Funnel Activation for Visual Recognition

Ningning Ma[1][0000−0003−4628−8831], Xiangyu Zhang[2⋆][0000−0003−2138−4608], and Jian Sun[2][0000−0002−6178−4166]

[1] Hong Kong University of Science and Technology
[2] MEGVII Technology
nmaac@cse.ust.hk, {zhangxiangyu,sunjian}@megvii.com

Abstract. We present a conceptually simple but effective funnel activation for image recognition tasks, called *Funnel activation (FReLU)*, that extends ReLU and PReLU to a 2D activation by adding a negligible overhead of spatial condition. The forms of ReLU and PReLU are $y = max(x, 0)$ and $y = max(x, px)$, respectively, while FReLU is in the form of $y = max(x, \mathbb{T}(x))$, where $\mathbb{T}(\cdot)$ is the 2D spatial condition. Moreover, the spatial condition achieves a pixel-wise modeling capacity in a simple way, capturing complicated visual layouts with regular convolutions. We conduct experiments on ImageNet, COCO detection, and semantic segmentation tasks, showing great improvements and robustness of FReLU in the visual recognition tasks. Code is available at https://github.com/megvii-model/FunnelAct.

Keywords: funnel activation, visual recognition, CNN

1 Introduction

Convolutional neural networks (CNNs) have achieved state-of-the-art performance in many visual recognition tasks, such as image classification, object detection, and semantic segmentation. As popularized in the CNN framework, one major kind of layer is the convolution layer, another is the non-linear activation layer.

First in the convolution layers, capturing the spatial dependency adaptively is challenging, many advances in more complex and effective convolutions have been proposed to grasp the local context adaptively in images [7,18]. The advances achieve great success especially on dense prediction tasks (e.g., semantic segmentation, object detection). Driven by the advances in more complex convolutions and their less efficient implementations, a question arises: *Could regular convolutions achieve similar accuracy, to grasp the challenging complex images?*

Second, usually right after capturing spatial dependency in a convolution layer *linearly*, then an activation layer acts as a scalar non-linear transformation. Many insightful activations have been proposed [31,14,5,25], but improving the performance on visual tasks is challenging, therefore currently the most widely used activation is still the Rectified Linear Unit (ReLU) [32]. Driven by the

⋆ Corresponding author

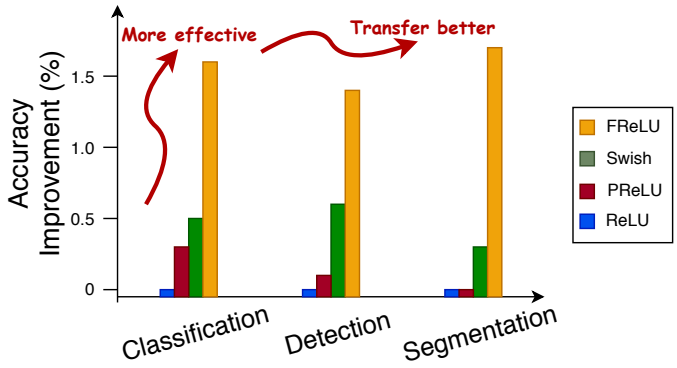

Fig. 1. Effectiveness and **generalization** performance. We set the ReLU network as the baseline, and show the *relative improvement* of accuracy on the three basic tasks in computer vision: image classification (Top-1 accuracy), object detection (mAP), and semantic segmentation (mean_IU). We use the ResNet-50 [15] as the backbone pre-trained on the ImageNet dataset, to evaluate the generalization performance on COCO and CityScape datasets. FReLU is more effective, and transfer better on all of the three tasks.

distinct roles of the convolution layers and activation layers, another question arises: *Could we design an activation specifically for visual tasks?*

To answer both questions raised above, we show that the simple but effective visual activation, together with the regular convolutions, can also achieve significant improvements on both dense and sparse predictions (e.g. image classification, see Fig. 1). To achieve the results, we identify spatially insensitiveness in activations as the main obstacle impeding visual tasks from achieving significant improvements and propose a new visual activation that eliminates this barrier. In this work, we present a simple but effective visual activation that extends ReLU and PReLU to a 2D visual activation.

Spatially insensitiveness is addressed in modern activations for visual tasks. As popularized in the ReLU activation, non-linearity is performed using a $max(\cdot)$ function, the condition is the hand-designed *zero*, thus in the scalar form: $y = max(x, 0)$. The ReLU activation consistently achieves top accuracy on many challenging tasks. Through a sequence of advances [31,14,5,25], many variants of ReLU modify the condition in various ways and relatively improve the accuracy. However, further improvement is challenging for visual tasks.

Our method, called **Funnel activation (FReLU)**, extends the spirit of ReLU/PReLU by adding a spatial condition (see Fig. 2) which is simple to implement and only adds a negligible computational overhead. Formally, the form of our proposed method is $y = max(x, \mathbb{T}(x))$, where $\mathbb{T}(x)$ represents the simple and efficient spatial contextual feature extractor. By using the spatial condition in activations, it simply extends ReLU and PReLU to a visual parametric ReLU with a pixel-wise modeling capacity.

Our proposed visual activation acts as an efficient but much more effective alternative to previous activation approaches. To demonstrate the effectiveness of the proposed visual activation, we replace the normal ReLU in classification networks, and we use the pre-trained backbone to show its generality on the other two basic vision tasks: object detection and semantic segmentation. The results show that FReLU not only improves performance on a single task but also transfers well to other visual tasks.

2 Related Work

Scalar activations Scalar activations are activations with single input and single output, in the form of $y = f(x)$. The Rectified Linear Unit (ReLU) [13,23,32] is the most widely used scalar activation on various tasks [26,38], in the form of $y = max(x, 0)$. It is simple and effective for various tasks and datasets. To modify the negative part, many variants have been proposed, such as Leaky ReLU [31], PReLU [14], ELU [5]. They keep the positive part identity and make the negative part dependent on the sample adaptively.

Other scalar methods such as the sigmoid non-linearity has the form $\sigma(x) = 1/(1+e^{-x})$, and the Tanh non-linearity has the form $tanh(x) = 2\sigma(2x)-1$. These activations are not widely used in deep CNNs mainly because they saturate and kill gradients, also involve expensive operations (exponentials, etc.).

Many advances followed [25,39,1,16,35,10,46], and recent searching technique contributes to a new searched scalar activation called Swish [36] by combing a comprehensive set of unary functions and binary functions. The form is $y = x * Sigmoid(x)$, outperforms other scalar activations on some structures and datasets, and many searched results show great potential.

Contextual conditional activations Besides the scalar activation which only depends on the neuron itself, conditional activation is a many-to-one function, which activates the neurons conditioned on contextual information. A representative method is Maxout [12], it extends the layer to a multi-branch and selects the maximum. Most activations apply a non-linearity on the linear dot product between the weights and the data, which is: $f(w^T x + b)$. Maxout computes the $max(w_1^T x + b_1, w_2^T x + b_2)$, and generalizes ReLU and Leaky ReLU into the same framework. With dropout [17], the Maxout network shows improvement. However, it increases the complexity too much, the numbers of parameters and multiply-adds has doubled and redoubled.

Contextual gating methods [8,44] use contextual information to enhance the efficacy, especially on RNN based methods, because the feature dimension is relatively smaller. There are also on CNN based methods [34], since 2D feature size has a large dimension, the method is used after a feature reduction.

The contextually conditioned activations are usually channel-wise methods. However, in this paper, we find the spatial dependency is also important in the non-linear activation functions. We use light-weight CNN technique depth-wise separable convolution to help with the reduction of additional complexity.

Spatial dependency modeling Learning better spatial dependency is challenging, Some approaches use different shapes of convolution kernels [41,42,40] to aggregate the different ranges of spatial dependences. However, it requires a multi-branch that decreases efficiency. Advances in convolution kernels such as atrous convolution [18] and dilated convolution [47] also lead to better performance by increasing the receptive field.

Another type of methods learn the spatial dependency adaptively, such as STN [22], active convolution [24], deformable convolution [7]. These methods adaptively use the spatial transformations to refine the short-range dependencies, especially for dense vision tasks (e.g. object detection, semantic segmentation). Our simple FReLU even outperforms them without complex convolutions.

Moreover, the non-local network provides the methods to capture long-range dependencies to address this problem. GCNet [3] provides a spatial attention mechanism to better use the spatial global context. Long-range modeling methods achieve better performance but still require additional blocks into the origin network structure, which decreases efficiency. Our method address this issue in the non-linear activations, solve this issue better and more efficiently.

Receptive field The region and size of receptive field are essential in vision recognition tasks [50,33]. The work on effective receptive field [29,11] finds that different pixels contribute unequally and the center pixels have a larger impact. Therefore, many methods have been proposed to implement the adaptive receptive field [7,51,49]. The methods achieve the adaptive receptive field and improve the performance, by involving additional branches in the architecture, such as developing more complex convolutions or utilizing the attention mechanism. Our method also achieves the same goal, but in a more simple and efficient manner by introducing the receptive field into the non-linear activations. By using the more adaptive receptive field, we can approximate the layouts in common complex shapes, thus achieve even better results than the complex convolutions, by using the efficient regular convolutions.

3 Funnel Activation

FReLU is designed specifically for visual tasks and is conceptually simple: the condition is a hand-designed zero for ReLU and a parametric px for PReLU, to this we modify it to a 2D funnel-like condition dependent on the spatial context. The visual condition helps extract the fine spatial layout of an object. Next, we introduce the key elements of FReLU, including the funnel condition and the pixel-wise modeling capacity, which are the main missing parts in ReLU and its variants.

ReLU We begin by briefly reviewing the ReLU activation. ReLU, in the form $max(x, 0)$, uses the $max(\cdot)$ to serve as non-linearity and uses a hand-designed $zero$ as the condition. The non-linear transformation acts as a supplement of the linear transformation such as convolution and fully-connected layers.

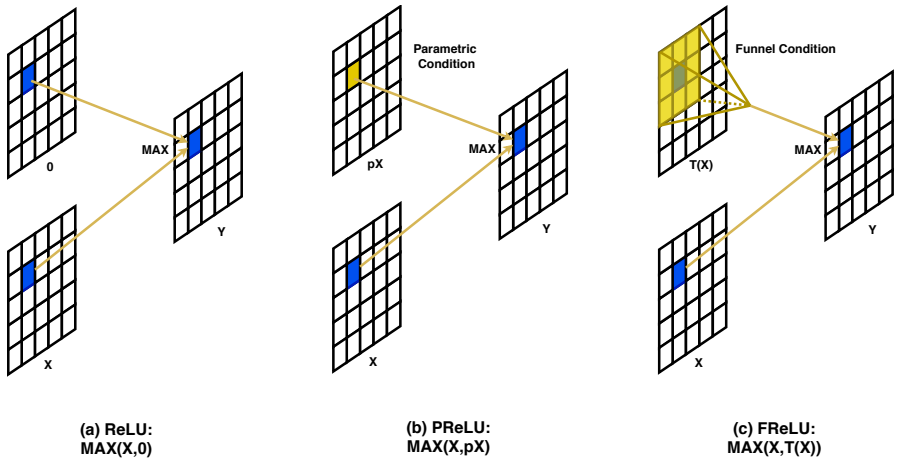

Fig. 2. Funnel activation. We propose a novel activation for visual recognition we call $FReLU$ that follows the spirit of ReLU/PReLU and extends them to 2D by adding a visual funnel condition $\mathbb{T}(x)$. (a) ReLU with a condition zero; (b) PReLU with a parametric condition; (c) FReLU with a visual parametric condition.

PReLU As an advanced variant of ReLU, PReLU has an original form $max(x, 0)+p\cdot min(x, 0)$, where p is a learnable parameter and initialized as 0.25. However, in most case $p < 1$, under this assumption, we rewrite it to the form: $max(x, px)$, $(p < 1)$. Since p is a channel-wise parameter, it can be interpreted as a 1x1 depth-wise convolution regardless of the bias terms.

Funnel condition FReLU adopts the same $max(\cdot)$ as the simple non-linear function. For the condition part, FReLU extends it to be a 2D condition dependent on the spatial context for each pixel (see Fig. 2). This is in contrast to most recent methods whose condition depends on the pixel itself (e.g. [31,14]) or the channel context (e.g. [12]). Our approach follows the spirit of ReLU that uses a $max(\cdot)$ to obtain the maximum between x and a condition.

Formally, we define the funnel condition as $\mathbb{T}(x)$. To implement the spatial condition, we use a **Parametric Pooling Window** to create the spatial dependency, specifically, we define the activation function:

$$f(x_{c,i,j}) = max(x_{c,i,j}, \mathbb{T}(x_{c,i,j})) \tag{1}$$

$$\mathbb{T}(x_{c,i,j}) = x^{\omega}_{c,i,j} \cdot p^{\omega}_{c} \tag{2}$$

Here, $x_{c,i,j}$ is the input pixel of the non-linear activation $f(\cdot)$ on the c-th channel, at the 2-D spatial position (i, j); function $\mathbb{T}(\cdot)$ denotes the funnel condition, $x^{\omega}_{c,i,j}$ denotes a $k_h \times k_w$ **Parametric Pooling Window** centered on $x_{c,i,j}$, p^{ω}_{c} denotes the coefficient on this window which is shared in the same channel, and $(\cdot)$ denotes dot multiply.

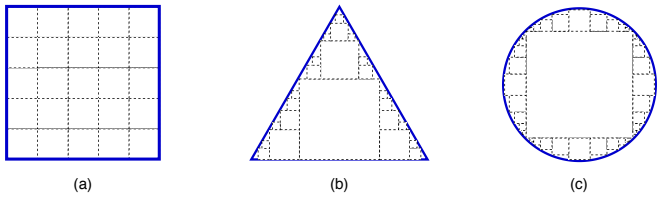

Fig. 3. Graphic depiction of how the per-pixel funnel condition can achieve *pixel-wise modeling capacity*. The distinct sizes of squares represent the distinct *activate fields* of each pixel in the top activation layers. (a) The normal activate field that has equal sizes of squares per-pixel, and can only describe the horizontal and vertical layouts. In contrast, the $max(\cdot)$ allows each pixel to choose *looking around or not* in each layer, after enough number of layers, they have many different sizes of squares. Therefore, the different sizes of squares can approximate (b) the shape of the oblique line, and (c) the shape of an arc, which are more common natural object layouts.

Pixel-wise modeling capacity Our definition of funnel condition allows the network to generate spatial conditions in the non-linear activations for every pixel. The network conducts non-linear transformations and creates spatial dependencies *simultaneously*. This is different from common practice which creates spatial dependency in the convolution layer and conducts non-linear transformations separately. In that case, the activations do not depend on spatial conditions explicitly; in our case, with the funnel condition, they do.

As a result, the pixel-wise condition makes the network has a pixel-wise modeling capacity, the function $max(\cdot)$ gives per-pixel a choice between *looking at the spatial context or not*. Formally, consider a network $\{F_1, F_2, ..., F_n\}$ with n FReLU layers, each FReLU layer F_i has a $k \times k$ parametric window. For brevity, we only analyze the FReLU layers regardless of the convolution layers. Because the max selection between 1×1 and $k \times k$, each pixel after F_1 has a $activate\ filed$ set $\{1, 1 + r\}$ $(r = k - 1)$. After the F_n layer, the set becomes $\{1, 1 + r, 1 + 2r, ..., 1 + nr\}$, which gives more choices to each pixel and can approximate any layouts if n is sufficiently large. With many distinct sizes of the activate field, the distinct sizes of squares can approximate the shape of the oblique line and arc (see Fig. 3). As we know, the layout of the objects in the images are usually not horizontal or vertical, they are usually in the shape of the oblique line or arc, therefore extracting the spatial structure of objects can be addressed naturally by the pixel-wise modeling capacity provided by the spatial condition. We show by experiments that it captures **irregular and detailed object layouts** better in complex tasks (see Fig. 4).

3.1 Implementation Details

Our proposed change is simple: we avoid the hand-designed condition in activations, we use a simple and effective spatial 2D condition to replace it. The visual activation leads to significant improvements as shown in Fig. 1. We first change the ReLU activations in the classification task on the ImageNet dataset.

We use ResNet [15] as the classification network and use the pre-trained network as backbones for other tasks: object detection and semantic segmentation.

All the regions $x^{\omega}_{c,i,j}$ in the same channel share the same coefficient p^{ω}_{c}, therefore, it only adds a slight additional number of parameters. The region represented by $x^{\omega}_{c,i,j}$ is a sliding window, the size is default set to a 3×3 square, and we set the 2-D padding to be 1, in this case,

$$x^{\omega}_{c,i,j} \cdot p^{\omega}_{c} = \sum_{i-1 \le h \le i+1, j-1 \le w \le j+1} x_{c,h,w} \cdot p_{c,h,w} \tag{3}$$

Parameter initialization We use the gaussian initialization to initialize the hyper-parameters. Therefore we get the condition values close to zero, which does not change the origin network's property too much. We also investigate the cases without parameters, (e.g. max pooling, average pooling), which do not show improvement. That shows the importance of the additional parameters.

Parameter computation We assume there is a $K'_h \times K'_w$ convolution with the input feature size of $C \times H \times W$ input, and the output size of $C \times H' \times W'$, then we compute the number of parameters to be $CCK'_hK'_w$, and the FLOPs (floating point operations) to be $CCK'_hK'_wHW$. To this we add our funnel condition with window $K_h \times K_w$, the additional number of parameters is CK_hK_w, and the additional number of FLOPs is CK_hK_wHW. We assume $K = K_h = K_w$,$K' = K'_h = K'_w$ for simplification.

Therefore the original complexity of parameters is $O(C^2K'^2)$, after adopting FReLU, it becomes $O(C^2K'^2 + CK^2)$); and the original complexity of FLOPs is $O(C^2K'^2HW)$, after adopting the visual activation, it becomes $O(C^2K'^2HW + CK^2HW)$. Usually, C is much larger than K and K', therefore the additional complexity can be negligible. Actually in practice the additional part is negligible (more details in Table 1). Moreover, the funnel condition is a $k_h \times k_w$ sliding window, and we implement it using the highly optimized depth-wise separable convolution operator followed with a BN [21] layer.

4 Experiments

4.1 Image Classification

To evaluate the effectiveness of our visual activation, first, we conduct our experiments on ImageNet 2012 classification dataset[9,37], which comprises 1.28 million training images and 50K validation images.

Our visual activation is easy to adopt on the network structures, by simply changing the ReLU in the original CNN structure. First, we evaluate the activation on different sizes of ResNet [15]. For the network structure, we use the original implementation. Spatial dependency is important especially in the shallow layers, for the small 224×224 input size, we replace the ReLUs in all the stages except the last stage, which has a small 7×7 feature map size. For the

Table 1. Comparisons with other effective activations [14,36] on ResNets [15] in ImageNet 2012. Image size 224x224. Single crop. We evaluate the Top-1 error rate on the test set.

Model	Activation	#Params	FLOPs	Top-1 Err.
ResNet-50	ReLU	25.5M	3.86G	24.0
	PReLU	25.5M	3.86G	23.7
	Swish	25.5M	3.86G	23.5
	FReLU	25.5M	3.87G	**22.4**
ResNet-101	ReLU	44.4M	7.6G	22.8
	PReLU	44.4M	7.6G	22.7
	Swish	44.4M	7.6G	22.7
	FReLU	44.5M	7.6G	**22.1**

training settings, we use a batch size of 256, 600k iterations, a learning rate of 0.1 with linear decay schedule, a weight decay of 1e-4, and a dropout [17] rate of 0.1. We present the Top-1 error rate on the validation set. For a fair comparison, we run all the results on the same code base.

Comparisons with scalar activations We conduct a comprehensive comparison on ResNets [15] with different depths (e.g. ResNet-50, ResNet-101). We take ReLU as the baseline and take one of its variants PReLU for comparison. Further, we compare our visual activation with the activation Swish [36] searched by the NAS [52,53] technique. Swish has shown its positive influence on various model structures, comparing with many scalar activations.

Table 1 shows the comparison, our visual activation still outperforms all of them with a negligible additional complexity. Our visual activation improves 1.6% and 0.7% top-1 accuracy rates on ResNet-50 and ResNet-101. It's remarkable that with the increase of model size and model depth, other scalar activations show limited improvement, while visual activation still has significant improvement. For example, Swish and PReLU improve the accuracy of 0.1% on ResNet-101, while visual activation increases still significantly on ResNet-101 with an improvement of 0.7%.

Comparison on light-weight CNNs Besides deep CNNs, we compare the visual activation with other effective activations on recent light-weight CNNs such as MobileNets [19] and ShuffleNets [30]. We use the same training settings in [30]. The model sizes are extremely small, we use a window size of $1\times3+3\times1$ to reduce the additional parameters. Moreover, for MobileNet we slightly refine the width multiplier from 0.75 to 0.73 to maintain the model complexity. Table 2 shows the comparison results on ImageNet dataset. Our visual activation also boosts accuracy on light-weight CNNs. ShuffleNetV2 $0.5\times$ can improve 2.5% top-1 accuracy by only adding a slight additional FLOPs.

Table 2. Comparisons among other effective activations [14,36] on light-weight CNNs (MobileNet [19], ShuffleNetV2 [30]) in ImageNet 2012. Image size 224x224. Single crop. We evaluate the Top-1 error rate on the test set.

Model	Activation	#Params	FLOPs	Top-1 Err.
MobileNet 0.75	ReLU	2.5M	325M	29.8
	PReLU	2.5M	325M	29.6
	Swish	2.5M	325M	28.9
	FReLU	2.5M	328M	**28.5**
ShuffleNetV2	ReLU	1.4M	41M	39.6
	PReLU	1.4M	41M	39.1
	Swish	1.4M	41M	38.7
	FReLU	1.4M	45M	**37.1**

4.2 Object Detection

To evaluate the generalization performance of visual activation on different tasks, we conduct object detection experiments on COCO dataset [28]. The COCO dataset has 80 object categories. We use the *trainval*35*k* set for training and use the *minival* set for testing.

Table 3. Comparisons of different activations in COCO **object detection**. We use ResNet-50 [15] and ShuffleNetV2 (1.5×) [30] with different activations as the pre-trained backbones. We use the RetinaNet [27] detector.

Model	Activation	#Params	FLOPs	mAP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
ResNet-50	ReLU	25.5M	3.86G	35.2	53.7	37.5	18.8	39.7	48.8
	Swish	25.5M	3.86G	35.8	54.1	38.7	18.6	40.0	49.4
	FReLU	25.5M	3.87G	**36.6**	**55.2**	**39.0**	**19.2**	**40.8**	**51.9**
ShuffleNetV2	ReLU	3.5M	299M	31.7	49.4	33.7	15.3	35.1	45.2
	Swish	3.5M	299M	32.0	49.9	34.0	16.2	35.2	45.2
	FReLU	3.7M	318M	**32.8**	**50.9**	**34.8**	**17.0**	**36.2**	**46.8**

We present the result on RetinaNet [27] detector. For a fair comparison, we train all the models in the same code base with the same settings. We use a batch size of 2, a weight decay of 1e-4 and a momentum of 0.9. We use anchors for 3 scales and 3 aspect ratios and use a 600-pixel train and test image scale. For the backbone, we use the pre-trained model in Section 4.1 as a feature extractor, and compare the generality among different activations.

Table 3 shows the comparison among different activations. The comparison shows that our visual activation increases 1.4% mAP comparing to the ReLU backbone, and increases 0.8% mAP comparing to the Swish backbone. It is worth mentioning that, on all the small, medium, and large objects, FReLU outperforms all the other counterparts significantly.

Table 4. Comparisons on the **semantic segmentation** task in CityScape dataset. We use the PSPNet [48] as the the framework and use the ResNet-50 [15] as backbone. The pre-trained backbones are from Table 1.

	ReLU	Swish[36]	FReLU
mean_IU	77.2	77.5	**78.9**
road	98.0	98.1	98.1
sidewalk	84.2	85.0	84.7
building	92.3	92.5	92.7
wall	55.0	56.3	59.5
fence	59.0	59.6	60.9
pole	63.3	63.6	64.3
traffic light	71.4	72.1	72.2
traffic sign	79.0	80.0	79.9
vegetation	92.4	92.7	92.8

	ReLU	Swish	FReLU
terrain	65.0	64.0	64.5
sky	94.7	94.9	94.8
person	82.1	83.1	83.2
rider	62.3	65.5	64.7
car	95.1	94.8	95.3
truck	77.7	70.1	79.8
bus	84.9	84.0	87.8
train	63.3	68.8	74.6
motorcycle	68.3	69.4	69.8
bicycle	78.2	78.4	78.7

We also show the comparison on the light-weight CNNs. As the comparison of ResNet-50, we use pre-trained ShuffleNetV2 backbones adopted with different activations. We mainly compare FReLU with ReLU and the effective activation Swish [36]. Table 3 shows visual activation also outperforms much better than ReLU and Swish backbones, to which it improves 1.1% mAP and 0.8% mAP respectively. Moreover, it increases the performance of all the sizes of objects.

4.3 Semantic Segmentation

We further present the semantic segmentation results on CityScape dataset [6]. The dataset is a semantic urban scene understanding dataset, contains 19 categories. It has 5,000 finely annotated images, 2,975 for training, 500 for validation and 1525 for testing.

We use the PSPNet [48] as the segmentation framework, for the training settings we use the poly learning rate policy [4] where the base is 0.01 and the power is 0.9, we use a weight decay of 1e-4, and 8 GPUs with a batch size of 2 on each GPU.

To evaluate the generality of the previous pre-trained models in Section 4.1, we use the pre-trained ResNet-50 [15] backbone models with different activations, we compare FReLU with Swish and ReLU respectively.

In Table 4, we show the comparison with scalar activations. From the result, we observe that our visual activation outperforms the ReLU and the searched Swish 1.7% and 1.4% mean_IU, respectively. Moreover, our visual activation has significant improvements in both large and small objects, especially on categories such as 'train', 'bus', 'wall', etc.

For better visualization of the improved performance, Fig. 4 shows the predict results on the testing dataset. It shows that by only changing the backbone activations, the results have obvious improvement. The boundaries of both the large and the small objects are well-segmented because the pixel-wise modeling

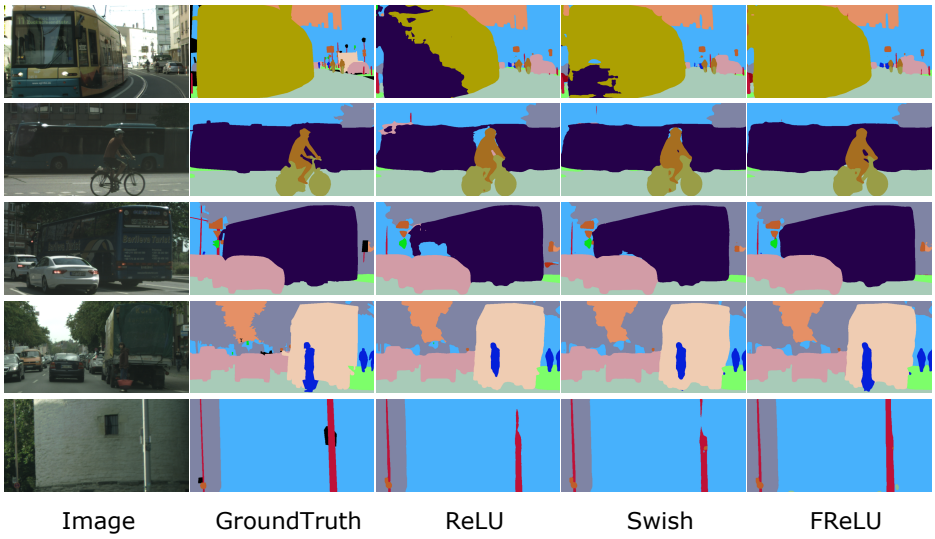

Fig. 4. Visualization of semantic segmentation on ResNet-50[15]-PSPNet[48] with different activations in backbone. We clip the CityScape images to make the differences more clear (better view enlarge images). FReLU has better long-range (large or slender objects) and short-range (small objects) understandings due to its better context capturing capacity. It captures irregular and detailed object layouts in complex cases much better. We note that modern frameworks are finely optimized with ReLU, however, it has obvious improvements by only changing the backbones, thus having the potential for further gains if redesign the frameworks for the visual activation.

capacity can handle both global and detailed regions (see Fig. 3). We note that the modern recognition frameworks are finely designed with the ReLU activation, therefore the visual activation still has great potential for further improving the results, which is beyond the focus of this work.

5 Discussion

The previous sections demonstrate the optimum performance comparing with other effective activations. To further investigate our visual activation, we conduct ablation studies. We first discuss the properties of the visual activation, then we discuss the compatibility with existing methods.

5.1 Properties

Our funnel activation mainly has two components: 1) funnel condition, and 2) $max(\cdot)$ non-linearity. Separately, we investigate the effect of each component.

Table 5. Ablation on the different **spatial condition manners**, and the different **non-linear manners**. The experiments are conducted on ResNet-50 [15]. Model A, B, C compare different visual conditions with/without parameters. Model D replaces *max* with *sum*, to this we add a ReLU, or it will not converge. Model E separates and evaluates the performance of the spatial condition itself. DW(x) represent the 3x3 depth-wise separable convolution.

Model	Activation	Top-1 Err.
A	Max(x, ParamPool(x))	**22.4**
B	Max(x, MaxPool(x))	24.4
C	Max(x, AvgPool(x))	24.5
A	Max(x, ParamPool(x))	**22.4**
D	Sum(x, ParamPool(x))	23.6
E	Max(DW(x), 0)	23.7

Table 6. Ablation on different normalization methods after the spatial condition layer. We adopt Batch Normalization (BN) [21], Layer Normalization (LN) [2], Instance Normalization (IN) [43] and Group Normalization (GN) [45] after the spatial condition layer which is implemented by depth-wise convolution. ImageNet results on ShuffleNetV2 0.5×.

Normalization	Top-1 Err.
-	37.6
BN	37.1
LN	36.5
IN	38.0
GN	36.5

Ablation on the spatial condition First, we compare the different manners of the spatial condition. Besides the manner of parametric pooling that we used, to investigate the importance of the additional parameters, we compare other pooling manners without additional parameters, they are max pooling and average pooling. We simply replace the parametric pooling with the other two non-parametric manners and evaluate the results on the ImageNet dataset.

Table 5 (A, B, C) shows the importance of the parametric pooling. Without additional parameter, the results decrease more than 2% top-1 accuracy, even perform worse than the baseline that does not use spatial condition. Table 6 shows the comparison of different normalization after the spatial condition.

Ablation on the non-linearity Second, we also compare the use of non-linearity. In our method, we use the $max(\cdot)$ function to perform the non-linearity, *simultaneously* capturing visual dependency. In contrast, we compare with the manners that *separately* capture visual dependency and non-linearity.

For the spatial context capturing, we use two manners: 1) use the parametric pooling as before, then linearly add up with the original feature, 2) simply add a depth-wise separable convolution layer. For the non-linear transformation, we use the ReLU function. Table 5 (A,D,E) show the results. Comparing with the baseline, the spatial context itself improves about 0.3% accuracy, but together as the non-linear condition in our method, it further increases more than 1%. Therefore, performing the spatial dependency and non-linearity *separately* has not an ideal effect as doing them *simultaneously*.

Ablation on the window size In the parametric pooling window, the size of the window decides the size of the area each pixel *looks*. We simply change the window size in the funnel condition and compare different sizes among $\{1 \times$

Table 7. Ablation on the **window size**. We simply change the window size in the funnel condition. We evaluate the top-1 error rate on ImageNet dataset using the ResNet-50 [15] structure.

Model	Window size	Top-1 Err.
A	1×1	23.7
B	3×3	**22.4**
C	5×5	22.9
D	7×7	23.0
E	Sum(1×3,3×1)	22.6
F	Max(1×3, 3×1)	**22.4**

Table 8. Ablation on **different layers**. We replace the ReLU with FReLU after the 1×1 convolution and the 3×3 convolution. Results are performed on ResNet-50 [15] and MobileNet [19].

	1×1 conv.	3×3 conv.	Top-1 Err.
ResNet-50	✓		22.9
		✓	23.0
	✓	✓	**22.4**
MobileNet	✓		29.2
		✓	29.0
	✓	✓	**28.5**

Table 9. Ablation of visual activation on **different stages** (Stage {2-4} in ResNet-50 [15]). In each stage we replace each ReLU with our visual activation. The results are the top-1 error rates on ImageNet. Image size 224x224.

Stage 2	Stage 3	Stage 4	Top-1 Err.
✓			23.1
	✓		23.0
		✓	23.3
✓	✓		22.8
	✓	✓	23.0
✓	✓	✓	**22.4**

Table 10. Ablation comparisons of the compatibility between FReLU and SENet [20] on ResNet-50 [15]. The results are the top-1 error rates on ImageNet. Image size 224x224. Single crop.

Model	#Params	FLOPs	Top-1
ReLU	25.5M	3.9G	24.0
FReLU	25.5M	3.9G	22.4
ReLU+SE	26.7M	3.9G	22.8
FReLU+SE	26.7M	3.9G	**22.1**

$1, 3\times3, 5\times5, 7\times7\}$. The case of 1×1 does not have the spatial condition and it is the case of PReLU since the parameter value is smaller than 1. Table 7 shows the comparison results. We conclude that 3×3 is the best choice. The larger window sizes also show benefits but do not outperform 3×3.

Further, we consider the case using an irregular window instead of squares. We use multiple windows of sizes 1×3 and 3×1, we consider to use the sum and max of them as the condition. Table 7 {B,E,F} show the comparison. The results show that irregular window sizes also have the optimum performance since they have a more flexible pixel-wise modeling capacity (Fig. 3).

5.2 Compatibility with Existing Methods

To adopt the new activation into the convolutional networks, we have to choose which layers, and which stages to adopt. Moreover, we also investigate the compatibility with existing effective approaches such as SENet.

Compatibility with different convolution layers First, we compare the positions after different convolution layers. That is, we investigate the effect of FReLU in different positions after 1×1 and 3×3 convolutions. We conduct

experiments on ResNet-50 [15] and ShuffleNetV2 [30]. We replace the ReLU after the 1×1 convolution and the 3×3 convolution and observe the improvement. Table 8 shows the results, in the bottleneck of the above two networks. From the results, we can see that the improvements on different layers are comparable, and it has the optimum performance when adopting both of them.

Compatibility with different stages Secondly, we investigate the compatibility with different stages in the CNN structures. The visual activations are important especially on the layer with high spatial dimensions. For the classification network whose shallow layers have larger spatial dimensions and deeper layers have large channel dimensions, there may be differences when we apply visual activations on different stages. For Stage 5 of ResNet-50 with 224x224 input, it has a relatively small 7x7 feature size, which mainly contains channel dependency instead of spatial dependency. Therefore, we adopt visual activations on Stage {2-4} on ResNet-50, as Table 9 shows. The results reveal that adopting the shallow layers has a larger effect, while a deeper layer has a smaller effect. Moreover, adopting FReLU on all of them has the optimum top-1 accuracy.

Compatibility with SENet At last, we compare the performance with SENet [20] and show the compatibility with it. Without the complex advances in CNN architecture, it achieves significant improvements on all the three vision tasks, simply together with the regular convolution layers. We further compare visual activation with recent effective attention module SENet, since SENet is one of the most effective attention modules recently.

Table 10 shows the result, although SENet uses an additional block to enhance the model capacity, it is remarkable that the simple visual activation even outperforms SENet. We also wish the visual activation we proposed can co-exist with other techniques, such as the SE module. We adopt the SE module on the last stage in ResNet-50 to avoid overfitting. Table 10 also shows the co-existence between FReLU and SE module. Together with SENet, funnel activation improves 0.3% accuracy further.

6 Conclusions

In this work, we present a funnel activation specifically designed visual tasks, which easily captures complex layouts using the pixel-wise modeling capacity. Our approach is simple, effective, and finely compatible with other techniques, that provides a new alternative activation for image recognition tasks. We note that ReLU has been so influential that many state-of-the-art architectures have been designed for it, however, their settings may not be optimal for the funnel activation. Therefore, it still has a large potential for further improvements.

Acknowledgements This work is supported by The National Key Research and Development Program of China (No. 2017YFA0700800) and Beijing Academy of Artificial Intelligence (BAAI).

References

1. Agostinelli, F., Hoffman, M., Sadowski, P., Baldi, P.: Learning activation functions to improve deep neural networks. arXiv preprint arXiv:1412.6830 (2014)
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
3. Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H.: Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 0–0 (2019)
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence **40**(4), 834–848 (2017)
5. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289 (2015)
6. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
7. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 764–773 (2017)
8. Dauphin, Y.N., Fan, A., Auli, M., Grangier, D.: Language modeling with gated convolutional networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 933–941. JMLR. org (2017)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
10. Elfwing, S., Uchibe, E., Doya, K.: Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Neural Networks **107**, 3–11 (2018)
11. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. pp. 249–256 (2010)
12. Goodfellow, I.J., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y.: Maxout networks. arXiv preprint arXiv:1302.4389 (2013)
13. Hahnloser, R.H., Sarpeshkar, R., Mahowald, M.A., Douglas, R.J., Seung, H.S.: Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature **405**(6789), 947–951 (2000)
14. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
16. Hendrycks, D., Gimpel, K.: Bridging nonlinearities and stochastic regularizers with gaussian error linear units (2016)
17. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)

18. Holschneider, M., Kronland-Martinet, R., Morlet, J., Tchamitchian, P.: A real-time algorithm for signal analysis with the help of the wavelet transform. In: Wavelets, pp. 286–297. Springer (1990)
19. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
20. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
21. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
22. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: Advances in neural information processing systems. pp. 2017–2025 (2015)
23. Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y.: What is the best multi-stage architecture for object recognition? In: 2009 IEEE 12th international conference on computer vision. pp. 2146–2153. IEEE (2009)
24. Jeon, Y., Kim, J.: Active convolution: Learning the shape of convolution for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4201–4209 (2017)
25. Klambauer, G., Unterthiner, T., Mayr, A., Hochreiter, S.: Self-normalizing neural networks. In: Advances in neural information processing systems. pp. 971–980 (2017)
26. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
27. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
28. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
29. Luo, W., Li, Y., Urtasun, R., Zemel, R.: Understanding the effective receptive field in deep convolutional neural networks. In: Advances in neural information processing systems. pp. 4898–4906 (2016)
30. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 116–131 (2018)
31. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: Proc. icml. vol. 30, p. 3 (2013)
32. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10). pp. 807–814 (2010)
33. Noh, H., Araujo, A., Sim, J., Weyand, T., Han, B.: Large-scale image retrieval with attentive deep local features. In: Proceedings of the IEEE international conference on computer vision. pp. 3456–3465 (2017)
34. Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al.: Conditional image generation with pixelcnn decoders. In: Advances in neural information processing systems. pp. 4790–4798 (2016)
35. Qiu, S., Xu, X., Cai, B.: Frelu: Flexible rectified linear units for improving convolutional neural networks. In: 2018 24th International Conference on Pattern Recognition (ICPR). pp. 1223–1228. IEEE (2018)

36. Ramachandran, P., Zoph, B., Le, Q.V.: Searching for activation functions. arXiv preprint arXiv:1710.05941 (2017)
37. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
38. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
39. Singh, S., Krishnan, S.: Filter response normalization layer: Eliminating batch dependence in the training of deep neural networks. arXiv preprint arXiv:1911.09737 (2019)
40. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-first AAAI conference on artificial intelligence (2017)
41. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
42. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
43. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
44. Wu, Y., Zhang, S., Zhang, Y., Bengio, Y., Salakhutdinov, R.R.: On multiplicative integration with recurrent neural networks. In: Advances in neural information processing systems. pp. 2856–2864 (2016)
45. Wu, Y., He, K.: Group normalization. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018)
46. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853 (2015)
47. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)
48. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2881–2890 (2017)
49. Zhao, H., Zhang, Y., Liu, S., Shi, J., Change Loy, C., Lin, D., Jia, J.: Psanet: Point-wise spatial attention network for scene parsing. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 267–283 (2018)
50. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Object detectors emerge in deep scene cnns. arXiv preprint arXiv:1412.6856 (2014)
51. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9308–9316 (2019)
52. Zoph, B., Le, Q.V.: Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578 (2016)
53. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8697–8710 (2018)