Dilated Convolutions with Lateral Inhibitions for Semantic Image Segmentation

Yujiang Wang\textsuperscript{1}  \hspace{1cm} Mingzhi Dong\textsuperscript{2}  \hspace{1cm} Jie Shen\textsuperscript{1,\dagger}  \hspace{1cm} Yiming Lin\textsuperscript{1}  \hspace{1cm} Maja Pantic\textsuperscript{1,3}
\textsuperscript{1}Imperial College London  \hspace{1cm} \textsuperscript{2}University College London  \hspace{1cm} \textsuperscript{3}Facebook AI

yujiang.wang14@imperial.ac.uk  \hspace{0.5cm} mingzhi.dong.13@ucl.ac.uk
\{jie.shen07, yiming.lin15, m.pantic\}@imperial.ac.uk

\textbf{Abstract}

Dilated convolutions are widely used in deep semantic segmentation models as they can enlarge the filters’ receptive field without adding additional weights nor sacrificing spatial resolution. However, as dilated convolutional filters do not possess positional knowledge about the pixels on semantically meaningful contours, they could lead to ambiguous predictions on object boundaries. In addition, although dilating the filter can expand its receptive field, the total number of sampled pixels remains unchanged, which usually comprises a small fraction of the receptive field’s total area. Inspired by the Lateral Inhibition (LI) mechanisms in human visual systems, we propose the dilated convolution with lateral inhibitions (LI-Convs) to overcome these limitations. Introducing LI mechanisms improves the convolutional filter’s sensitivity to semantic object boundaries. Moreover, since LI-Convs also implicitly take the pixels from the laterally inhibited zones into consideration, they can also extract features at a denser scale. By integrating LI-Convs into the Deeplabv3+ architecture, we propose the Lateral Inhibited Atrous Spatial Pyramid Pooling (LI-ASPP) and the Lateral Inhibited MobileNet-V2 (LI-MNV2). Experimental results on three benchmark datasets (PASCAL VOC 2012, CelebAMask-HQ and ADE20K) show that our LI-based segmentation models outperform the baseline on all of them, thus verify the effectiveness and generality of the proposed LI-Convs.

\textbf{1 Introduction}

Since the introduction of the pioneering Fully Convolutional Networks (FCN) [27], deep Convolutional Neural Networks (CNNs) [9, 55, 26, 54, 20] have made impressive progress in semantic image segmentation, a task that performs per-pixel classifications. In deep CNN models, a series of convolutions and spatial poolings are applied to obtain progressively more abstract and more representative feature descriptors with decreasing resolutions. As a consequence, the deepest features can have significantly lower resolution than the original image (e.g. only 1/16 or 1/32 of the input size in FCN [27]), hence it would be difficult to decode these features into the segmentation map at the same size of the input image without losing details. This is a crucial challenge in the semantic segmentation task.

Dilated convolutions [19], which are first applied to the semantic segmentation task by [52, 6], can effectively overcome such difficulties and thus are widely employed in state-of-the-art segmentation methods [26, 9, 51, 8, 46]. By inserting zeros (dilation) into the convolutional filters, dilated convolutions can observe features from larger areas without increasing the kernel parameters, which is important to the extractions of global semantic features. Besides, it can also produce feature maps

\dagger Corresponding author.

Preprint. Under review.
that are invariant input resolutions. In practice, dilated convolutions can be utilised to retain the resolution of the feature maps when encoding representations in the backbone network [53, 46], typically by replacing certain convolutional layers with dilated ones. It can also be employed during the decoding stage to generate more robust semantic labels, e.g. the Atrous Spatial Pyramid Pooling (ASPP) [8, 7] adopts three parallel dilated convolutions with different dilation rates to aggregate the multi-scale contextual information.

Despite its broad applications, dilated convolutions still have several limitations. The pixels around semantically meaningful contours separate different objects and possess stronger semantic information. In dilated convolution, however, the importance of those pixels are not explicitly accentuated, and therefore such positional significance has to be implicitly learnt. This can lead to ambiguous and misleading boundary labels. Various approaches have been proposed to compensate for such problems and to refine the contour predictions, including the Conditional Random Fields (CRF) [7, 4] and the decoder component in Deeplabv3+ [9]. However, dilated convolution’s sensitivity on spotting semantically meaningful edges still leaves room for improvement.

Additionally, although the receptive field of dilated filters is enlarged, the total number of sampled pixels stay the same, which only consist of a small fraction of pixels in the area. The sparse sampling can somehow impair the potentials for dense prediction tasks like semantic segmentation. Similar concerns were address in [41, 46, 11, 50], and the proposed improvements include a denser Gaussian sampling process [41], a hybrid dilated convolution module [46] and the deformable convolutional filters [11].

In this paper, we propose to overcome the drawbacks in dilated convolutions from a biologically-inspired perspective, which is to leverage the Lateral Inhibition (LI) mechanism in the human visual system. Lateral inhibition [17, 56, 45] is a neurobiological phenomenon that a neuron’s excitation to a stimulus can be suppressed by the activation of its surrounding neurons. Because of the LI mechanism, our retina cells are sensitive to the spatially varying stimulus such as the semantic borderlines between objects, which is crucial to the inborn segmentation abilities of our eyes. See Fig. 1 (Left) for an intuitive illustration of the LI mechanism.

Motivated by such observations, we propose a dilated convolution with lateral inhibitions (LI-Convs) to enhance the convolutional filter’s sensitivity to semantic contours. The LI-Convs also sample the receptive window in a denser fashion by implicitly making inferences on pixels within the lateral inhibited zones. To evaluate LI-Convs, we follow the Deeplabv3+ [9] segmentation models and present two LI-based variants which are 1. the Lateral Inhibited Atrous Spatial Pyramid Pooling (LI-ASPP) for decoding semantic features, and 2. the Lateral Inhibited MobileNet-V2 (LI-MNV2) as the backbone network for encoding features. The performance of LI-ASPP and LI-MNV2 surpass the baseline on three segmentation benchmark dataset: PASCAL VOC 2012 [12], CelebAMask-HQ [23] and ADE20K [56], which verifies the effectiveness and generality of the proposed LI-Convs.

2 Related Works

Semantic Image Segmentation

Fully Convolutional Networks (FCN) [27] is the pioneering work of using deep models for semantic segmentation. The fully connected layers in deep image classification models are replaced with convolutional ones to produce semantic heat maps for segmentation predictions. The resolution of such heat maps is typically much smaller than that of the input image (e.g. 1/32), and various works are proposed to compensate the information loss during decoding such features, including the de-convolutional layers [33, 37, 34], the skip-connections of low-level features [2, 16] and dilated convolutions [52, 8, 51, 26, 5]. Yu et al. [52] stacks dilated convolutional layers with different dilation rates in a cascaded manner, leading to a context module for aggregating the multi-scale contextual information. Deeplabv3 [8] builds an Atrous Spatial Pyramid Pooling (ASPP) module consisting of three parallel dilated convolutions, one 1*1 convolution and one image-level pooling, and it also employs dilated convolutions in the backbone network. DenseASPP [51] introduces dense connection into the ASPP module, while the technique of Neural Architecture Search [58] is utilised by [5] to search for an optimal decoding structure of organising dilated convolutions layers. For other segmentation practice [35, 47, 28, 48], readers are referred to [31] for more details.

Dilated Convolutions

Dilated convolutions, also known as atrous convolutions, is first introduced by Holschneider et al. [19] in signal analysis and have broad applications such as object detection [24, 52], lip-reading [49, 50] and optical flow [57, 44]. It is first applied to semantic segmentation...
by authors of [52] to enlarge filter’s receptive fields without sacrificing the spatial resolution. Conditional Random Fields (CRF) are involved in [4] as a post-processing procedure to refine the ambiguous semantic contour predictions. Similar ideas can be found in DeepLabv3+ [9], which designs a decoding module to incorporate low-level backbone features to improve the qualities of contouring pixels. Deformable convolutions [11] introduce the offsets into the sampling grids of filters to better model the spatial relationships. Gaussian kernels are adopted by [41] to obtain pixels at a wider range in dilated convolutions. Wang et al. [46] observe the gridding effects brought by the fixed sampling locations in dilated kernels and demonstrate a hybrid dilated convolution with different dilated rates. Different from those approaches, we employ the lateral inhibition (LI) mechanisms [17] to enhance the dilated convolutions’ sensitivity on semantically meaningful contours and to implicitly sample features in a denser fashion.

Lateral Inhibitions The study on the eyes of horseshoe crab (Limulus) performed by Hartline et al. [17] reveals the lateral inhibition (LI) effects in visual systems, where the excitation of neighbouring neurons can suppress a cell’s response to the stimuli. Although lateral inhibitions are mainly studied in the field of neuroscience [38, 43, 36], the computer vision community has also shown interests in this mechanism. The recurrent neural network with lateral inhibitions is studied in [29] and it is shown that LI can improve the robustness and efficiency of the network. Authors of [13] introduce LI into a shallow CNN to improve image classification. Similar ideas can be found in the work for colour video segmentation [44]. Those network architectures are somehow too shallow to be useful for recent methods using deep backbones like MobileNet-V2 (MNV2) [40] or ResNet [18]. Recently, authors of [3] employ LI in VGG model [42] to improve the performance on saliency detection. However, none of the previous works has evaluated LI’s potentials for semantic segmentation, while their methods of integrating LI do not touch the core mechanisms in deep CNNs such as the convolutional operations. In this work, however, lateral inhibitions work closely with the convolutional filters to fundamentally augment the model’s segmentation powers.

3 Dilated Convolutions with Lateral Inhibitions

3.1 Definition

Define $\Psi_k = \mathbb{Z}^2 \cap [-k, k]^2$ where $k \in \mathbb{Z}_{\geq 0}$, and let a discrete function $F: \Psi_k \mapsto \mathbb{R}$ represents a convolutional filter of size $(2k + 1)^2$. Define another discrete function $G: \mathbb{Z}^2 \mapsto \mathbb{R}$ representing features of arbitrary sizes. Let $d$ be the dilation rate, a dilated convolutional operator $\ast_d$ is written as

$$ (F \ast_d G)(p) = \sum_{dm+n=p} F(m)G(n) $$

(1)

where $p, m, n \in \mathbb{Z}^2$. Note that $\ast_d$ turns into a regular convolutional operator when $d = 1$, i.e. no dilation is inserted.

With the introduction of lateral inhibitions (LI), the activation of each sampled pixel, i.e. $G(n)$ in Eq. 1 would be suppressed by its neighbours within a certain range. Let the lateral inhibitions come from a square region of size $(2t + 1)^2$ centred on $n$ where $t \in \mathbb{Z}_{\geq 0}$, and refer this region as the lateral inhibition zone (the LI zone). Define $\Psi_t = \mathbb{Z}^2 \cap [-t, t]^2$ and let $L: \Psi_t, \mathbb{Z}^2 \mapsto \mathbb{R}$ be a discrete function describing the spatially-varying inhibition intensities in the LI zones, the amount of the total inhibitions received by a sampled pixel $G(n)$ can be described as $\sum_{u+v=n} L(u, n)G(v)$ where $u, v \in \mathbb{Z}^2$. Consequently, a dilated convolutional operator with lateral inhibition $\ast_d$ (LI-Convs) can be defined as

$$ (F \ast_d G)(p) = \sum_{dm+n=p} F(m)\phi(G(n)) - \sum_{u+v=n} L(u, n)G(v)). $$

(2)

where $\phi$ represents an activation function like ReLu. The introduced LI terms and non-linearity distinguish LI-Convs in Eq. 2 with Eq. 1. Fig. 3 (Middle, Right) provides an intuitive comparison between dilated convolution and the proposed LI-Convs.

We can also "dilate" the lateral inhibition zone to efficiently expand its field-of-views, in similar way to that of dilated convolutions. Denote the dilation rate in LI zones as $e$, a generalised LI-Convs operator $\ast_{d,e}$ is denoted as

$$ (F \ast_{d,e} G)(p) = \sum_{dm+n=p} F(m)\phi(G(n)) - \sum_{e(u+v=n)} L(u, n)G(v)). $$

(3)
Figure 1: **Left**: A toy example to illustrate the lateral inhibition mechanisms where the LI intensity is set to 0.25. The difference between the two neurons at the centre (representing a semantic contour) becomes more significant after LI. **Middle**: A $3 \times 3$ convolutional filter where $d = 4$. The sampled pixels (denoted as red dots) only comprises a small fraction of all pixels in the receptive field. **Right**: An illustration of the proposed LI-Convs with $3 \times 3$ lateral inhibition zones. Each sampled pixel receives inhibition signals from eight neighbours to enhance sensitivity on semantic contours and to extract information at a denser scale.

Although a wide variety of kernel forms can be taken by the LI intensity descriptor $L$, we opt for an intuitive formulation that is also easy to implement. In particular, $L(u, n)$ in Eq. 3 simply takes the production of a differentiable weight $W_L \in [0, 1] \cap \mathbb{R}$ and an exponentially decaying factor that is related to the distance between $u$ and $n$, which can be described as

$$L(u, n) = W_L \exp\left(-\frac{D^2(u, n)}{2\sigma^2}\right)$$

where $\sigma$ is a parameter representing the standard deviation and $D(u, n)$ refers to a certain distance measurement between $u$ and $n$. Here we employ the Euclidean distance.

### 3.2 Implementation of LI-Convs

We take a straight-forward approach to implement the LI-Convs in Eq. 5. We first design a Lateral Inhibition layer (the LI layer) to perform pixel-wise lateral inhibitions, while a dilated convolutional layer is subsequently applied to the inhibited features. The LI layer is essentially a light-weight module that can be flexibly inserted into deep models, while it can be easily implemented as a dilated convolutional layer with specifically shaped filters. In particular, let a discrete function $K : \Psi_t \mapsto \mathbb{R}$ represent one such LI filter, $K$ can be described as:

$$K(u) = \begin{cases} 
1.0 & u = 0, \\
-W_L \exp\left(-\frac{D^2(u, 0)}{2\sigma^2}\right) & u \neq 0.
\end{cases}$$

Note that the LI filter $K$ has identical size with the LI zones which is $(2t + 1)^2$, and applying $K$ with a stride of 1 can generate pixel-wise inhibited features. We empirically set $\sigma$ in Eq. 5 to a fixed value during training, thus there is only one weight $W_L$ to learn for each LI filter, which is significantly less than that of regular convolutional filters. In practice, we learn the lateral inhibition weights in a channel-wise manner, i.e. each LI filter learns a separate $W_L$. Therefore, a LI layer will introduce a total of $C$ learnable weights where $C$ is the channel number of the input tensor.

Figure 2: The structure of LI-Convs. The lateral inhibitions is first calculated by the LI layer, and the inhibited features are fed into the dilated convolution layer. The dilated convolution part can be any kind of convolution implementations such as the depthwise one [10].
3.3 LI-ASPP and LI-MNV2

We introduce the proposed LI-Convs into the state-of-the-art segmentation model Deeplabv3+ \cite{9} to evaluate the proposed LI-Convs. As shown in Fig. 3 (Left), we have replaced the three 3 × 3 parallel dilated convolution operations in Atrous Spatial Pyramid Pooling (ASPP) \cite{9} with the proposed LI-Convs, leading to the LI-ASPP model. Besides, we also investigate the potentials of LI layer in the backbone network such as the MobileNet-V2 (MNV2). In particular, we insert the LI layer into the residual bottleneck (RB) of MobileNet-V2 \cite{40}, between the 1 × 1 expansion convolution and 3 × 3 depthwise convolution, as illustrated in Fig. 3 (Right). We refer to this structure as the LI bottleneck layer. In the original MNV2 architecture, there is a total of 17 residual bottleneck layers, and we replace the 10th, 13th and 16th residual bottlenecks with the LI bottlenecks to obtain the LI-MNV2 network.
Table 1: The performance of different LI-Conv’s parameters on Pascal Voc 2012 validation set for LI-ASPP and LI-MNV2, respectively. "RB" refers to the Residual Bottleneck in MNV2 [40].

<table>
<thead>
<tr>
<th>Backbone</th>
<th>Decoding model</th>
<th>Positions of LI-Convs</th>
<th>LI Zone Sizes</th>
<th>LI Rates $e$</th>
<th>$W_L$ Init. Range</th>
<th>mIoU (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNV2</td>
<td>LI-ASPP</td>
<td>Three dilated convolution layers in ASPP with rates ${6, 12, 18}$</td>
<td>$3 \times 3$</td>
<td>$1$</td>
<td>$[0.0, 0.0]$</td>
<td>72.49</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$3 \times 3$</td>
<td>${1,3,5}$</td>
<td>$[0.0, 0.0]$</td>
<td>72.56</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$5 \times 5$</td>
<td>$1$</td>
<td>$[0.05, 0.15]$</td>
<td>72.43</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$3 \times 3$</td>
<td>$1$</td>
<td>$[0.05, 0.35]$</td>
<td>72.64</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$3 \times 3$</td>
<td>$1$</td>
<td>$[0.0, 0.15]$</td>
<td>72.93</td>
</tr>
<tr>
<td></td>
<td>LI-MNV2</td>
<td>${1 - 6}^{th}$ RB</td>
<td>$3 \times 3$</td>
<td>$1$</td>
<td>$[0.0, 0.0]$</td>
<td>72.07</td>
</tr>
<tr>
<td></td>
<td>ASPP</td>
<td>$16^{th}$ RB</td>
<td>$3 \times 3$</td>
<td>$1$</td>
<td>$[0.0, 0.0]$</td>
<td>72.21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>${3, 6, 10, 13, 16}^{th}$ RB</td>
<td>$3 \times 3$</td>
<td>$1$</td>
<td>$[0.0, 0.0]$</td>
<td>71.78</td>
</tr>
<tr>
<td></td>
<td></td>
<td>${13, 16}^{th}$ RB</td>
<td>$3 \times 3$</td>
<td>$1$</td>
<td>$[0.0, 0.0]$</td>
<td>72.43</td>
</tr>
<tr>
<td></td>
<td></td>
<td>${10, 13, 16}^{th}$ RB</td>
<td>$3 \times 3$</td>
<td>$1$</td>
<td>$[0.0, 0.0]$</td>
<td>72.5</td>
</tr>
</tbody>
</table>

4.2 Experimental Setup

**Evaluation metric**  Mean Intersection-over-Union (mIoU) is the most widely used evaluation metric for the segmentation task, and we adopt it to evaluate the quality of model predictions. We also report the model parameters and the FLOPs to provide more comprehensive analyses.

**Training Settings**  We generally follow the training settings in Deeplabv3+ [9], while we have also made some modifications to suit our needs. Particularly, we use the ImageNet [39] checkpoint provided by MobileNet-V2 authors [40] to initialise LI-MNV2, while the weights of LI-ASPP are randomly initialised. Note that we do not use the MS COCO dataset [25] to pre-train the model. During training, we set the image crop size to be $513 \times 513$ for the PASCAL VOC 2012 and CelebAMask-HQ datasets and $257 \times 257$ for ADE20K dataset. We train for 120 epochs using a batch size of 16 and the SGD method [22] is applied to optimise the pixel-wise cross-entropy loss with L2-regularisation. The initial learning rate is set to 0.01 with the decaying policy described in [9]. The **output stride**, which is defined in [8] denoting the ratio of original input resolution to the final feature’s resolution, is set to be 16 for all datasets. We adopt strategies in [9, 8] to use the BatchNorm layers [21] and to randomly scale the training data for augmentation. Depthwise convolution [10] is used in the ASPP implementations following [9]. During evaluations, we set the **output stride** to be 16 for all datasets, and we use a crop size of $257 \times 257$ for ADE20K and $513 \times 513$ for PASCAL VOC 2012 and CelebAMask-HQ.

**LI Layer Settings**  A lateral inhibition layer has several key hyper-parameters that can affect the performance. We fine-tune those parameters on the Pascal Voc 2012 validation set to determine a best-performing combination. Particularly, for LI-ASPP, we set the size of LI zones to be $3 \times 3$, the value for the standard deviation $\sigma$ in Eq. 5 is selected to be 1.0, the LI rate $e$ in Eq. 3 is set to 1, and we uniformly initialise the LI intensity $W_L$ in Eq. 5 between 0.0 and 0.15 during training. Almost identical LI settings are applied in LI-MNV2, except that all LI intensities $W_L$ are initialised as 0.0 such that the training can start smoothly without breaking any existing organisations in ImageNet checkpoint. Moreover, we evaluate different positions of adding LI bottlenecks in the MNV2 architectures, and a general trend can be observed that adding LI to higher layers can produce better performance than to bottom ones. Besides, we select ReLu as the activation $\phi$ in Eq. 3.

**Implementations**  We implement our method in the Tensorflow framework [1]. For the implementation of the baseline Deeplabv3+ [9] model, we directly use the code provided by authors. To ensure a fair comparison, the decoder module in Deeplabv3+ [9] is disabled for all experiments. It takes around one day per GPU (2080TI) to train a model (LI-MNV+LI-ASPP) on Pascal Voc 2012 dataset, and it requires about 2.5/0.6 days to do so on CelebAMask-HQ and ADE20K datasets.
Table 2: Performance of different methods on the Pascal VOC 2012 and ADE20K (validation set) and on the CelebAMask-HQ (test set). To provide a fair comparison, we repeat each experiment for three times to report the means and the standard deviations, and the model parameters and FLOPs (for crop size $513 \times 513$) are also included.

<table>
<thead>
<tr>
<th>Method</th>
<th>Pascal VOC 2012</th>
<th>CelebA Mask-HQ</th>
<th>ADE-20K</th>
<th>Parameters (Kilo)</th>
<th>FLOPs (Mega)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNV2+ASPP (Deeplabv3 [8])</td>
<td>71.85±0.30</td>
<td>74.69±0.31</td>
<td>30.02±0.11</td>
<td>2568.02</td>
<td>6479</td>
</tr>
<tr>
<td>MNV2 + LI-ASPP</td>
<td>72.63±0.24</td>
<td>75.25±0.05</td>
<td>30.43±0.27</td>
<td>2568.98</td>
<td>6498</td>
</tr>
<tr>
<td>LI-MNV2 + ASPP</td>
<td>72.47±0.34</td>
<td>75.29±0.15</td>
<td>30.44±0.36</td>
<td>2569.94</td>
<td>6517</td>
</tr>
<tr>
<td>LI-MNV2 + LI-ASPP</td>
<td><strong>72.95±0.24</strong></td>
<td><strong>75.55±0.13</strong></td>
<td><strong>30.52±0.18</strong></td>
<td>2570.52</td>
<td>6528</td>
</tr>
</tbody>
</table>

4.3 Results

In Table 1, we demonstrate the performance of different LI parameters on Pascal VOC 2012 validation set for LI-ASPP and LI-MNV2, respectively. In the LI-ASPP experiments, we investigate the performance of different settings of LI hyper-parameters such as the size of LI Zones, the LI rates $e$ and the initialisation range for $W_L$. As shown in Table 1, using a $3 \times 3$ LI zone and let $e = 1$ can generally yield better performance than other settings like a $5 \times 5$ LI Zone or $e = 5$. LI-ASPP achieves the best performance when $W_L$ is randomly initialised within $[0.0, 0.15]$, and therefore we opt for this setting for LI-ASPP.

In addition, we evaluate different options of adding LI-Convs in the Residual Bottleneck (RB) layers of the MNV2 architecture [40]. It can be spotted from Table 1 that adding LI mechanisms to the early RB layers (e.g. the earliest six RB layers) cannot promote the accuracy. In contrast, LI-Convs integrated with top layers such as the $\{10, 13, 16\}$th RB layers can produce higher mIoUs. This observation is somehow in line with the expectations since the higher-level layers are generally encoding more semantic representations, which can better benefit from the improved sensitivity to semantic contours introduced by LI layers.

In Table 2, we report the evaluation results of different methods on the three segmentation benchmark datasets, where each experiment is repeated for three times to present the mean and standard deviations (SD). Compared with the baseline method (MNV2+ASPP i.e. Deeplabv3 [8]), LI-MNV2 and LI-ASPP both demonstrate superior performance when used solely, while the best mIoUs on three datasets are all achieved by using them together. Particularly, our method (LI-MNV2+LI-ASPP) gains a relative improvement of 1.53%, 1.1% and 1.6% over the baseline on Pascal VOC 2012, CelebAMask-HQ and ADE-20K datasets, respectively, with more stabilised performance featured by smaller SDs. The model’s parameters and FLOPs, however, are slightly increased by 0.10% and 0.75%, which is arguably acceptable considering the accuracy compensations. Basing on those results, the effectiveness and generality of the proposed LI-Convs are therefore justified.

4.4 Discussion

How the LI layer works

To intuitively understand the LI mechanisms, we dive into the channel-level features to visualise the patterns before and after LI layers. As demonstrated in Fig. 4, we plot several feature channels before and after the LI layers in LI-ASPP on CelebAMask-HQ dataset. It can be discovered that although the intensity of activation is suppressed globally after the LI layer, the inhibited feature exhibits more recognisable patterns with clarified and emphasised contours, which can be more desirable in the segmentation domain.

What interests the model

In Fig. 5, we visualise the class-level heat maps and the segmentation predictions generated by the baseline (MNV2+ASPP) and our method (LI-MNV2+LI-ASPP) on CelebAMask-HQ. We utilise deeper reds to denote higher positive neurons responses (more model attention) in heat maps, and vice versa for deeper blues. Compared with the baseline, the semantically meaningful contouring areas receive more attention from our model, e.g. the "glasses" and "skin" heat maps in Fig. 5. Such kind of contour sensitivity can be reasonably attributed to the proposed...
Figure 4: Visualisations of the channel-level features before and after LI layers on CelebAMask-HQ. Although the activation is inhibited globally, the feature patterns after LI layer are generally easier to recognise mainly due to the clarifications on semantic contours.

LI-Convs. Besides, the segmentation predictions generated by our method have better visual qualities, which also verifies the superiority of the LI-Convs.

5 Conclusion

We describe a dilated convolution with lateral inhibitions (LI-Convs) to enhance the model’s sensitivity to semantic contours and to extract features at denser scales. The performance of the proposed LI-ASPP and LI-MNV2 architectures is shown to outperform the baseline method on three segmentation benchmark datasets, which verify the effectiveness and generality of the LI-Convs. We also investigate and try to understand the working mechanisms hidden behind. The proposed LI-Convs can be seamlessly integrated into deep models for other tasks, such as lip-reading and object detection, that require explicit awareness of the semantic boundaries.

Figure 5: Visualisations of the class-level heat maps and semantic predictions of the baseline (MNV2+ASPP) and our method (LI-MNV2+LI-ASPP) on CelebAMask-HQ. Deeper reds in heat maps represent higher positive responses or more attention from the model, and vice versa for deeper blues. Our method allocates more attention to shape the semantic boundary areas and thus can produce predictions with higher visual qualities.
**Broader Impact**

This work introduces a biologically inspired improvement to the dilated convolution algorithm for semantic segmentation, of which we do not see any direct or potential hazards that may lead to social or ethical issues. The proposed method can be potentially applied in various machine vision tasks to boost the performance.

**References**


