Delving Deeper into Anti-Aliasing in ConvNets

Xueyan Zou[1] · Fanyi Xiao[2] · Zhiding Yu[3] · Yuheng Li[1] · Yong Jae Lee[1]

Received: 15 September 2021 / Accepted: 10 August 2022 / Published online: 2 October 2022

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 (Zhang in: ICML, 2020). 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 image classification, semantic segmentation, instance segmentation, video instance segmentation, and image-to-image translation. Both 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/.

Keywords Computer vision · Anti-aliasing · Neural network consistency · Neural network architecture

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 (Mnih et al., 2015; He et al., 2015; Tan & Le, 2019; Rajpurkar et al., 2016). 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) (Shankar et al., 2019; Azulay & Weiss, 2018; Tan & Le, 2019). This phenomenon was demonstrated to be in part due to *aliasing* when downsampling in ConvNets (Zhang, 2020).

Aliasing refers to the phenomenon that high frequency information in a signal is distorted during subsampling (Gonzales & Woods, 2002). 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:

$$001100110011 \xrightarrow[\text{maxpool}]{\text{k=2, stride=2}} 010101 \tag{1}$$

$$011001100110 \xrightarrow[\text{maxpool}]{\text{k=2, stride=2}} 111111 \tag{2}$$

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.

Communicated by William Smith.

✉ Xueyan Zou
xueyan@cs.wisc.edu

Fanyi Xiao
fanyix@fb.com

Zhiding Yu
zhidingy@nvidia.com

Yuheng Li
li2464@wisc.edu

Yong Jae Lee
yongjaelee@cs.wisc.edu

[1] University of Wisconsin-Madison, Madison, USA

[2] Meta AI, Menlo Park, USA

[3] NVIDIA, Santa Clara, USA

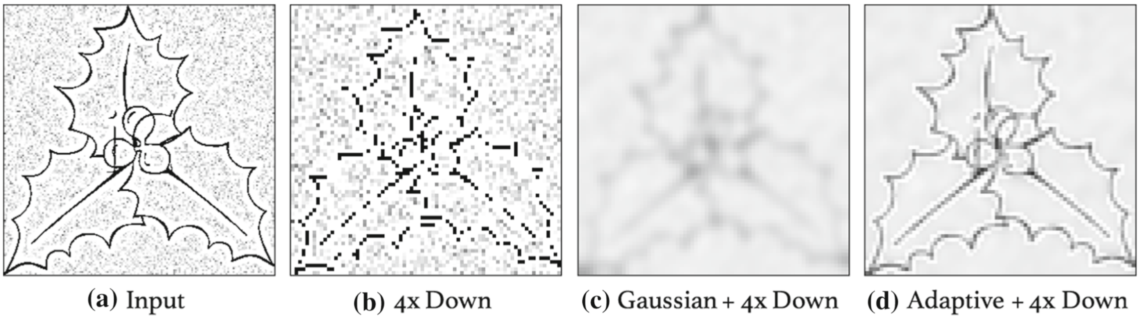

Fig. 1 Toy example demonstrating the effect of adaptive filtering for anti-aliasing. **a** Input image. **b** Result of direct downsampling. **c** Result of downsampling after applying a single Gaussian filter tuned to match the frequency of the noise. **d** Result of downsampling after applying spatially-adaptive Gaussian filters (stronger blurring for background noise and weaker for edges). We generate the background impulse noise using a Bernoulli distribution (with $P = 0.5$) per pixel location with a normal distribution determining the impulse noise magnitude. [We then overlay the foreground image over the background noise. For (c), the fixed filter value is generated by $g(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$, where σ is the standard deviation of the Gaussian filter, and (x, y) is the index of the filter location with $(0, 0)$ as the filter center. For (d), the filter "strength" is varied by σ as well as the kernel size]

To tackle this, Zhang (2020) 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. 1a 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), video instance segmentation (YoutubeVIS), and domain generalization (ImageNet to ImageNet VID, COCO to YoutubeVIS). 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.

This paper expands upon our previous conference paper (Zou et al., 2020) with the following new contributions:

- We propose a novel consistency metric for video instance segmentation and evaluate the robustness of our approach to video natural perturbation on the YoutubeVIS dataset (Sects. 4.5 and 4.8.3).
- We conduct experiments on the image-to-image translation task using pix2pixHD (Wang et al., 2018) as a baseline. Results in Sects. 4.6 and 4.8.4 show that our approach can generate more realistic images both qualitatively and quantitatively.
- We identify our adaptive filtering layer as a variant of the sliding window self-attention in vision transformers (Sect. 3.5).
- We give more comprehensive related work analysis in Sect. 2.
- We discuss some limitations of our approach in Sect. 5.

2 Related Work

Anti-aliasing Aliasing is a well-known problem in signal processing, and lowpass filters are often designed according to the Nyquist theorem to counter it Shannon (1949); Proakis and Manolakis (1992). In addition, the phenomenon has been studied under the scope of invariance in pattern recognition (Wood, 1996; Caelli & Liu, 1988; Li, 1992). More recently, it has been shown that aliasing also widely exists in deep neural networks and has non-negligible effect on the network predictions. For example, Zhang (2020) made the observation that network predictions are not consistent to shifting inputs and pointed out that these phenomena are caused by aliasing when a feature map is downsampled. Our subsequent work (Zou et al., 2020) further proposed *adaptive* filtering layers in place of the fixed low-pass filtering layers proposed in Zhang (2020) to better address the shift inconsistency problem. Recently, several concurrent works have either addressed aliasing issues in GANs using a continuous interpretation (Karras et al., 2021), or target the design of truly shift-invariant convnets with adaptive polyphase sampling (Chaman et al., 2021). Anti-aliasing is also highly related to geometric transformation invariance, which is explored in several recent works (Zhang et al., 2019; Lee et al., 2019; Bloem-Reddy & Teh, 2020; Rowley et al., 1998).

Network Robustness Current deep neural networks are vulnerable to input perturbations without special training recipes. These perturbations can be malicious such as adversarial attacks (Szegedy et al., 2013; Kurakin et al., 2017), or naturally occurring such as input translation (Mairal et al., 2014; Bietti & Mairal, 2017; Ye et al., 2019; Zhang, 2020), natural perturbations (Shankar et al., 2019), domain gaps (Muandet et al., 2013; Li et al., 2017), or out-of-distribution samples (Lee et al., 2017, 2018). One underlying reason is that networks tend to pick up superficial patterns instead of learning truly compositional representations (Geirhos et al., 2019), and their vulnerability to input perturbations can also lead to prediction inconsistencies. Adversarial defense methods via novel training pipelines (Madry et al., 2018; Liao et al., 2018), losses (Kannan et al., 2018) and architectures (Xie et al., 2019) have been proposed to obtain adversarially robust networks. Mairal et al. (2014); Bietti and Mairal (2017) propose new algorithms to learn more shift-invariant representations. In addition, data augmentation is an effective way to improve network robustness (Zhang et al., 2017; Zhang, 2020; Yun et al., 2019) and generalization. Finally, domain generalization methods (e.g., (Wang et al., 2019, ?; Huang et al., 2020)) have been proposed to increase a model's robustness to domain differences in the data.

Image Filtering Low-pass filters like box (Rosenberg, 1974) and Gaussian (Gonzales & Woods, 2002) are classic *content agnostic* smoothing filters; i.e., their filter weights are fixed regardless of spatial location and image content. Bilateral (Paris et al., 2009) and guided (He et al., 2010) filters are *content aware* as they can simultaneously preserve edge information while removing noise. Recent works integrate such classic filters into deep networks (Zhang, 2020; Xie et al., 2019). 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). (Su et al., 2019; Jia et al., 2016) introduced the dynamic filtering layer, whose weights are predicted by convolution layers conditioned on pre-computed feature maps. Our method differs 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, (Wang et al., 2019) introduces an adaptive convolution layer for upsampling, whereas we focus on downsampling with an adaptive low-pass filtering layer.

Applications The application of anti-aliasing covers a variety of visual recognition tasks, ranging from classification (Deng et al., 2009), dense prediction (He et al., 2017; Chen et al., 2017), video analysis (Yang et al., 2019) to generation tasks (Wang et al., 2018). We find that anti-aliasing techniques are especially effective for dense prediction tasks including instance segmentation (He et al., 2017; Zhou et al., 2019) and semantic segmentation (Long et al., 2015; Chen et al., 2018). These tasks 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, (Zhang, 2020), focused mainly on image classification. In addition, video insconsistency caused by motion blur, natural perturbations, etc. has also been widely observed (Shankar et al., 2019; Gu, 2021; Li et al., 2010). We specifically explore the consistency problem in video instance segmentation (Yang et al., 2019) to demonstrate the effectiveness of our approach. Finally, generative models also have sampling operations in their encoder and/or decoder architecture (Wang et al., 2018; Richardson et al., 2021; Park et al., 2019). Thus, we also investigate our approach in this area.

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

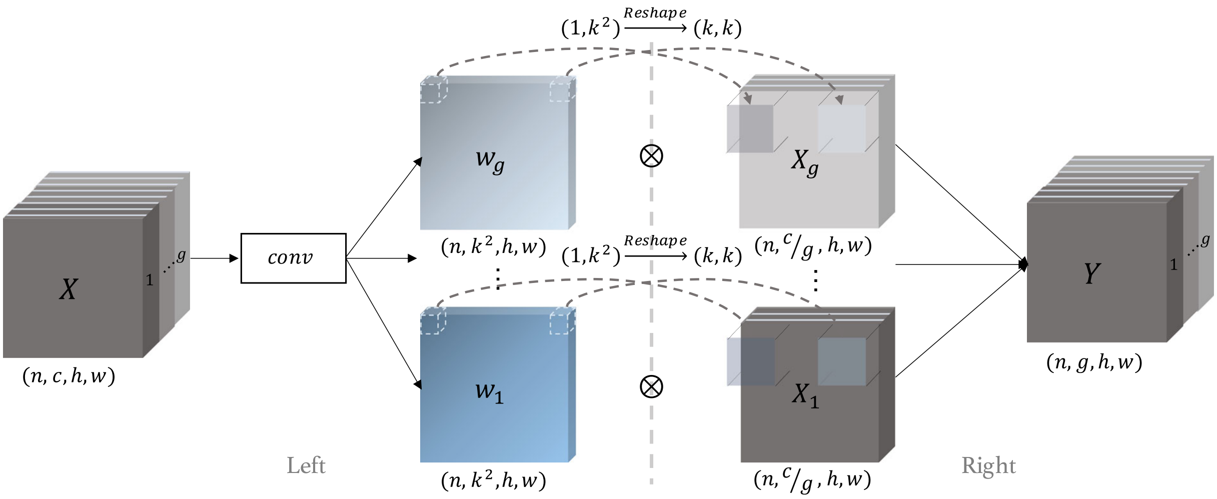

Fig. 2 Method overview. (Left) For each spatial location and feature channel group in the input X, we predict a $k \times k$ filter w. (Right) We apply the learned filters on X to obtain content aware anti-aliased features. See text for more details

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).

3.1 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 down-sampled, 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:

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

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.

3.2 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 (Wu & He, 2018), 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:

$$Y_{i,j}^{g} = \sum_{p,q \in \Omega} w_{i,j,g}^{p,q} \cdot X_{i+p,j+q}^{c}, \tag{4}$$

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.

3.3 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.

3.4 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

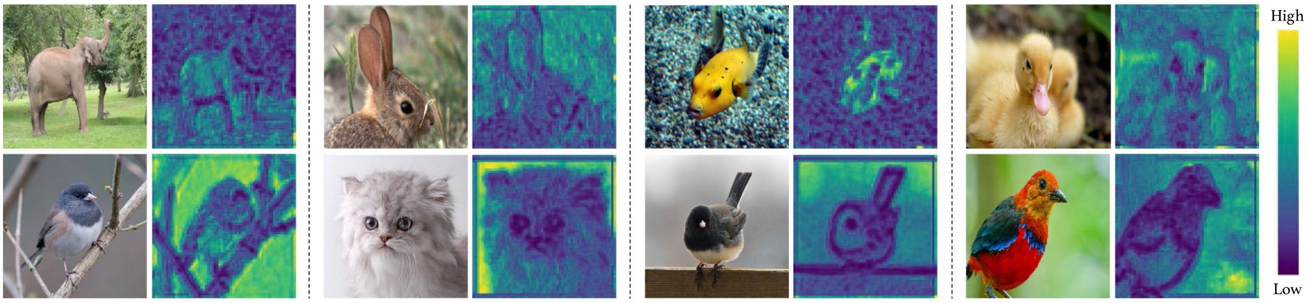

Fig. 3 Variance of the learned filter weights across spatial locations. Low variance corresponds to more blur, while high variance corresponds to less blur. Our model correctly learns to blur high frequency content (e.g., edges) more to prevent aliasing, and blur low frequency content less to preserve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

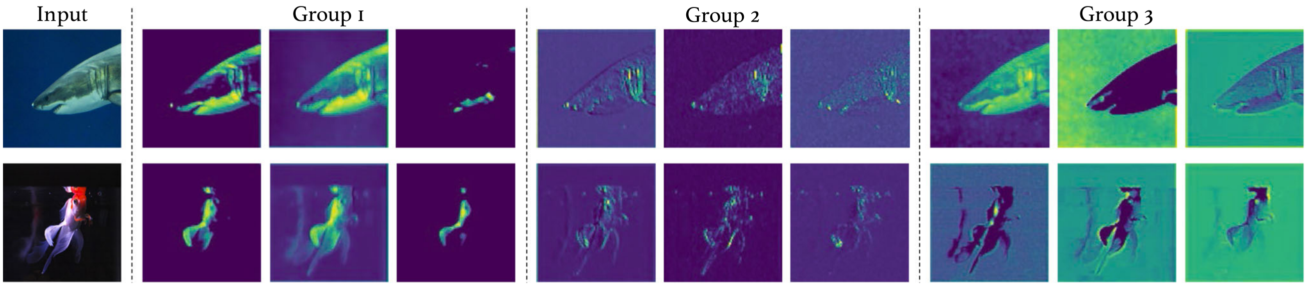

Fig. 4 Visualization of predicted feature maps within and across groups. The features within each group are more similar to each other than to those in other groups. Each group captures a different aspect of the image (e.g., edges, color blobs) (Color figure online)

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. Figure 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.

3.5 Relation with Self-attention

Recently, the transformer architecture (Vaswani et al., 2017) has emerged as a state-of-the-art alternative to convolutional networks on various vision tasks including classification (Dosovitskiy et al., 2021), detection (Carion et al., 2020), and segmentation (Xie et al., 2021). To deal with the transformer's quadratic complexity to input length, more efficient architectures such as the SwinTransformer (Liu et al., 2021) and LongFormer (Beltagy et al., 2020) have been proposed. Their key idea is to apply both sparse global and local attention. In this section, we show that our proposed anti-aliasing module can be interpreted as a form of sliding window local attention.

Given feature map X with dimension $h \times w \times d$, a sliding window local attention will apply local self-attention within each $k \times k$ feature patch window. It can be represented by the following equation:

$$Attention(x_c) = softmax(\frac{\phi_q(x_c)\phi_k(x)^T}{\sqrt{d}})\phi_v(x) \tag{5}$$

where x is the feature patch with size $k \times k \times d$, x_c is the center point of the feature patch x, and ϕ represents linear projection. The self-attention layer will first compute the cross similarity between x_c to each feature point in x (k^2 total), apply a softmax to normalize the similarity values to sum to one, and finally, use the resulting weights to compute a weighted sum over the projected values ($\phi_v(x)$) of the k^2 points.

In the above equation, we can consider replacing the linear projections ($\phi_q(\cdot)$ and $\phi_k(\cdot)$) and the dot product between

Table 1 Image classification accuracy, consistency on ImageNet (Deng et al., 2009), and domain generalization results ImageNet → ImageNet VID (Deng et al., 2009). We compare to strong ResNet-101 (He et al., 2016) and LPF (low-pass filter) (Zhang, 2020) baselines

		Accuracy			Consistency		Generalization	
Methods	Filter size	Top-1 Abs	Top-5 Abs	Delta	Abs	Delta	Abs	Delta
ResNet-101 (He et al., 2016)	–	77.7	93.8	–	90.6	–	67.6	–
LPF (Zhang, 2020)	3 × 3	78.4	94.1	+0.7	91.6	+1.0	68.8	+1.2
	5 × 5	77.7	93.9	+0.0	91.8	+1.2	67.0	−0.6
Ours	3 × 3	**79.0**	**94.4**	**+ 1.3**	91.8	+1.2	**69.9**	**+2.3**
	5 × 5	78.6	94.3	+0.9	**92.2**	**+1.6**	69.1	+1.5

Our method shows consistent improvement in accuracy, consistency, and generalization
Bold values indicate statistical significance

them, with a conv layer to compute the summing weights:

$$Attention(x_c) = softmax(conv(x))\phi_i(x) \tag{6}$$

where ϕ_i is identity projection. This equation exactly represents our proposed anti-aliasing module.

In both cases (Eqs. 5 and 6), the output is a weighted sum of its input value tensor, and demonstrates that our approach can be viewed as a form of sliding window self-attention.

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 (He et al., 2016), Deeplab v3+ (Chen et al., 2018), Mask R-CNN on large scale datasets including ImageNet, ImageNet VID (Deng et al., 2009), MS COCO (Lin et al., 2014), PASCAL VOC (Everingham et al., 2015) and Cityscapes (Cordts et al., 2016). 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 (Deng et al., 2009), 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 (Zhang, 2020) 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 Zhang (2020).

Consistency metric We use the consistency metric defined in Zhang (2020), which measures how often the model outputs the same top-1 class given two different shifts on the same test image:

$$Consist = \mathbb{E}_{X,h_1,w_1,h_2,w_2}\ \mathbb{I}\{F(X_{h_1,w_1}) = F(X_{h_2,w_2})\} \tag{7}$$

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 (Zhang, 2020), 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.

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

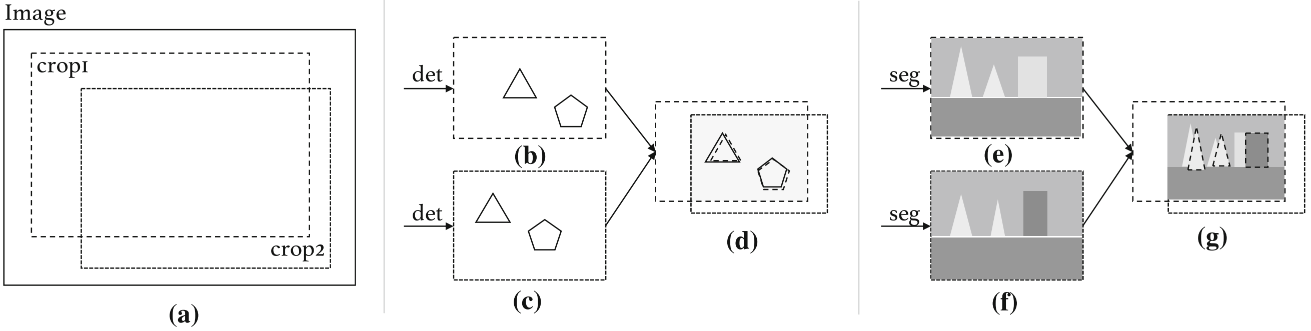

Fig. 5 Our new consistency metrics. (**b**,**c**,**d**): mean Average Instance Segmentation Consistency (mAISC). (**e**,**f**,**g**): mean Average Semantic Segmentation Consistency (mASSC). Both metrics first crop two patches from the input image (**a**) and then perform detection/segmentation (det/seg) on its content (**b**,**c**,**e**,**f**). Then, the overlapping part from the two patches are selected out (**d**,**g**) for evaluating the consistency score

Table 2 Instance segmentation results on MS COCO. We compare to Mask R-CNN (He et al., 2017) **and LPF** (Zhang, 2020)

	Mask				Box			
method	mAP	Delta	mAISC	Delta	mAP	Delta	mAISC	Delta
Mask R-CNN (He et al., 2017)	36.1	–	62.9	–	40.1	–	65.1	–
LPF (Zhang, 2020)	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**

Our approach consistently improves over the baselines on both mask and box detection accuracy. Our model performs especially well on shift consistency, with a 5.1 and 4.7 point improvement over Mask R-CNN on mAISC mask and box, respectively

Bold values indicate statistical significance

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 (Shankar et al., 2019).

4.3 Instance Segmentation

Experimental settings In this section, we present results on MS-COCO for instance segmentation (Lin et al., 2014). MS-COCO contains 330k images, 1.5M object instances and 80 categories. We use Mask R-CNN (He et al., 2017) as our base architecture. We adopt the hyperparameter settings from the implementation of Massa and Girshick (2018). 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.

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 demon-

Table 3 Semantic segmentation on PASCAL VOC 2012 (Everingham et al., 2015) and Cityscapes (Cordts et al., 2016)

	PASCAL VOC				Cityscapes			
method	mIOU	Delta	mASSC	Delta	mIOU	Delta	mASSC	Delta
Deeplab v3+ (Chen et al., 2018)	78.5	–	95.5±0.11	–	78.5	–	96.0±0.10	–
LPF (Zhang, 2020)	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**

We compare to Deeplab v3+ (Chen et al., 2018) and LPF (Zhang, 2020). Our approach leads to a large improvement in accuracy on PASCAL VOC and Cityscapes (1.8 point and 1.0 point, respectively). Under the mASSC consistency metric, our approach also shows improvement upon the two baselines. The results are averaged over three runs

Bold values indicate statistical significance

strates 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 (Everingham et al., 2015) and Cityscapes (Cordts et al., 2016) semantic segmentation with Deeplab v3+ (Chen et al., 2018) as the base model. We extend implementations from Hu et al. (2020, ?) and (aaa, 2020). 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 Chen et al. (2017).

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:

$$Consist(X, Y) = \mathbb{E}_{i \in [0,h)} \mathbb{E}_{j \in [0,w)} \ \mathbb{I}[S(X)_{i,j} = S(Y)_{i,j}] \quad (8)$$

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

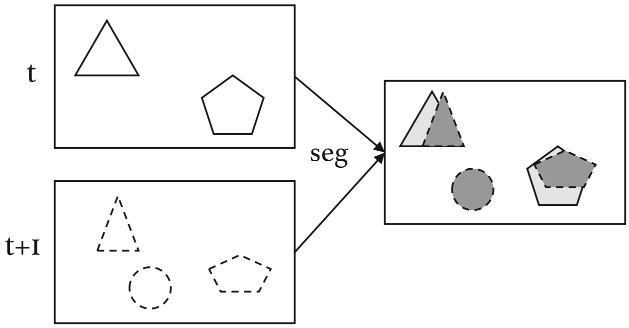

Fig. 6 Video instance segmentation consistency metric. For any two consecutive frames, if the object is detected in both frames, we record it as a positive pair

4.5 Video Consistency

Experimental settings We next validate our method's generalization to video data and its robustness to natural perturbations in video. For this, we perform the video instance segmentation task on the YoutubeVIS dataset (Yang et al., 2019) using the model trained in Sect. 4.3. We only evaluate on the 20 overlapping classes between COCO and YoutubeVIS. Since the validation set of YoutubeVIS does not have ground-truth annotation for all frames, we randomly select 260 videos in the training set to validate video consistency.

Consistency metric (mAVISC) To measure an instance segmentation model's robustness to natural perturbations in video, we propose a new mean Average Video Instance Segmentation Consistency (mAVISC) metric. For each video sequence, for all pairs of consecutive frames, and for each object that appears in each pair of frames, we first determine whether the object is detected according to a predetermined IOU threshold. If so, we record it as a positive pair, as shown

Table 4 Video instance segmentation consistency on YoutubeVIS (Yang et al., 2019). We evaluate video instance segmentation consistency for IOU thresholds (α) ranging from 0.5 to 0.8

	mAVISC							
Method	$\alpha = 0.5$	Delta	$\alpha = 0.6$	Delta	$\alpha = 0.7$	Delta	$\alpha = 0.8$	Delta
Mask R-CNN (He et al., 2017)	90.09	–	89.29	–	88.14	–	87.29	–
LPF (Zhang, 2020)	90.11	+ 0.02	88.96	−0.33	88.38	+0.24	87.32	+0.03
Ours	**90.71**	**+ 0.62**	**89.61**	**+ 0.32**	**88.79**	**+ 0.65**	**87.68**	**+ 0.39**

Our approach consistently increases video consistency with a good margin (+0.62/ + 0.32/ + 0.65/ + 0.39) for all IOU thresholds, whereas LPF increases it with a relatively smaller margin (+0.02/ + 0.32/ + 0.03) or can even decrease video consistency (−0.33 when $\alpha = 0.6$)
Bold values indicate statistical significance

Table 5 Image-to-Image translation results.

datasets	Cityscapes						Facades			
methods	FID ↓	Delta	mIoU	Delta	mAcc	Delta	PSNR	Delta	SSIM	Delta
pix2pixHD (Wang et al., 2018) / pix2pix (Isola et al., 2017)	52.21	–	71.23	–	78.97	–	60.33	–	1.08	–
LPF (Zhang, 2020)	52.68	+0.47	67.61	−3.62	75.61	−3.36	61.14	+0.81	1.37	+0.29
Ours (Zou et al., 2020)	**50.21**	**−2.0**	**71.99**	**+0.67**	**80.23**	**1.26**	**61.50**	**+1.17**	**1.41**	**+0.33**

On the Cityscapes dataset, the generated images of LPF have worse performance on both image quality and semantic segmentation, while the images generated by our approach tend to be more realistic (FID) and semantically accurate (mIoU, mAcc). On the Facades dataset, for shifted image pairs, our approach generates more consistent images compared to the baseline approaches for both pixel (PSNR) and patch (SSIM) metrics
Bold values indicate statistical significance

in Fig. 6. Below is the equation for computing mAVISC:

$$\frac{1}{NM_iQ_i}\sum_{i=1}^{N}\sum_{j=1}^{M_i}\sum_{t=1}^{Q_i}\mathbb{I}\{\mathbb{I}\{IOU(GT_{i,j,t}, P_{i,j,t}) > \alpha\} = \mathbb{I}\{IOU(GT_{i,j,t+1}, P_{i,j,t+1}) > \alpha\}\} \tag{9}$$

where N, M_i, Q_i is the number of video sequences, objects in the i'th video, and frames in the i'th video, respectively. GT represents the ground truth video object bounding boxes, P represents the bounding box predictions, and α is the IOU threshold to determine whether the ground truth object is detected.

Results Table 4 shows video consistency results on the YoutubeVIS dataset for Mask R-CNN, Mask R-CNN with LPF, and Mask R-CNN with our approach using the proposed mAVISC metric. We evaluate on IOU thresholds ranging from $\alpha = 0.5$ to $\alpha = 0.8$. (We do not include $\alpha = 0.9$ because at this very strict threshold, there are too few correct detections for any method, making difficult to make reliable conclusions.) As shown in Table 4, our approach consistently increases video consistency with a good margin (+0.62/+0.32/+0.65/+0.39) across all IOU thresholds, where LPF increases with fairly small margin (+0.02/ + 0.32/ + 0.03) or even decreases video consistency (−0.33 when $\alpha = 0.6$).

4.6 Image-to-Image Translation

Experiment Settings We evaluate image-to-image translation on the Cityscapes (Cordts et al., 2016) and Facades (Tyleček & Šára, 2013) datasets using Pix2PixHD (Wang et al., 2018) and Pix2Pix (Isola et al., 2017) as the baseline models, respectively. On Cityscapes, following (Wang et al., 2018), we use 2976 images for training and 500 images for evaluation. On Facades, we use a total of 400 images for training and evaluation following (Isola et al., 2017). For both Pix2Pix (Isola et al., 2017) and Pix2PixHD (Wang et al., 2018), we insert our module before each downsampling layer and upsampling layer following (Zhang, 2020). For downsampling, we simply insert our adaptive module with $stride = 2$. For upsampling, we first use nearest neighbor interpolation to upsample the feature map and then apply our adaptive filtering layer with $stride = 1$. We follow all the training settings from Wang et al. (2018); Isola et al. (2017).

On the Cityscapes dataset, we focus on image generation quality as well as our model's generalization capability to the segmentation task. We use mIoU, mAcc, and FID to evaluate the generated image quality. For mIoU and mACC, we first run the DeepLab V3+ semantic segmentation model (trained in Sect. 4.4) on the generated images, following (Wang et al., 2018). We compare the resulting segmentation maps with the ground truth segmentation maps. For FID, we use the publicly available codebase at https://github.com/mseitzer/pytorch-fid to compare the distributions of the generated image features and the real image features. On the Facades dataset, we

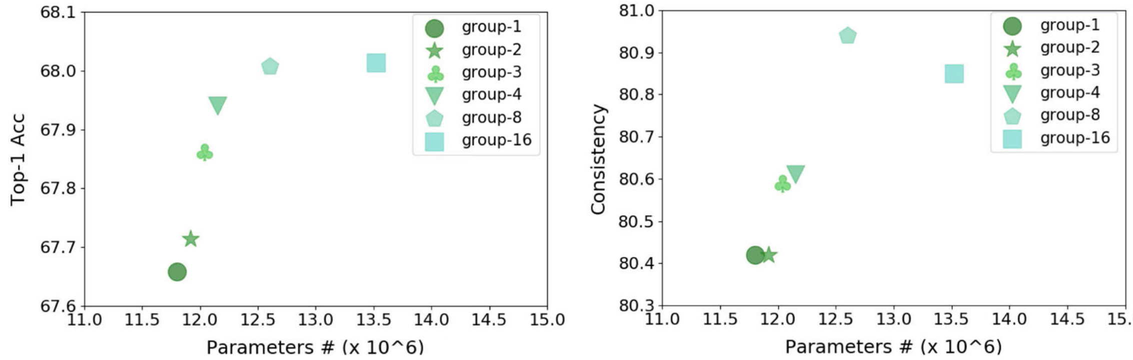

Fig. 7 Effect of number of groups on top-1 accuracy and consistency. As the group number increases, both Top-1 accuracy and consistency first increase then decrease. The performance saturates with group number 8

Table 6 Filter ablations. Gaussian blur is better than no blur (ResNet)

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**

Learning the blur filter globally (Image Ada.), spatially (Spatial Ada.), and over channels (Ours) progressively does better
Bold values indicate statistical significance

follow (Zhang, 2020; Karras et al., 2021) to evaluate the shift consistency of the image generation model. To evaluate the similarity of two shifted images, we compute both PSNR and SSIM to evaluate both pixel-wise and patch-wise similarity. Results

In Table 5, we first compare pix2pixHD (Wang et al., 2018), pix2pixHD together with LPF (Zhang, 2020), and pix2pixHD with our approach on the Cityscapes dataset. Overall, our approach generates more realistic images (e.g., FID score decreases by 2 points) and has better mIOU and mAcc scores than both pix2pixHD and LPF. In addition, we compare pix2pix (Isola et al., 2017), pix2pix together with LPF (Zhang, 2020), and pix2pix with our approach on the Facades dataset. The results show that our model is more consistent on image shift compared to the baseline approaches.

4.7 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 Sect. 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. 7, 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 12.90 M, 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 12.60 M *vs* 12.90 M parameters). This shows that our adaptive anti-aliasing method does not gain performance by simply scaling up its capacity.

Type of filter. In Table 6, 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 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, 4.5 M to 4.63 M). As for runtime, on a RTX2070 GPU, our method (R-101 backbone) takes 6.4 ms to forward a 224 × 224 image whereas a standard ResNet-101 takes 4.3 ms.

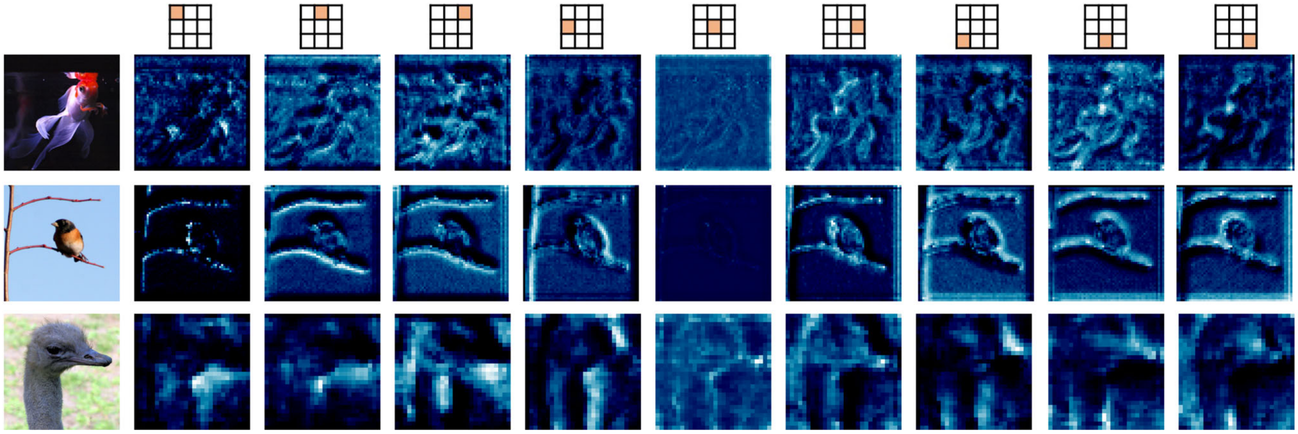

Fig. 8 Visualization of learned filter weights at each spatial location. We can see that the learned filter weight is adaptive to different visual content. Specifically, our model tends to "grow" edges so that it is easier for them to be preserved. For example, the learned filter tends to integrate more information from left to right (see center-left and bottom-left weights in the second row of this figure) on the vertical tree branch and thus grow it to be thicker. This way, it is easier for the tree branch contours to be preserved after downsampling

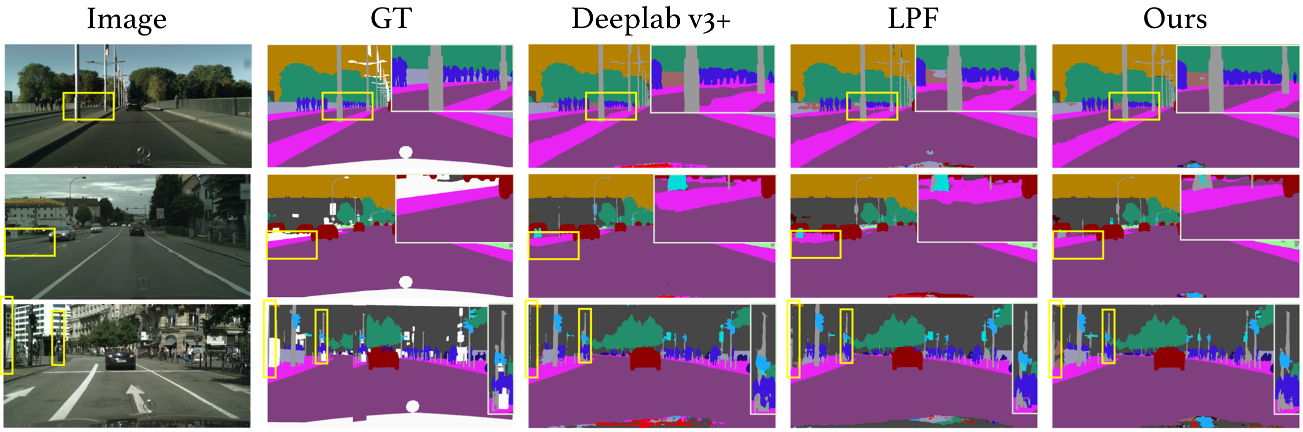

Fig. 9 Qualitative results for semantic segmentation on Cityscapes. In the first row, within the yellow box region, our method clearly distinguishes the road edge compared to Deeplab v3+ and LPF. Similar behavior (better segmented road contours) is also observed in the second row. This holds for other objects as well - the light pole has better delineation compared to both baselines in the third row (Color figure online)

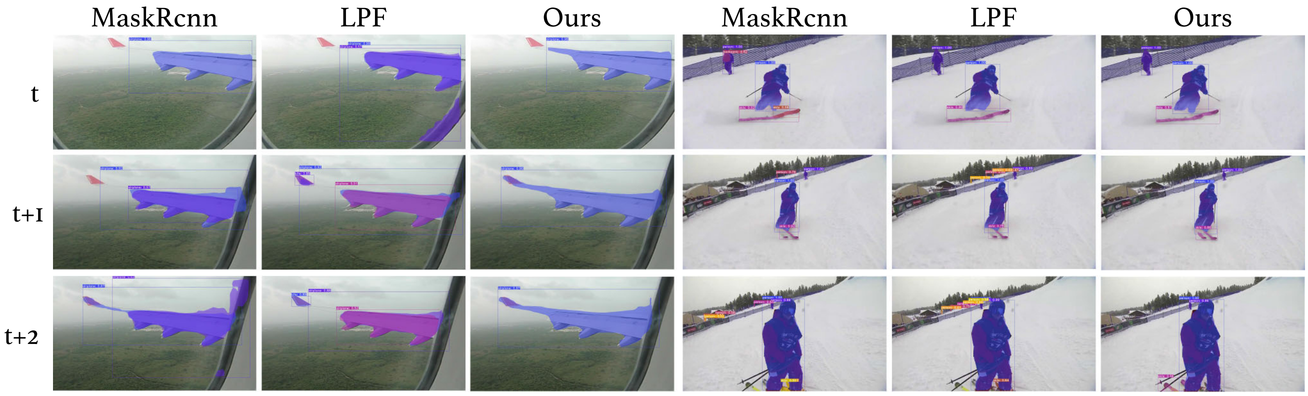

Fig. 10 Qualitative results of video instance segmentation consistency. As shown in the first three columns, our method produces more consistent instance segmentation of the airplane wing whereas Mask-RCNN and LPF produce more inconsistent results that fluctuate over frames. In the last three columns, we observe the existence of redundant detections for both Mask-RCNN and LPF

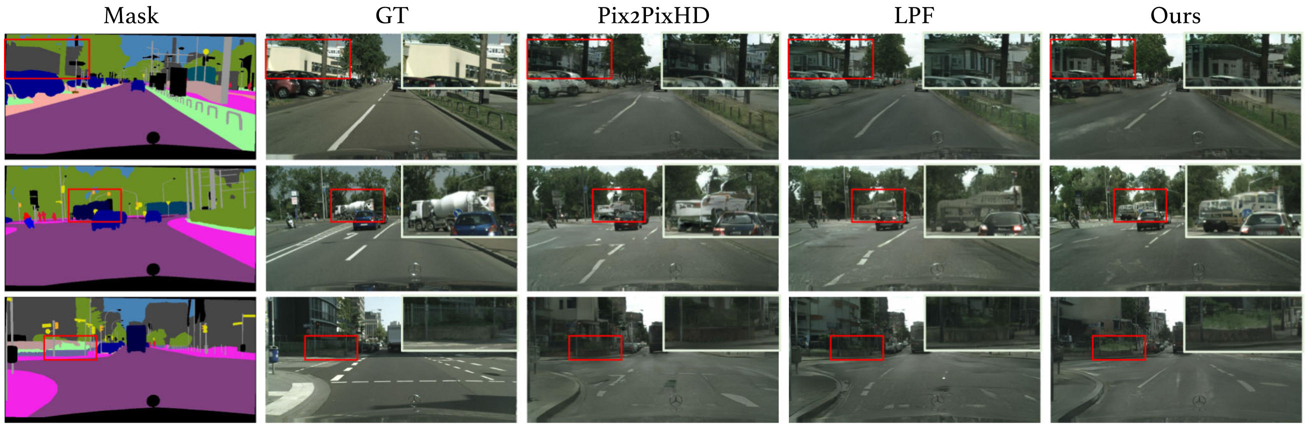

Fig. 11 Qualitative results for image to image translation on Cityscapes. In the first row, our approach generates clear boundaries along the roof. In contrast, the other methods produce blurry boundaries. In the second row, our approach not only produces a clear edge on the car, it also generates a very tiny traffic light (see region inside the red rectangle). The other two methods fail in this situation. In the last row, our approach clearly identifies the boundary between the wall and bushes whereas the other two approaches' produce very blurry and dark generations (Color figure online)

Type of Backbone We compare Top-1 accuracy and Consistency with two additional backbone networks, VGG (Simonyan & Zisserman, 2015) and DenseNet-121 (Huang et al., 2017), on the Cifar-10 dataset (Krizhevsky et al., 2009). For VGG, our approach achieves 94.0 Top-1 accuracy and 97.2 Consistency, and for DenseNet, our approach achieves 95.6 Top-1 accuracy and 97.4 Consistency. Similar to the ResNet101 results, our approach improves Top-1 accuracy with a good margin compared to the baseline network, which does not have any anti-aliasing (+0.6 for VGG and +1.7 for DenseNet) as well as LPF (+0.4 for VGG and +1.1 for DenseNet). Our method's consistency is also improved upon the baseline network (+0.6 for VGG and +0.1 for DenseNet) although it does not outperform the LPF method (−0.4 for VGG and −0.9 for DenseNet). As Cifar-10 has relatively low resolution (32^2 pixels) images in comparison with ImageNet (224^2 pixels), there can be a trade-off between accuracy and consistency. Specifically, we find that decreasing the content frequency for anti-aliasing to improve shift consistency may have a side effect on classification accuracy when the image resolution is already very low. Thus, the consistency performance may not be improved as much in comparison with higher resolution images such as those in ImageNet, as we had shown in Table 1.

4.8 Qualitative Results

4.8.1 Semantic Segmentation

We show qualitative results for semantic segmentation in Fig. 9 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.

4.8.2 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 in the second row) 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.

4.8.3 Video Instance Segmentation Consistency

In addition to image results, we also show qualitative results on a video dataset. In Sect. 4.5, we quantitatively demonstrated that our method provides additional robustness to natural perturbations. Here we show qualitative results to illustrate its effectiveness. In Fig. 10, each row represents a different time stamp. In the left airplane example, we can observe that while all three methods can detect the airplane's wing, the detections of Mask R-CNN (He et al., 2017) and LPF (Zhang, 2020) fluctuate over time (e.g. multiple detections on the airplane's wing) whereas our detections are quite stable. In the right skiing example, both Mask R-CNN and LPF generate lots of redundant detections compared to our approach that is likely caused by the aliasing effects of downsampling.

4.8.4 Image-to-Image Translation

Finally, we show qualitative results of applying our approach to generative models. In Fig. 11, we compare with pix2pixHD (Wang et al., 2018) and pix2pixHD together with LPF (Zhang, 2020) on image-to-image translation using the Cityscapes dataset. We find that our adaptive filters are better at preserving boundaries in image generation. In the first row, our approach generates clear boundaries along the roof. In contrast, the other methods produce blurry boundaries. In the second row, our approach not only produces a clear edge on the car, it also generates a very tiny traffic light (see region inside the red rectangle). The other two methods fail in this situation. In the last row, our approach clearly identifies the boundary between the wall and bushes whereas the other two approaches' produce very blurry and dark generations. We attribute this property to the fact that with LPF or the original conv filters, the filter weights are fixed at all spatial locations. This means that it will be difficult for neighbouring in the higher resolution output to have different values within a small local region. And this could potentially cause the unclear boundary effect shown in Fig. 11.

5 Limitations

We have shown in this paper that our approach is effective for various discriminative and generative tasks. However, it also has some limitations. First, although both the computation and parameter overhead is marginal, with our current implementation, GPU memory overhead is not negligible as it involves the unfold function in PyTorch which is memory intensive. Second, we empirically found the optimal group number of filter weights to be 8 for our tasks. However, it may not be optimal for other tasks and thus is a hyperparameter that needs to be tuned.

6 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. Despite some of the limitations observed in Sect. 5, we believe our work can be a promising foundation for exploring anti-aliasing on other tasks (e.g., video recognition) as well as other forms of input noise.

Acknowledgements This work was supported in part by ARO YIP W911NF17-1-0410, NSF CAREER IIS-2150012, NSF IIS-2204808, NSF CCF-1934568, GCP research credit program, and AWS ML research award.

References

Azulay, A. & Weiss, Y. (2018). Why do deep convolutional networks generalize so poorly to small image transformations? In *JMLR*.

Beltagy, I., Peters, M.E. & Cohan, A. (2020). Longformer: The long-document transformer. arXiv:2004.05150.

Bietti, A. & Mairal, J.(2017). Invariance and stability of deep convolutional representations. In *NeurIPS*.

Bloem-Reddy, B. & Teh, Y. W. (2020). Probabilistic symmetries and invariant neural networks. In *JMLR*.

Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). YOLACT: real-time instance segmentation. In *ICCV*.

Caelli, T. M. & Liu, Z. Q. (1988). On the minimum number of templates required for shift, rotation and size invariant pattern recognition. In *Pattern recognition*.

Carion, N., Massa, F., Synnaeve, G., Usunier, N. Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *ECCV*.

Chaman, A. & Dokmanic, I. (2021). Truly shift-invariant convolutional neural networks. In *CVPR*.

Chen, L. C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. In *CVPR*.

Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016) The cityscapes dataset for semantic urban scene understanding. In *CVPR*.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *CVPR*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., & Uszkoreit, J. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. In *IJCV*.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., & Brendel, W. (2019). Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*.

Gonzales, R. C. & Woods, R. E. (2002). *Digital image processing*. Prentice Hall.

Gu, Z. (2021). Spatiotemporal inconsistency learning for deepfake video detection. In arXiv:2109.01860.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *ICCV*.

He, K., Sun, J., & Tang, X. (2010). Guided image filtering. In *ECCV*.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*.

Hu, P., Heilbron, F. C., Wang, O., Lin, Z., Sclaroff, S., & Perazzi, F. (2020). Temporally distributed networks for fast video semantic segmentation. In *CVPR*.

Hu, P., Perazzi, F., Heilbron, F. C., Wang, O., Lin, Z., Saenko, K., & Sclaroff, S. (2020). Real-time semantic segmentation with fast attention. In *ECCV workshop*.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *CVPR*.

Huang, Z., Wang, H., Xing, E. P., & Huang, D. (2020). Self-challenging improves cross-domain generalization. In *ECCV*.

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *CVPR*.

Jia, X., De Brabandere, B., Tuytelaars, T., & Gool, L. V. (2016). Dynamic filter networks. In *NeurIPS*.

Kannan, H., Kurakin, A., & Goodfellow, I. (2018). Adversarial logit pairing. arXiv:1803.06373.

Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., & Aila, T. (2021). Alias-free generative adversarial networks. arXiv:2106.12423.

Krizhevsky, A. & Hinton, G. (2009) Learning multiple layers of features from tiny images, Citeseer.

Kurakin, A., Goodfellow, I., & Bengio, S. (2017). Adversarial examples in the physical world. In *ICLR Workshop*.

Lee, J., Lee, Y., Kim, J., Kosiorek, A., Choi, S., & Teh, Y. W. (2019). Set transformer: A framework for attention-based permutation-invariant neural networks. In *ICML*.

Lee, K., Lee, H., Lee, K., & Shin, J. (2017). Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *ICLR*.

Lee, K., Lee, K., Lee, H., & Shin, J. (2018). A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*.

Li, D., Yang, Y., Song, Y. Z. & Hospedales, T. M. (2017). Deeper, broader and artier domain generalization. In *ICCV*.

Li, S., Ma, L., Zhang, F., & Ngan, K. N. (2010). Temporal inconsistency measure for video quality assessment. In *28th picture coding symposium*.

Li, Y. (1992). Reforming the theory of invariant moments for pattern recognition. In *Pattern recognition*.

Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., & Zhu, J. (2018). Defense against adversarial attacks using high-level representation guided denoiser. In *CVPR*.

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *ECCV*.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *CVPR*.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *ICLR*.

Mairal, J., Koniusz, P., Harchaoui, Z., & Schmid, C. (2014). Convolutional kernel networks. In *NeurIPS*.

Massa, F. & Girshick, R. (2018). maskrcnn-benchmark: fast, modular reference implementation of instance segmentation and object detection algorithms in PyTorch. https://github.com/facebookresearch/maskrcnn-benchmark. Accessed: [Oct.10 2019].

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. In *Nature*.

Muandet, K., Balduzzi, D., & Schölkopf, B. (2013). Domain generalization via invariant feature representation. In *ICML*.

Paris, S., Kornprobst, P., Tumblin, J., & Durand, F. (2009). Bilateral filtering: Theory and applications. *Foundations and Trends in Computer Graphics and Vision, 4*(1), 1–73.

Park, T., Liu, M. Y., Wang, T. C., & Zhu, J. Y. (2019). Semantic image synthesis with spatially-adaptive normalization. In *CVPR*.

Proakis, J. G. & Manolakis, D. G. (1992). Digital signal processing. In *MPC*.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*.

Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., & Cohen-Or, D. (2021). Encoding in style: a stylegan encoder for image-to-image translation. In *CVPR*.

Rosenberg, D. (1974). Box filter. US Patent 3,815,754.

Rowley, H. A., Baluja, S., & Kanade, T. (1998). Rotation invariant neural network-based face detection. In *CVPR*.

Shankar, V., Dave, A., Roelofs, R., Ramanan, D., Recht, B., & Schmidt, L. (2019). A systematic framework for natural perturbations from videos. arXiv:1906.02168.

Shannon, C. E. (1949). Communication in the presence of noise. In *Proceedings of the IRE*.

Simonyan, K. & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Su, H., Jampani, V., Sun, D., Gallo, O., Learned-Miller, E., & Kautz, J. (2019). Pixel-adaptive convolutional neural networks. In *CVPR*.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. arXiv:1312.6199

Tan, M. & Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*.

Tyleček, R. & Šára, R. (2013). Spatial pattern templates for recognition of objects with regular structure. In *German conference on pattern recognition*, pp. 364–374. Springer.

VainF: DeepLabv3Plus-Pytorch (2020). https://github.com/VainF/DeepLabV3Plus-Pytorch.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017). Attention is all you need. In *NeurIPS*.

Wang, H., Ge, S., Xing, E. P. & Lipton, Z. C. (2019). Learning robust global representations by penalizing local predictive power. In *NeurIPS*.

Wang, H., He, Z., Lipton, Z. C. & Xing, E. P. (2019). Learning robust representations by projecting superficial statistics out. In *ICLR*.

Wang, J., Chen, K., Xu, R., Liu, Z., Loy, C. C., & Lin, D. (2019). Carafe: Content-aware reassembly of features. In *ICCV*.

Webber, C. J. (1994) Self-organisation of transformation-invariant detectors for constituents of perceptual patterns. *Network: Computation in Neural Systems, 5*(4), 471–496. https://doi.org/10.1088/0954-898X_5_4_004

Wang, T. C., Liu, M. Y., Zhu, J. Y., Tao, A., Kautz, J., Catanzaro, B. (2018). High-resolution image synthesis and semantic manipulation with conditional gans. In *ICCV*.

Wood, J. (1996). Invariant pattern recognition: a review. In *Pattern recognition*.

Wu, Y. & He, K. (2018). Group normalization. In *ECCV*.

Xie, C., Wu, Y., Maaten, L. v. d., Yuille, A. L., & He, K. (2019). Feature denoising for improving adversarial robustness. In *CVPR*.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (2021). Segformer: Simple and efficient design for semantic segmentation with transformers. arXiv:2105.15203.

Yang, L., Fan, Y. & Xu, N. (2019). Video instance segmentation. In *ICCV*.

Ye, M., Zhang, X., Yuen, P. C. & Chang, S. F. (2019). Unsupervised embedding learning via invariant and spreading instance feature. In *CVPR*.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*.

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. In *ICLR*.

Zhang, R. (2020). Making convolutional networks shift-invariant again. In *ICML*.

Zhang, Z., Hua, B. S., Rosen, D. W., & Yeung, S. K. (2019). Rotation invariant convolutions for 3d point clouds deep learning. In *3DV*.

Zou, X., Xiao, F., Yu, Z., & Lee, Y. J. (2020). Delving deeper into anti-aliasing in convnets. In *BMVC*.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.