Bryan Russell, Antonio Torralba, Ce Liu, Rob Fergus, and William Freeman. Object recognition by scene alignment. *NeurIPS*, 20:1241–1248, 2007. 3

[43] Joseph Tighe and Svetlana Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. In *ECCV*, pages 352–365, 2010. 3

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 1, 2

[45] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2020. 7

[46] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018. 1, 2

[47] Yangxin Wu, Gengwei Zhang, Yiming Gao, Xiajun Deng, Ke Gong, Xiaodan Liang, and Liang Lin. Bidirectional graph reasoning network for panoptic segmentation. In *CVPR*, pages 9080–9089, 2020. 2, 5

[48] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 1492–1500, 2017. 4, 7

[49] Jimei Yang, Brian Price, Scott Cohen, and Ming-Hsuan Yang. Context driven scene parsing with attention to rare classes. In *CVPR*, pages 3294–3301, 2014. 3

[50] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes. In *CVPR*, pages 3684–3692, 2018. 2, 8

[51] Changqian Yu, Jingbo Wang, Changxin Gao, Gang Yu, Chunhua Shen, and Nong Sang. Context prior for scene segmentation. In *CVPR*, pages 12416–12425, 2020. 1, 2, 8

[52] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, pages 2636–2645, 2020. 6

[53] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 2

[54] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, 2020. 1, 2, 7, 8

[55] Fan Zhang, Yanqin Chen, Zhihang Li, Zhibin Hong, Jingtuo Liu, Feifei Ma, Junyu Han, and Errui Ding. Acfnet: Attentional class feature network for semantic segmentation. In *ICCV*, pages 6798–6807, 2019. 2, 7, 8

[56] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context rncoding for semantic segmentation. In *CVPR*, pages 7151–7160, 2018. 1

[57] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017. 1, 2, 5, 6, 7

[58] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014. 2

[59] Long Zhu, Yuanhao Chen, Antonio Torralba, William Freeman, and Alan Yuille. Part and appearance sharing: Recursive compositional models for multi-view. In *CVPR*, pages 1919–1926, 2010. 3

[60] S. Zhu and D. Mumford. A stochastic grammar of images. *Foundations and Trends in Comp. Graphics and Vision*, 2(4):259–362, 2006. 3

[61] Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *ICCV*, pages 593–602, 2019. 1, 2