Pattern Recognition 123 (2022) 108369

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

ADCNN: Towards learning adaptive dilation for convolutional neural networks

Jie Yao[a,1], Dongdong Wang[b,1], Hao Hu[c,1], Weiwei Xing[a,**], Liqiang Wang[b,*]

[a] *School of Software Engineering, Beijing Jiaotong University, Beijing, 100044, China*
[b] *Department of Computer Science, University of Central Florida, 32816, USA*
[c] *KTH Royal Institute of Technology, Stockholm, 11428, Sweden*

ARTICLE INFO

Article history:
Received 19 May 2021
Revised 29 August 2021
Accepted 9 October 2021
Available online 16 October 2021

Keywords:
Adaptive dilated convolution
Representation learning
Image classification

ABSTRACT

Dilated convolution kernels are constrained by their shared dilation, keeping them from being aware of diverse spatial contents at different locations. We address such limitations by formulating the dilation as trainable weights with respect to individual positions. We propose Adaptive Dilation Convolutional Neural Networks (ADCNN), a light-weighted extension that allows convolutional kernels to adjust their dilation value based on different contents at the pixel level. Unlike previous content-adaptive models, ADCNN dynamically infers pixel-wise dilation via modeling feed-forward inter-patterns, which provides a new perspective for developing adaptive network structures other than sampling kernel spaces. Our evaluation results indicate ADCNNs can be easily integrated into various backbone networks and consistently outperform their regular counterparts on various visual tasks.

1. Introduction

Convolutional kernels are critical components for Convolutional Neural Networks (CNNs), which have been dominant approaches for majority of computer vision tasks in recent years [1]. Their power relies on the ability of hierarchically representing spatial features over input regions called Receptive Fields (RFs), by stacking a number of convolutional layers into deep structures [2]. Nowadays, among common practices for designing CNN architectures, which usually prefer large RFs in order to achieve superior performances, Dilated Convolutional Kernels (DCKs) serve as a popular choice not only because of their simplicity, but also effectiveness [3,4]. Unlike their conventional equivalents, DCKs are able to exponentially enlarge RFs without increasing kernel sizes. CNN models with dilated kernels also report the impressive results on fundamental tasks such as semantic segmentation [4]. Moreover, DCKs perform well in some more specific tasks such as object detection with mutil-model [5], demonstrating significant performance gain by employing dilated convolutional kernels.

To further improve the dilated kernels, two obvious problems that universally reside in most of existing dilated CNN structures need to be properly tackled: fixed RF sizes and manually selected dilation range. First, the dilation value for a convolutional layer is shared across all pixels, which means that every output location has the same size of RF. However, this could be very counter-intuitive: sizes of Region of Interest (ROIs) usually vary dramatically over different positions, and thus, sizes of RFs are also expected to be adjusted accordingly to encode diverse spatial information. Therefore it is reasonable to believe a monosized RF across every position is hard to capture such enormous intra and inter sample diversities especially for large-scale, high-resolution image datasets.

Second, the mainstream approaches of selecting a dilation value is mainly feature-independent; for each dilated convolution layer, we need to specify dilation values arbitrarily before it can be integrated into the base structure. This usually requires a strong domain knowledge about input and output contexts for hand-crafting; and for many specific tasks, there is no clear guidance available for selecting proper dilation values in practice. In recent years, deformable convolutional neural networks [6,7] have been proposed to enhance the transformation modeling capability of CNNs by augmenting the spatial sampling locations in the modules with additional offsets and learning the offsets from the target tasks. However, they set a small value such as 1 for offset as the upper bound, which means that it usually needs to stack deformable convolutional layers to enlarge the RFs and get better performance. On the other aspect, if we choose a larger value as the upper bound of the offset, it will degenerate the deformable

[*] Corresponding author.
[**] Co-corresponding author.
E-mail addresses: wwxing@bjtu.edu.cn (W. Xing), lwang@cs.ucf.edu (L. Wang).
[1] Contributed equally to this work.

https://doi.org/10.1016/j.patcog.2021.108369
0031-3203/© 2021 Elsevier Ltd. All rights reserved.

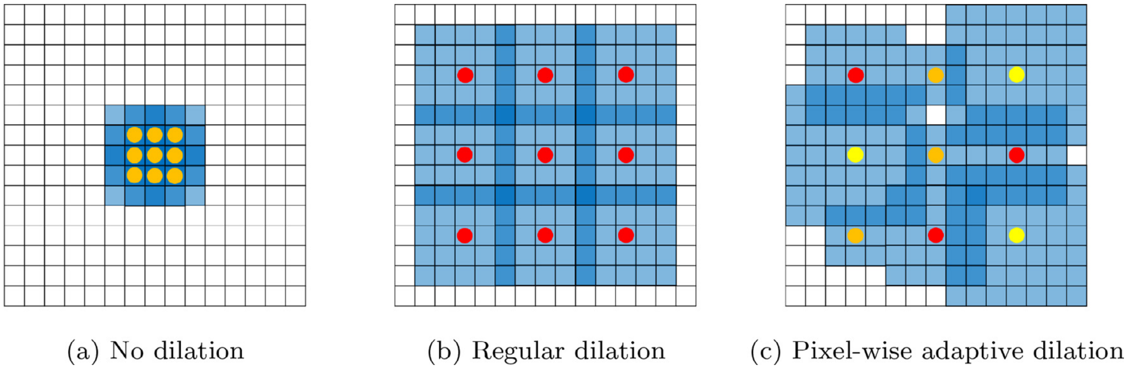


Fig. 1. Comparison of regular and pixel-wise adaptive dilation. Different colors stand for different dilation.

convolutional layer into an attention mechanism due to some incorrect focus on minute details due to deformable convolution endows flexibility to the kernel, and the flexibility is enhanced with the increase of the offset, which makes learning a proper offset need either a well-prepared dataset or an adequate training process.

In this paper, we answer the above challenges by combining the dilation selection with conventional CNN modules and incorporating them into a unified data-driven framework. We propose Adaptive Dilation Convolutional Neural Networks (ADCNN), a simple yet powerful extension for general DCKs, which treats dilation values as learnable weights and can be jointly optimized with other CNN weights in an end-to-end fashion. As shown in Fig. 1, in the newly formulated ADCNN kernels, dilation is learned to change at different input positions to reflect input spatial diversity, resulting in dynamic RFs with irregular shapes in a single layer. In practice, there are two major difficulties to overcome.

How to decide the dilation value online. We handle this by regarding the dilation as a function of input at individual pixels. More specifically, the function samples dilation values through certain probability distributions that are conditioned by pixel-wise input features. To solve non-differentiable nature of general sampling process, we approximate it by employing Gumbel-Softmax [8] as a differentiable estimation to keep ADCNN end-to-end trainable.

What are proper dilation values for inputs. Since there is no clear explanation on how network layers work, we believe that it still remains an open question and can only be answered with valid hypotheses. For ADCNN kernels, we make the assumption that dilation values are related to inter-layer patterns between convolution layers due to their hierarchical nature. In such cases, RF size at each location is adjusted based on information flows between corresponding inter-layer pixels during forward propagation.

Following the strategies described above, ADCNN-kernels evolve into light-weighted modules that can be easily plugged into various CNN architectures. Moreover, sampling dilation space through inter-layer pattern modeling also demonstrate that adaptive networks can be achieved in a simpler manner without engaging high dimensional spaces. We evaluate the proposed ADCNNs via several fundamental tasks including large-scale, fine-grained visual classification, semantic segmentation and optical flow estimation. Moreover, several ablation studies are performed to examine various properties of ADCNNs. Our experimental results indicate in most cases ADCNNs are able to consistently yield better performances across various popular backbone architectures with trivial cost.

The rest of this paper is organized as follows. We review relevant literature in Section 2, then ADCNNs are elaborated in Section 3. Sections 4–6 demonstrate experimental results. Section 7 concludes the proposed method and discusses the limitations and future work.

2. Related work

2.1. Content-adaptive networks

This research direction is focused on building dynamic internal structures via data-driven approaches to better leverage larger spatial variations from inputs. A set of related techniques tend to develop differentiable approximations for traditional image-adaptive filters and integrate them as end-to-end trainable layers for CNN models. For example, Liang et al. [9] proposed Spatio-Temporal adaptive and Channel selective Correlation Filters (STCCF) for robust tracking. Zhang et al. [10] introduced their learning modulation filter networks (LMFNs) to improve detection performance. These approaches conduct content-adaptive enhancements in separate layers without interacting with convolution kernels. Another set of techniques propose the idea of directly generating kernel weights based on layer inputs and extend it with attention mechanism as well as other task-specific improvements. For example, Jia et al. [11] proposed the Dynamic Filter Network, where filters are generated dynamically conditioned on an input in a sample-specific way; Su et al. [12] proposed a pixel-adaptive convolution (PAC) operation in which the filter weights are multiplied with a spatially varying kernel that depends on learnable, local pixel features; Wu et al. [13] proposed a dynamic filtering strategy with large sampling field for ConvNets (LS-DFN) to learn dynamic position-specific kernels and takes advantage of very large receptive fields and local gradients. Besides, there are some researches focus on how to effectively enlarge receptive fields (RFs) in order to achieve better performance. Zhen et al. [14] used two affine transformation layers to operate feature maps, so the RFs in the following layers will be changed accordingly. Shelhamer et al. [15] introduced their free-form filters and structured Gaussian filters to optimize the RFs. However, most of them rely on additional modules with large kernel sizes, being incapable of scaling up to more general structures.

2.2. Dilated convolutional networks

Comparing to the above approaches to build content-adaptive networks, dilated convolution kernels [3], which support exponential expansion of the receptive field without loss of resolution or coverage, become a popular choice as it can exponentially increase RF sizes while maintaining small kernel sizes. Various works benefit from this characteristic. For example, Li et al. [16] proposed an end-to-end learning framework for monocular depth estimation using dilated convolution and hierarchical feature fusion to learn the scale-aware depth cues. Wang et al. [17] constructed a new learning architecture using the dilated convolutional residual network to generate high-frequency details and eliminate color discrepancies for ensuring visual consistency in the completed im-

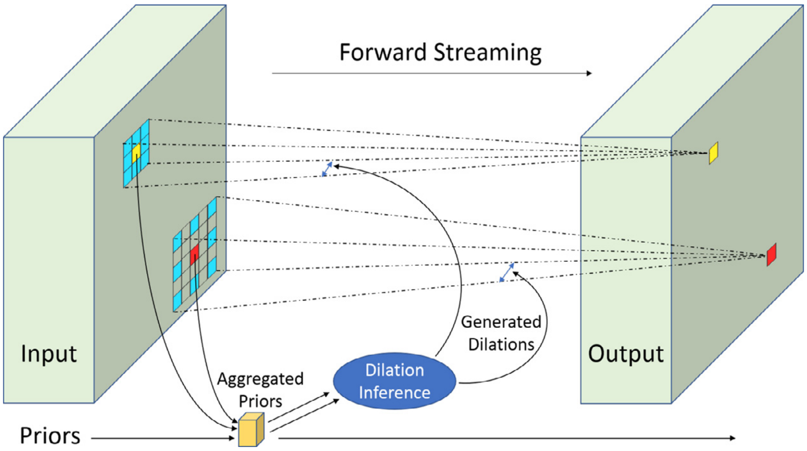


Fig. 2. Overview of a ADCNN kernel.

age. Chen et al. [18] designed location-aware multi-dilation module (LAMD) in the classifiers for robust detection. However, dilated convolutional kernels could also lead to negative impacts, such as sparsity and “gridding” effect [4]. Unlike static RFs produced by dilation, in recent years, a new kind of dynamic convolutional network, which is Deformable Convolutional Networks [6], has been proposed to enhance the transformation modeling capability of CNNs. Based on the idea of augmenting the spatial sampling locations in the modules with additional offsets, Deformable Convolutional kernels learn such offsets from the target tasks, without additional supervision. Later, Zhu et al. [7] proposed a reformulation of Deformable ConvNets that improves its ability to focus on pertinent image regions. However, most of them rely on sufficient training data and the training pattern will be different from the original backbone network.

3. Pixel-wise adaptive dilated convolution

Now we elaborate the proposed approach for extending conventional dilated convolution kernels into ADCNN kernels. Without loss of generality, we assume all the convolutions in the rest of this paper are 2D operations. Suppose we are considering the $(l-1)$-th layer, whose input is $\mathbf{X}^{l-1}$ with $\mathbf{X}^{l-1} \in \mathbb{R}^{w^{l-1} \times h^{l-1}}$. w^{l-1}, h^{l-1} are the width and height of the input x^{l-1} respectively. $\mathcal{K}_{\mathbf{W};d}$ is a dilated convolutional kernel with dilation value d and weights $\mathbf{W}$. The output of convolution between $\mathcal{K}$ and $\mathbf{X}$ is

$$\mathbf{Y}^l_{i,j} = \sum_{m=0}^{K}\sum_{n=0}^{K} \mathbf{w}_{m,n} \times \mathbf{X}^{l-1}_{i+dm,j+dn} \tag{1}$$

where K is the kernel size and i, j are coordinates for dimensions w and h, respectively. Apparently, d is a constant variable independent to i and j. Our goal is to convert d into a function $\mathcal{D}_{i,j}$ such that the output of $\mathcal{D}_{i,j}$ could be aware of location-specific contents. More specifically, we treat $\mathcal{D}_{i,j}$ as an inference process that generates dilation values by sampling from position-dependent hidden distributions. Fig. 2 sketches the basic idea of a ADCNN kernel.

3.1. Dilation inference

Sampling dilation values directly from categorical distributions is straightforward. However, gradients are unable to backpropagate through sampled nodes in such cases, making the entire training process intractable. Inspired by [19], we employ Gumbel-Softmax (GS) [8] as $\mathcal{D}_{i,j}$ to approximate the inference of discrete dilation values. Suppose that there are D valid options for dilation value, and $\mathbf{d}_{i,j} \in [0,1]^D$ is the estimation of one-hot vector that corresponds to the dilation value at position (i, j), then sampling $\mathbf{d}_{i,j} \sim \mathrm{GS}(\mathbf{h}_{i,j})$ can be achieved by

$$\mathbf{d}_{i,j} = \mathcal{D}_{i,j}(\mathbf{h}) = \frac{\exp((\mathbf{h}_{i,j} + \mathbf{g}_{i,j})/\tau)}{\sum \exp((\mathbf{h}_{i,j} + \mathbf{g}_{i,j})/\tau)} \tag{2}$$

where $\sum$ means summation of all tensor elements here; $\mathbf{h}, \mathbf{h}_{i,j}$ are content-related hidden priors and their subtensors at each positions, respectively; $\mathbf{g}_{i,j} \in \mathbb{R}^D$ are i.i.d. samples drawn from the Gumbel$(0,1)$ distribution and τ controls how much the GS is close to a true categorical distribution.

3.2. Hidden prior generation

As mentioned in Section 1, we believe dilation adaptation should be governed by feature hierarchy, hence build up our dilation inference mechanism upon inter-layer pattern modeling to capture dependencies between abstraction levels. We consider aggregation as a feasible way and will generate hidden priors $\mathbf{h}$ through sequentially aggregating multiple $\mathbf{Y}$ from hierarchical layers. Let l denote the newly added layer index, there are several aggregation options for inter-layer patterns modeling.

Recurrent Aggregation. A straightforward way for sequential aggregation can be written as

$$\mathbf{h}^l_{i,j} = f(\mathbf{W}^l_h \mathbf{h}^{l-1}_{i,j} + \mathbf{U}^l_h \mathbf{Y}^{l-1}_{i,j}) \tag{3}$$

where $\mathbf{W}^l_h$ and $\mathbf{U}^l_h$ are 1×1 kernels weights with output channel of D; $f(\cdot)$ is a non-linear activation function. In this case, $\mathbf{h}^l_{i,j}$ continuously accumulates information from each layer as l goes deeper, implying layers are highly dependent on each other to mutually decide proper RF sizes.

Gated Aggregation. To model inter-layer pattern smarter, we introduce a gate variable $\mathbf{a}^l_h$ to modulate information from each layer in a data-driven manner. We use a similar way to [20] for computing $\mathbf{a}^l_h$, with which the entire aggregation can be formulated as following

$$\mathbf{h}^l_{i,j} = f(\mathbf{a}^l_h \circ (\mathbf{W}^l_h \mathbf{h}^{l-1}_{i,j}) + (1 - \mathbf{a}^l_h) \circ (\mathbf{U}^l_h \mathbf{Y}^{l-1}_{i,j})) \tag{4}$$

$$\mathbf{a}^l_h = \sigma(\mathbf{W}^l_a \mathbf{h}^{l-1}_{i,j} + \mathbf{U}^l_a \mathbf{Y}^{l-1}_{i,j}) \tag{5}$$

where $\sigma(\cdot)$ is the sigmoid activation and $\circ$ means element-wise multiplication. In this way, layers are not strictly dependent on their hierarchical order and will impact dilation sampling in a more complicated way.

Markov Aggregation. An important extreme case of Recurrent Aggregation, Markov Aggregation sets the kernel weights $\mathbf{W}_h^l$ from Eq. (3) to $\mathbf{0}$.

$$\mathbf{h}_{i,j}^l = f(\mathbf{U}_h^l \mathbf{Y}_{i,j}^{l-1}) \tag{6}$$

Similar to the Markov model [21], this means RF sizes are dominated by the last layer. No other inter-layer patterns need to be aggregated for multiple hierarchical layers.

3.3. Dilation adaption vs. kernel adaption

To better understand advantages of proposed adapted dilation, it is worth comparing ADCNN with other related approaches. Recently, there are several works [7,13] also targeting on learning dynamic kernels based on different input contents. We give their approaches a unified name called kernel adaption, since they achieve content-awareness via directly manipulating the kernel space. For example, the modulated deformable convolution [7] can be expressed as

$$y(p) = \sum_{k=1}^{K} w_k \cdot x(p + p_k + \Delta p_k) \cdot \Delta m_k \tag{7}$$

where x is the input feature map and y is the output feature map at location p. Δp_k and Δm_k are the learnable offset and modulation scalar for the k-th location, respectively. This method changes the shape of convolutional kernel by using the offsets and learning these offsets from the target task. More specifically, kernel adaption tends to learn a mapping function $\mathcal{F}$ such that $\mathbf{W}_{m,n} = \mathcal{F}_{m,n}(\mathbf{X})$, where m and n are the pixel index of the convolutional kernel, respectively.

Compared with kernel adaption, ADCNN kernels do so through a more indirect way of engaging dilation rate. Instead of kernel space, dilation function $\mathcal{D}$ sets the target on a dilation space, which contains all D possible dilation values. Theoretically, mapping inputs to dilation space rather than kernel space could have several benefits.

Low dimensional vs. high dimensional complexity. It is easy to see from previous discussions that the dimension of dilation space equals to the number of all dilation options D, while kernel space needs to keep a dimension of $C^{l-1} \times C^l$ such that it can be consistent with input and output channel size. Practically speaking, there is no need to keep a large group of dilation candidates due to their ability of exponentially enlarging RFs [3,4]. Meanwhile, channel size usually increases dramatically as network goes deeper in order to capture more complicated high level abstractions. These facts make D significantly smaller than $C^{l-1} \times C^l$ and leads to an easier learning process with less need of worrying about feature sparsity. Besides, low dimensional complexity also allows ADCNN kernels to be deployed to a wider level range of layers.

Dilation space sharing vs. kernel space orthogonality. Basically, kernel adaption generates kernel values using a single function for a convolution layer. So generated kernels could be highly correlated with each other. However, recent work [22] indicates spaces regularized by orthogonality constrains lead to better results and more stable training process. Therefore, it is hard to balance kernel generation and space orthogonality at the same time. Unlike kernel adaption approaches, ADCNNs mainly rely on dilation spaces, which are not only separated from individual kernel spaces but also can be shared by all convolution layers of a CNN. This means inter-layer patterns are easier to be carried over multiple layers and are able to be more coherently propagated into deeper layers through shared dilation space. Thus compared to kernel adaption, it is expected that ADCNN kernels could be aware of different input contents without interfering the orthogonality among kernel spaces.

4. ADCNNs For semantic segmentation

Since the proposed ADCNN module is highly related to RF adaptation, dense prediction tasks could be ideal to test its effectiveness. Thus, we first evaluate ADCNNs through semantic segmentation to explore their properties from various aspects. We will show that ADCNNs is designed for general purpose and can be applied to solve more problems in later sections.

4.1. Default experimental configurations

We implement ADCNNs with various backbone architectures via PyTorch library. In the following sections, unless otherwise specified, we will employ VGG-16 [23] as backbone net and follow the same training protocol of FCN-8s [24] as task specific framework for evaluation. All ADCNN kernels will follow Markov Aggregation with three available dilation options $\{1, 2, 4\}$ $(D = 3)$. And the τ of GS is set to 1000 by default to generate a smooth distribution. The default dataset is Pascal VOC 2012 [25] and we report mean Intersection over Union (mIoU) on its validation set as evaluation results. All the models will be optimized via Adam optimizer.

4.2. Feature level study

In this section, we conduct several experiments to answer the question: Which convolution level is suitable for ADCNN kernels? For example, considering the convolution blocks, conv3, conv4 and conv5, of a VGG-16 backbone network, if either one is evolved into ADCNN kernel, then which one can yield largest RF on the top layer (conv5-3 in this case) after training? Although for static dilation, RF size of conv5-3 should be the same no matter which block is dilated, this might not hold for ADCNN kernels with multiple dilation candidates, since dilation values are subject to various level of sensitivities due to hierarchical representations. To confirm this, we investigate several cases including both individual and combined ADCNN kernels.

Table 1 summarizes the mIoU for different cases. When only one block is modified, mIoU increases when the feature level for ADCNN changes from low to high. This matches our expectation that ADCNN kernels for higher level features perform better than ADCNN kernels in lower level, as low-level ADCNN kernels are more sensitive to local variances and tend to focus on capturing information in a smaller region; while high-level kernels are usually related to complicated and abstract concepts, leading them to be more responsive for larger input regions. To further support such a claim, we visualize both RFs and Effective RFs (ERFs) [2] for a randomly picked image and put them along with their segmentation results in Fig. 3. As we can see, both RFs and ERFs continuously expand their sizes as feature level for ADCNN goes higher; meanwhile, visually better segmentation results can be achieved with larger RFs and ERFs. This provides us a supportive example that encourages ADCNN extension for higher feature level in practice.

Besides, we also test several cases of combining multiple extended blocks into more complicated ADCNN architectures (the

Table 1
mIoU for feature level study. $\sigma^2(\mathbf{d}_{i,j})$ is variance of pixel dilation sampling.

Method	conv3	conv4	conv5	$\sigma^2(\mathbf{d}_{i,j})$	mIoU
Vanilla	-	-	-	-	64.7
ADCNN	✓			1.96×10^{-4}	63.9
		✓		1.84×10^{-4}	64.7
			✓	4.01×10^{-6}	**66.5**
	✓	✓		2.45×10^{-4}	65.4
		✓	✓	1.24×10^{-4}	66.1
	✓	✓	✓	1.93×10^{-4}	65.9

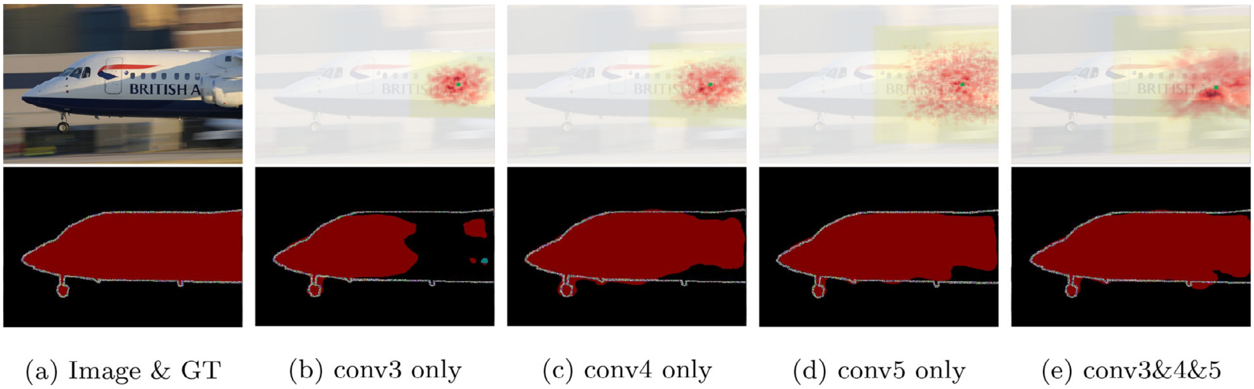


Fig. 3. The top row indicates the input image and its visualized RFs and ERFs on conv5-3 layer of LSD-VGG16 with different conv blocks modified. Patches means RFs and red dots inside are ERFs. The bottom row shows the ground truth and corresponding segemtation results. GT stands for groundtruth. conv3&4&5 means "conv3+conv4+conv5". (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

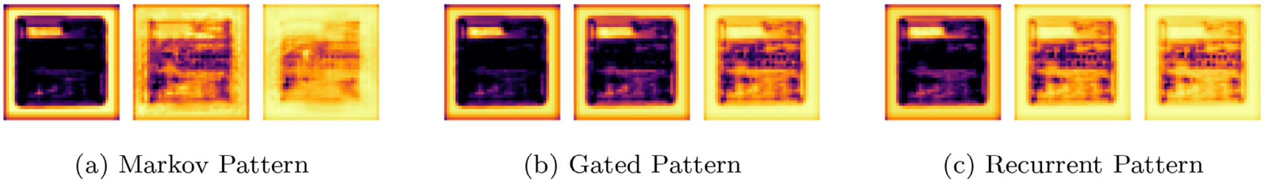


Fig. 4. Mathematical expectation of dilation sampling at each pixel for individual sub-layers (from left to right: conv5-1 to conv5-3). Brighter color means higher dilation and vise versa. The input is the same as the one in Fig. 1.

Table 2
Aggregation study on different backbones.

Task	Semantic Segmentation	
Backbone	VGG-16	ResNet-101
Vanilla Non-Aggregation	64.7	75.1
ADCNN Markov Aggregation	**66.5**	**77.2**
ADCNN Gated Aggregation	65.5	76.7
ADCNN Recurrent Aggregation	65.3	75.6

last three lines of Table 1). To our surprise, stacking additional ADCNN-blocks may result in inferior performances to single block even with better ERF. This indicates it will increase the burden of the model to identify positive and negative features due to the too large RF. We further investigate possible explanations by calculating the variances of dilation sampling for each case. We find performances always decrease when conv5 is combined with more ADCNN-blocks, along with notable variance increments. Such increments brought by additional sampling might be the reason for performance downgrading as they produce some extra burden to the convergence of the dilation sampling process.

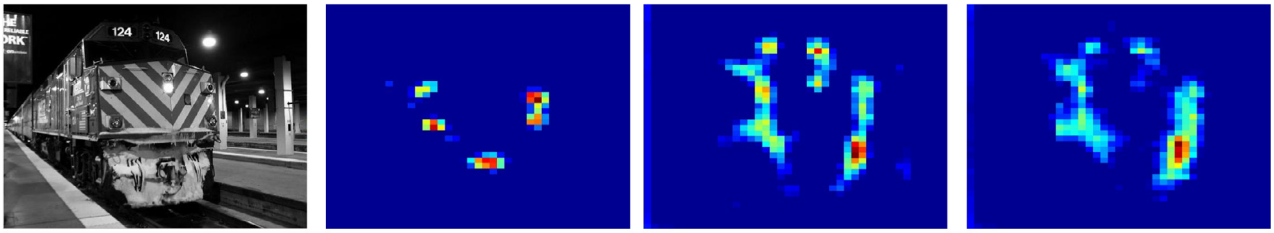
4.3. Pattern aggregation study

Now we focus on studying the impacts brought by each pattern aggregation strategy described in Section 3.2. As suggested from Section 4.2, we only extend the conv5 block of a VGG-16 backbone into ADCNN kernels to avoid too much dilation sampling. All three (conv5-1, conv5-2 and conv5-3) sub-layers are upgraded with ADCNN kernels and connected as each aggregation asks. We also include ResNet-101 [26] combined with DeeplabV3+ [27] as an additional backbone to see if skip connections may result in different impacts.

The results are concluded in Table 2. Basically, all three strategies have better results than backbone networks. However, for both cases Markov Aggregation always yields a better result than other two options. To further dig up the roots behind such phenomenon, in Fig. 4, we calculate and visualize the mathematical expectations at each pixel for all three sub-convolution layers of ADCNN-VGG16. According to the visualization results, we can find that the borders of feature maps are given a large dilation ratio for all patterns. The image in the border area contains little information as the main object of the image usually appears in the central area of the image. This suggests that the kernels in the boundary area are more inclined to use a large dilation value to obtain more effective information. Moreover, for those convolution kernels which are closer to the center area, they gradually start to touch the target object. In addition to learning the features of the object, some of they also need to learn the boundary information of the object. Therefore, they are more likely to choose a smaller dilation value to achieve a better focusing effect. Meanwhile, we can see that during the streaming from conv5-1 to conv5-3, ADCNN with Markov Aggregation is more likely to choose larger dilation everywhere without carrying spatial patterns of input; while both Gated and Recurrent Aggregation are more willing to adjust RF sizes according to spatial structures from input and reserve some spatial clues for dilation sampling. In such cases, information aggregated by lower level features could be too local-sensitive, forcing next layer to put its RF in a smaller region in order to capture such local variations. Thus, our results for semantic segmentation indicate Markov Aggregation is the best option among the three without overly aggregating inter-layer patterns.

4.4. Dilation boundary determination

In this section, we aim to figure out if ADCNN kernels are able to learn a proper range of actually sampled dilation, or they tend to always pick the maximal available dilation value as more options are available. We setup experiments for comparing mIoUs of a VGG-16 backbone with one, two and three available dilation options for their conv5 blocks, respectively. Based on the discussion in Section 4.3, we only consider the cases with Markov Aggregation to get rid of impacts from multiple inter-layer patterns. Other settings remain default.



(a) Input image (b) 1 dilation option (c) 2 dilation options (d) 3 dilation options

Fig. 5. Activation maps for ADCNNs with different number of dilation options.

Table 3
Ablation study of dilation boundary determination on conv5 of FCN-8s.

dilation=1	dilation=2	dilation=4	mIoU
✓			64.7
✓	✓		66.2
✓	✓	✓	66.5

Our results are shown in Table 3, where we gradually increase the available dilation options based on their values from top to bottom and compare the changes of mIoU. Note that the case with the single dilation value 1 is identical to a vanilla backbone network. Apparently, there is a significant performance boost as the number of dilation options is increased from one to two. However, the third dilation option only brings a minor improvement. This suggests that major performance gain is brought by the second one with value 2. We also visualize the output of ADCNN blocks with a randomly picked input for each case in Fig. 5. We can see when options are increased from one to two, a large amount of extra neurons are activated. However, such a number of additional activated neurons is significantly dropped when options increase to three. This means more dilation options may not further improve the performance, as ADCNN can automatically decide the best dilation boundary without worrying about overlarge candidates. And the selection set of dilation values in subsequent experiments is set to three options based on this finding.

4.5. Performance boosting for backbone architectures

Finally, we verify ADCNNs can be easily combined with various popular base architectures to further improve their performance. In addition to VGG-16, we also employ another four representative architectures, ResNet-101 [26], Dilated Residual Networks (DRN) [4], Xception [32] and MobileNet-v2 [33], as additional backbone nets. We combined these base structures with FCN [24] and Deeplabv3+ [27] framework and evaluate them on Cityscapes [34], a more challenging dataset.

We report mIoUs for each backbone network and corresponding ADCNN in Table 4, respectively, along with other state-of-the-art results for comparison. From these results we can see ADCNNs could always yield better results for every backbone structure on both datasets, exhibiting strong robustness and versatility. We also visualize part of segmentation results in Fig. 6, which coincides with mIoU that ADCNNs have more correctly labeled pixels and more details preserved. And the results on class Iou of Cityscapes is shown in Table 5.

5. ADCNNs for image classification

In this section, we demonstrate that the proposed ADCNNs are not only suitable for dense prediction tasks such as semantic segmentation, but also available for more general applications. More specifically, two fundamental tasks, large-scale and fine-grained image classification will be performed to evaluate the performance of ADCNNs with several backbone architectures. We show that ADCNNs can constantly yield better results than their regular counterparts with little extra costs.

5.1. Large-scale image classification

As an important yet challenging work, large-scale image classification usually requires a CNN model with more layers in order to achieve better performances. Unfortunately, it also makes the model significantly increase its model size. We believe ADCNNs could properly address such limitations as light-weighted extensions for their base nets, with better performance and similar training efficiency. To prove this, we select six popular CNN architectures, VGG-16, ResNet-50, ResNet-101, Wide-ResNet101-2 [35], DRN-C-26 and MobileNet-v2, as backbone nets and run experiments on ILSVRC-2012 dataset [36]. Similar to segmentation experiments, we only consider Markov Aggregation in the following experiments since our pilot studies indicate it always yields better results.

We report both top-1 and top-5 classification accuracies for every pair of vanilla and ADCNN in Table 6, along with the comparison of their model complexity changes. Considering the millions of

Table 4
Semantic Segmentation Experiments on validation sets of VOC 2012 and Cityscapes.

Pascal VOC 2012			Cityscapes		
Method	mIoU		Method	mIoU	
	regular	ADCNN		regular	ADCNN
SSDD [28]	64.9	-	Multiscale DEQ [29]	80.3	-
VGG-16+FCN-32s	62.8	**65.1**	RepVGG-B2 [30]	80.6	-
VGG-16+FCN-8s	64.7	**66.5**	OCR(ResNet-101-FCN) [31]	80.6	-
ResNet-101+Deeplabv3+	75.1	**77.2**	MobileNetv2+Deeplabv3+	70.3	**71.5**
Xception+Deeplabv3+	73.5	**74.4**	Xception+Deeplabv3+	77.5	**79.0**
DRN-D-54+Deeplabv3+	75.4	**77.2**	ResNet-101+Deeplabv3+	80.1	**80.7**

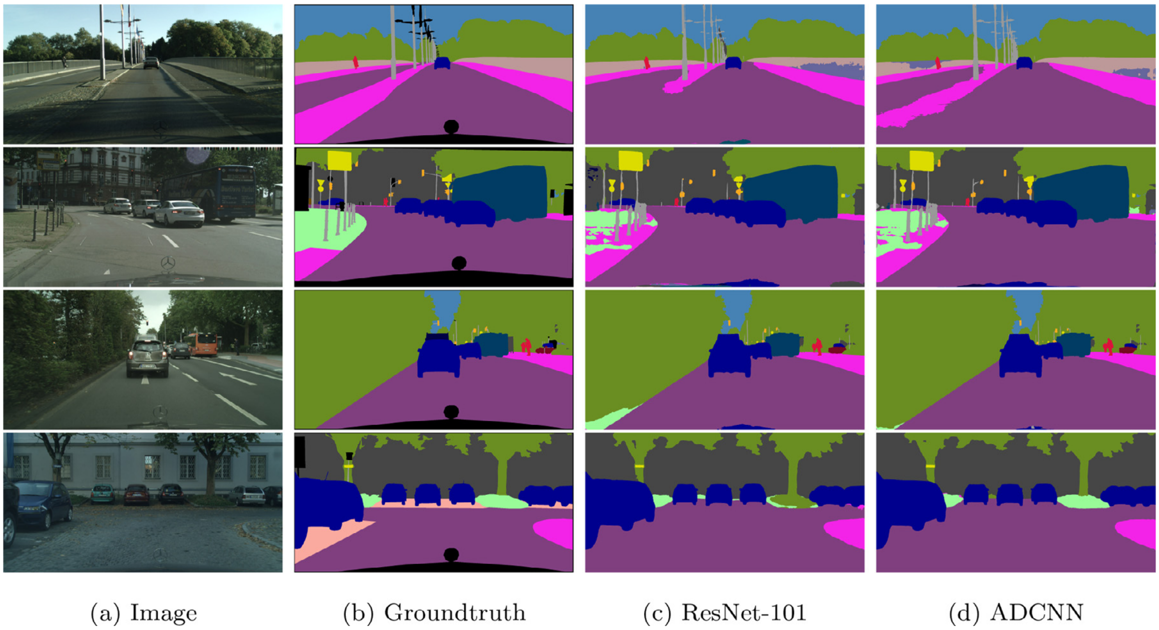

(a) Image (b) Groundtruth (c) ResNet-101 (d) ADCNN

Fig. 6. Semantic segmentation results on Cityscapes dataset.

Table 5
Performance of ADCNN-ResNet-101 on the Cityscapes validation set.

Backbone	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain
ADCNN-ResNet-101	**0.984**	**0.867**	**0.934**	0.610	**0.654**	**0.668**	**0.737**	**0.817**	**0.930**	0.653
ResNet-101	0.983	0.860	0.931	**0.625**	0.638	0.648	0.726	0.801	0.929	**0.659**
Backbone	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	mIoU
ADCNN-ResNet-101	**0.954**	**0.840**	**0.674**	**0.956**	**0.810**	**0.919**	0.808	**0.722**	**0.796**	**0.807**
ResNet-101	0.953	0.833	0.658	0.953	0.797	0.912	**0.815**	0.720	0.787	0.801

Table 6
Accuracies for large-scale image classification on ILSVRC 2012 and corresponding model complexities. p# means model size and Δp# is the number of weights introduced by ADCNNs; (Δp#)/(p#) is the percentage of model size that ADCNNs have increased.

Metric	Accuracy (%)				Model Complexity		
	Top@1		Top@5				
Backbone	Regular	ADCNN	Regular	ADCNN	p#	Δp#	(Δp#)/(p#) (%)
VGG-16	73.0	**74.5**	91.2	**92.0**	138M	21K	0.016
DRN-C-26	75.1	**75.9**	92.4	**92.6**	21.1M	4K	0.019
MobileNetV2	71.8	**72.6**	91.0	90.8	3.5M	2.7K	0.078
ResNet-50	76.0	**76.9**	93.0	**93.4**	25.5M	1K	0.004
ResNet-101	78.3	**78.8**	94.0	**94.2**	178.2M	116.7K	0.066
Wide-ResNet101-2	78.8	**79.1**	94.3	**94.4**	507.5M	233.5K	0.046

parameters that backbone models contain, several thousand extra weights introduced by ADCNN kernels are trivial burdens regarding total model complexity. Meanwhile, we can observe around 1% improvement of top-1 accuracies for each ADCNN and slight top-5 accuracy improvements for most cases, suggesting new modules with less than 0.1% size overhead bring 10 times of performance boosting. This provides us a strong evidence to demonstrate the efficiency of ADCNNs for large-scale classification problem. Besides, the training curves shown in Fig. 7 also confirm that ADCNNs have very similar or even quicker learning progresses to their conventional counterparts, indicating additional introduced weights don't cost extra training iterations to converge.

5.2. Fine-grained image classification

Unlike general classification problem, fine-grained task puts a special emphasis on mining subtle discriminative information in order to recognize objects from different sub-categories. In this section we empirically demonstrate the proposed ADCNNs could properly handle such challenges via their dynamically dilated kernels. Meanwhile, we carry out some comparative experiments with a well-known kernel adaption method, Deformable Convolutional Neural Network [7]. We use all backbones from Section 5.1 except for VGG-16 due to its extremely huge size and initialize corresponding networks with their pretrained weights. Experiments are conducted on Stanford Cars [37] and FGVC-Aircraft [38] datasets following their default protocol with an input size of 448.

All of our experimental results are summarized in Table 7, where we compare the top-1 accuracy for each pair of ADCNN and its vanilla equivalent. We can observe that both of the adaption methods can make some contributes to the performance while ours can have higher increases for all backbone networks on both datasets, demonstrating ADCNNs are not only versatile to be integrated into various backbone nets, but also competent to dis-

Table 7
Top-1 Accuracy (%) for Fine-Grained Visual Classification on different databases with input size of 448.

Task	Stanford Cars			FGVC-Aircrafts		
Backbone	Regular	Deformable	ADCNN	Regular	Deformable	ADCNN
ResNet-50	89.9	92.8	**93.3**	87.9	89.4	**90.1**
ResNet-101	90.9	91.6	**91.7**	88.5	89.3	**89.6**
Wide-ResNet101-2	91.9	92.1	**92.5**	89.4	91.1	**91.3**
DRN-C-26	90.1	91.1	**92.4**	86.8	87.2	**89.7**

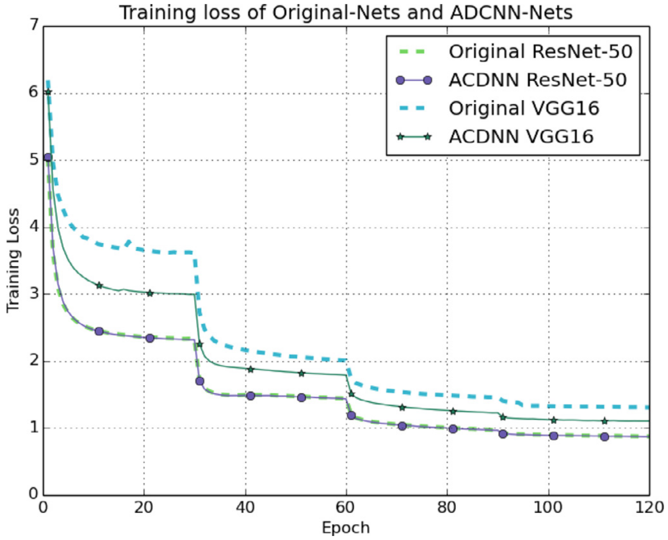
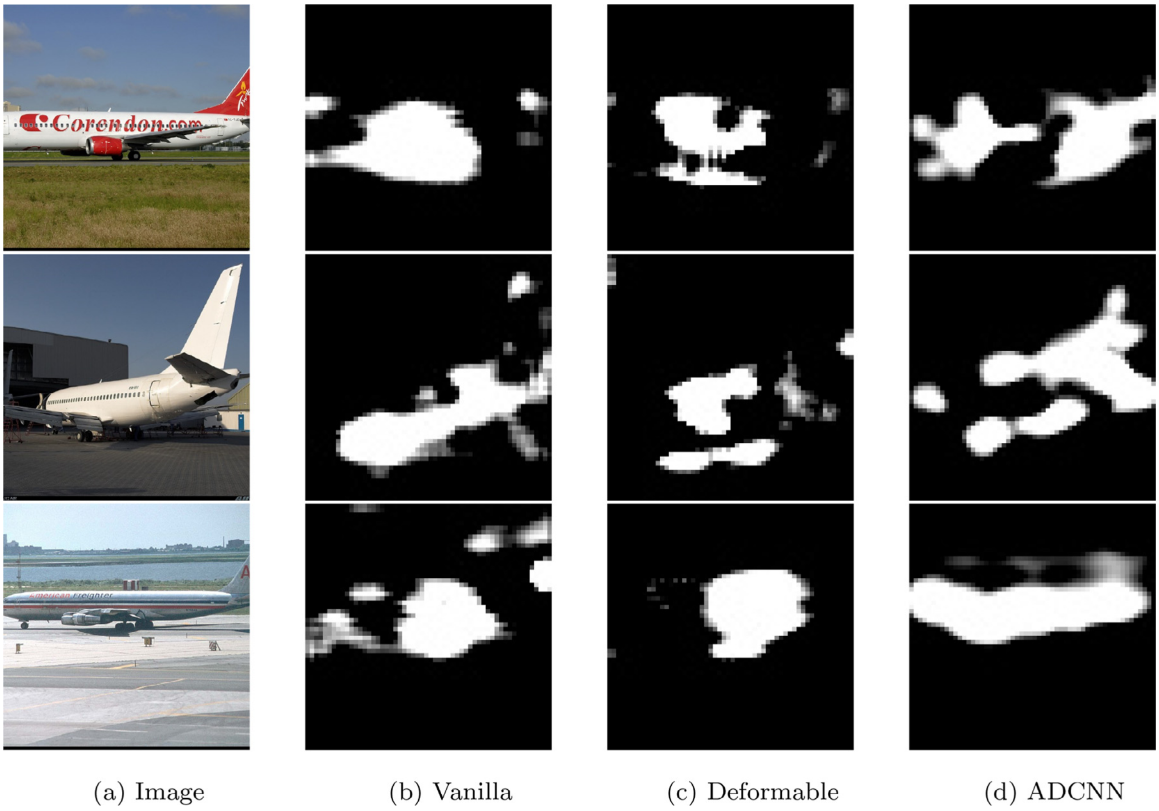


Fig. 7. The training curve of large-scale image classification using ADCNNs based on the VGG-16 and ResNet-50.

tinguish subtle differences. Moreover, we also extract class activation maps from trained DRN-C-26 networks using FGVC-Aircrafts dataset as shown in Fig. 8. From the activation maps we can observe that the activated area of our method is larger than the other two approaches. Meanwhile, the deformable convolution tends to put attention on central details. This might be caused by the flexibility of deformable convolutional kernels. During the training process, deformable convolutional kernels are updated based on the information from the loss function. As the dataset is fragmentary, the information will be limited and kernels are likely to put attention to some partial details. So adding some restrictions while enlarging the RFs, which is the proposed approach, can boost the training process.

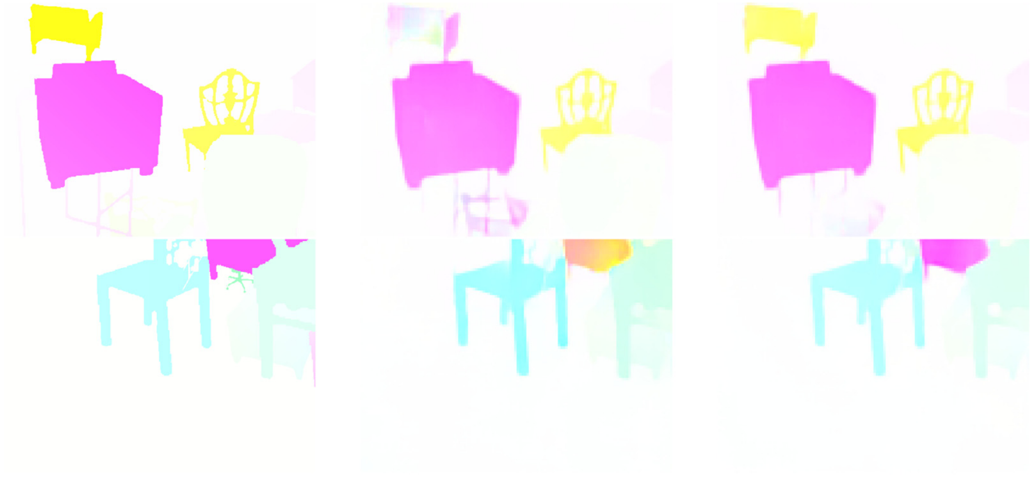
6. ADCNNs for optical flow estimation

Besides semantic segmentation and image classification, we also demonstrate the proposed ADCNNs can perform well on other dense prediction tasks. We conduct experiments on optical flow estimation using the FlyingChairs dataset [39], which consists of 22,872 image pairs and corresponding flow fields. In our experiments, we use two variants of FlowNet introduced by [40],



(a) Image (b) Vanilla (c) Deformable (d) ADCNN

Fig. 8. Class activation maps extracted from DRN-C-26 while we use test set of FGVC-Aircrafts dataset to generate these activation maps. (8 a) are the testing images from dataset. (8 b) are the activation maps based on the vanilla DRN-C-26 network. (8 c) are the activation maps based on the deformable convolutional network. (8 d) are the activation maps based on the our method.



(a) Ground Truth (b) FlownetC (c) ADCNN

Fig. 9. Optical Flow Estimation results on FlyingChairs. (9 a) are the ground truth images from the testing dataset. (9 b) are the estimation results based on the FlownetC and (9 c) are the estimation results based on the proposed method.

Table 8
Average End-Point-Error (aEPE) on FlyingChair dataset.

Method	aEPE	Method	aEPE
FlowNetS [40]	2.78	FlowNetC [40]	2.19
FlowNetS+SegAware [41]	2.36	FlowNetC+LS-DFN,s=7 [13]	2.11
FlowNetS+LS-DFN,s=7 [13]	2.34	FlowNetC+LS-DFN,s=9 [13]	2.06
FlowNetS+Inception-v4 [42]	2.21	FlowNetC+Inception-v4 [42]	1.93
FlowNetS+ADCNN	**1.84**	**FlowNetC+ADCNN**	**1.71**

FlowNetS and FlowNetC, as baseline models. Both of them follow a similar process which firstly learns the semantic features of input images and then upsamples the features to estimate the optical flow.

In experiments, we introduce ADCNN kernels at the convolutional layers before and after the image pairs are merged for both baselines. We use average End-Point-Error (aEPE) to quantitatively measure the performance of the optical flow estimation. As shown in Table 8, ADCNN-enabled models further reduce aEPEs by a large margin compared to their regular counterparts, and significantly outperform state-of-the-art methods. In addition, qualitative results such as generated samples of FlownetC are shown in Fig. 9, where ADCNN gives better estimations than regular models.

7. Conclusions and discussions

In this paper we formulate the dilation as a learnable weight for convolution kernels such that its value can be dynamically decided during the running time. This leads to ADCNNs, a light-weighted, end-to-end trainable framework that allows their kernels to adjust pixel-wise RFs in a data-driven manner. To infer proper dilation values based on feature hierarchy, we model inter-layer patterns via several sequential aggregation strategies. Our studies on semantic segmentation explore various properties of ADCNNs. Results indicate better performance can be achieved with all three aggregation strategies when ADCNN kernels are with higher feature levels, and dilation boundary can be learned to avoid overlarge RFs. We also demonstrate ADCNNs can consistently boost performances over several popular backbone architectures, and be a valuable option for more general visual tasks such as large-scale and fine-grained image classifications.

Although most of our experimetal results indicate ADCNNs could continuously improve the performance across multiple vision tasks, we also observe two significant limitations that might be associated with the employment of GS in our current design. One of them is found during our sensitivity analysis (Fig. 10) of τ, which is a hyper-parameter of GS. From the figure we can see both mean and variance of mIoU change significantly as the value of τ varies: larger τ values usually yield higher mean and lower variance. This suggests τ needs to be large enough in order to acquire good performance and stability. However, this also implies the dilation inference is more biased away from approximating desired distribution. Thus, it becomes necessary to find a proper trade-off between theoretical properties and practical performance, which could be tricky under certain circumstances. Besides, the instability caused by GS also prevents ADCNN kernels from being deployed to more convolution layers. As reported in Section 4.2, stacking more ADCNN blocks leads to increased dilation variance and downgraded performance. To reduce the instability, we plan to explore the possibility of replacing GS with more deterministic, quantization-based techniques in the future. Also, as we discussed in Section 3.3, exploring the inner-relationship between kernel generation and space orthogonality and then propos-

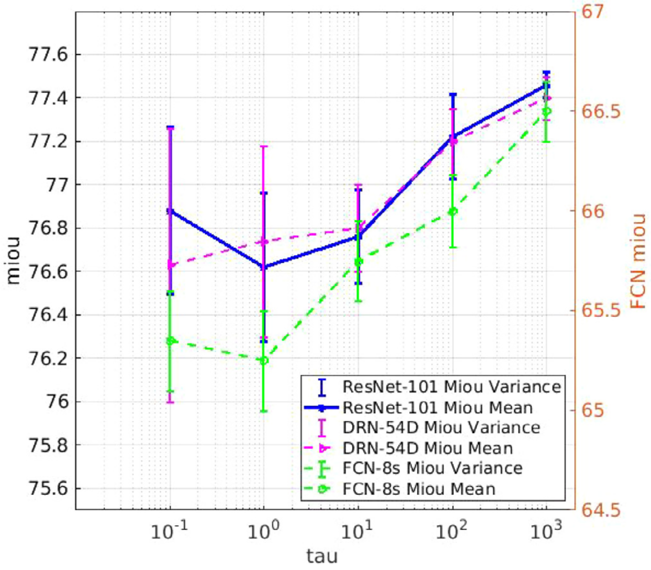


Fig. 10. The sensitivity analysis on τ by performing semantic segmentation task on VOC 2012 validation set with three backbone nets. The mean and variance at each τ value are computed by repeating 5 times with same settings.

ing a regularization method [43] to balance them is another interesting research direction. In addition to further research on the method itself, our method can also be applied to many real-world applications. As our method can boost the feature response capability shown in Section 5.2, the proposed method has the potential to be applied in some scenarios, such as pedestrian detection which requires the model to have a strong feature discrimination ability [44].

Declaration of Competing Interest

We wish to draw the attention of the Editor to the following facts which may be considered as potential conflicts of interest and to significant financial contributions to this work. [OR] We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.

The main research interests are as follows:

1. Deep learning;
2. Deep neural network robustness;
3. Computer vision;
4. Data mining and big data analysis.

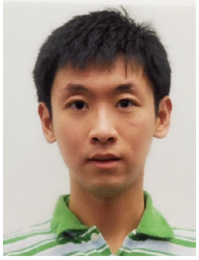
Acknowledgment

This work is supported by the National Science Foundation [grant number 1704309], the National Natural Science Foundation of China [grant number 61876018] and the Digital Futures research center.

References

[1] J. Yao, W. Xing, D. Wang, J. Xing, L. Wang, Active dropblock: method to enhance deep model accuracy and robustness, Neurocomputing 454 (2021) 189–200.
[2] W. Luo, Y. Li, R. Urtasun, R. Zemel, Understanding the effective receptive field in deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2016, pp. 4898–4906.
[3] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, in: 4th International Conference on Learning Representations, ICLR, 2016.
[4] F. Yu, V. Koltun, T. Funkhouser, Dilated residual networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 472–480.
[5] J. Xu, W. Wang, H. Wang, J. Guo, Multi-model ensemble with rich spatial information for object detection, Pattern Recognit 99 (2020) 107098.
[6] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 764–773.
[7] X. Zhu, H. Hu, S. Lin, J. Dai, Deformable convnets v2: More deformable, better results, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9308–9316.
[8] E. Jang, S. Gu, B. Poole, Categorical reparameterization with gumbel-softmax, in: 5th International Conference on Learning Representations, ICLR, OpenReview.net, 2017.
[9] Y. Liang, Y. Liu, Y. Yan, L. Zhang, H. Wang, Robust visual tracking via spatio-temporal adaptive and channel selective correlation filters, Pattern Recognit 112 (2021) 107738.
[10] D. Zhang, W. Ding, B. Zhang, C. Liu, J. Han, D. Doermann, Learning modulation filter networks for weak signal detection in noise, Pattern Recognit 109 (2021) 107590.
[11] X. Jia, B. De Brabandere, T. Tuytelaars, L.V. Gool, Dynamic filter networks, in: Advances in Neural Information Processing Systems, 2016, pp. 667–675.
[12] H. Su, V. Jampani, D. Sun, O. Gallo, E. Learned-Miller, J. Kautz, Pixel-adaptive convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 11166–11175.
[13] J. Wu, D. Li, Y. Yang, C. Bajaj, X. Ji, Dynamic filtering with large sampling field for convnets, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 185–200.
[14] Z. Wei, Y. Sun, J. Wang, H. Lai, S. Liu, Learning Adaptive Receptive Fields for Deep Image Parsing Network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2434–2442.
[15] E. Shelhamer, D. Wang, T. Darrell, Blurring the line between structure and learning to optimize and adapt receptive fields, arXiv preprint arXiv:1904.11487 (2019).
[16] B. Li, Y. Dai, M. He, Monocular depth estimation with hierarchical fusion of dilated cnns and soft-weighted-sum inference, Pattern Recognit 83 (2018) 328–339.
[17] Q. Wang, H. Fan, G. Sun, Y. Cong, Y. Tang, Laplacian pyramid adversarial network for face completion, Pattern Recognit 88 (2019) 493–505.
[18] Q. Chen, P. Wang, A. Cheng, W. Wang, Y. Zhang, J. Cheng, Robust one-stage object detection with location-aware classifiers, Pattern Recognit 105 (2020) 107334.
[19] H. Hu, L. Wang, G. Qi, Learning to adaptively scale recurrent neural networks, in: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI, 2019, pp. 3822–3829, doi:10.1609/aaai.v33i01.33013822.
[20] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555 (2014).
[21] P.A. Gagniuc, Markov Chains: from Theory to Implementation and Experimentation, John Wiley & Sons, 2017.
[22] N. Bansal, X. Chen, Z. Wang, Can we gain more from orthogonality regularizations in training deep networks? in: Advances in Neural Information Processing Systems, 2018, pp. 4261–4271.
[23] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: 3rd International Conference on Learning Representations, ICLR, 2015.
[24] J. Long, E. Shelhamer, T. Darrell, Fully Convolutional Networks for Semantic Segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.
[25] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, Int J Comput Vis 88 (2) (2010) 303–338.
[26] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[27] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 801–818.
[28] W. Shimoda, K. Yanai, Self-supervised difference detection for weakly-supervised semantic segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 5208–5217.
[29] S. Bai, V. Koltun, J.Z. Kolter, Multiscale deep equilibrium models, NeurIPS, 2020.
[30] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, J. Sun, Repvgg: Making vgg-style convnets great again, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13733–13742.
[31] Y. Yuan, X. Chen, X. Chen, J. Wang, Segmentation transformer: Object-contextual representations for semantic segmentation, in: European Conference on Computer Vision (ECCV), volume 1, 2021.
[32] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251–1258.
[33] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520.
[34] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 3213–3223.
[35] S. Zagoruyko, N. Komodakis, Wide residual networks, in: Proceedings of the British Machine Vision Conference 2016, BMVC, 2016.
[36] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int J Comput Vis 115 (3) (2015) 211–252.
[37] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3d object representations for fine-grained categorization, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2013, pp. 554–561.
[38] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, A. Vedaldi, Fine-grained visual classification of aircraft, arXiv preprint arXiv:1306.5151 (2013).
[39] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, T. Brox, Flownet: Learning optical flow with convolutional networks, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 2758–2766.
[40] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, T. Brox, Flownet 2.0: Evolution of optical flow estimation with deep networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2462–2470.
[41] A.W. Harley, K.G. Derpanis, I. Kokkinos, Segmentation-aware convolutional networks using local attention masks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5038–5047.
[42] C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: Thirty-first AAAI Conference on Artificial Intelligence, 2017.
[43] C. Song, J. Wu, L. Zhu, X. Zuo, Weight correlation reduction and features normalization: improving the performance for shallow networks, Vis Comput (2021) 1–10.
[44] Q. Mou, L. Wei, C. Wang, D. Luo, S. He, J. Zhang, H. Xu, C. Luo, C. Gao, Unsupervised domain-adaptive scene-specific pedestrian detection for static video surveillance, Pattern Recognit 118 (2021) 108038.

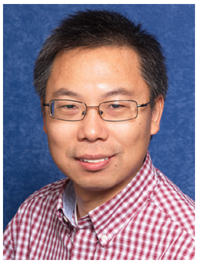
Jie Yao received the B.S. degree in Software Engineer from Beijing Jiao-tong University, China, in 2016. During 20192020, he was a visiting student at University of Central Florida. Currently, he is a Ph.D. candidate of School of Software Engineering at Beijing Jiaotong University, China. His research interests include image processing, computer vision and deep model robustness.



Dongdong Wang is a Ph.D. candidate in computer science at University of Central Florida. He earned his Master of Science in Environmental Science from Duke University in 2017. His research explores the approaches for knowledge distillation and deep neural network optimization and the application field is focused on computer vision. His work is developed upon numerical analysis and optimization. He has published several peer-reviewed papers in leading conferences such as CVPR and AAAI.

Hao Hu is a postdoc researcher at the Robotic, Perception and Learning Division, KTH Royal Institute of Technology in Stockholm, Sweden. Before joining KTH, he worked for FX Palo Alto Laboratory, Inc. as a research scientist. Hao obtained his PhD and master degree in Computer Science from University of Central Florida, U.S. and Bachelor degree of Information science from Nankai University, China. His research interests across multiple topics of deep learning and computer vision, with special focuses on self-supervised feature learning, temporal modeling and applied deep learning. His works are, and have been supported by multiple grant sources including KTH digital future, AAAI travel grants, etc. He is also serving for multiple machine learning and computer vision venues including CVPR, ICCV, ICML, NeurIPS, etc.

Weiwei Xing received her B.S. degree in Computer Science and Technology and Ph.D. degree in Signal and Information Processing from Beijing Jiaotong University, in 2001 and 2006 respectively. During 20112012, she was a visiting scholar at University of Pennsylvania. Currently, she is an associate professor at School of Software Engineering, Beijing Jiaotong University. Her research interests mainly include intelligent information processing and artificial intelligence.



Liqiang Wang is an associate professor in the Department of Computer Science at the University of Central Florida. He is the director of Big Data Lab. He was a faculty member (20062015) in the Department of Computer Science at the University of Wyoming. He received Ph.D. in Computer Science from Stony Brook University in 2006. He was a visiting Research Scientist in IBM T.J. Watson Research Center during 20122013. His research focuses on big data techniques, which include the following aspects: (1) improving the accuracy, security, privacy, and fairness of big data analytics; (2) optimizing performance, scalability and resilience of big data processing, especially on Cloud and GPU platforms; (3) using program analysis and deep learning techniques to detect and avoid programming errors, execution anomaly, as well as performance defects in big data programs. He received NSF CAREER Award in 2011 and Castagne Faculty Fellowship (20132015).