Delving Deeper into Anti-aliasing in ConvNets

Xueyan Zou[1]
zxyzou@ucdavis.edu
Fanyi Xiao[1]
fyxiao@ucdavis.edu
Zhiding Yu[2]
zhidingy@nvidia.com
Yong Jae Lee[1]
yongjaelee@ucdavis.edu

[1] University of California, Davis
Davis, CA, USA
[2] NVIDIA
Santa Clara, CA, USA

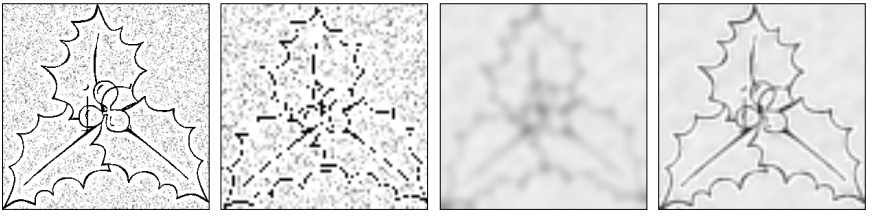
Abstract

Aliasing refers to the phenomenon that high frequency signals degenerate into completely different ones after sampling. It arises as a problem in the context of deep learning as downsampling layers are widely adopted in deep architectures to reduce parameters and computation. The standard solution is to apply a low-pass filter (e.g., Gaussian blur) before downsampling [37]. However, it can be suboptimal to apply the same filter across the entire content, as the frequency of feature maps can vary across both spatial locations and feature channels. To tackle this, we propose an adaptive content-aware low-pass filtering layer, which *predicts separate filter weights for each spatial location and channel group* of the input feature maps. We investigate the effectiveness and generalization of the proposed method across multiple tasks including ImageNet classification, COCO instance segmentation, and Cityscapes semantic segmentation. Qualitative and quantitative results demonstrate that our approach effectively adapts to the different feature frequencies to avoid aliasing while preserving useful information for recognition. Code is available at https://maureenzou.github.io/ddac/.

1 Introduction

Deep neural networks have led to impressive breakthroughs in visual recognition, speech recognition, and natural language processing. On certain benchmarks such as ImageNet and SQuAD, they can even achieve "human-level" performance [11, 24, 26, 31]. However, common mistakes that these networks make are often quite *unhuman* like. For example, a tiny shift in the input image can lead to drastic changes in the output prediction of convolutional neural networks (ConvNets) [1, 28, 31]. This phenomenon was demonstrated to be partially due to *aliasing* when downsampling in ConvNets [37].

Aliasing refers to the phenomenon that high frequency information in a signal is distorted during subsampling [9]. The Nyquist theorem states that the sampling rate must be at least twice the highest frequency of the signal in order to prevent aliasing. Without proper anti-aliasing techniques, a subsampled signal can look completely different compared to its input. Below is a toy example demonstrating this problem on 1D signals:



(a) Input (b) 4x Down (c) Gaussian + 4x Down (d) Adaptive + 4x Down

Figure 1: **Effect of adaptive filtering for anti-aliasing.** (a) Input image. (b) Result of direct downsampling. (c) Result of downsampling after applying a single Gaussian filter tuned to match the frequency of the noise. (d) Result of downsampling after applying spatially-adaptive Gaussian filters (stronger blurring for background noise and weaker for edges).

$$\begin{aligned} 001100110011 &\xrightarrow[\text{maxpool}]{\text{k=2, stride=2}} 010101 \\ 011001100110 &\xrightarrow[\text{maxpool}]{\text{k=2, stride=2}} 111111 \end{aligned} \tag{1}$$

Here k is the kernel size (1×2). Because of aliasing, a one position shift in the original signal leads to a completely different sampled signal (bottom) compared to the original sampled one (top). As downsampling layers in ConvNets are critical for reducing parameters and inducing invariance in the learned representations, the aliasing issue accompanying these layers will likely result in a performance drop as well as undesired shift variance in the output if not handled carefully.

To tackle this, [37] proposed to insert a Gaussian blur layer before each downsampling module in ConvNets. Though simple and effective to a certain degree, we argue that the design choice of applying a universal Gaussian filter is not optimal – as signal frequencies in a natural image (or feature map) generally vary throughout spatial locations and channels, different blurring filters are needed in order to satisfy the Nyquist theorem to avoid aliasing. For example, the image in Fig. 1 (a) contains high frequency impulse noise in the background and relatively lower frequency edges in the foreground. Directly applying a downsampling operation produces discontinuous edges and distorted impulse noise shown in (b) due to aliasing. By applying a Gaussian filter before downsampling, we can avoid aliasing as shown in (c). However, as the high frequency impulse noise needs to be blurred more compared to the lower frequency edges, when using a single Gaussian filter tuned for the impulse noise, the edges are over-blurred leading to significant information loss. To solve this issue, what we need is to apply different Gaussian filters to the foreground and background separately, so that we can avoid aliasing while preserving useful information, as in (d).

With the above observation, we propose a *content-aware anti-aliasing* module, which adaptively predicts low-pass filter weights for different spatial locations. Furthermore, as different feature channels can also have different frequencies (e.g., certain channels capture edges, others capture color blobs), we also predict different filters for different channels. In this way, our proposed module adaptively blurs the input content to avoid aliasing while preserving useful information for downstream tasks. To summarize, our contributions are:

- We propose a novel adaptive and architecture-independent low-pass filtering layer in ConvNets for anti-aliasing.
- We propose novel evaluation metrics, which measure shift consistency for semantic and

instance segmentation tasks; i.e., a method's robustness to aliasing effects caused by shifts in the input.

- We conduct experiments on image classification (ImageNet), semantic segmentation (PASCAL VOC and Cityscapes), instance segmentation (MS-COCO), and domain generalization (ImageNet to ImageNet VID). The results show that our method outperforms competitive baselines with a good margin on both accuracy and shift consistency.
- We demonstrate intuitive qualitative results, which show the interpretability of our module when applied to different spatial locations and channel groups.

2 Related Work

Network robustness In deep learning, the robustness problem related to adversarial attacks [18, 30], input translation [2, 22, 36], and natural perturbations [28] has been widely studied. The crux of these studies is how small variations in the input image can lead to large variations in the predictions. In order to obtain a stable and robust network, [17, 19, 35] introduce novel losses or network architectures to defend against adversarial attacks. Apart from adversarial defense, [2, 22] propose new algorithms to learn more shift-invariant representations. Finally, [37] provides a new perspective on obtaining shift-invariant features in the context of anti-aliasing. Unlike [37], which applies a single hand-coded low-pass filter regardless of content, we adaptively learn the low-pass filter in a content-aware way and demonstrate it leads to improvement in both recognition accuracy and network robustness.

Image filtering Low-pass filters like box [27] and Gaussian [9] are classic *content agnostic* smoothing filters; i.e., their filter weights are fixed regardless of spatial location and image content. Bilateral [25] and guided [10] filters are *content aware* as they can simultaneously preserve edge information while removing noise. Recent works integrate such classic filters into deep networks [14, 35, 37]. However, directly integrating these modules into a neural network requires careful tuning of hyperparameters subject to the input image (e.g., σ_s and σ_r in bilateral filter or r and ε in guided filter). [16, 29] introduced the dynamic filtering layer, whose weights are predicted by convolution layers conditioned on pre-computed feature maps. We differ from them in two key aspects: 1) our filter weights vary across both spatial and channel groups, and 2) we insert our low-pass filtering layer before every downsampling layer for anti-aliasing, whereas the dynamic filtering layer is directly linked to the prediction (last) layer in order to incorporate motion information for video recognition tasks. Finally, [33] introduces an adaptive convolution layer for upsampling, whereas we focus on downsampling with an adaptive low-pass filtering layer.

Pixel classification tasks such as semantic segmentation [5, 21] and instance segmentation [8, 13] require precise modeling of object boundaries, so that pixels from the same object instance can be correctly grouped together. Thus, while blurring can help reduce aliasing, it can also be harmful to these tasks (e.g., when the edges are blurred too much or not blurred enough hence resulting in aliasing). We investigate the effect of anti-aliasing in these pixel-level tasks, whereas our closest work, [37], focused mainly on image classification.

3 Approach

To enable anti-aliasing for ConvNets, we apply the proposed *content-aware anti-aliasing* module before each downsampling operation in the network. Inside the module, we first

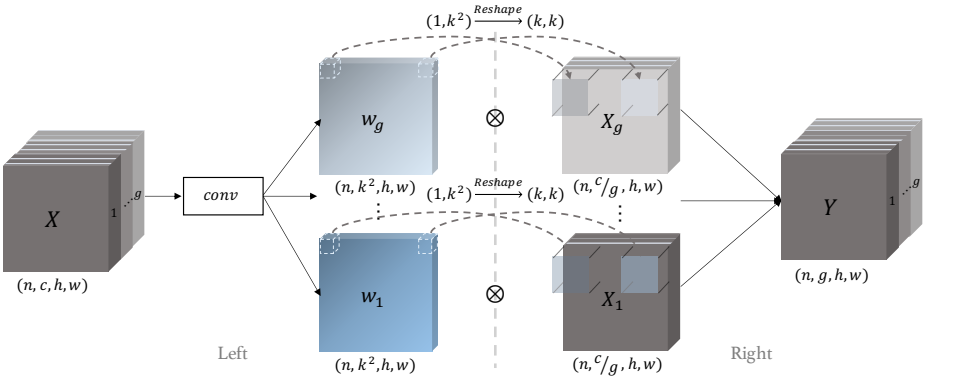


Figure 2: **Method overview.** (Left) For each spatial location and feature channel group in the input X, we predict a $k \times k$ filter w. (Right) We apply the learned filters on X to obtain content aware anti-aliased features. See text for more details.

generate low-pass filters for *different spatial locations and channel groups* (Fig. 2 left), and then apply the predicted filters back onto the input features for anti-aliasing (Fig. 2 right).

Spatial adaptive anti-aliasing. As frequency components can vary across different spatial locations in an image, we propose to learn different low-pass filters in a content-aware manner across spatial locations. Specifically, given an input feature X that needs to be downsampled, we generate a low-pass filter $w_{i,j}$ (e.g., a 3×3 conv filter) for each spatial location (i,j) on x. With the predicted low-pass filter $w_{i,j}$, we can then apply it to input X:

$$Y_{i,j} = \sum_{p,q \in \Omega} w_{i,j}^{p,q} \cdot X_{i+p,j+q}, \tag{2}$$

where $Y_{i,j}$ denotes output features at location (i,j) and Ω points to the set of locations surrounding (i,j) on which we apply the predicted smooth filter. In this way, the network can learn to blur higher frequency content more than lower frequency content, to reduce undesirable aliasing effects while preserving important content as much as possible.

Channel-grouped adaptive anti-aliasing. Different channels of a feature map can capture different aspects of the input that vary in frequency (e.g., edges, color blobs). Therefore, in addition to predicting different filters for each spatial location, it can also be desirable to predict different filters for each *feature channel*. However, naively predicting a low-pass filter for each spatial location and channel can be computationally very expensive. Motivated by the observation that some channels will capture similar information [34], we group the channels into k groups and predict a single low-pass filter $w_{i,j,g}$ for each group g. Then, we apply $w_{i,j,g}$ to the input X:

$$Y_{i,j}^{g} = \sum_{p,q \in \Omega} w_{i,j,g}^{p,q} \cdot X_{i+p,j+q}^{c}, \tag{3}$$

where g is the group index to which channel c belongs. In this way, channels within a group are learned to be similar, as shown in Fig. 4.

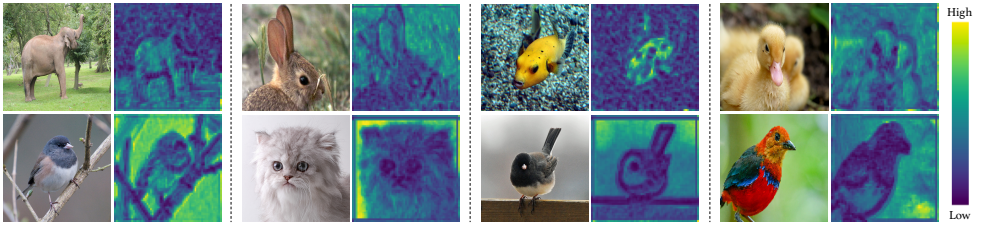


Figure 3: **Variance of the learned filter weights across spatial locations.** Low variance corresponds to more blur, while high variance corresponds to less blur. Our model correctly learns to blur high frequency content (e.g., edges) more to prevent aliasing, and blur low frequency content less to preserve useful information.

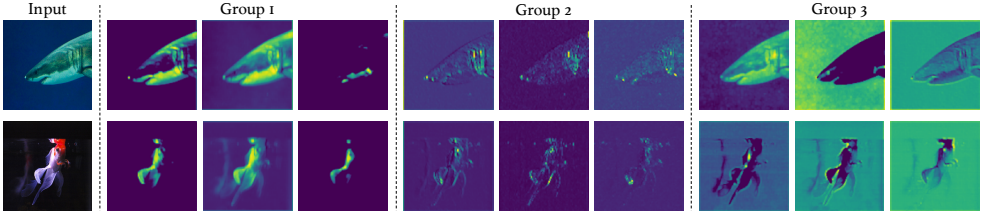


Figure 4: **Visualization of predicted feature maps within and across groups.** The features within each group are more similar to each other than to those in other groups. Each group captures a different aspect of the image (e.g., edges, color blobs).

Learning to predict filters. To dynamically generate low-pass filters for each spatial location and feature channel group, we apply a convolutional block (conv + batchnorm) to the input feature $X \in R^{n\times c\times h\times w}$ to output $w \in R^{n\times g\times k^2\times h\times w}$, where g denotes the number of channel groups and each of the k^2 channels corresponds to an element in one of $k \times k$ locations in the filters. For grouping, we group every c/g consecutive channels, where c is the total number of channels. Finally, to ensure that the generated filters are low-pass, we constrain their weights to be positive and sum to one by passing it through a softmax layer.

Analyzing the predicted filters. In this section, we analyze the behavior of our learned filters. First, we analyze how the filters spatially adapt to different image content. For this, we compute the variance of the learned filter weights across different spatial locations. A $k \times k$ average filter with $1/k^2$ intensity in each element will have zero variance whereas an identity filter with one in the center and zeros everywhere else will have high variance. From Fig. 3, one can clearly see that when the image content has high frequency information (e.g., elephant background trees, bird contours), the learned filters' variance tends to be smaller; i.e., more blur is needed to prevent aliasing. Conversely, the filters' variance is larger when the content is relatively smoother (e.g., background in bird images); i.e., less blur is needed to prevent aliasing. In this way, the learned filters can reduce aliasing during sampling while preserving useful image content as much as possible.

We next analyze how the filters adapt to different content across different feature groups. Fig. 4 shows this effect; e.g., group 1 captures relatively low frequency information with smooth areas, while group 2 captures higher frequency information with sharp intensity transitions. In this way, the learned filters can adapt to different frequencies across feature channels, while saving computational costs by learning the same filter per group.

methods	Filter	accuracy Abs	accuracy Delta	consistency Abs	consistency Delta	generalization Abs	generalization Delta
ResNet-101 [12]	-	77.7	-	90.6	-	67.6	-
LPF [37]	3 x 3	78.4	+ 0.7	91.6	+ 1.0	68.8	+1.2
	5 x 5	77.7	+ 0.0	91.8	+ 1.2	67.0	- 0.6
Ours	3 x 3	**79.0**	**+ 1.3**	91.8	+ 1.2	**69.9**	**+2.3**
	5 x 5	78.6	+ 0.9	**92.2**	**+ 1.6**	69.1	+1.5

Table 1: Image classification accuracy, consistency on ImageNet [7], and domain generalization results ImageNet → ImageNet VID [7]. We compare to strong ResNet-101 [12] and LPF (low-pass filter) [37] baselines. Our method shows consistent improvement in accuracy, consistency, and generalization.

4 Experiments

We first introduce our experimental settings and propose consistency metrics for image classification, instance segmentation, and semantic segmentation. We compare to strong baselines including ResNet [12], Deeplab v3+ [5], Mask R-CNN on large scale datasets including ImageNet, ImageNet VID [7], MS COCO [20], PASCAL VOC [8] and Cityscapes [6]. We also conduct ablation studies on our design choices including number of groups, parameter counts, as well as filter types. Finally, we present qualitative results demonstrating the interpretability of our anti-aliasing module.

4.1 Image Classification

Experimental settings We evaluate on ILSVRC2012 [7], which contains 1.2M training and 50K validation images for 1000 object classes. We use input image size of 224×224, SGD solver with initial learning rate 0.1, momentum 0.9, and weight decay 1e-4. Full training schedule is 90 epochs with 5 epoch linear scaling warm up. Learning rate is reduced by 10x every 30 epochs. We train on 4 GPUs, with batch size 128 and batch accumulation of 2. For fair comparison, we use the same set of hyperparameters and training schedule for both ResNet-101, LPF [37] baselines as well as our method. The number of groups is set to 8 according to our ablation study. We extend the code base introduced in [37].

Consistency metric We use the consistency metric defined in [37], which measures how often the model outputs the same top-1 class given two different shifts on the same test image: $\textit{Consistency} = \mathbb{E}_{X,h_1,w_1,h_2,w_2}\, \mathbb{I}\{F(X_{h_1,w_1}) = F(X_{h_2,w_2})\}$, where $\mathbb{E}$ and $\mathbb{I}$ denote expectation and indicator function (outputs 1/0 with true/false inputs). X is the input image, h_1, w_1 (height/width) and h_2, w_2 parameterize the shifts and $F(\cdot)$ denotes the predicted top-1 class.

Results and analysis As shown in Table 1, our adaptive anti-aliasing module outperforms the baseline ResNet-101 without anti-aliasing with a 1.3 point boost (79.0 vs 77.7) in top-1 accuracy on ImageNet classification. More importantly, when comparing to LPF [37], which uses a fixed blurring kernel for anti-aliasing, our method scores 0.6 points higher (79.0 vs 78.4) on top-1 accuracy. Furthermore, our method not only achieves better classification accuracy, it also outputs more consistent results (+0.2/+0.4 consistency score improvements for 3×3 and 5×5 filter sizes) compared to LPF. These results reveal that our method preserves more discriminative information for recognition when blurring feature maps.

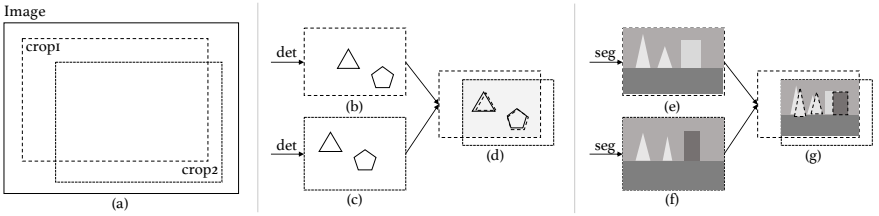


Figure 5: Our new consistency metrics. (b,c,d): mean Average Instance Segmentation Consistency (mAISC). (e,f,g): mean Average Semantic Segmentation Consistency (mASSC).

4.2 Domain Generalization

Experimental settings ImageNet VID is a video object detection dataset, which has 30 classes that overlap with 284 classes in ImageNet (some classes in ImageNet VID are the super class of ImageNet). It contains 3862/1315 training/validation videos. We randomly select three frames from each validation video, and evaluate Top-1 accuracy on them to measure the generalization capability of our model which is pretrained on ImageNet (i.e. it has never seen any frame in ImageNet VID). As a video frame may contain multiple objects in different classes, we count a prediction as correct as long as it belongs to one of the ground-truth classes.

Results and analysis Table 1 reveals that our method generalizes better to a different domain compared to the ResNet-101 baseline (+2.3% points increase in top-1 accuracy for 3×3 filter) and LPF model (+1.1%) which adopts a fixed blur kernel. We hypothesize that the better generalization capability comes from the fact that we learn a representation that is less sensitive to downsampling (i.e., more robust to shifts). This is particularly useful for video frames, as they can be thought of as having natural shift perturbations of the same content across frames [28].

4.3 Instance Segmentation

Experimental settings In this section, we present results on MS-COCO for instance segmentation [20]. MS-COCO contains 330k images, 1.5M object instances and 80 categories. We use Mask R-CNN [13] as our base architecture. We adopt the hyperparameter settings from the implementation of [23]. When measuring consistency, we first resize images to 800×800 and then take a crop of 736×736 as input.

Consistency metric (mAISC) We propose a new mean Average Instance Segmentation Consistency (mAISC) metric to measure the shift invariance property of instance segmentation methods. As shown in Fig. 5, given an input image (a), we randomly select two crops (b) and (c), and apply an instance segmentation method on them separately. $M(b)$ and $M(c)$ denote the predicted instances in the overlapping region of image (b) and (c). To measure consistency, for any given instance m_b in $M(b)$ we find its highest overlapping counterpart m_c in $M(c)$. If the IOU between m_b and m_c is larger than a threshold (0.9 in our experiments), we regard m_b as a positive (consistent) sample in $M(b)$. (A sample m_c from $M(c)$ can only be considered a counterpart of any instance in $M(b)$ once.) We compute the final mAISC score as the mean percentage of positive samples in $M(b)$ over all input image pairs.

	Mask				Box			
method	mAP	Delta	mAISC	Delta	mAP	Delta	mAISC	Delta
Mask R-CNN [13]	36.1	-	62.9	-	40.1	-	65.1	-
LPF [37]	36.8	+ 0.7	66.0	+ 4.1	40.9	+ 0.8	68.8	+ 3.7
Ours	**37.2**	**+ 1.1**	**67.0**	**+ 5.1**	**41.4**	**+ 1.3**	**69.8**	**+ 4.7**

Table 2: Instance segmentation results on MS COCO. We compare to Mask R-CNN [13] and LPF [37]. Our approach consistently improves over the baselines for both mask and box accuracy and consistency.

	PASCAL VOC				Cityscapes			
method	mIOU	Delta	mASSC	Delta	mIOU	Delta	mASSC	Delta
Deeplab v3+ [5]	78.5	-	95.5±0.11	-	78.5	-	96.0±0.10	-
LPF [37]	79.4	+ 0.9	95.9±0.07	+ 0.4	78.9	+ 0.4	96.1±0.05	+ 0.1
Ours	**80.3**	**+ 1.8**	**96.0±0.13**	**+ 0.5**	**79.5**	**+ 1.0**	**96.3±0.07**	**+ 0.3**

Table 3: Semantic segmentation on PASCAL VOC 2012 [8] and Cityscapes [6]. We compare to Deeplab v3+ [5] and LPF [37]. Our approach leads to improved accuracy and consistency.

Results and analysis We evaluate mAP and mAISC for both mask and box predictions. As shown in Table 2, while simply applying a fixed Gaussian low-pass filter improves mAP by +0.7/+0.8 points for mask/box, our adaptive content-aware anti-aliasing module is more effective (further +0.4/+0.5 point improvement over LPF for mask/box). This demonstrates that it is important to have different low-pass filters for different spatial locations and channel groups. More interestingly, by introducing our adaptive low-pass filters, mAISC increases by a large margin (+5.1/+4.7 for mask/box over the baseline, and +1.0/+1.0 over LPF). This result demonstrates that 1) an anti-aliasing module significantly improves shift consistency via feature blurring, and 2) edges (higher frequency) are better preserved using our method (compared to LPF) during downsampling which are critical for pixel classification tasks.

4.4 Semantic Segmentation

Experimental settings We next evaluate on PASCAL VOC2012 [8] and Cityscapes [6] semantic segmentation with Deeplab v3+ [5] as the base model. We extend implementations from [15] and [32]. For Cityscapes, we use syncBN with a batch size of 8. As for PASCAL VOC, we use a batch size of 16 on two GPUs without syncBN. We report better performance compared to the original implementation for DeepLab v3+ on PASCAL VOC. For Cityscapes, our ResNet-101 backbone outperforms the Inception backbone used in [4].

Consistency metric (mASSC) We propose a new mean Average Semantic Segmentation Consistency (mASSC) metric to measure shift consistency for semantic segmentation methods. Similar to mAISC, we take two random crops (e,f) from the input image (a) in Fig. 5. We then compute the Semantic Segmentation Consistency between the overlapping regions X and Y of the two crops: $Consistency(X,Y) := \mathbb{E}_{i\in[0,h)}\mathbb{E}_{j\in[0,w)}\,\mathbb{I}[S(X)_{i,j} = S(Y)_{i,j}]$, where $S(X)_{i,j}$ and $S(Y)_{i,j}$ denote the predicted class label of pixel (i,j) in X and Y, and h,w is the height and width of the overlapping region. We average this score for all pairs of crops in an image, and average those scores over all test images to compute the final mASSC.

Results and analysis As shown in Table 3, our method improves mIOU by 1.8 and 1.0 points on PASCAL VOC and Cityscapes compared to the strong baseline of DeepLab v3+. Furthermore, our method also consistently improves the mASSC score (+0.5 and +0.3 for VOC and Cityscapes) despite the high numbers achieved by the baseline method (95.5/96.0).

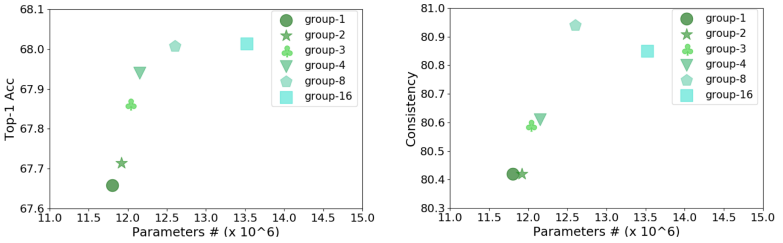


Figure 6: Effect of number of groups on top-1 accuracy and consistency.

methods	top-1 Acc	consistency
ResNet	66.5	79.1
Gaussian	66.7	79.8
Image Adaptive	66.7	78.7
Spatial Adaptive	67.7	80.3
Ours	**68.0**	**80.9**

Table 4: Filter ablations. Gaussian blur is better than no blur (ResNet). Learning the blur filter globally (Image Ada.), spatially (Spatial Ada.), and over channels (Ours) progressively does better.

Finally, to measure the variance of our mASSC results, we report the standard deviation over three runs with different random seeds.

4.5 Ablation Studies

Experimental settings For efficiency, we perform all ablation studies using ResNet-18 with input image size 112×112 and batch size 200 on ImageNet. All other hyperparameters are identical to those used in Sec. 4.1.

Number of channel groups. We vary the number of channel groups and study its influence on image classification accuracy. As shown in Fig. 6, the trend is clear – increasing the number of groups generally leads to improved top-1 accuracy. This demonstrates the effectiveness of predicting different filters across channels. However, there exists a diminishing return in this trend – the performance saturates when the group number goes beyond 8. We hypothesize this is caused by overfitting.

Number of parameters. We further compare the effects of directly increasing the number of parameters in the base network *vs* adding more groups in our content-aware low-pass filters. To increase the number of parameters for the base network, we increase the base channel size in ResNet-18. We find that directly increasing the number of parameters barely improves top-1 accuracy – when the number of parameters increases from 12.17M to 12.90M, top-1 accuracy increases only by 0.1%. Also, with similar (or less) number of parameters, our method yields a higher performance gain compared to naively increasing network capacity (68.0% *vs* 67.7% top-1 accuracy for 12.60M *vs* 12.90M parameters). This shows that our adaptive anti-aliasing method does not gain performance by simply scaling up its capacity.

Type of filter. In Table 4, we ablate our pixel adaptive filtering layers with various baseline components. Applying the same low-pass filter (Gaussian, Image Adaptive) across the entire image performs better than the vanilla ResNet-18 without any anti-aliasing. Here, Image Adaptive refers to the baseline which predicts a single low-pass filter for the entire image. By adaptively learning a spatially variant low-pass filter, performance improves further (Spatial Adaptive). Overall, our method achieves the best performance which demonstrates the

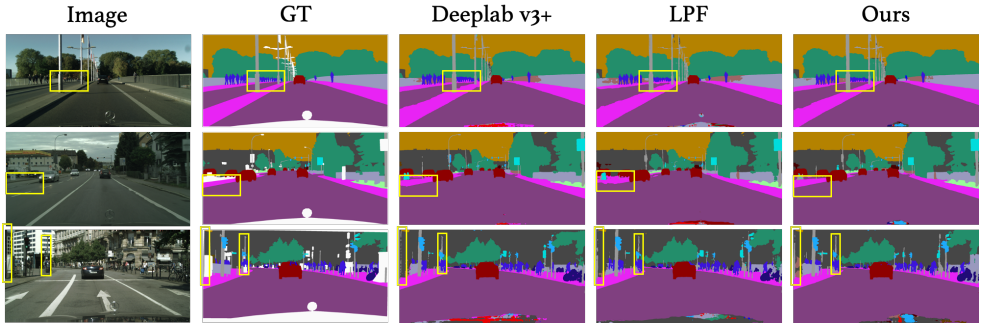


Figure 7: Qualitative results for semantic segmentation on Cityscapes.

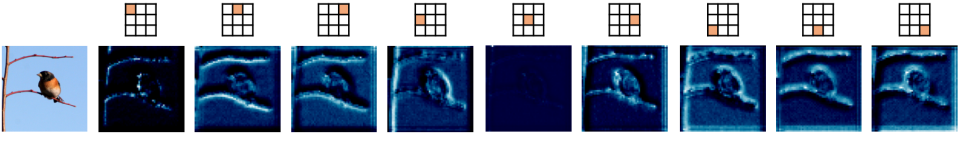


Figure 8: Visualization of learned filter weights at each spatial location.

benefits of predicting filters that are both spatially varying and channel adaptive.

Overhead. Finally, with our spatial/channel adaptive filtering added, the number of parameters increases by 2.9-7.8% for ResNet models (e.g., 4% for R-101, 4.5M to 4.63M). As for runtime, on a RTX2070 GPU, our method (R-101 backbone) takes 6.4 ms to forward a 224x224 image whereas a standard ResNet-101 takes 4.3 ms.

4.6 Qualitative Results

Semantic Segmentation. We show qualitative results for semantic segmentation in Fig. 7 to demonstrate that our module better preserves edge information. For example, in the first row, within the yellow box region, our method clearly distinguishes the road edge compared to Deeplab v3+ and LPF. Similar behavior (better segmented road contours) is also observed in the second row. This holds for other objects as well – the light pole has better delineation compared to both baselines in the third row.

Low-pass filter weights. To further understand our adaptive filtering module, we visualize the low-pass filter weights for each spatial location. As shown in Fig. 8, our model tends to "grow" edges so that it's easier for them to be preserved. For example, the learned filter tends to integrate more information from left to right (see center-left and bottom-left weights in Fig. 8) on the vertical tree branch and thus grow it to be thicker. This way, it's easier for tree branch contours to be preserved after downsampling.

5 Conclusion

In this paper, we proposed an adaptive content-aware low-pass filtering layer, which predicts separate filter weights for each spatial location and channel group of the input. We quantitatively demonstrated the effectiveness of the proposed method across multiple tasks and qualitatively showed that our approach effectively adapts to the different feature frequencies to avoid aliasing while preserving useful information for recognition.

6 Acknowledgements

This work was supported in part by ARO YIP W911NF17-1-0410, NSF CAREER IIS-1751206, NSF CCF-1934568, GCP research credit program, and AWS ML research award.

References

[1] Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? In *JMLR*, 2018.

[2] Alberto Bietti and Julien Mairal. Invariance and stability of deep convolutional representations. In *NeurIPS*, 2017.

[3] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. YOLACT: real-time instance segmentation. In *ICCV*, 2019.

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

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

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

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

[8] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. In *IJCV*, 2015.

[9] Rafael C Gonzales and Richard E Woods. Digital image processing, 2002.

[10] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. In *ECCV*, 2010.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[14] Ping Hu, Bing Shuai, Jun Liu, and Gang Wang. Deep level sets for salient object detection. In *CVPR*, 2017.

[15] Ping Hu, Fabian Caba Heilbron, Oliver Wang, Zhe Lin, Stan Sclaroff, and Federico Perazzi. Temporally distributed networks for fast video semantic segmentation. In *CVPR*, 2020.

[16] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *NeurIPS*, 2016.

[17] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

[18] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[19] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *CVPR*, 2018.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[21] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[22] Julien Mairal, Piotr Koniusz, Zaid Harchaoui, and Cordelia Schmid. Convolutional kernel networks. In *NeurIPS*, 2014.

[23] Francisco Massa and Ross Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. https://github.com/facebookresearch/maskrcnn-benchmark, 2018. Accessed: [Oct.10 2019].

[24] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(1):529–533, 2015.

[25] Sylvain Paris, Pierre Kornprobst, Jack Tumblin, Frédo Durand, et al. Bilateral filtering: Theory and applications. *Foundations and Trends® in Computer Graphics and Vision*, 4(1):1–73, 2009.

[26] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*, 2016.

[27] D Rosenberg. Box filter, June 11 1974. US Patent 3,815,754.

[28] Vaishaal Shankar, Achal Dave, Rebecca Roelofs, Deva Ramanan, Benjamin Recht, and Ludwig Schmidt. A systematic framework for natural perturbations from videos. *arXiv preprint arXiv:1906.02168*, 2019.

[29] Hang Su, Varun Jampani, Deqing Sun, Orazio Gallo, Erik Learned-Miller, and Jan Kautz. Pixel-adaptive convolutional neural networks. In *CVPR*, 2019.

[30] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[31] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *ICML*, 2019.

[32] VainF. DeepLabv3Plus-Pytorch, 2020. URL https://github.com/VainF/DeepLabV3Plus-Pytorch.

[33] Jiaqi Wang, Kai Chen, Rui Xu, Ziwei Liu, Chen Change Loy, and Dahua Lin. Carafe: Content-aware reassembly of features. In *ICCV*, 2019.

[34] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018.

[35] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *CVPR*, 2019.

[36] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *CVPR*, 2019.

[37] Richard Zhang. Making convolutional networks shift-invariant again. In *ICML*, 2020.