# PAD-NETS: LEARNING DYNAMIC RECEPTIVE FIELDS VIA PIXEL-WISE ADAPTIVE DILATION

**Anonymous authors**
Paper under double-blind review

## 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 respect to individual positions. We introduce Pixel-wise Adaptive Dilation (PAD), a light-weighted extension that allows convolution kernels to flexibly adjust receptive fields based on different contents at pixel level. By inferring dilation via modeling inter-layer patterns, PAD-Nets also provide a possible way to partially understand the hierarchical representations of CNNs. Our evaluation results indicate PAD-Nets can consistently outperform their conventional counterparts on various visual tasks.

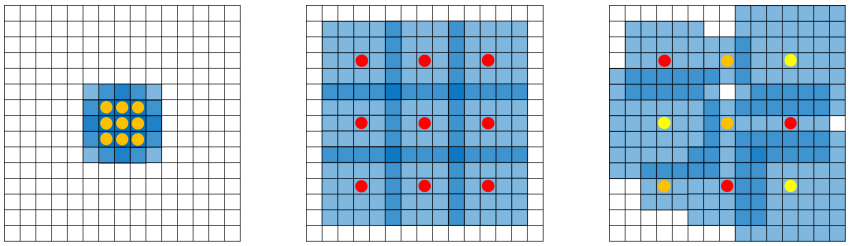
## 1 INTRODUCTION

The power of prestigious Convolutional Neural Nets (CNN) (Simonyan & Zisserman, 2014; He et al., 2016; Huang et al., 2017) relies on the ability of hierarchically representing spatial features across input regions called Receptive Fields (RFs) (Luo et al., 2016). Common practices usually prefer large RFs in order to achieve superior performances, making Dilated Convolution Kernels (DCKs) (Yu & Koltun, 2015; Yu et al., 2017) a favorable choice due to the ability of exponentially enlarging RFs while keeping kernels small. To further improve the dilated kernels, two obvious problems, which universally reside in most of existing dilated CNN structures, need to be properly tackled. First, all the weights share a single dilation value across all pixels. This could be very counter-intuitive as resultant monosized RFs are less capable of encoding huge spatial variances from inputs. Second, dilation selection is data-independent, requiring strong knowledge on input contexts to pick the proper value, which may only be suitable to a certain set of experimental settings.

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 Pixel-wise Adaptive Dilated Nets (PAD-Nets), 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 Figure 1, in the newly formulated PAD 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 convolution 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 indifferentiable nature of general sampling process, we approximate it by employing Gumbol-Softmax (Jang et al., 2016) as a differentiable estimation in order to keep PAD 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 PAD 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.



(a) No dilation (b) Regular dilation (c) Pixel-wise adaptive dilation

Figure 1: Comparison of regular and pixel-wise adaptive dilation. Colors stands for distinct dilation.

Following strategies described above, PAD-kernels evolve into light-weighted modules that can be easily plugged into various CNN architectures. Moreover, learning dilation through inter-layer pattern modeling also provides a chance to potentially unveil the mechanisms of CNNs in part. We evaluate the proposed PAD-Nets via several fundamental tasks including semantic segmentation, large-scale and fine-grained visual classification. Moreover, several ablation studies are performed to examine various properties of PAD-Nets. Our experimental results indicate in most cases, PAD-Nets are able to consistently yield better performances across various popular backbone architectures with trivial cost.

The rest of this paper is organized as following. We review relevant literature in Section 2, then PAD-Nets are elaborated in Section 3. Sections 4 and 5 demonstrate experimental results, and Section 6 concludes the paper.

## 2 Related Work

**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, Jampani et al. (2016) includes bilateral filters (Aurich & Weule, 1995; Tomasi & Manduchi, 1998) in CNN models as a layer for character recognition; Wang et al. (2018) and Wu et al. (2018a) introduce their trainable version of non-local means filters (Buades et al., 2005) and guided filters (He et al., 2012), respectively. 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 (Xue et al., 2016; Jia et al., 2016; Su et al., 2019), and extend it with attention mechanism (Wu et al., 2018b) as well as other task-specific improvements. However, most of them rely on additional modules with large kernel sizes, being incapable of scaling up to more general network structures.

**Dynamic receptive fields.** Comparing to the above approaches to build content-adaptive nets, there is much less work aiming at enabling the content-aware ability via adjusting receptive fields (RFs). Majority of RF-related researches focus on how to effectively enlarge RFs in order to achieve better performance. Among them, dilated convolution kernels (Yu & Koltun, 2015) become a popular choice as it can exponentially increase RF sizes while maintaining small kernel sizes. However, this could also lead to negative impacts, such as sparsity and "gridding" effect (Yu et al., 2017). Unlike static RFs produced by dilation, recent works such as Dai et al. (2017) and Zhu et al. (2019) argue that RFs should be more diverse in order to capture rich spatial variations. They propose deformable CNNs that learn to adjust the positions for convolving, resulting in free-form RFs that are totally data-dependent. Besides, Shelhamer et al. (2019) attempts to create diverse yet controllable RFs by composing the structured Gaussian kernels and unstructured ordinary convolution kernels.

## 3 Pixel-wise Adaptive Dilated Convolution

Now we elaborate the proposed approach for extending conventional dilated convolution kernels into PAD kernels. Without loss of generality, we assume all the convolutions in the rest of this paper

are 2D operations. Suppose $\mathcal{K}_{\mathbf{W},\mathbf{b};d}$ is a dilated convolutional kernel with dilation value $d$, and $\mathbf{X} \in \mathbb{R}^{w \times h \times c}$ is input. The output of convolution between $\mathcal{K}$ and $\mathbf{X}$ is

$$\mathbf{Y}_{i,j} = \sum_{m=0}^{K} \sum_{n=0}^{K} (\mathbf{W}_{m,n} \mathbf{x}_{i+dm,j+dn} + \mathbf{b}_{m,n}) \tag{1}$$

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. Figure 2 sketches the basic idea of a PAD 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 van den Oord et al. (2017); Gal et al. (2017) and Hu et al. (2019), we employ Gumbol-Softmax (GS) (Jang et al., 2016; Maddison et al., 2016) 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 $\mathbf{d}_{i,j} \sim \text{GS}(\mathbf{h}_{i,j})$ can be achieved by

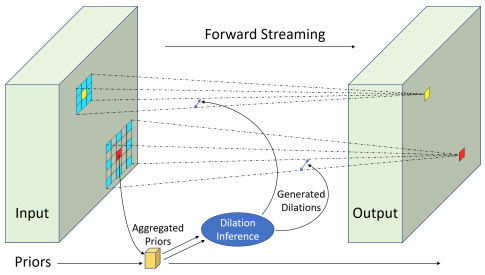



Figure 2: Overview of a PAD kernel.

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

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 $\tau$ 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 and build up our dilation inference mechanism upon inter-layer pattern modeling to capture dependencies between abstraction levels. Inspired by Lin et al. (2017); He et al. (2017) and Kirillov et al. (2019), we consider aggregation as a feasible way and will generate hidden priors $\mathbf{h}$ through sequentially aggregating multiple $\mathbf{Y}$ from hierarchical layers. Suppose $l$ is the newly added layer index, then 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 $1 \times 1$ 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 layers as $l$ goes deeper, implying layers are highly dependent with 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 Hochreiter & Schmidhuber (1997) and Chung et al. (2014) for computing $\mathbf{a}^l_h$, with which the entire aggregation can be formulated as following

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

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

| CONV3 | CONV4 | CONV5 | $\sigma^2(\mathbf{d}_{i,j})$ | MIOU |
|---|---|---|---|---|
| ✓ | | | $1.96 \times 10^{-4}$ | 63.9 |
| | ✓ | | $1.84 \times 10^{-4}$ | 64.7 |
| | | ✓ | $4.01 \times 10^{-6}$ | **66.5** |
| ✓ | ✓ | | $2.45 \times 10^{-4}$ | 65.4 |
| | ✓ | ✓ | $1.24 \times 10^{-4}$ | 66.1 |
| ✓ | ✓ | ✓ | $1.93 \times 10^{-4}$ | 65.9 |

Table 2: mIoU for pattern aggregation study. VGG-16 backbone is combined with FCN-8s and ResNet-101 is with Deeplab v3+.

| AGGREGATION | VGG-16 | RESNET-101 |
|---|---|---|
| MARKOV | **66.5** | **77.2** |
| GATED | 65.5 | 76.7 |
| RECURRENT | 65.3 | 75.6 |
| BACKBONE | 64.7 | 75.1 |

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

where $\sigma(\cdot)$ is the sigmoid activation and $\circ$ means element-wise multiplication. In this way, layers are not strictly dependent following 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 equation (3) to $\mathbf{0}$. Similar to the Markov model (Gagniuc, 2017), 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 DISCUSSION

Basically, PAD is a *light-weighted extension* for general CNN structures, which means it introduces little extra weights for dilation inference and is independent to feature channel size. More specifically, all the weights brought with PAD have an output channel of $D$, while other RF adaptation works such as Dai et al. (2017) often rely on additional modules that need to match the same output channel size of features. Since $D$ is usually much smaller than feature channel size, PAD kernels are easier to be deployed at higher level convolution layers.

Moreover, the proposed dilation inference grants *a manageable way to partially understand the feature hierarchy*. Through learning RF adaptations based on inter-layer pattern aggregations, we can better explore the principles behind inter-layer interactions via analyzing dilation changes. It is also expected that inferring proper dilation values according to inter-layer patterns may help comprehend the differences between low and high level convolution layers. Such information could be invaluable knowledge for designing interpretable architectures. Furthermore, the flexibility of inter-layer pattern modeling allow PAD-Nets to be attuned to various tasks and applications.

## 4 PAD-NETS FOR SEMANTIC SEGMENTATION

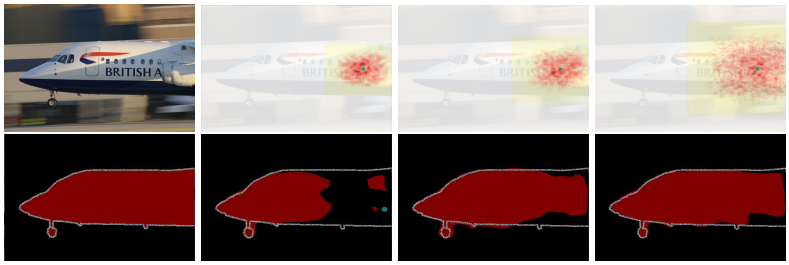
Since the proposed PAD module is highly related to RF adaptation, dense prediction tasks could be ideal to test its effectiveness. Thus, we first evaluate PAD-Nets through semantic segmentation to explore their properties from various aspects. We will show that PAD-Nets is designed for general purpose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

### 4.1 GENERAL EXPERIMENTAL CONFIGURATIONS

We implement PAD-Nets with various backbone architectures via PyTorch library (Paszke et al., 2017). Unless otherwise specified, we will employ VGG-16 (Simonyan & Zisserman, 2014) as backbone net and follow the same training protocol as FCN-8s (Long et al., 2015) for evaluation. The default dataset is Pascal VOC 2012 (Everingham et al., 2010) and we report mean Intersection over Union (mIoU) on its validation set as evaluation results. All the models will be optimized via Adam optimizer (Kingma & Ba, 2014). All architectural descriptions can be found in Appendix A.

### 4.2 FEATURE LEVEL STUDY

In this section, we conduct several experiments to answer the question: which convolution level is suitable for PAD kernels? Although in static cases RF size for a single layer should keep the same regardless of lower level dilation, this might not be hold for PAD kernels since dilation values are subject to various level of sensitivities due to hierarchical feature representations. To confirm this, we individually extend each convolution block of a vanilla VGG-16 backbone net with PAD kernels and Markov Aggregation, and run experiments following default settings described in Section 4.1.



(a) Image & GT (b) conv3 only (c) conv4 only (d) conv5 only

Figure 3: The top row indicates the input image and its visualized RFs and ERFs for PAD-VGG16 with different conv blocks. Patches means RFs and red dots inside are ERFs. The bottom row shows the ground truth and corresponding segemtation results. GT stands for groundtruth.

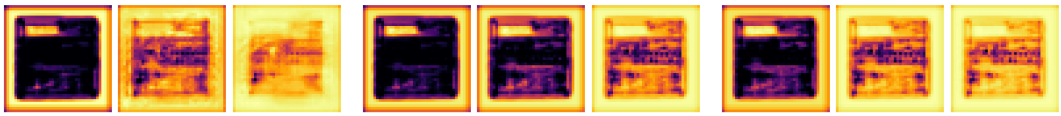
Table 1 summarizes the mIoU for different cases. When only one block is modified, mIoU increases when the feature level for PAD changes from low to high. This matches our expectation that PAD kernels for higher level features perform better than PAD kernels in lower level, as low-level PAD kernels are 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) (Luo et al., 2016) for a randomly picked image and put them along with their segmentation results in Figure 3. As we can see, both RFs and ERFs continuously expand their sizes as feature level for PAD goes higher; meanwhile, visually better segmentation results can be achieved with larger RFs and ERFs. This provides us a supportive example that encourages PAD extension for higher feature level in practice.

Besides, we also test several cases of combining multiple extended blocks into more complicated PAD-Net architectures (the last three lines of Table 1). To our surprise, stacking additional PAD-blocks to single PAD-block may result in inferior performances. We further investigate possible explanations by calculating the variances of dilation sampling for each cases. We find performances always decrease when conv5 is combined with more PAD-blocks, along with notable variance increments. Such increments brought by additional sampling might be the reason for performance downgrading as they make the entire structure more unstable.

### 4.3 PATTERN AGGREGATION STUDY

Now we focus on studying the impacts brought by each pattern aggregation strategies described in Section 3.2. As suggested from Section 4.2, we only extend conv5 block of a VGG-16 backbone into PAD kernels to avoid too much dilation sampling. All three (conv5-1, conv5-2 and conv5-3) sub-layers are upgraded with PAD kernels and connected as each aggregation asks. We also include ResNet-101 (He et al., 2016) combined with DeeplabV3+ (Chen et al., 2018) 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. However, for both cases Markov Aggregation always yields a better result than other two options, implying too much redundant inter-layer patterns might be accumulated. To further explore the root behind such phenomenon, in Figure 4, we calculate and visualize the mathematical expectations at each pixel for all three sub-convolution layers. We can see that during the streaming from conv5-1 to conv5-3, PAD-Net 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

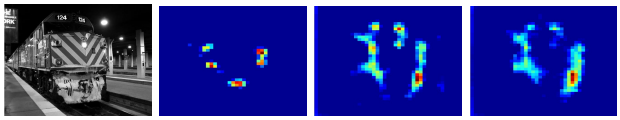


(a) Markov Pattern (b) Gated Pattern (c) Recurrent Pattern

Figure 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 Figure 1.

Table 3: mIoU of cases with available dilation options change.

| $d_{i,j}$=1 | $d_{i,j}$=2 | $d_{i,j}$=4 | MIOU |
|---|---|---|---|
| ✓ | | | 64.7 |
| ✓ | ✓ | | 66.2 |
| ✓ | ✓ | ✓ | 66.5 |



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

Figure 5: Activation maps for PAD-Nets with different number of dilation options.

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 whether more dilation options for a PAD kernel can always lead to better performance or not. 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.

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 net. 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 PAD blocks with a randomly picked input for each case in Figure 5. When options increased to two and three, There are more neurons being activated than single dilation with similar spatial distributions. This means more dilation options may not further improve the performance, as PAD-Net can intelligently decide the best boundary for dilation values without worrying about overlarge candidates.

### 4.5 Performance Boosting for Backbone Architectures

Finally, we verify PAD-Nets can be easily combined various popular base architectures to further improve their performance. In addition to VGG-16, we also employ another four popular architectures, ResNet-101 (He et al., 2016), Dilated Reside Nets (DRN) (Yu et al., 2017), Xception (Chollet, 2017) and MobileNet-v2 (Sandler et al., 2018), as additional backbone nets. We combined these base structures with FCN (Long et al., 2015) and Deeplabv3+ (Chen et al., 2018) framework and evaluate them on Cityscapes (Cordts et al., 2016), a more challenging dataset. Detailed architectures and configurations be found in Appendix A (Table 8).

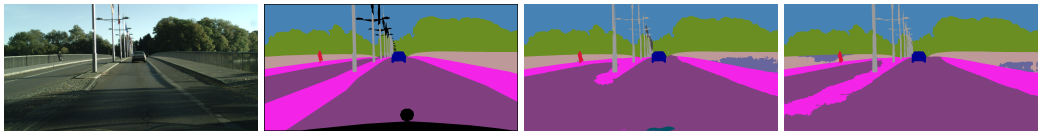
We report mIoUs for each backbone net and corresponding PAD-Net in Tables 4 and 5, respectively, along with other state-of-the-art results for comparison. From these two tables we can see PAD-Nets could always yield better results for every backbone structure on both datasets, exhibiting

Table 4: mIoU for VOC 2012 validation set.

| METHOD | MIOU | |
|---|---|---|
| | REGULAR | PAD |
| DPN (LIU ET AL., 2015) | 67.8 | - |
| CRF+RNN (ZHENG ET AL., 2015) | 69.6 | - |
| DCNN (DAI ET AL., 2017) | 75.1 | - |
| VGG-16+FCN-32S | 62.8 | **65.1** |
| VGG-16+FCN-8S | 64.7 | **66.5** |
| RESNET-101+DEEPLABV3+ | 75.1 | **77.2** |
| XCEPTION+DEEPLABV3+ | 73.5 | **74.4** |
| DRN-D-54+DEEPLABV3+ | 75.4 | **77.2** |

Table 5: mIoU for CityScape validation set.

| METHOD | MIOU | |
|---|---|---|
| | REGULAR | PAD |
| DANET-101 (FU ET AL., 2019) | 77.6 | - |
| PSP-NET (ZHAO ET AL., 2017) | 78.8 | - |
| IN-ABN (ROTA BULÒ ET AL., 2018) +RESNEXT-101 (XIE ET AL., 2017) | 79.2 | - |
| PSP-NET + GFF (LI ET AL., 2019) | 80.4 | - |
| GSCNN (TAKIKAWA ET AL., 2019) | 80.8 | - |
| MOBILENETV2+DEEPLABV3+ | 70.3 | **71.5** |
| XCEPTION+DEEPLABV3+ | 77.5 | **79.0** |
| RESNET-101+DEEPLABV3+ | 80.1 | **80.7** |



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

Figure 6: Semantic segmentation results on Cityscapes dataset.

strong robustness and versatility. We also visualize part of segmentation results in Figure 6, which coincides with mIoU that PAD-Nets have more correctly labeled pixels and more details preserved. More results on class IoU and segmentation can be found in Appendix B.

## 5 PAD-NETS FOR IMAGE CLASSIFICATION

In this section, we demonstrate that the proposed PAD-Nets 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 PAD-Nets with several backbone architectures. We show that PAD-Nets 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 PAD-Nets 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 four popular CNN architectures, VGG-16, ResNet-50, DRN-C-26 and MobileNet-v2, as backbone nets and run experiments on ILSVRC-2012 dataset (Russakovsky et al., 2015). Similar to segmentation experiments, we only consider Markov Aggregation in following experiments since our pilot studies indicate it always yields better results. Other configuration details can be found in Appendix A (Table 9).

We report both top-1 and top-5 classification accuracies for every pair of vanilla and PAD-Net in Table 6, along with the comparison of their model complexity changes. Considering millions of parameters that backbone models contain, several thousand extra weights introduced by PAD kernels are trivial burdens regarding to total model complexity. Meanwhile, we can observe around $1\%$ improvement of top-1 accuracies for each PAD-Net 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 PAD-Nets for large-scale classification problem. In addition, training curve comparison can be found in Appendix C.

Table 6: Accuracies for lage-scale image classification on ILSVRC 2012 and corresponding model complexities. p# means model size and $\Delta$p# is the number of weights introduced by PAD-Nets; ($\Delta$p#)/(p#) is the percentage of model size that PAD-Nets have increased.

| Metric | Accuracy (%) | | | | Model Complexity | | |
|---|---|---|---|---|---|---|---|
| Backbone | Top@1 | | Top@5 | | p# | $\Delta$p# | ($\Delta$p#)/(p#) (%) |
| | Regular | PAD | Regular | PAD | | | |
| VGG-16 | 73.0 | **74.5** | 91.2 | **92.0** | 138M | 21K | 0.016 |
| ResNet-50 | 76.0 | **76.9** | 93.0 | **93.4** | 25.5M | 1K | 0.004 |
| 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 |

Table 7: Top-1 Accuracy for Fine-Grained Visual Classification on different databases.

| Task | Fine-Grained Classification | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Dataset | Stanford Cars | | | | FGVC-Aircrafts | | | |
| Crop Size | 224 | | 448 | | 224 | | 448 | |
| | Regular | PAD | Regular | PAD | Regular | PAD | Regular | PAD |
| ResNet-50 | 91.2 | **91.6** | 92.3 | **93.5** | 86.1 | **86.6** | 87.9 | **90.1** |
| DRN-C-26 | 91.0 | **91.4** | 90.3 | **92.4** | 86.3 | **86.5** | 86.8 | **89.6** |
| MobileNetV2 | 88.7 | **88.8** | 80.6 | **82.7** | 83.2 | **84.1** | 80.5 | **87.6** |

### 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 PAD-Nets could properly handle such challenges via their dynamically dilated kernels. We use all backbones from Section 5.1 except for VGG-16 due to its extremely huge size and initialize corresponding PAD-Nets with their pretrained weights. Experiments are conducted on Stanford Cars (Krause et al., 2013) and FGVC-Aircraft (Maji et al., 2013) datasets following their default protocol with two input sizes, 224 and 448. Similarly, we put other configuration details in Appendix A (Table 10).

All of our experimental results are summarized in Table 7, where we compare the top-1 accuracy for each pair of PAD-Net and its vanilla equivalent. We can only observe trivial improvements with input size 224 for both datasets. However, performance gain increases to over $2\%$ for nearly every type of PAD-Net, indicating PAD-Nets could be a much better option for high-resolution images to get higher performance without more significant resource burdens. Moreover, by comparing the activation maps of DRN and PAD-DRN, we further discover that PAD-Nets can better capture parts information and preserve more details than baselines with fixed, predefined dilation values. We provide such evidence with more visualized activation maps for DRN-C-26 and its PAD-extension in Appendix B.

## 6 Conclusion

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 PAD-Nets, 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 PAD-Nets and indicate better performance can be achieved when PAD kernels are with higher feature levels and Markov Aggregation. We also demonstrate PAD-Nets 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.

## REFERENCES


Volker Aurich and Jörg Weule. Non-linear gaussian filters performing edge preserving diffusion. In *Mustererkennung 1995*, pp. 538–545. Springer, 1995.

Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pp. 60–65. IEEE, 2005.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 764–773, 2017.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3146–3154, 2019.

Paul A Gagniuc. *Markov chains: from theory to implementation and experimentation*. John Wiley & Sons, 2017.

Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout. In *Advances in Neural Information Processing Systems*, pp. 3581–3590, 2017.

Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence*, 35(6):1397–1409, 2012.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Hao Hu, Liqiang Wang, and Guo-Jun Qi. Learning to adaptively scale recurrent neural networks. *arXiv preprint arXiv:1902.05696*, 2019.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Varun Jampani, Martin Kiefel, and Peter V Gehler. Learning sparse high dimensional filters: Image filtering, dense crfs and bilateral neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4452–4461, 2016.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *Advances in Neural Information Processing Systems*, pp. 667–675, 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6399–6408, 2019.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 554–561, 2013.

Xiangtai Li, Houlong Zhao, Lei Han, Yunhai Tong, and Kuiyuan Yang. Gff: Gated fully fusion for semantic segmentation. *arXiv preprint arXiv:1904.01803*, 2019.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. In *Proceedings of the IEEE international conference on computer vision*, pp. 1377–1385, 2015.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 4898–4906, 2016.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Adam Paszke, Sam Gross, Soumith Chintala, and Gregory Chanan. Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration. *PyTorch: Tensors and dynamic neural networks in Python with strong GPU acceleration*, 6, 2017.

Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5639–5647, 2018.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

Evan Shelhamer, Dequan Wang, and Trevor Darrell. Blurring the line between structure and learning to optimize and adapt receptive fields. *arXiv preprint arXiv:1904.11487*, 2019.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Hang Su, Varun Jampani, Deqing Sun, Orazio Gallo, Erik Learned-Miller, and Jan Kautz. Pixel-adaptive convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11166–11175, 2019.

Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. *arXiv preprint arXiv:1907.05740*, 2019.

Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Iccv*, volume 98, pp. 2, 1998.

Aaron van den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pp. 6306–6315, 2017.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, 2018.

Huikai Wu, Shuai Zheng, Junge Zhang, and Kaiqi Huang. Fast end-to-end trainable guided filter. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1838–1847, 2018a.

Jialin Wu, Dai Li, Yu Yang, Chandrajit Bajaj, and Xiangyang Ji. Dynamic sampling convolutional neural networks. *arXiv preprint arXiv:1803.07624*, 2018b.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

Tianfan Xue, Jiajun Wu, Katherine Bouman, and Bill Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Advances in neural information processing systems*, pp. 91–99, 2016.

Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 472–480, 2017.

Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2403–2412, 2018.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017.

Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 1529–1537, 2015.

Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9308–9316, 2019.

## A ARCHITECTURE AND CONFIGURATIONS

With respect to segmentation tasks, five deep neural networks, as backbones, are modified by incorporating PAD layers. Specific architecture and configuration information is listed in Table 8. For classification tasks, four deep models, VGG-16, ResNet-50, DRN-C-26, and MobileNet-v2, are modified to PAD-Nets and their architectures and configurations on corresponding PAD-Nets are listed by Table 9. Concerning fine-grained tasks, three deep models, including ResNet-50, DRN-C-26, and MobileNet-V2, are modified to PAD-Nets, and specific architecture and configuration information is listed in Table 10. All PAD-Nets are trained by following Markov Aggregation.

Table 8: PAD-Nets on semantic segmentation tasks. (Section 4.5)

| **PAD-Nets** | **Architectures Description** | **Configurations** |
|---|---|---|
| FCN8s/FCN32s + VGG-16 | Conv-5 is modified to PAD layers. Dilation variables are learnt with 1, 2, and 4. | Follow FCN8s/FCN32s training scheme with no augmentation or image pre-processing or post-processing. FCN8s is trained in at once mode. |
| ResNet-101 + DeeplabV3+ | The 3x3 convolution layers in Layer-4 block are modified to PAD layers. Dilation variables are learnt with 1, 2, and 4. | Follow deeplab VOC-2012 training scheme with batch size of 8 and outstride of 16. |
| DRN-D-54 + DeeplabV3+ | Layer-7 and Layer-8 are modified to PAD layers. Dilation variables are learnt with 1, 2, and 4. | Follow deeplab VOC-2012 training scheme with batch size of 8 and outstride of 16. |
| Xception + DeeplabV3+ | The 3x3 convolution layers are modified in Middle-flow and Exit-flow where strides are 1. | Follow deeplab VOC-2012 training scheme with batch size of 8, outstride of 16, no BN fine-tuning, and multi-scale testing. |
| MobileNetV2 + DeeplabV3+ | The 3x3 convolution layers with stride 1 in depth-wise modules with output channels of 160 and 320 are modified to PAD layers. | Follow deeplab VOC-2012 training scheme with batch size of 8, outstride of 16, no BN fine-tuning, and multi-scale testing. |

Table 9: PAD-Nets on large-scale image classification tasks. (Section 5.1)

| **PAD-Nets** | **Architectures Description** | **Configurations** |
|---|---|---|
| VGG-16 | Only Conv-5 is incorporated with PAD units. The dilation variables are learnt based on 1, 2, and 4. | Follow Pytorch ImageNet training default settings with 128 batch size and 120 epochs. |
| ResNet-50 | The last three convolution layers in Layer-2 block are modified with PAD units. The dilation variable are learnt based on 1, 2, and 4. | Follow DRN ImageNet training scheme with 192 batch size and 120 epochs. |
| DRN-C-26 | Layer-8 is modified with PAD units. The dilation variables are learnt based on 1, 2, and 4. | Follow original DRN reported training scheme with 192 batch size and 120 epochs. |
| MobileNetV2 | The Layer-2 with t=6, c=32, n=3, and s=2 is modified with PAD units. The dilation variables are learnt based on 1, 2, and 4. | Follow the setting of 1.0-224 reported in Sandler et al. (2018) with 256 batch size and 300 epochs. |

Table 10: PAD-Nets on fine-grained image classification tasks. (Section 5.2)

| **PAD-Nets** | **Architectures Description** | **Configurations** |
|---|---|---|
| ResNet-50 | The last three convolution layers in Layer-2 block are modified with PAD units. The dilation variable are learnt based on 1, 2, and 3. | The setting follows that of DLA reported in Yu et al. (2018) with two crop sizes, i.e., 224 and 448. |
| DRN-C-26 | Layer-7 and Layer-8 are modified with PAD units. The dilation variables are learnt based on 1, 2, and 3. | The same as the above. |
| MobileNetV2 | The Layer-2 with t=6, c=32, n=3, and s=2 is modified with PAD units.The dilation variables are learnt based on 1, 2, and 3. | The same as the above. |

Table 11: PAD-FCN8s IoUs on VOC-2012 across all classes

| | background | aeroplane | bicycle | bird | boat | bottle | bus | car | cat | chair | cow |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PAD-FCN8s | **0.914** | **0.833** | **0.388** | 0.751 | **0.627** | 0.740 | **0.802** | 0.744 | **0.805** | 0.252 | **0.805** |
| FCN8s | 0.908 | 0.798 | 0.363 | **0.776** | 0.581 | **0.742** | 0.775 | **0.749** | 0.799 | **0.292** | 0.712 |
| | dining table | dog | horse | motorbike | person | potted plant | sheep | sofa | train | tv/monitor | mIoU |
| PAD-FCN8s | **0.474** | **0.724** | **0.729** | **0.783** | **0.791** | **0.510** | **0.729** | **0.370** | 0.773 | **0.600** | **0.665** |
| FCN8s | 0.375 | 0.684 | 0.673 | 0.765 | 0.780 | 0.490 | 0.760 | 0.344 | **0.789** | 0.572 | 0.647 |

## B More results on semantic segmentation and fine-grained image classification

More quality results on semantic segmentation tasks are shown in Figure 7, Figure 8, and Figure 9. More class IoUs are included in Table 11, Table 12, Table 13, and Table 14. For fine-grained classification tasks, detailed feature maps for fine-grained image classification are shown in Figure 11.

## C Training Curve for Large-scale Image Classification

The training curve for large-scale image classification is shown in Figure 10. We can see PAD-Nets have similar and even better convergence rates comparing to their conventional counterparts.

Table 12: PAD-ResNet-101 IoUs on VOC-2012 across all classes

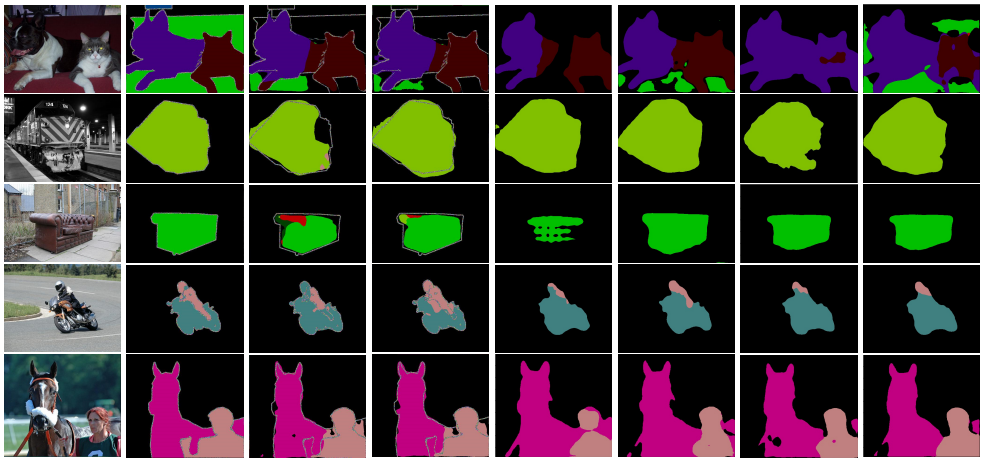
| | background | aeroplane | bicycle | bird | boat | bottle | bus | car | cat | chair | cow |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PAD-ResNet-101 | **0.932** | **0.838** | **0.393** | 0.848 | 0.622 | **0.756** | 0.908 | **0.848** | **0.918** | **0.373** | **0.874** |
| ResNet-101 | 0.922 | 0.770 | 0.388 | **0.853** | **0.626** | 0.698 | **0.913** | 0.836 | 0.886 | 0.225 | 0.835 |
| | dining table | dog | horse | motorbike | person | potted plant | sheep | sofa | train | tv/monitor | mIoU |
| PAD-ResNet-101 | **0.584** | **0.879** | **0.851** | 0.805 | **0.833** | **0.554** | **0.852** | **0.534** | **0.835** | 0.648 | **0.772** |
| ResNet-101 | 0.568 | 0.862 | 0.791 | **0.810** | 0.815 | 0.452 | 0.764 | 0.461 | 0.824 | **0.691** | 0.751 |

Table 13: PAD-DRN-54 IoUs on VOC-2012 across all classes

| | background | aeroplane | bicycle | bird | boat | bottle | bus | car | cat | chair | cow |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PAD-DRN-54-D | **0.927** | **0.823** | **0.384** | 0.845 | **0.668** | **0.729** | **0.915** | 0.838 | 0.852 | 0.294 | **0.876** |
| DRN-54-D | 0.921 | 0.799 | 0.345 | **0.846** | 0.660 | 0.723 | 0.868 | **0.848** | **0.884** | **0.313** | 0.820 |
| | dining table | dog | horse | motorbike | person | potted plant | sheep | sofa | train | tv/monitor | mIoU |
| PAD-DRN-54-D | **0.568** | 0.839 | **0.836** | **0.814** | **0.813** | **0.491** | **0.805** | 0.434 | **0.781** | **0.693** | **0.772** |
| DRN-54-D | 0.528 | **0.840** | 0.801 | 0.805 | 0.800 | 0.475 | 0.739 | **0.492** | 0.750 | 0.675 | 0.754 |

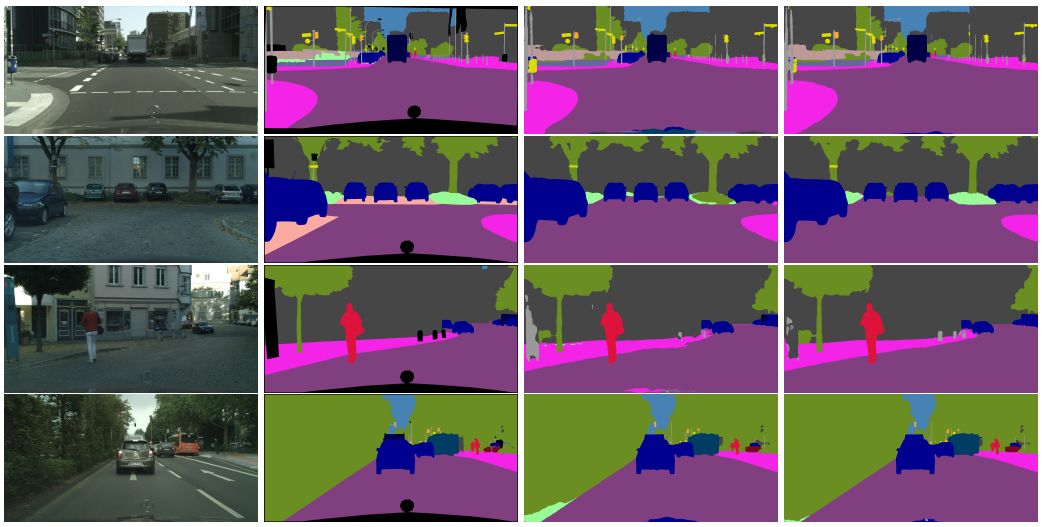
Table 14: PAD-ResNet-101 IoUs on Cityscapes across all classes

| | road | sidewalk | building | wall | fence | pole | light | sign | vegetation | terrain |
|---|---|---|---|---|---|---|---|---|---|---|
| PAD-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** |
| | sky | person | rider | car | truck | bus | train | motorcycle | bicycle | mIoU |
| PAD-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 |



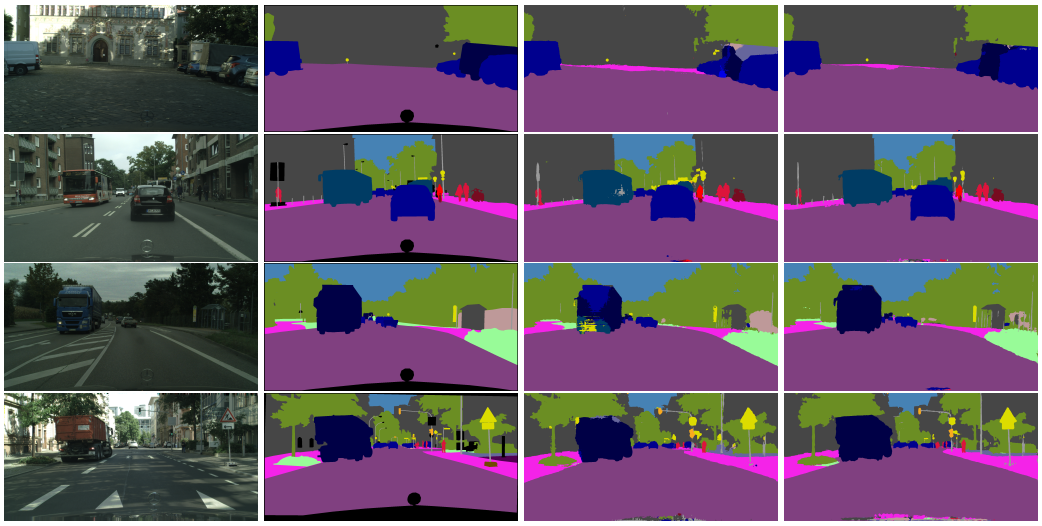
(a) Image (b) GT (c) FCN8s (d) PAD-FCN8s (e) ResNet-101 (f) PAD-ResNet-101 (g) DRN (h) PAD-DRN

Figure 7: Semantic segmentation results on Pascal VOC 2012.



(a) Image (b) GT (c) ResNet-101 (d) PAD-ResNet-101

Figure 8: Semantic segmentation results from ResNet-101 and PAD-ResNet-101 on Cityscapes.



(a) Image (b) GT (c) Xception (d) PAD-Xception

Figure 9: Semantic segmentation results from Xception and PAD-Xception on Cityscapes.

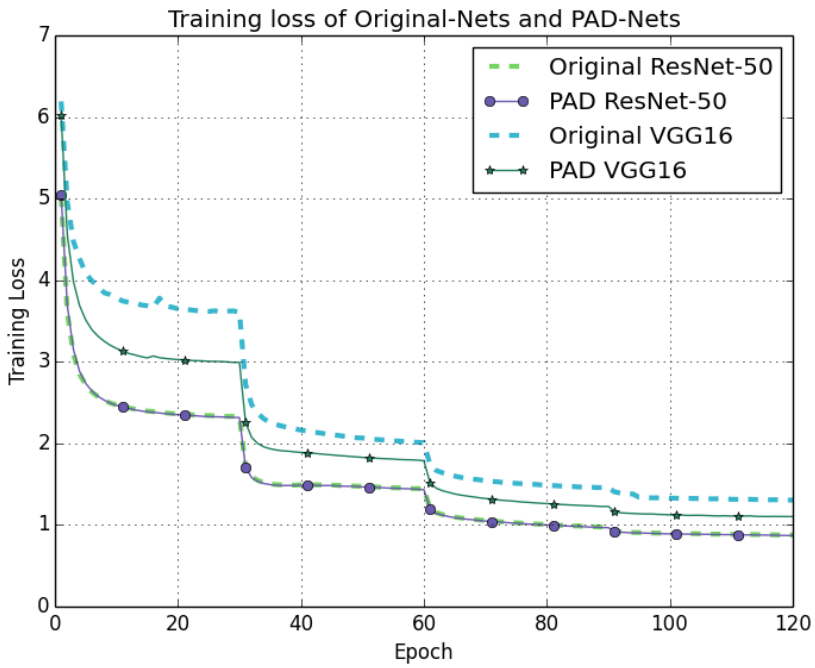
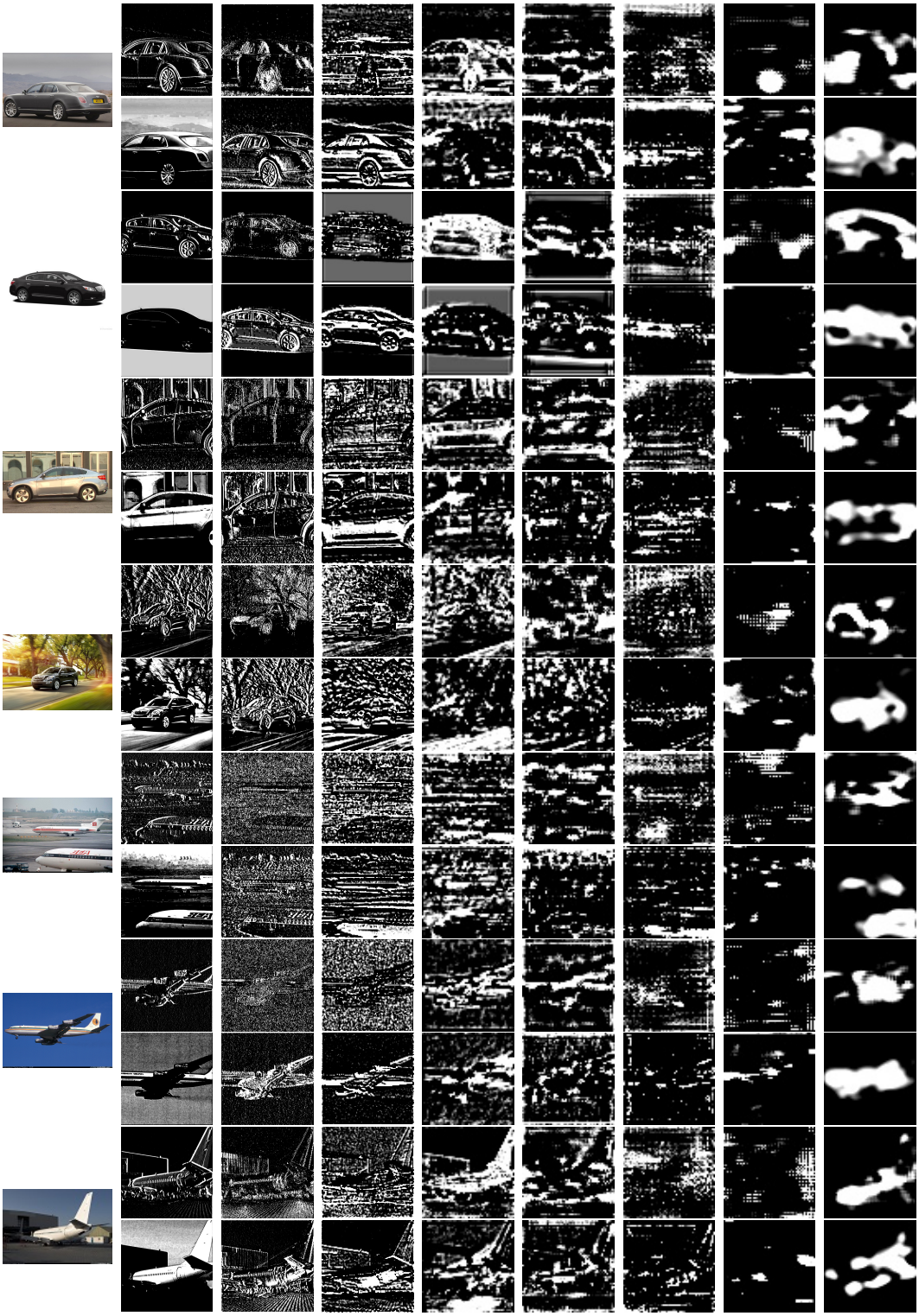



Figure 10: The training curve of large-scale image classification using PAD-Nets based on the VGG-16 and ResNet-50 .



(a) Image (b) Layer1 (c) Layer2 (d) Layer3 (e) Layer4 (f) Layer5 (g) Layer6 (h) Layer7 (i) Layer8

Figure 11: Activation maps of regular DRN-C-26 (Odd rows) and PAD-DRN-C-26 (Even row) for samples from Stanford Cars and FGVC-Airecrafts.