Object detection with location-aware deformable convolution and backward attention filtering

Chen Zhang
Illinois Institute of Technology
Chicago, USA
czhang57@hawk.iit.edu

Joohee Kim
Illinois Institute of Technology
Chicago, USA
joohee@ece.iit.edu

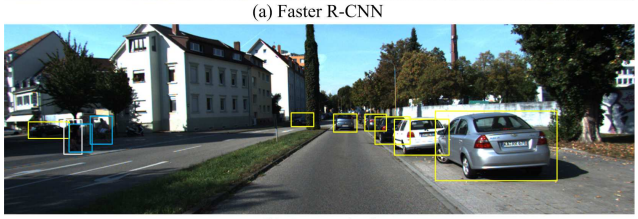
Abstract

Multi-class and multi-scale object detection for autonomous driving is challenging because of the high variation in object scales and the cluttered background in complex street scenes. Context information and high-resolution features are the keys to achieve a good performance in multi-scale object detection. However, context information is typically unevenly distributed, and the high-resolution feature map also contains distractive low-level features. In this paper, we propose a location-aware deformable convolution and a backward attention filtering to improve the detection performance. The location-aware deformable convolution extracts the unevenly distributed context features by sampling the input from where informative context exists. Different from the original deformable convolution, the proposed method applies an individual convolutional layer on each input sampling grid location to obtain a wide and unique receptive field for a better offset estimation. Meanwhile, the backward attention filtering module filters the high-resolution feature map by highlighting the informative features and suppressing the distractive features using the semantic features from the deep layers. Extensive experiments are conducted on the KITTI object detection and PASCAL VOC 2007 datasets. The proposed method shows an average 6% performance improvement over the Faster R-CNN baseline, and it has the top-3 performance on the KITTI leaderboard with the fastest processing speed.

1. Introduction

Vision-based object detection is one of the most active research areas in computer vision for a long time. For applications such as autonomous driving, accurate real-time multi-class object detection is required to understand the driving situation and avoid hitting other traffic participants. Traditional object detection systems rely on hand-crafted feature extraction and machine learning based classification. Recently, object detectors based on deep convolutional neural networks (CNN) [1] have shown a huge performance improvement in the benchmark such as KITTI [2] that focuses on driving scenes.

(a) Faster R-CNN



(b) Our method

Figure 1. Comparison between (a) Faster R-CNN and (b) our proposed method on KITTI object detection dataset. Different box color indicates different object category. Our method successfully detects the cyclists behind the pedestrian, and two small cars, which are not detected by Faster R-CNN.

A typical driving scene is shown in Figure 1, which is captured by a car-mounted camera. Three main objects that should be detected accurately in the driving scene are pedestrians, cyclists, and cars. These traffic participants interact with the autonomous car all the time and must be detected in real-time to avoid accident. Multi-class object detection for these traffic participants is challenging because they have different distance to the camera, which results in high scale variation. Also, pedestrians, cyclists and cars interact with each other frequently, as a result, occlusion occurs quite often. Furthermore, a street scene in a modern city contains cluttered backgrounds with various visual attributes, which makes the object detection even harder.

According to previous studies [3] [4] [5], context infor-

mation and high-resolution features are crucial for detecting multi-scale objects under complex scenes. The most common solution to extract context features is to increase the receptive field so that a larger area can be seen by the convolutional layers [6]. However, it is observed that the distribution of context information is uneven and not fixed. To capture the context information, not only a large receptive field is needed, but also an adaptive geometric structure of inputs is desired. The standard convolution has a fixed input sampling grid that is not flexible to handle the high variation of context distribution. The deformable convolution [7] breaks the fixed geometry of the standard convolution by introducing a set of offsets to shift the location of each input sample, which makes it a good approach to adaptively extract context features. Another important aspect for a successful detection is to utilize high-resolution features to handle small objects. However, high-resolution features found in shallow CNN layers are cluttered and distractive in a street scene. To keep the detector focused on the target, it is desired to highlight the informative features while suppressing distractive ones. One good solution is to use deep convolutional layers with high level of semantic features as the attention map to filter the high-resolution feature maps generated from shallow convolutional layers.

In this paper, we propose a location-aware deformable convolution and a backward attention filtering to improve the detection performance. The contributions can be described as follows: (1) We propose a location-aware deformable convolution to extract context features that do not have a fixed geometric distribution. The context features extracted by the proposed deformable convolution is used to enhance the standard convolutional features for improving the object detection performance. (2) We propose a backward attention filtering module to filter the feature map of shallower layers using deeper layer features. The filtered feature maps make the informative features stand out for classification and bounding box regression and also make the region proposal network (RPN) easier to generate reasonable ROIs. Thus, the number of ROIs needed is reduced, and the detection speed is improved. (3) We combine the location-aware deformable convolution and the backward attention filtering module into a forward-backward object detection network. The proposed detection network achieves the top performance for multi-class object detection on KITTI and PASCAL VOC dataset with the shortest runtime among the top-performing methods.

2. Related works

2.1. Convolutional neural network for object detection

In the recent years, deep learning-based object detectors have shown significantly improved performance over the traditional hand-crafted models [8] [9]. Region-based convolutional neural networks (R-CNN) is presented in [10] for object detection task, which is improved by Fast R-CNN [11] with a faster speed. Faster R-CNN [12] replaces the traditional non-CNN based ROI generation scheme with RPN to construct an two-stage object detector, which first generates ROIs using RPN and then performs classification and bounding box regression for each ROI.

2.2. Context information

In MultiPath network [3], four field-of-view are employed for each ROI to capture different levels of contextual information around the object. By increasing the padding ratio of ROIs such that the actual pooled region is larger than the object proposal itself, MS-CNN [13] can exploit the contextual information for object detection. The use of recurrent neural network (RNN) is another way to extract contextual information. In [5], four-direction IRNN [14] is applied to gather contextual information from four directions. Rolling recurrent network (RRN) [4] explores the contextual information from different convolutional layers in a rolling fashion.

2.3. Deformable model

Deformable part model (DPM) [15] is a widely used part-based method for highly variable object detection. Later it was formulated as a CNN in [16]. Spatial transform networks (STN) [17] introduces the spatial transformer to warp feature maps. The active convolution unit (ACU) was proposed in [18] to learn the shape of convolution through backpropagation so that the generalization of convolution can be achieved. The deformable convolution was proposed in [7] to break the limitation of the fixed geometric structure in the standard convolution. The deformable convolution features a convolutional layer that estimates 2D offsets to the regular grid sampling locations, and the sampling locations are adjusted based on the offsets to achieve a spatially adaptive convolution operation.

2.4. Multi-scale object detection

In MS-CNN [13], the RPN has multiple branches for detecting objects with different scales. In Inside-Outside Net [5] and MultiPath network [3], skip pooling is performed on multiple convolutional layers to obtain high-resolution features for small object detection like multi-stage features in [19]. In SSD [20], multi-scale object detection is achieved by assigning different detection convolutional layers on feature maps with different levels of resolution. In scale-dependent pooling (SDP) network [21], ROI pooling, classification, and bounding box regression for a certain object scale are performed on the convolutional feature map that has the most suitable level of resolution and abstraction for detection.

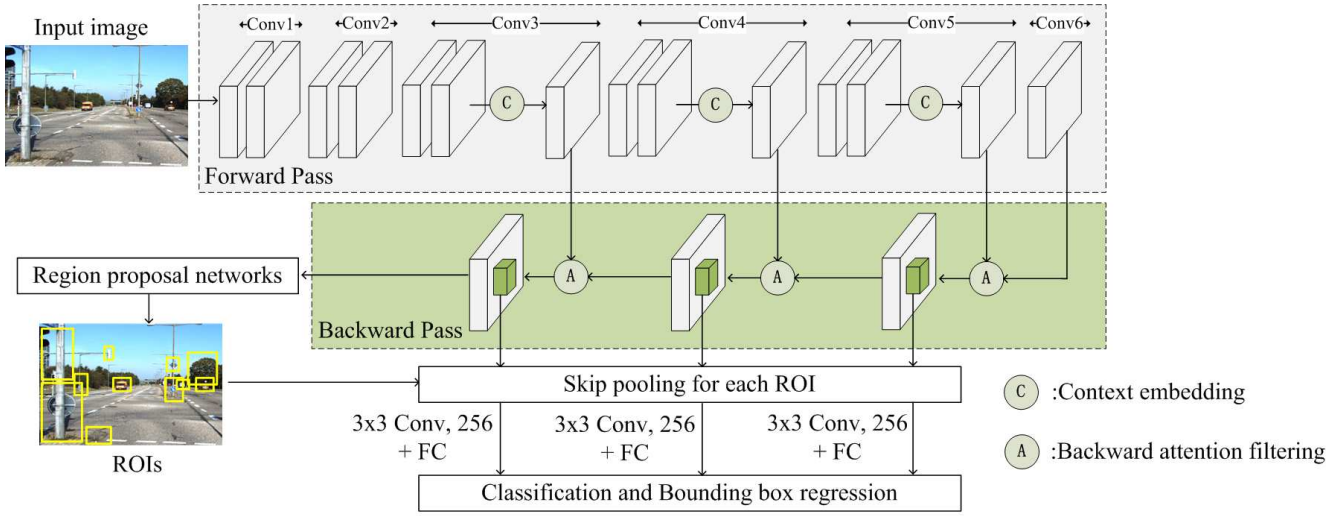


Figure 2. The overall architecture of the proposed network. During the forward pass, the input image is fed into a VGG-16 based feed-forward network to generate the feature maps. Context features are embedded using the location-aware deformable CNN. After the Conv6 layer is obtained, the backward pass applies the proposed backward attention filtering to filter feature maps from deep layers to shallow layers. ROIs are generated by region proposal networks, and ROI pooling is carried out for each ROI on three filtered feature maps. The pooled features are fed into the classification and bounding box regression subnetworks to obtain the detection result.

2.5. Attention mechanism

Attention mechanism has been utilized in many computer vision tasks. The diversified visual attention network (DVAN) was proposed in [22] to search the area with high attention value and zoom in the image for fine-grained object classification. In [23], weak semantic segmentation is applied as the attention map to regularize the feature map for pedestrian detection. In RON [24], an objectness map is generated and used as the attention map to suppress the features that belong to background areas. Aspect ratio attention bank and sub-region attention bank were proposed in [25] to refine pooled features for each ROI. Residual attention network was proposed in [26] for image classification. The attention map produced in [26] is both spatial and channel-wise, which means features on different location and channel are filtered differently.

3. Proposed method

3.1. Overview

The proposed method can be applied to different backbone networks such as VGG [27], ResNet [28], MobileNet [29], GoogleNet [30] and Inception ResNet [31]. Here, we use VGG-16 as the example to describe the overall architecture as shown in Figure 2. The network consists of three major components: the forward pass, the backward pass, and the object detection subnetworks. During the forward pass, the input image is fed into the backbone network that includes 14 convolutional layers. Three context embedding modules are inserted before Conv3_3, Conv4_3 and Conv5_3. In these modules, context features generated by the proposed location-aware deformable convolution are embedded with the features from standard convolution to obtain the enhanced Conv3_3, Conv4_3, and Conv5_3 layers. During the backward pass, the proposed backward attention filtering is carried out from deep layers to shallow layers. There are three backward attention filtering modules in the backward pass. Each module filters the input feature map using the output from the predecessor module. After the backward pass, three filtered feature maps (Conv3_3, Conv4_3, and Conv5_3) are obtained. They are fed into the RPN to generate the ROIs. For each ROI, ROI pooling is carried out on all three filtered feature maps. These pooled features are processed by additional layers and fused at the fully-connected layer. Finally, the fused features for each ROI are sent to the classification and bounding box regression subnetworks. The classification subnetwork predicts the class (pedestrian, cyclist, car, or background), while the bounding box regression subnetwork predicts the ROI's bounding box offsets with respect to the anchor box.

3.2. Location-aware deformable convolution

The standard convolutional unit [1] samples the input feature map at fixed locations and generates the output by computing the weighted sum of the samples. Recently, deformable convolution has been proposed to overcome the limitations of standard convolution. In deformable convolution [7], 2D offsets to the regular grid sampling locations in the standard convolution are estimated using an additional

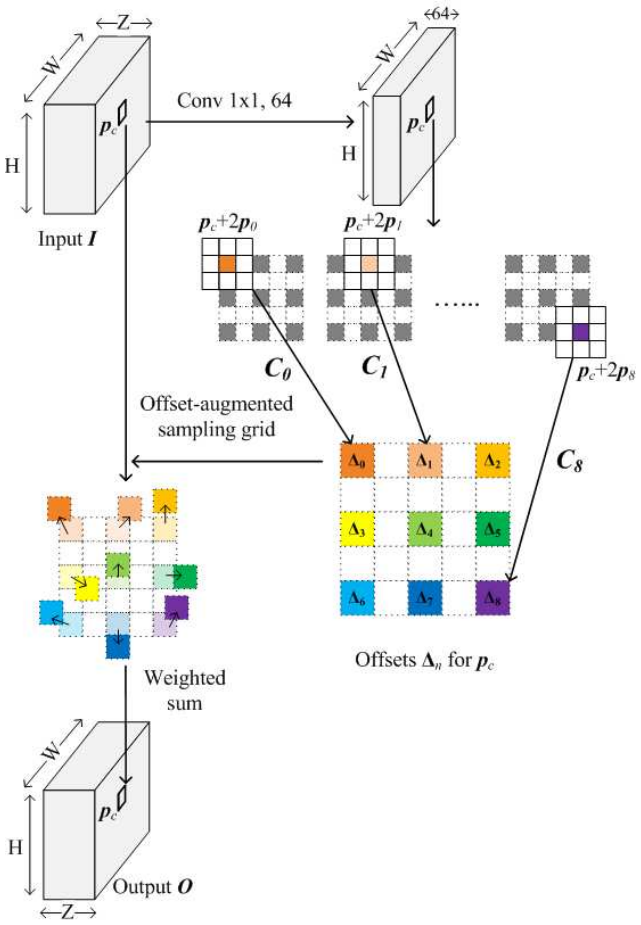


Figure 3. Example of 3×3 location-aware deformable convolution with the dilation size $D = 2$. H, W, and Z are the input feature map's height, width, and number of channels, respectively. Better viewed in color.

convolutional layer and the weighted sum of the samples at offset-augmented locations is computed to obtain the output feature.

In the deformable convolution module, there is only one convolutional layer for estimating all the offsets, which is based on the same receptive field as in the standard convolution. Estimating the offset for each input sample using the same receptive field and convolutional layer does not fully utilize the unique characteristics of each input, which may cause sub-optimal offset estimation. Besides, the receptive field is so small that surrounding features are not seen during the offset estimation, which makes it hard to capture useful context information. In this paper, we propose a location-aware deformable convolution module to capture the unevenly distributed context features. The proposed method adjusts the receptive field in offset estimation adaptively based on each input sample's location and surroundings.

Our proposed location-aware deformable convolution module for context feature extraction is depicted in Figure 3. Note that the offset estimation and the offset-augmented sampling take place in 2D spatial domain. Assume that the input feature map is $\boldsymbol{I}$, and the output feature map is $\boldsymbol{O}$, for each 2D location $\boldsymbol{p}_c$=(x_c, y_c) on the output feature map. The 3×3 deformable convolution that is centered on $\boldsymbol{p_c}$ is defined as:

$$\boldsymbol{O}(\boldsymbol{p}_c) = \sum_{n=0}^{8} \boldsymbol{W}(\boldsymbol{p}_n) \cdot \boldsymbol{I}(\boldsymbol{p}_c + D \cdot \boldsymbol{p}_n + \boldsymbol{\Delta}_n), \quad (1)$$

where $\boldsymbol{W}$ is the weight matrix. $\boldsymbol{p}_n \in G$ is a location in the 3×3 regular sampling grid G, and D is the dilation size. The input sample on the regular sampling grid without offset-augmentation is located at $\boldsymbol{p}_c + D \cdot \boldsymbol{p}_n$. After the offset $\boldsymbol{\Delta}_n$ for each input sample is estimated, the offset-augmented input sample is located at $\boldsymbol{p}_c + D \cdot \boldsymbol{p}_n + \boldsymbol{\Delta}_n$, which has an irregular and adaptive geometric structure to capture context information that does not have a fixed distribution. The sampling grid G has nine elements and is defined as:

$$G = \{(-1,-1),(-1,0),...,(0,0),...,(1,1)\}. \quad (2)$$

Before estimating the offsets for each input sample, a 1×1 convolution is applied to the input feature map to reduce the channel size to 64. Reducing the channel size is necessary to keep the computation cost cheap because the offset estimation is done individually to each input sample. After the 1×1 convolution, nine 3×3 convolutional layers $\boldsymbol{C}_n, n \in \{0,1,,8\}$ are applied to estimate the offset for each input sample. Unlike [7] where the convolution for offset estimation is always centered on $\boldsymbol{p}_c$, the center of the 3×3 convolutional layer $\boldsymbol{C}_n$ in the proposed method is located at $\boldsymbol{p}_c$+D·$\boldsymbol{p}_n$. Thus, each input sample's offset estimation is determined by its location and unique surroundings. For each input sample $\boldsymbol{p}_c + D \cdot \boldsymbol{p}_n$, $\boldsymbol{C}_n$ outputs the offset $\boldsymbol{\Delta}_n = (\Delta x_n, \Delta x_n)$, where Δx_n is the x coordinate of the offset, and Δy_n is the y coordinate of the offset. Note that the nine offset estimation convolutional layers do not share parameters and they are trained individually. By having a 3×3 convolutional layer centered on each input sample, the receptive field for offset estimation is extended to cover the area outside the original 3×3 sampling grid G.

After all nine offsets $\boldsymbol{\Delta}_n, n \in \{0,1,,8\}$ are obtained, the offset-augmented input samples are located at $\boldsymbol{p}_c + D \cdot \boldsymbol{p}_n + \boldsymbol{\Delta}_n$. Since the estimated offset $\boldsymbol{\Delta}_n$ is often a fractional number, interpolation is carried out to obtain the feature value of the fractionally sampled input. The weighted sum over all offset-augmented input samples is computed based on Equation (1) to obtain the output feature value $\boldsymbol{O}(\boldsymbol{p}_c)$. The output feature map $\boldsymbol{O}$ is obtained by estimating the offset and computing the convolution using Equation (1) to all inputs from the input feature map $\boldsymbol{I}$.

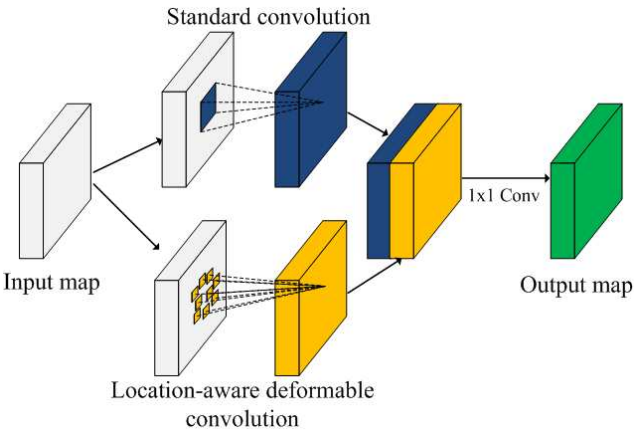


Figure 4. Context embedding module.

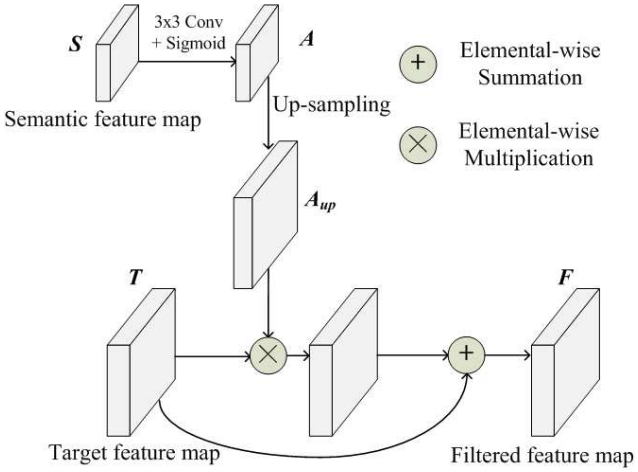


Figure 5. Backward attention filtering module.

3.3. Context feature embedding

The proposed context feature embedding module is illustrated in Figure 4. There are two links from the input feature map to the output feature map. The top link is the standard convolution which generates the standard feature map with the regular sampling grid. The bottom link applies the proposed location-aware deformable convolution which generates the context feature map using Equation (1). After the standard feature map and the context feature map are generated, these two features maps are concatenated, and a 1×1 convolution is applied on the concatenated map to generate the output feature map.

During the forward pass, context feature embedding is carried out on multiple convolutional layers to exploit the context information of different resolutions. Specifically, context feature embedding is performed to generate Conv3_3, Conv4_3, and Conv5_3 layers by using Conv3_2, Conv4_2, and Conv5_2 as the input, respectively. The dilation size D in location-aware deformable convolution is set to 2 to have a large and adaptive receptive field for a better context feature extraction. Section 4.3 includes more detailed information regarding the dilation setups.

3.4. Backward attention filtering

Multi-scale object detection, especially small object detection, relies heavily on the high-resolution features from shallow convolutional layers. While high-resolution features provide informative clues for small objects, they also contains distractive features which are not beneficial for RPN, classification, and bounding box regression subnetworks. To suppress the distraction while highlighting the informative high-resolution features, we filter the high-resolution feature maps with the low-resolution feature maps which are rich in semantically meaningful information.

The architecture of the proposed backward attention filtering module is given in Figure 5. There are two inputs for the attention filtering module: one is the target feature map $\boldsymbol{T}$ to be filtered, and the other is the semantic feature map $\boldsymbol{S}$ from the deeper convolutional layer, which is used to generate the attention map.

First, the semantic feature map $\boldsymbol{S}$ is processed by a 3×3 convolutional layer. The output $\boldsymbol{A}$ has the same channel size as the target feature map so that elemental-wise operation can be performed. The sigmoid function is used for non-linear activation. The attention map $\boldsymbol{A_{up}}$ is obtained by up-sampling $\boldsymbol{A}$ to the same spatial size as the target feature map $\boldsymbol{T}$. Elemental-wise multiplication is carried out between the attention map $\boldsymbol{A_{up}}$ and the target feature map $\boldsymbol{T}$. The filtered feature map $\boldsymbol{F}$ is obtained by elemental-wise summation between $\boldsymbol{T}$ and $\boldsymbol{T}\cdot\boldsymbol{A_{up}}$. Assume that the feature at the spatial location (x, y) and channel c on the target feature map is $\boldsymbol{T}(x, y, c)$. The filtered feature on the output feature map $\boldsymbol{F}$ can be formulated as:

$$\boldsymbol{F}(x, y, c) = (1 + \boldsymbol{A_{up}}(x, y, c)) \cdot \boldsymbol{T}(x, y, c) \qquad (3)$$

Equation (3) is similar in spirit to residual-connection [28], which is used to prevent the filtered feature value from degradation. The filtered feature map $\boldsymbol{F}$ is used as the semantic feature map for the next filtering module as well as the feature map for the object detection subnetworks. In the proposed network, there are three such attention filtering modules to filter the Conv 5_3, Conv 4_3, and Conv 3_3 layers in a backward connection as shown in Figure 2. After the backward filtering is complete, three filtered feature maps are obtained and sent to the object detection subnetworks.

3.5. Object detection subnetworks

The object detection subnetworks are based on the Faster R-CNN [12] architecture, which is a two-stage detector. The first stage is to generate object proposals or ROIs with the RPN. The setup of the RPN is similar to the original one proposed in [12]. The second stage is to perform ROI pooling for each ROI from the three feature maps obtained from the backward attention filtering. We use ROI pooling from multiple feature maps of different resolutions to improve the performance for multi-scale object detection. As shown in Figure 2, for each ROI, we use skip pooling [5] to extract a fixed-length feature descriptor from the filtered Conv3_3, Conv4_3, Conv5_3 feature maps.

As in [3] and [32], we apply a late feature fusion that performs feature concatenation at fully-connected layers. Specifically, we employ a 3×3 convolutional layer and a fully-connected layer for the pooled Conv3_3, Conv4_3, and Conv5_3 features. Each convolutional layer and fully-connected layer are trained individually to exploit the uniqueness of each pooled feature. The output size of the fully-connected layer is 1024, which gives a good balance between performance and speed. After all fully-connected features are obtained, they are concatenated into a vector, resulting in a final feature size of 3072.

After the concatenated features are obtained, the classification and bounding box regression subnetworks take the features as the input and make the final prediction on the ROI's class and its bounding box offsets. The classification subnetwork outputs a class score C_{class}. The bounding box regression subnetwork outputs the bounding box offsets $t = [t_x, t_y, t_w, t_h]$, where t_x, t_y, t_w, t_h are the offsets with respect to the ROI's x coordinate, y coordinate, width, and height, respectively. They are parameterized using the method in [33]. The total loss function L is a multi-task loss defined as:

$$L = L_{cls}(C_{class}, C_{GT}) + \alpha \times L_{bbox}(t, t_{GT}), \quad (4)$$

where C_{GT} is the ground truth for multi-class classification and t_{GT} is the ground truth for bounding box regression. The classification loss L_{cls} is the cross-entropy loss and the bounding box regression loss L_{bbox} is the smooth L_1 loss. α is equal to 1 when the C_{GT} is the non-background class. Otherwise, α is equal to 0.

4. Experiments

4.1. Dataset

The KITTI benchmark dataset [2] is a real-world computer vision dataset for autonomous driving. The 2D object detection benchmark consists of 7481 training images and 7518 testing images. The object categories are cars, pedestrians, and cyclists. The evaluation metric is based on the average precision (AP). Since KITTI dataset only provides the ground truth annotation for training images, to evaluate the design or optimize the network setup, one creates a validation set from the training images. In our case, we divide the training set into two parts. One half contains 3741 images, which are used as the training set. The other half contains 3740 images and used as the validation set. In addtion, we also evaluate our proposed method on PASCAL VOC2007 dataset [34] for general object detection. The dataset contains 9963 images, and there are 20 object classes.

4.2. Implementation details

VGG-16 and ResNet. The proposed method can be integrated with different backbone networks. We have implemented the proposed method based on VGG-16 and ResNet-18 for KITTI dataset. ResNet-18 is chosen because it is a computationally cheap network and therefore is suitable for real-time applications such as autonomous driving. For experiments on PASCAL VOC dataset, VGG-16 and ResNet-101 are used. To switch the backbone from VGG-16 to ResNet, all that need to be done is to connect the context embedding module and the backward attention filtering module accordingly. During the ROI pooling process, the spatial size of the pooled features is set to 3×3. After the detection scores and the bounding box offsets for each ROI are obtained, non-maximum suppression with the IoU threshold of 0.5 is carried out to generate the final detection results.

Training Details. During the training stage, positive examples are defined as the sampled regions that have an IoU above 0.5 with the ground truth annotation. Meanwhile, sampled regions with the IoU below 0.3 are taken as the negative examples. The optimization of the network is done by stochastic gradient descent (SGD). The learning rate is set to 0.0005 for the VGG-16 version, and 0.00025 for the ResNet-18 version. The momentum is 0.9. The maximum iteration is set to 20000 for RPN and 40000 for the classification and bounding box regression subnetworks.

Software and Hardware. Caffe deep learning toolbox [35] with MATLAB interface is used as the software. The hardware is based on the Intel Core i7-6700 CPU and the NVIDIA Titan X GPU with 12GB memory.

4.3. Design evaluation

In this section, we evaluate the effectiveness of each proposed component and compare the performance with reference methods on KITTI validation set.

Comparison with Faster R-CNN. We compare the proposed method with Faster R-CNN [12] for VGG-16 and ResNet-18 backbone networks. The comparison results are given in Table 1. To evaluate the effectiveness of the proposed context embedding and backward attention filter-

Method	Backbone	AP (%) Car			AP (%) Cyclist			AP (%) Pedestrian			PASCAL
		Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard	VOC 2007
Faster R-CNN [12]	VGG-16	87.35	87.09	72.50	89.05	74.06	70.57	77.10	73.08	65.19	73.2
Ours (a)	VGG-16	91.39	90.66	80.17	**91.71**	81.83	77.27	82.04	78.09	68.83	75.7
Ours (b)	VGG-16	90.70	90.32	75.95	90.03	78.44	72.38	80.77	77.80	68.18	74.2
Ours (c)	VGG-16	**92.22**	**91.74**	**81.51**	91.64	**83.03**	**78.77**	**83.29**	**79.18**	**70.68**	76.1
Faster R-CNN [12]	ResNet-18/101	85.56	83.56	69.12	85.51	72.59	68.22	76.37	72.07	63.74	76.4
Ours (a)	ResNet-18/101	90.49	89.45	79.65	86.71	79.76	73.27	79.68	75.66	68.16	79.0
Ours (b)	ResNet-18/101	90.12	89.50	78.96	86.34	78.59	70.22	77.75	73.25	65.51	77.9
Ours (c)	ResNet-18/101	91.73	90.24	80.08	87.75	80.23	75.27	80.06	76.93	68.47	**79.8**

Table 1. Comparison with the baseline Faster R-CNN [12] on VGG-16 [27] and ResNet [28] backbone networks. Note that ResNet-18 is for KITTI datset, and ResNet-101 is for PASCAL VOC 2007 dataset.

Method	D_1	D_2	AP (%) at moderate difficulty		
			car	cyclist	pedestrian
Method A (Deformable convolution [7])	1	1	86.70	74.89	73.96
	2	2	89.26	80.84	77.25
Method B (Location-aware deformable convolution)	1	1	87.82	75.24	74.93
	1	2	87.35	75.19	74.25
	2	1	**90.66**	**81.83**	**78.09**
	2	2	90.62	80.47	77.35
Method C (Standard convolution)	n/a	n/a	87.09	74.06	73.08

Table 2. Comparison with the original deformable convolution on different dilation setups on KITTI validation set. D_1 indicates the dilation size of the deformable convolution. D_2 indicates the dilation size of the convolution for offset estimation.

Method	AP (%) at moderate difficulty		
	car	cyclist	pedestrian
Weak segmentation [23] [24]	86.83	74.59	73.55
Residual attention [26]	88.92	77.28	76.76
Ours	**90.32**	**78.44**	**77.80**

Table 3. Comparison with reference attention mechanisms on KITTI validation set.

Method	Runtime	Car	Cyclist	Pedestrian
Faster R-CNN [12]	2 sec	79.11	62.81	65.91
RRC [4]	3.6 sec	90.19	76.47	75.33
Sensekitt [38]	4.5 sec	90.00	72.50	67.28
SDP+RPN [21]	0.4 sec	89.42	73.08	70.20
SubCNN [39]	2 sec	88.86	70.77	71.34
MS-CNN [13]	0.4 sec	88.83	74.45	73.62
Deep3DBox [40]	1.5 sec	88.86	73.48	n/a
DeepStereo [41]	3.4 sec	88.75	65.72	67.32
Ours/VGG-16	0.22 sec	88.99	74.65	73.96
Ours/ResNet-18	0.14 sec	86.61	72.22	71.85

Table 4. Comparison with other state-of-the-art methods on KITTI test set at moderate difficulty.

Method	Backbone	mAP(%)
Faster R-CNN [12]	VGG-16/ResNet-101	73.2/76.4
ION [5]	VGG-16	75.6
Deformable CNN [7]	ResNet-101	78.7
SSD [20]	VGG-16	74.3
YOLO9000 [42]	DarkNet-19	73.7
RON [24]	VGG-16	75.4
FPN [43]	ResNet-101	80.5
R-FCN [44]	ResNet-101	76.6
Ours	VGG-16/ResNet-101	76.1/79.8

Table 5. Comparison with other state-of-the-art methods on PASCAL VOC 2007 test set.

ing module, we conduct experiments on three setups. The setup **Ours (a)** features the context embedding module using location-aware deformable convolution only. The setup **Ours (b)** features the backward attention filtering module only. The setup **Ours (c)** features both modules. The object detection subnetworks for Faster R-CNN has the same architecture as described in Section 3.5. From the results in Table 1, it can be seen that both the context embedding and the backward attention filtering improve the performance on both KITTI and PASCAL VOC datasets. On KITTI validation set, context embedding has an average 4.8% and 5.1% AP improvement on VGG-16 and ResNet-18, respectively. The backward attention filtering achieves a 2.8% and 3.6% improvement in AP on VGG-16 and ResNet-18, respectively. On PASCAL VOC test set, context embedding has a 2.5% and 2.6% AP improvement on VGG-16 and ResNet-101, respectively. The backward attention filtering achieves a 1.0% and 1.5% improvement in AP on VGG-16 and ResNet-101, respectively. By combining both modules, the setup **Ours (c)** has the best performance.

Location-aware deformable convolution. Here, we evaluate the effectiveness of the location-aware deformable convolution module by comparing it with the original deformable convolution [7] under different dilation setups.

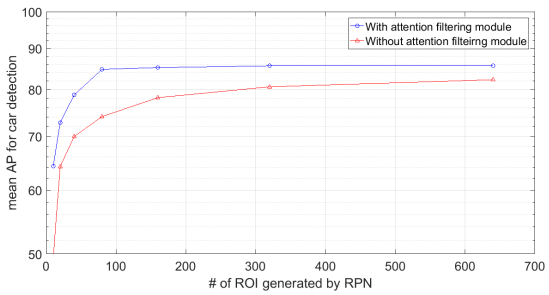


Figure 6. Effect of changing the number of ROIs on KITTI validation set. Car detection is evaluated because cars appear more frequently than cyclists and pedestrians. The mean AP is calculated from all difficulties.

The network setup is the same as the VGG-16 version of **Ours (a)** in Table 1, except that Method B uses the proposed location-aware deformable convolution, and Method A uses the original deformable convolution to extract context features. We test different combinations of dilation setups. Note that the original deformable convolution uses the same dilation setup for offset estimation and convolution. The comparison results are shown in Table 2. We can observe that a decent performance improvement is obtained when the deformable convolution's dilation size D_1 is set to 2. This observation suggests the usefulness of context information by increasing the receptive field of convolution. Among all dilation setups, the best result is obtained when the dilation size is 2 for deformable convolution and 1 for offset estimation convolution layers.

Backward Attention filtering. We compare the performance of the proposed attention filtering module with two popular attention mechanisms that can be used for object detection: a weak semantic segmentation-based attention module [23] [24] and a residual attention module [26]. Since the original methods in [23] [24] [26] are not tested on KITTI dataset, we implement them based on the description given in the corresponding publication. The weak semantic segmentation subnetwork is built based on FCN [36], and it is trained by labeling all pixels inside the positive bounding box as 1 and background as 0. For the residual attention method, we replace the proposed backward attention module with the feed-forward residual attention module. All other components are the same as the VGG-16 version of **Ours (b)** in Table 1. Table 3 shows the comparison results. It is observed that the backward attention filtering has the best performance.

ROI reduction. We investigate how the backward attention filtering module helps speed up the detection. In a Faster R-CNN based method, a larger number of ROIs slow down the processing speed dramatically [37]. By highlighting the features for the target object using the attention filtering module, the number of ROIs needed to reach a good performance can be reduced. Figure 6 compares the mean AP for car detection based on the number of ROIs with and without the backward attention filtering module. It can be observed that after applying the attention filtering module, the number of ROIs required to achieve a good performance is reduced by over 50%. As a result of processing a smaller number of ROIs for each frame, the runtime is reduced. For the evaluation on the validation and testing set, the number of ROIs for each frame is set to 150.

4.4. Comparison with state-of-the-art methods

We compare the performance with other state-of-the-art vision-based multi-class detection methods on KITTI and PASCAL VOC 2007 test set. The proposed network is trained using all images from the training set. All hyper-parameters are the same as the training setup described in Section 4.2. Table 4 and Table 5 show the performance comparison results in terms of mAP. On KITTI dataset, our method has the second-best performance on pedestrian and cyclist categories. On car detection task, our method has the 4th best performance. Since a small number of ROIs are needed after applying the backward attention filtering, the proposed method has the fastest speed among the top-performing methods. Especially, our method achieves a runtime of 0.14 seconds per frame with a comparable average precision based on ResNet-18. On PASACL VOC test set, our method outperforms all other methods except for FPN.

5. Conclusion

In this paper, we propose a location-aware deformable convolution and a backward attention filtering module to improve the performance of multi-class, multi-scale object detection for autonomous driving. The location-aware deformable convolution adaptively extracts unevenly distributed context features, which are embedded with the standard convolutional features to build strong and comprehensive features for detecting objects in a complex scene. To further improve the performance and reduce the number of ROIs needed, the backward attention filtering module utilizes the high-level semantic features from deep convolutional layers to highlight informative high-resolution features and suppress the distractive ones. By combining the two proposed methods into a forward-backward network, the proposed detection network achieves good performance on KITTI and PASCAL VOC dataset with a fast speed among the top-performing methods.

Acknowledgement. This work is supported by the Industrial Core Technology Development Program of MOTIE/KEIT, KOREA. [#10083639, Development of Camera-based Real-time Artificial Intelligence System for Detecting Driving Environment and Recognizing Objects on Road Simultaneously].

References

[1] A. Krizhevsky, I. Sutskever, and G.E. Hinton. ImageNet classification with deep convolutional neural networks. in *Proceedings of Neural Information Processing Systems*, 2012.

[2] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[3] S. Zagoruyko, A. Lerer, T. Y. Lin, P. O. Pinheiro, S. Gross, S. Chintala, and P. Doll. A MultiPath network for object detection. in *Proceedings of British Machine Vision Conference*, 2016.

[4] J. Ren, X. Chen, J. Liu, W. Sun, J. Pang, Q. Yan, Y. Tai, and L. Xu. Accurate single stage detector using recurrent rolling convolution. in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[5] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick. Inside-Outside net: Detecting objects in context with skip pooling and recurrent neural networks. in *arXiv: 1512.04143*, 2015.

[6] F. Yu, and V. Koltun. Multi-scale context aggregation by dilated convolutions. in *Proceedings of International Conference on Learning Representations*, 2016.

[7] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. in *Proceedings of IEEE International Conference on Computer Vision*, 2017.

[8] N. Dalal, and B. Triggs. Histograms of oriented gradients for human detection. in *Proceedings of IEEE Conference on Computer Vision Structure Recognition*, 2005.

[9] P. Dollr, Z. Tu, P. Perona, and S. Belongie. Integral channel features. in *Proceedings of British Machine Vision Conference*, 2009.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[11] R. Girshick. Fast R-CNN. in *Proceedings of IEEE International Conference on Computer Vision*, 2015.

[12] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[13] Z. Cai, Q. Fan, R.S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. in *Proceedings of European Conference on Computer Vision*, 2016.

[14] Q.V. Le, N. Jaitly, and G.E. Hinton. A simple way to initialize recurrent networks of rectified linear units. in *arXiv: 1504.00941*, 2015.

[15] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[16] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. in *arXiv preprint arXiv:1409.5403*, 2014.

[17] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. in *Proceedings of Neural Information Processing Systems*, 2015.

[18] Y. Jeon, and J. Kim. Active convolution: Learning the shape of convolution for image classification. in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[19] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun. Pedestrian detection with unsupervised multi-stage feature learning. in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg. SSD: Single shot multibox detector. in *Proceedings of European Conference on Computer Vision*, 2016.

[21] F. Yang, W. Choi, and Y. Lin. Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers. in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[22] B. Zhao, X. Wu, J. Feng, Q. Peng and S. Yan. Diversified visual attention networks for fine-grained object classification. in *IEEE Transactions on Multimedia*, 2017.

[23] G. Brazil, X. Yin, and X. Liu. Illuminating pedestrians via simultaneous detection & segmentation. in *Proceedings of IEEE International Conference on Computer Vision*, 2017.

[24] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen. RON: Reverse connection with objectness prior networks for object detection. in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[25] Y. Zhai, J. Fu, Y. Lu, and H. Li. Feature selective networks for object detection. in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[26] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[27] K. Simonyan, and A. Zisserman. Very deep convolutional networks for large-scale image recognition. in *arXiv: 1409.1556*, 2014.

[28] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[29] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. in *arXiv:1704.04861*, 2017.

[30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[31] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, Inception-Resnet and the impact of residual connections on learning. in *arXiv preprint arXiv:1602.07261*, 2016.

[32] S. Gidaris, and N. Komodakis. Object detection via a multi-region & semantic segmentation-aware CNN model. in *Proceedings of IEEE International Conference on Computer Vision*, 2015.

[33] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[34] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, The PASCAL visual object classes challenge: A retrospective. in *International Journal of Computer Vision*, 2015.

[35] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Derrell. Caffe: Convolutional architecture for fast feature embedding. in *arXiv: 1408.5093*, 2014.

[36] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[37] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[38] B. Yang, J. Yan, Z. Lei, and S. Li. Craft objects from images. in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[39] Y. Xiang, W. Choi, Y. Lin, and S. Savarese. Subcategory-aware convolutional neural networks for object proposals and detection. in *Proceedings of IEEE Winter Conference on Applications of Computer Vision*, 2017.

[40] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka. 3D bounding box estimation using deep learning and geometry. in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[41] C. Pham, and J. Jeon. Robust object proposals re-ranking for object detection in autonomous driving using convolutional neural networks. in *Signal Processing: Image Communication*, 2017.

[42] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. in *Proceedings of IEEE International Conference on Computer Vision*, 2017.

[43] T. Lin, P. Dollr, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature Pyramid Networks for Object Detection. in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[44] J. Dai, Y. Li, K. He, J. Sun. R-FCN: Object Detection via Region-based Fully Convolutional Networks. in *Proceedings of Neural Information Processing Systems*, 2016.