Pattern Recognition Letters
journal homepage: www.elsevier.com

Disentangled Convolution for Optimizing Receptive Field

Takumi Kobayashi[a,**]

[a]*National Institute of Advanced Industrial Science and Technology, 1-1-1 Umezono, Tsukuba, 305-8560, Japan*

Article history:
Received **
Received in final form **
Accepted **
Available online **

Communicated by **

Keywords: Convolution, Receptive Field, Gaussian Smoothing, CNN, Image Classification

ABSTRACT

Convolutional neural network (CNN) is one of the primary techniques for high-performance image recognition. The convolution operation with small-sized filters is a key ingredient of CNN and the receptive field of the whole CNN is enlarged by stacking lots of convolution layers. The convolution layer, however, is problematic in terms of the receptive field. It provides *fixed* small receptive field due to the *fixed* filter size in convolution, which requires us to manually control it in advance. Besides, the larger-sized convolution filters significantly increase computation cost. Thus, in this study, we propose a method to *adaptively* tune the receptive field of the convolution operation in an end-to-end manner as well as to enlarge the receptive field in a low computation cost. Based on the biological studies and scale-space theory, we can disentangle convolution operation into Gaussian envelope filtering for smoothing and derivative-related filtering, both of which are heterogeneously parameterized. Those two types of filters are jointly optimized in the end-to-end CNN training and the receptive field of the convolution is adequately optimized via learning the Gaussian envelope with a low extra computation cost. The experimental results on image classification tasks demonstrate that the proposed method effectively enlarges receptive field to improve performance.

1. Introduction

Convolutional neural networks (CNNs) have being successfully applied to various fields of image processing and recognition [18, 8]. The convolution operation is well suited for constructing natural images [27] and thus CNN can extract effective image features [8]. It also performs on various signals beyond images, such as audio signals [28] and point clouds [32].

The convolution efficiently reduces the parameter size of neural networks due to so-called weight sharing in which the fixed-size convolution filter works on an input image and feature maps in CNNs. Recently, CNNs are constructed by stacking lots of convolution layers of even small filters [21]; modern CNN models [8, 34] generally stack 3×3 convolutions deeply as the 3×3 filter is most effective from the practical computing viewpoint [4]. In these CNNs, convolution operation is problematic in terms of *receptive field*. The receptive field size of convolution is pre-fixed as the filter size, demanding us to manually control it for properly characterizing target appearance of various sizes. In addition, convolution of larger-sized filters increases computation cost significantly, due to which it is practically hard to directly enlarge the receptive field by larger filters.

Receptive field of filtering is closely connected to the scale-space theory [31] which provides local scale analysis by means of *Gaussian* smoothing. Image characteristics of various scales can be analyzed through controlling the scale of Gaussian filter. The Gaussian filter is also applied to CNNs toward smooth local pooling since it works as anti-aliasing filtering [19, 22]. The scale of Gaussian, however, is manually pre-fixed in those networks. Inspired from the biological study on visual cortex [14, 16], the convolution filters are supposed to be smooth enough for extracting effective visual features [6], and derivatives of Gaussian can be employed as filter bases to build the convolution filters in CNNs [12, 15]. Those basis filters are constructed by the Gaussian function of pre-fixed scale, and thus the receptive field of each convolution is accordingly *pre-fixed*. Thus, though the abovementioned approaches partly ad-

**Corresponding author:
e-mail: takumi.kobayashi@aist.go.jp (Takumi Kobayashi)

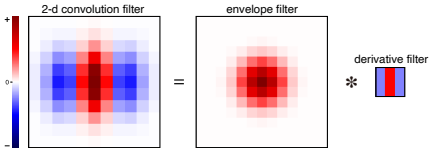


Fig. 1. Disentanglement of 2-d convolution filter. For visual recognition, most convolution filters can be composed of envelope and derivative filters.

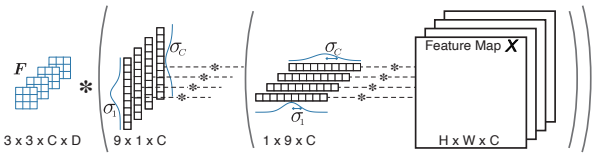


Fig. 2. Disentangled convolution. Standard 3×3 convolution with a filter F follows envelope filtering which is implemented by two successive channel-wise convolutions. The envelope filtering controls the receptive field via scale parameter σ_c. The trainable parameters are both $\{\sigma_c\}_{c=1}^{C}$ and F.

dress the receptive field issue of convolution, they employ pre-fixed scales, lacking flexibility based on image characteristics.

To remedy the prefixed receptive field size in convolution, we propose a method to adaptively optimize the receptive field of convolution on the basis of Gaussian which is fundamental representation for filters according to the biological studies and the scale-space theory. In the method, the standard convolution operation is disentangled into Gaussian envelope (smoothing) and derivative-related filtering, considering that the derivatives of Gaussian play key roles for convolution in CNNs [12, 15]. We *parameterize* the Gaussian envelope *analytically* while the derivative-related filtering is implemented by *discrete* filters as in the standard convolution (Fig. 2); the Gaussian envelope is equipped with the scale parameter σ. Thus, the proposed convolution is composed of these two *heterogeneous* filters which are jointly optimized in an end-to-end manner. By granting trainability to the Gaussian scale which is responsible for receptive field, we can adaptively tune the receptive field of the convolution. It is noteworthy that our method can efficiently enlarge the receptive field with a low computation cost through the Gaussian envelope filtering in contrast to enlarging the convolution filters. Our contributions are summarized as follows:

- We derive *parametric* Gaussian envelope functions through disentangling convolution to control the receptive field.
- The Gaussian scales are *automatically* optimized as trainable parameters in an end-to-end learning.
- The method works as convolution of larger-sized receptive field in a *low* computation cost.

The method may be related to the curriculum by smoothing (CBS) [26] which inserts Gaussian blurring layer after convolution to smooth out the feature maps for regularizing networks during training. In CBS, however, the scale of Gaussian filter is *manually* tuned so as to be gradually decayed according to curriculum learning [5], which clearly contrasts with our adaptive approach. Our key technical contribution of adaptively optimizing Gaussian scale is also distinctive compared to the prior works [19, 22, 12, 15] which pre-fixes the scale.

Notation. We denote feature map tensors by X and Y. Convolution *functions* are described such as by $\mathtt{f}(x,y)$ and $\mathtt{g}(x,y)$ which are practically implemented as *filter weights* $\boldsymbol{F}\in\mathbb{R}^{k\times k}$ and $\boldsymbol{G}\in\mathbb{R}^{k_g\times k_g}$. These representations are interchangeably used.

2. Method

2.1. Disentanglement of convolution

In CNNs, convolution filter of $\tilde{k}\times\tilde{k}\times C$ works on an input feature map of $H\times W\times C$ to produce a map of $H\times W$; actually, D filters are applied to output D-channel feature maps. We focus on the convolution for the c-th input channel.

As implied in some biological works [14, 16], for visual recognition, 2-d convolution filters are naturally composed of envelope and derivative-related filters as shown in Fig. 1. It inspires us to *disentangle* the CNN convolution filter $\mathtt{w}$ as

$$\mathtt{w}(x,y)=\frac{1}{Z}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)*\mathtt{f}(x,y)=\mathtt{g}(x,y;\sigma)*\mathtt{f}(x,y), \quad (1)$$

where $*$ indicates a convolution operator. The first term is the Gaussian envelope function $\mathtt{g}$ with the normalization constant Z and the scale parameter σ, which encapsulates the convolution into local compact support region. That is, the receptive field size of the convolution $\mathtt{w}$ is controlled by the parameter σ. On the other hand, the second term $\mathtt{f}$ is the filter to extract detailed image characteristics by means of derivative filtering, e.g., $\mathtt{f}(x,y)=\frac{\partial}{\partial x}\frac{\partial}{\partial y}$. Such a form of image filters in (1) can be found in the biological [14, 16] and theoretical [31] works and is also connected to the empirical observation in CNNs [15].

While the envelope function $\mathtt{g}$ is analytically formulated by means of Gaussian with a single parameter σ, the derivative-related filter $\mathtt{f}$ is hard to define its analytic form in advance; it is cumbersome to enumerate all the derivative orders that are effective for image features. Even though re-parameterizing it by using small number of analytic basis filters [12, 15], it is less computationally efficient for a small-sized filter, such as 3×3, also involving a risk of information loss. Thus, we model the filter function $\mathtt{f}$ as a discrete filter $\boldsymbol{F}\in\mathbb{R}^{k\times k}$ as in ordinary convolution filters of CNNs. As shown in Fig. 2, our disentangled convolution is composed of *analytic* envelope function $\mathtt{g}$ and *discrete* filter $\boldsymbol{F}$, both of which are trained in an end-to-end learning framework. It should be noted that the receptive field of the convolution is optimized through learning the parameter σ in the envelope, in contrast to the ordinary fixed-size convolution that pre-fixes the receptive field of the convolution.

The disentanglement (1) is different from the simple filter decomposition of $\mathtt{w}(x,y)=\mathtt{g}(x,y)\cdot\mathtt{f}(x,y)$ [15] which is based on the *multiplicative* combination of the envelope $\mathtt{g}$ and filtering $\mathtt{f}$. The simple form requires those two filters $\mathtt{g}$ and $\mathtt{f}$ to have the identical domain, i.e., the same filter size, making it impossible to define $\mathtt{g}$ and $\mathtt{f}$ heterogeneously. In contrast, the receptive field of convolution in our formulation (1) is controlled by the analytic envelope with σ in disregard of $\mathtt{f}$ which can be implemented by smaller-sized filters (Fig. 2).

2.2. Disentangled convolution on feature map

Forward path. The disentangled convolution (1) works on a feature map by the following successive convolution (Fig. 2);

$$\mathtt{w}(x,y)*X(x,y)=\boldsymbol{F}(x,y)*\{\mathtt{g}(x,y)*X(x,y)\}, \quad (2)$$

where an input feature map is denoted by $X \in \mathbb{R}^{H\times W}$. The first convolution by the envelope function g works as smoothing over the input feature map. For smoothing the feature maps independently across channels, we equip the envelope function g with channel-wise parameter σ_c to provide the convolution for C-channel feature maps by

$$\bar{X}(x,y,c) = \mathsf{g}(x,y;\sigma_c) * X(x,y,c), \forall c \in \{1,\cdots,C\}, \quad (3)$$

$$Y(x,y,d) = \sum_{c=1}^{C} \boldsymbol{F}(x,y,c,d) * \bar{X}(x,y,c), \forall d \in \{1,\cdots,D\}, \quad (4)$$

where an input feature map X of C-channels is first processed by $\mathsf{g}(x,y;\sigma_c)$ in a channel-wise manner and then convolved with the discrete filter $\boldsymbol{F} \in \mathbb{R}^{k\times k\times C\times D}$ over the whole channel to output D-channel map Y. While the channel-wise (depth-wise) convolution has been found in the other works [1, 9], our channel-wise convolution (3) performs by using the analytical filter $\mathsf{g}(x,y,;\sigma_c)$ with a channel-wise trainable parameter σ_c. After that, the ordinary convolution with trainable discrete filter $\boldsymbol{F}$ is applied in (4) as derivative-related filtering. As a result, our convolution layer contains two types of trainable parameters of $\{\sigma_c\}_{c=1}^{C}$ and $\boldsymbol{F} \in \mathbb{R}^{k\times k\times C\times D}$.

In (2) the smoothing via envelope function is followed by the convolution with $\boldsymbol{F}$. It is also conceivable to place the smoothing *after* the convolution as in the CBS [26] which applies Gaussian smoothing with hand-tuned σ after convolution.

$$\bar{Y}(x,y,d) = \sum_{c=1}^{C} \boldsymbol{F}(x,y,c,d) * X(x,y,c), \quad (5)$$

$$Y(x,y,d) = \mathsf{g}(x,y;\sigma_d) * \bar{Y}(x,y,d), \forall d \in \{1,\cdots,D\}, \quad (6)$$

This procedure of 'conv→env' tunes smoothness of the *output* feature map, which could effectively convey information to the subsequent layers. On the other hand, the approach of 'env→conv' in (2) requires lower computation cost since the number of input channel C is generally smaller than that of output channels D in most CNNs. We empirically compare those two types of processing orders in the experiment (Sec. 3.1).

Backward path. Since the envelope filtering (3) is followed by the standard convolution (4), we focus on the backward path through the envelope, given $\frac{\partial l}{\partial \bar{X}}$ where l indicates a loss. The loss derivatives with respect to the feature map X and the scale parameter σ are

$$\frac{\partial l}{\partial X(x,y,c)} = \mathsf{g}(x,y;\sigma_c) * \frac{\partial l}{\partial \bar{X}(x,y,c)}, \quad (7)$$

$$\frac{\partial l}{\partial \sigma_c} = \int\Big[\frac{\partial \mathsf{g}(x,y;\sigma_c)}{\partial \sigma_c} * X(x,y,c)\Big]\frac{\partial l}{\partial \bar{X}(x,y,c)}dxdy, \quad (8)$$

where we use the symmetry of Gaussian envelope function $\mathsf{g}(x,y;\sigma_c) = \mathsf{g}(-x,-y;\sigma_c)$. The feature map is updated through the envelope smoothing as in the forwarding path, which would filter out the high-frequency noises in the backward information. For updating σ_c via (8), the loss derivatives $\frac{\partial l}{\partial \bar{X}}$ are weighted by the convolution $\frac{\partial \mathsf{g}}{\partial \sigma_c} * X$ which indicates the saliency map on the feature map X by means of Gaussian derivative [31] (or DoG [20]). Thus, σ_c is learned based on how the salient features on X contributes to the loss l.

2.3. Implementation

Discrete filter $\boldsymbol{F}$. In discretized derivative operation, the filter size is related to the order of derivatives; for example, the 1st-order derivative $\frac{\partial}{\partial x}$ is realized by 2-pixel size, and the 2nd-order one requires at least 3-pixels. Based on the biological work [16] and the modern CNN architectures [8, 34], we apply 3×3 filters ($k = 3$) for the discrete filter $\boldsymbol{F}$ which can extract up to 2nd order derivatives well capturing various local image characteristics.

Gaussian envelope g. The *analytic* envelope function g that controls receptive field of convolution is implemented by *large-sized discrete* filters. To mitigate the computation issue of the large filtering, the Gaussian envelope is decomposed into separable filters (Fig. 2); $\mathsf{g}(x,y;\sigma) * X = \mathsf{g}(y;\sigma) * \{\mathsf{g}(x;\sigma) * X\}$ where $\mathsf{g}(x;\sigma) = \frac{1}{Z_\sigma}\exp(-\frac{x^2}{2\sigma^2})$. The 1-dimensional envelope of the size $k_g = 2r+1$, e.g., $k_g = 9$ in Fig. 2, is practically implemented by the discrete filter $\boldsymbol{g}_\sigma \in \mathbb{R}^{k_g}$ of

$$\boldsymbol{g}_\sigma[x] = \frac{\exp(-\frac{x^2}{2\sigma^2})}{\sum_{\xi=-r}^{r}\exp(-\frac{\xi^2}{2\sigma^2})} \Rightarrow \boldsymbol{g}_\sigma = \mathtt{softmax}\Big[\Big\{-\frac{x^2}{2\sigma^2}\Big\}_{x=-r}^{r}\Big], \quad (9)$$

where due to the discretization, the analytic normalization via Z_σ is replaced by softmax operation which is well established in a CNN literature. In addition, for training non-negative parameter σ, we introduce auxiliary parameter $\tilde{\sigma} \in \mathbb{R}$ such that $\sigma^2 = \log(1+\exp(\tilde{\sigma})) = \mathtt{softplus}(\tilde{\sigma})$ on $-\infty < \tilde{\sigma} < +\infty$. From a viewpoint of filtering, the discrete filter has $k_g \times k_g$ receptive field but it should noted that the Gaussian envelope (9) controls an *effective* receptive field [21] up to $k_g \times k_g$ where valid positive weights are assigned, as shown in Fig. 4b. We hereafter refer to this effective receptive field as receptive field for simplicity.

2.4. Comparison to other methods

Our method can be related to the other approaches from the following aspects. These comparisons highlight our technical contribution to optimize the scale parameter in an analytic Gaussian envelope function for tuning the receptive field size.

Smoothing. The method of curriculum-by-smoothing (CBS) [26] also applies Gaussian smoothing after convolution with *manually-tuned* scale (σ); σ is exponentially decayed in the manner of curriculum learning [5]. In contrast to CBS, the proposed method *automatically* tunes σ through an end-to-end learning in an adaptive manner for training images and tasks without manually exploring σ nor curriculum scheduling.

As mentioned in Sec. 2.1, smooth filters can be coupled with a convolution filter in a *multiplicative* manner of

$$\mathtt{w}(x,y) = \mathsf{g}(x,y) \cdot \mathtt{f}(x,y), \quad (10)$$

where the smooth envelope g is defined on the *same* receptive field as $\mathtt{f}$. In contrast, our approach is capable of enlarging the field size of g beyond that of $\mathtt{f}$ via the convolutional form (1).

While we analytically define envelope function g as a parametric Gaussian form (9), it could be described by means of ensemble approach used in [4]. As shown in Fig. 3, the envelope function g is approximated by a linear combination of several *fixed* smoothing functions by

$$\mathsf{g}(x,y;\sigma) * X(x,y) \approx \Big[\sum_{i=1}^{m}\alpha_i \mathsf{g}(x,y;\hat{\sigma}_i)\Big] * X(x,y), \quad (11)$$

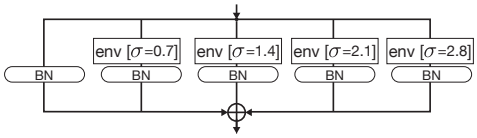


Fig. 3. Ensemble of smoothing filters of $m = 5$ scales.

where $\{\hat{\sigma}_i\}_{i=1}^m$ indicate m fixed scales. This approach is implemented with batch-normalization [11] as shown in Fig. 3, which converts the optimization of scales $\{\sigma_c\}_{c=1}^C$ in (9) into that of BatchNorm parameters [4].

Attention. We have so far discussed the envelope function from the smoothing viewpoint. It is also possible to relate the non-negative weight g with a local *attention* map [29]; our framework in Fig. 2 first aggregates local features by means of Gaussian attention map g and then applies a standard convolution. From that perspective, the Gaussian attention is generalized into a parametric attention map $\hat{\mathsf{g}}$ such that

$$\int \hat{\mathsf{g}}(x,y)dxdy = 1 \ \wedge \ \hat{\mathsf{g}}(x,y) \geq 0 \ \forall (x,y), \tag{12}$$

which is simply implemented by $\mathtt{softmax}(\boldsymbol{G})$ with an auxiliary weight parameter $\boldsymbol{G} \in \mathbb{R}^{k_g \times k_g}$ to be optimized. The local attention map $\boldsymbol{G}$ is constant and shared at any positions.

On the other hand, an adaptive local attention map is conceivable according to the attention-based approach [30] by constructing the filter $\boldsymbol{G}_{x,y}$ based on local features around $\boldsymbol{X}(x,y)$ as

$$\bar{\boldsymbol{X}}(x,y) = \sum_{(i,j)\in\mathcal{N}} G_{x,y}[i,j]\boldsymbol{X}(x+i,y+j), \tag{13}$$

$$\boldsymbol{G}_{x,y} = \mathtt{softmax}(\{\boldsymbol{w}_c^\top \boldsymbol{X}(x,y) + \boldsymbol{w}_n^\top \boldsymbol{X}(x+i,y+j)\}_{i,j}), \tag{14}$$

where $\boldsymbol{w}_c$ and $\boldsymbol{w}_n \in \mathbb{R}^C$ are projection vectors from the center feature vector $\boldsymbol{X}(x,y)$ and neighboring one $\boldsymbol{X}(x+i,x+j)$ into an attention weight [30], respectively. This approach adaptively assigns a local attention map to each position (x,y) while ours and the above-mentioned one (12) use constant envelope.

Note that these methods demand considerable computation cost due to the parameter size $\boldsymbol{G}$ and the projection (14).

Convolution. From the architectural viewpoint (Fig. 2), our method stacks successively two convolution layers for envelope smoothing (3) and feature extraction (4). It should be noted that the first convolution is constrained to the analytic Gaussian function parameterized by σ_c for controlling receptive field of the convolution. The parametric Gaussian envelope can be architecturally extended into the trainable depth-wise (DW) convolution [1, 9] composed of $1 \times k_g$ and $k_g \times 1$ filters in accordance with our implementation (Sec. 2.3) which are optimized without any analytic constraints through the end-to-end learning; this DW approach optimizes $1 \times k_g$ and $k_g \times 1$ filters at each channel in addition to the standard convolution with a filter $\boldsymbol{F}$.

As to the parameter size, our method introduces C additional parameters, $\{\sigma_c\}_{c=1}^C$, per convolution. It is fairly compared to enhanced convolution which is equipped with additional parameters. For example, additional projection consuming D parameters $\{w_d\}_{d=1}^D$ is introduced by

$$\boldsymbol{Y}(x,y,d) = \sum_{c=1}^{C} \boldsymbol{F}(x,y,c,d) * \boldsymbol{X}(x,y,c) + w_d \frac{1}{C} \sum_{c=1}^{C} \bar{\boldsymbol{X}}_{avg}(x,y,c), \tag{15}$$

where $\bar{\boldsymbol{X}}_{avg}$ is constructed by applying average-pooling of size $k_g \times k_g$ to $\boldsymbol{X}$. It is also comparable to apply local squeeze-and-excitation (SE) [10] with C additional parameters $\{\alpha_c\}_{c=1}^C$ as

$$\boldsymbol{Y}(x,y,d) = \sum_{c=1}^{C} \mathtt{sigmoid}(\alpha_c \bar{\boldsymbol{X}}_{avg}(x,y,c)) \left[\boldsymbol{F}(x,y,c,d) * \boldsymbol{X}(x,y,c)\right]. \tag{16}$$

3. Experimental Results

We applied the proposed method to ResNet-50 [8] on ImageNet classification task [3]. Since the receptive field of convolution would be crucial at downsizing feature maps, the proposed method is embedded at the layers that perform downsizing by means of 2-strided convolution; ResNet contains *four* downsizing layers including the first convolution (Fig. 5b) and three residual blocks (ResBlocks, Fig. 5a).

Following the standard practice [8], the ResNet is trained on mini-batch size of 256 by SGD with momentum of 0.9, weight decay of 0.0001 and initial learning rate of 0.1 which is divided by 10 every 30 epoch over 120 training epochs. The scale parameter $\tilde{\sigma}_c$ is initialized as $\tilde{\sigma}_c = 0$, while the other parameters are initialized by the standard protocol.

3.1. Performance analysis on ImageNet training

We analyze the proposed method with the envelope size $k_g = 9$ from the following aspects.

ResBlock. As shown in Fig. 5a, the proposed convolution is applied at the ResBlock to two types of strided convolution; one is 1×1 convolution at the *identity path* and the other is 3×3 convolution at the *residual path*. We explore how the proposed convolution works on those paths in Table 1a. The performance of original ResNet (denoted by 'none' in Table 1a) is improved by applying our convolution to residual path, which is further enhanced by the application to both paths. Our convolutions at both paths equally contributes to performance improvement; our method at each path improves the performance by about 0.4 point. The proposed method can effectively control the receptive field size through not only the residual path but also the identity path of point-wise (1×1) convolution. We hereinafter apply our convolution to both paths at the ResBlocks.

Layer position. Then, in Table 1b, we compare performance in terms of layer positions at which the proposed method is embedded. The method at the deeper layers effectively contributes to performance improvement, while the one at the 1st layer is less effective; the method applied to the 2nd~4th layers produces almost the same performance as the full model applying our convolution to the 1st~4th layers. As discussed later, this is also verified through visualizing envelope function (Fig. 4b) to show that the receptive field of the first convolution layer is not enlarged by the Gaussian envelope. On the other hand, the

Table 1. Performance analysis by ResNet-50 [8] on ImageNet [3]. We report the top-1 error rate (%) with top-5 error rate.

(a) Paths at ResBlock

Path	none	residual	identity& residual
Err.	23.81 6.91	23.46 6.95	23.03 6.72

(b) Layer positions

Layers	1,2,3,4	2,3,4	3,4	4
Err.	23.03 6.72	23.08 6.67	23.44 6.81	23.57 7.01

(c) Processing order

Order	env→conv	conv→env
Err.	23.03 6.72	22.93 6.69

(d) Parameterization of envelope

Type	single σ	multiple $\{\sigma_c\}_{c=1}^{C}$
Err.	23.40 7.04	23.03 6.72

(e) Envelope size at inference

Size	9×9	3×3
Err.	23.03 6.72	22.98 6.70

(f) Computation complexity

Method	ResNet-50	Ours (9×9)	Ours (3×3)
GFLOPs	3.84	3.89	3.86
Speed (ms/img)	0.167	0.184	0.177

(g) Comparison to other methods

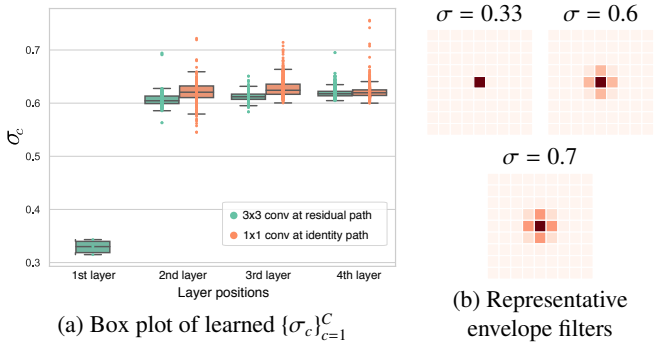
	Convolution			Attention		Smoothing			
	+DW [9]	+Proj. (15)	+SE (16)	Param. (12)	Adapt. (14)	CBS [26]	Mult. (10)	Ens.(11)	Ours
Err.	24.65 7.50	23.60 7.01	23.85 7.13	23.42 6.94	23.59 6.80	23.91 7.18	23.74 7.06	23.44 7.05	23.03 6.72

(h) Various fixed σ

σ	0	0.3	0.6	0.7	1	2	ours
Err.	23.81 6.91	23.67 6.97	23.23 6.84	23.21 6.79	23.77 7.22	26.79 8.86	23.03 6.72

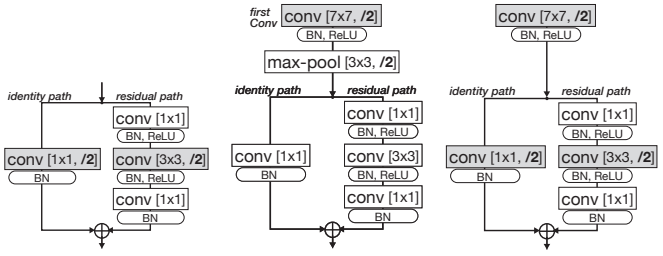
(i) Performance on various ResNet models

CNN	ResNet-34	ResNet-50	ResNet-101	ResNeXt-50 [34]
orig.	25.90 8.27	23.81 6.91	22.04 6.17	22.32 6.54
ours	25.36 7.97	22.58 6.29	21.56 5.94	21.93 6.25



(a) Box plot of learned $\{\sigma_c\}_{c=1}^{C}$ (b) Representative envelope filters

Fig. 4. Learned envelope filters at respective layers. In (a), each learned σ_c is depicted by the dot point, and (b) shows representative ones; $\sigma = 0.33$ for 1st layer (mean), 0.6 and 0.7 for other layers (mean and max).



(a) ResBlock (b) 1st layer (c) our 1st layer

Fig. 5. ResBlock equipped with the proposed convolution. The notion of '/2' indicates the stride of 2 pixels. The proposed convolution is applied to the gray-colored layers. (a) Standard ResBlock. (b) 1st layer of ResNet. (c) Our modified 1st layer of ResNet.

deeper layers demand the receptive field to be properly optimized for better performance. Enlargement of convolution receptive field at the deeper layers more effectively contributes to enlarge the receptive field of the whole CNN.

Before/After convolution. As mentioned in Sec. 2.2, the processing order of the Gaussian envelope $\mathbf{g}$ and the discrete filter $\boldsymbol{F}$ is analyzed; two types of ordering, envelope before/after convolution, are conceivable as shown in (3,4) and (5,6). Table 1c demonstrates that, though the model of 'conv→env' contains larger number of parameters ($\{\sigma_d\}_{d=1}^{D}$) due to $D > C$, there is less significant performance difference between the two approaches. Thus, for computation efficiency, we apply Gaussian envelope before convolution (env→conv) in (3,4).

Parameterization of Gaussian envelope. We apply the channel-wise Gaussian envelope (3) to capture spatial characteristics from respective feature maps, which is compared to the simple envelope parameterized only by single σ across channels. Table 1d shows that the simple approach deteriorates performance since it ignores the difference of feature map characteristics among channels, validating the effectiveness of our channel-wise smoothing.

Analysis of learned σ. We then qualitatively analyze the learned channel-wise scales of $\sigma_c^2 = \texttt{softplus}(\tilde{\sigma}_c)$. Fig. 4a shows the learned $\{\sigma_c\}_{c=1}^{C}$ at respective paths and layers. The learned scale at the first convolution is small with less diversity, which interestingly contrasts with the (hand-crafted) original larger convolution size of 7×7 in the ResNet architecture (Fig. 5b). On the other hand, the scales at the deeper layers are moderately larger, exhibiting larger variance across channels which also reflects the difference among channels. It should be noted that the receptive field of the whole CNN can be more effectively enlarged by the larger σ at the deeper layers. Fig. 4b visually depicts the envelope filters, demonstrating that the envelope at the first convolution is close to delta function without smoothing, while the deeper layers demand smoothing effects.

Shrinking envelope filter at inference. As shown in Fig. 4b, the *effective* receptive field of learned envelope filters are so compact as to be encapsulated even by 3×3 filter size, implying that the envelope filters are redundantly implemented by larger size of 9×9, $k_g = 9$ in (9), at a training phase. Thus, *after* training, the envelope filter size k_g can be reduced to fit the learned

effective receptive field; we apply $k_g = 3$ to the Gaussian envelope (9) with the learned σ_c *only at inference*. Table 1e&f show that the smaller envelope filter of 3×3 successfully reduces computation cost while keeping performance.

Computation complexity. The method is evaluated from the computation viewpoint. It roughly requires additional computation of $2k_gCHW$ per convolution layer due to the decomposed envelope computation in Sec. 2.3. Thus, as shown in Table 1f, the ResNet-50 model equipped with our method increases negligible computation cost compared to the baseline, especially in the case of $k_g = 3$ by the shrinking technique (Table 1e); the time speed is measured on a single NVIDIA Titan-V GPU.

Fixed σ. In Table 1h, our method automatically optimizing σ is compared with naive approaches to fix σ; we can fix scales based on the analysis in Fig. 4. The performance is improved via slightly enlarging σ and the favorable performance is given by $\sigma = 0.6$ and 0.7, the majority of our optimized σs in Fig. 4, though being inferior to ours. The performance is then suddenly degraded by $\sigma > 0.7$. Such a sensitivity to the scale σ makes it hard to manually tune σ in advance. Thus, our method is advantageous in both points of improving performance and eliminating laborious parameter tuning.

Comparison. We then compare the proposed method to the comparable approaches discussed in Sec. 2.4; the performance comparison is shown in Table 1g.

From the architectural viewpoint that we stack two convolution layers with C additional parameters, the method is compared with augmented convolution methods by depth-wise (DW) module [9] as well as projection (15) and SE module (16). Especially, the latter two approaches (15,16) that roughly adds C parameters to baseline provide fair comparison in terms of parameter size. Our method outperforms those approaches, demonstrating that the additional C parameters effectively works in the form of Gaussian envelope.

The Gaussian envelope can be related to local attention models. The analytic form of Gaussian is generalized to (trainable) parametric weight (12), while adaptive attention model (14) constructs local attention weights based on input neuron activations in contrast to our static Gaussian envelope. Those attention models improve performance, yet being inferior to ours. The analytic Gaussian envelope is simple and general to endow effective inductive bias with low computation cost, even compared to these local attention models based on a more general parametric map and an adaptive prediction (14).

Our method shares the motivation of smoothing with CBS [26] which manually tunes scale parameters σ_c in the curriculum learning framework [5]. The smoothing Gaussian envelope also has some alternatives of multiplicative envelope over convolution weights (10) and ensemble approach (11) in Fig. 3. The performance comparison validates the effectiveness of our analytic envelope with automatic optimization for the scale σ.

The performance results in Table 1g demonstrate that the proposed method works well by optimizing the receptive field by means of the parameterized Gaussian smoothing function.

Architectural modification on ResNet. As shown in the above experimental results, the method adaptively enlarges the receptive field of convolution, which inspires us to slightly modify the CNN architecture of ResNet (Fig. 5b). We exclude the max-pooling after the first convolution and apply the downsizing operation to the following ResBlock with our convolution as shown in Fig. 5c; thereby, the proposed convolution is embedded at *five* layers of ResNet in total. Table 1i shows performance results of various ResNet models equipped with the proposed convolution, exhibiting performance improvement.

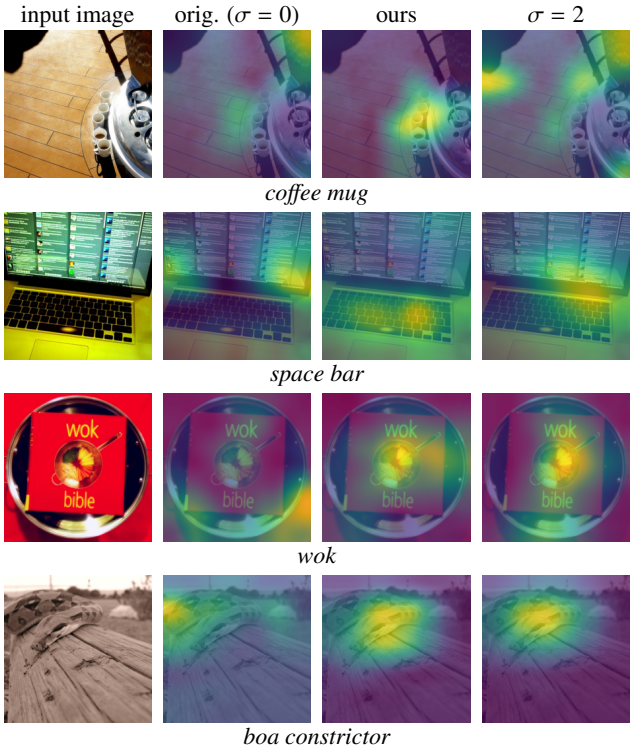


Fig. 6. GradCam [24] attention maps of ResNet-50 models.

Fig. 6 shows examples of attention maps produced by applying GradCam [24] to ResNet-50 models. In the original model, the features at image borders occasionally interfere in performance. On the other hand, the proposed method favorably suppresses those features by means of smooth envelope function at convolution layers to improve performance. The model fixing $\sigma = 2$ (Table 1h) produces similar maps to ours, though degrading the attention in some cases by the larger receptive field.

3.2. Transfer learning

To assess the effectiveness of feature representation learned by the method, the pre-trained CNNs on ImageNet are transferred to the other downstream image classification tasks. For fair comparison, only the final FC classifier in the ResNets is fine-tuned on the datasets while freezing the feature extraction layers before the global average pooling.

We evaluate performance on the datasets of Caltech-256 [7], MIT-67 [23], SUN-397 [33], FMD [25] and DTD [2] which provide various tasks related to image classification to assess the transferability of the pre-trained CNN models. The performance is measured by classification error rates (%) computed via the standard protocol provided in the respective datasets; for Caltech-256, we draw 60 training samples on each class, and for details of the other datasets, refer to the respective papers.

Table 2. Performance results (error rates %) by transfer learning.

Type	*object*	*scene*		*material*	
Dataset	Caltech-256	MIT-67	SUN-397	FMD	DTD
orig. ResNet-34	17.24	34.48	46.37	20.87	33.80
our ResNet-34	16.80	32.91	45.95	20.00	33.37
orig. ResNet-50	15.27	30.00	43.32	19.73	30.62
our ResNet-50	15.05	28.51	42.36	17.27	29.72
CBS [26] ResNet-50	15.27	28.73	42.81	19.67	30.22

Performance results are shown in Table 2. Our ResNet models of which receptive fields are optimized via envelope functions are superior to the original models. In particular the performance improvement is well found in the tasks which are rather apart from the object classification that is closely related to the ImageNet classification (pre-training) task. These results demonstrate that our method contributes to improve feature representation toward high generalization by properly optimizing the receptive fields of convolution to mitigate overly fitting to the target dataset/task. Such improvement can be discussed from the regularization perspective [26, 13, 17]. The Gaussian envelope renders smoothing *regularization* to CNN by reducing gradient noises to reach the optimizer of better generalization [13]. Smoothing is common artifact frequently observed in images and thus the regularization regarding the smoothing incorporates favorable inductive bias to CNNs for improving transferability in contrast to the other common regularization [17]. As shown in Table 2, the CBS [26] slightly improves performance due to the smoothing regularization injected during training, but is inferior to our method which optimizes the degree of smoothing via learning σ_c.

4. Conclusion

We have proposed a convolution method to adaptively optimize receptive field. Based on the biological insights and the scale-space theory, the standard convolution process is disentangled into two components of Gaussian envelope smoothing and derivative-related filtering, which are formulated in analytic and discrete manners, respectively. Through end-to-end learning, those two heterogeneously parameterized filters are jointly optimized while adequately tuning the receptive field of convolution via the Gaussian scale parameter in contrast to the standard convolution and prior works which pre-fixes the scale. Our method performs as a convolution of larger-sized receptive field with a low computation cost due to the efficient Gaussian envelope. In the experiments on the ImageNet classification task, the proposed method is thoroughly evaluated from various aspects and exhibits favorable performance improvement even on transfer learning scenarios using the other datasets.

References

[1] Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In: CVPR. pp. 1800–1807.

[2] Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A., 2014. Describing textures in the wild. In: CVPR.

[3] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255.

[4] Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., Sun, J., 2021. Repvgg: Making vgg-style convnets great again. In: CVPR.

[5] dn Jerome Louradour, Y. B., Collobert, R., Weston, J., 2009. Curriculum learning. In: ICML.

[6] Feinman, R., Lake, B. M., 2019. Learning a smooth kernel regularizer for convolutional neural networks. In: CogSci.

[7] Griffin, G., Holub, A., Perona, P., 2007. Caltech-256 object category dataset. Tech. Rep. 7694, Caltech.

[8] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: CVPR. pp. 770–778.

[9] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv, 1704.04861.

[10] Hu, J., Shen, L., Sun, G., 2018. Squeeze-and-excitation networks. In: CVPR. pp. 7132–7141.

[11] Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. Journal of Machine Learning Research 37, 448–456.

[12] Jacobsen, J.-H., van Gemert, J., Lou, Z., Smeulders, A. W. M., 2016. Structured receptive fields in cnns. In: CVPR. pp. 2610–2619.

[13] Jastrzebski, S., Szymczak, M., Fort, S., Arpit, D., Tabor, J., Cho, K., Geras, K., 2020. The break-even point on optimization trajectories of deep neural networks. In: ICLR.

[14] Jones, J. P., Palmer, L. A., 1987. An evaluation of the twodimensional gabor filter model of simple receptive fields in cat striate cortex. J. Neurophysiology 58 (6), 1233–1258.

[15] Kobayashi, T., 2018. Analyzing filters toward efficient convnets. In: CVPR. pp. 5619–5628.

[16] Koenderink, J. J., van Doorn, A. J., 1987. Representation of local geometry in the visual system. Biological cybernetics 55 (6), 367–375.

[17] Kornblith, S., Shlens, J., Le, Q. V., 2019. Do better imagenet models transfer better. In: CVPR.

[18] Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. In: NeurIPS. pp. 1097–1105.

[19] Lee, J., Won, T., Lee, T. K., Lee, H., Gu, G., Hong, K., 2020. Compounding the performance improvements of assembled techniques in a convolutional neural network. arXiv:2001.06268.

[20] Lowe, D. G., 2004. Distinctive image features from scale invariant features. International Journal of Computer Vision 60, 91–110.

[21] Luo, W., Li, Y., Urtasun, R., Zemel, R., 2016. Understanding the effective receptive field in deep convolutional neural networks. In: NeurIPS. pp. 9446–9454.

[22] Mairal, J., 2016. End-to-end kernel learning with supervised convolutional kernel networks. NeurIPS, 1399–1407.

[23] Quattoni, A., Torralba, A., 2009. Recognizing indoor scenes. In: CVPR. pp. 413–420.

[24] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In: ICCV. pp. 618–626.

[25] Sharan, L., Rosenholtz, R., Adelson, E. H., 2014. Accuracy and speed of material categorization in real-world images. Journal of Vision 14 (10).

[26] Sinha, S., Garg, A., Larochelle, H., 2020. Curriculum by smoothing. In: NeurIPS.

[27] Ulyanov, D., Vedaldi, A., Lempitsky, V., 2018. Deep image prior. In: CVPR. pp. 9446–9454.

[28] van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K., 2016. Wavenet: A generative model for raw audio. arXiv:1609.03499.

[29] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. In: NeurIPS.

[30] Wang, X., Girshick, R., Gupta, A., He, K., 2018. Non-local neural networks. In: CVPR. pp. 7794–7803.

[31] Witkin, A. P., 1983. Scale-space filtering. In: IJCAI.

[32] Wu, W., Qi, Z., Fuxin, L., 2019. Pointconv: Deep convolutional networks on 3d point clouds. In: CVPR. pp. 9621–9630.

[33] Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., Torralba, A., 2010. Sun database: Large-scale scene recognition from abbey to zoo. In: CVPR.

[34] Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K., 2017. Aggregated residual transformations for deep neural networks. In: CVPR. pp. 5987–5995.