Unconstrained Control of Feature Map Size Using Non-integer Strided Sampling

Donggyu Joo
jdg105@kaist.ac.kr
Junho Yim
junho.yim@kaist.ac.kr
Junmo Kim
junmo.kim@kaist.ac.kr

School of Electrical Engineering
Korea Advanced Institute of Science
and Technology (KAIST),
South Korea

Abstract

Convolutional neural network (CNN) is mainly composed of convolution, pooling, and non-linear activation layers. Nowadays, almost all networks use only 2×2 max pooling or convolution layers with stride of 2 for downsampling. This technique is known to be good at extracting good feature, but it also has the constraint that feature map size is always reduced dramatically to half. In this work, we propose a simple new sampling technique that we call non-integer strided sampling (NSS), which enables free feature map size change so that it is not always reduced to half. Using this NSS layer, we design a new type of network architecture, *GradualNet*, which makes the feature map size change more smoothly than it is in existing networks.

Our results show that NSS can improve the performance of networks without having more parameters. Moreover, we propose other interesting possibilities for a CNN architecture based on the NSS layer. This result reveals that previous networks have been stuck in a stereotype, and this could be an important discovery in CNN architecture that has the potential to resolve this stereotype.

1 Introduction

During the development of convolutional neural network (CNN) [3, 4, 9, 11, 17, 18], the basic form of CNN architecture underwent many changes in various parts, such as depth, or filter size. However, the downsampling method was not changed. Already downsampling with stride of 2 [2, 3, 4, 19] has become a stereotype of CNN. We only use 2×2 max pooling [6, 7, 10, 16, 22] or convolution with stride of 2 [2, 3, 4, 20, 21] to reduce feature map sizes. However, it is just one way of reducing the feature map size, and it is not necessarily the best way to reduce the feature map size by half every time. Thus, we need to perform more extensive investigations on how to downsample or upsample effectively in the CNN.

When we use convolution, the information we can obtain from 30×30 and 15×15 feature maps is quite different. In a 30×30 feature map, the 3×3 convolution filter sees only $1/10$th of the image at a time (Figure 2, left). However, if it is applied to a 15×15 feature map, then the filter sees $1/5$th of the image each time (Figure 2, right). This indicates that the characteristic and quality of information we can extract by convolution significantly depends

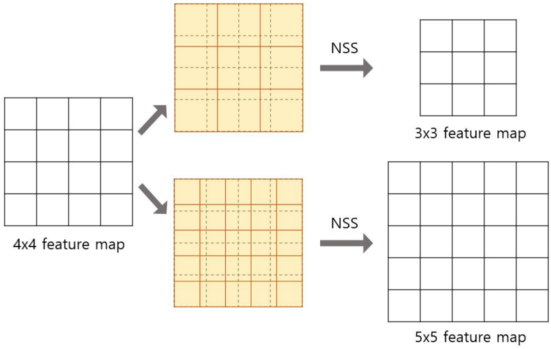


Figure 1: Simple description of the proposed non-integer strided sampling (NSS). In this figure, a 4×4 feature map is transformed to a 3×3 (upper) or 5×5 (lower) feature map by averaging the corresponding values. This can be applied to any middle feature map. The transformed feature map size is not restricted; it can be decreased or increased to any size.

on the spatial size of the feature maps to which the convolution is applied. Therefore, if convolutions can be applied to feature maps with a larger set of candidate spatial resolutions, we can potentially increase the set of the candidate information that can be extracted by convolutions. However, existing CNN typically exhibits only a certain size of feature maps as inputs for the convolution layer. In case of the CIFAR [8] dataset, typical networks only observe three levels of feature map sizes: 32, 16, and 8 [2, 3, 4, 20, 21]. It is missing the opportunity to obtain additional types of information that can come from more extensive set of candidate feature map sizes. For example, we can obtain more information at the first downsampling stage if we use feature map sizes with a finer granularity, such as 20 and 24, instead of using only 16.

We propose a non-integer strided sampling (NSS). It is a simple new idea that feature map size can be gradually reduced or increased in a network. As illustrated in Figure 1, when the 4×4 feature map is given, we can generate both a 3×3 feature map and a 5×5 feature map freely, depending on user's choice. The strength of our method mainly comes from two points. First, it does not need any additional learnable parameters. Second, feature map size can be freely changed. The proposed method can be used for both downsampling and upsampling, and it can induce a gradual change in feature map size throughout the network. Our method can be applied to any form of network. Therefore, ResNet [3, 4] and any other network can be transformed to have a gradually decreasing feature map size while preserving the number of parameters. We call this transformed network *GradualNet* or *GradNet*, which means that the feature map size is changed more gradually or smoothly than in existing networks. Therefore, GradNet is not a specifically defined type of network; rather, it is a generic term for networks that use NSS.

Our research makes the following contributions: **1.** We propose a novel technique, NSS, to change the feature map size freely without having more parameters. **2.** We suggest new types of network architecture to make feature map size change gradually throughout the network while existing networks experience radical changes. **3.** We show that it is not always necessary to reduce the feature map size in a classification task. By increasing the feature map size in the middle of the network, performance can be further improved. **4.** By applying the idea of a PyramidNet by Han et al. [2], we finally create a smooth network that changes both the feature map size and the number of channels gradually.

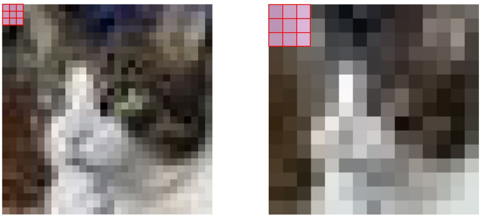


Figure 2: Same convolution filter which is applied to feature map of different sizes. It observes totally different information although it is the same convolution layers. **Left**: 30 × 30 feature map with the 3 × 3 convolution filter. **Right**: 15 × 15 feature map with the same 3× 3 convolution filter.

2 Related Work

Fractional max pooling (FMP) is the method proposed by Graham and Benjamin [1]. To the best of our knowledge, FMP [1] is the most famous technique that enables the free change of feature map size. FMP also claims the same idea with us that the feature map size could be changed freely. It aims to reduce the feature map size with a ratio between 1 and 2 by modifying the kernel size of max pooling. FMP uses 1×1, 1×2, 2×1, 2×2 max pooling randomly depending on the input and output feature map sizes. By using FMP layers, a new network is created that does not use downsampling with stride of 2. However, FMP tries to pick the largest values out of multiple regions with different sizes, so information is extracted in an uneven way. More importantly, FMP only works for downsampling with a ratio between 1 and 2 and is not applicable to upsampling.

Veit et al. [20] tried to verify the effect of removing an individual layer in the ResNet [3, 4]. As mentioned by these authors, removing individual layer of ResNet only causes a slight performance reduction compared with other plain networks like the VGG network [17]. However, even in the ResNet, removing the downsampling blocks causes a relatively high performance reduction because the layers double the number of channels at once. PyramidNet [2] addresses this phenomenon by increasing the number of channels gradually throughout the network and not abruptly. Therefore, the ensemble effect of PyramidNet is stronger than that of ResNet, and it is good at generalization. We think that this phenomenon is not only due to the abrupt change in the number of channels but also the sharp downsampling. Therefore, we focus on the feature map size, whereas PyramidNet focuses on the number of channels. We try to use both PyramidNet and GradNet in a combined way in this paper. Ultimately, we obtain a perfectly smooth form of network that is well balanced.

3 Effect of Downsampling

Before the main experiment, we conducted an experiment to compare ability of two networks using convolution with stride of 3 and stride of 2. This comparison will help us to understand the effect of the downsampling scales. In this section, the proposed NSS layer is not used.

Based on the 26-layer ResNet, we designed three networks to compare downsampling with stride of 2 and stride of 3 fairly for the CIFAR dataset [8]. The first network (**Net-A**) is modified to have three convolution layers with stride of 2 at 4th, 7th,and 10th residual modules. Then, this network has a 4×4 feature map at the last. Following the general setting,

Model	CIFAR-10	CIFAR-100	# of Parameters
Net-A (stride 2)	**92.77**	**69.55**	1.10M
Net-B (stride 3)	92.31	69.23	1.56M
Net-B’ (stride 3)	92.11	68.08	1.12M

Table 1: Top-1 accuracy on CIFAR dataset. This early experiment shows the effect of downsampling size. All three networks are based on the 26-layer ResNet. The first model uses stride of 2, and the other two models use stride of 3.

the number of channels starts from 16 and doubles when the feature map size reduces to half (# of channels: 16 → 32 → 64 → 128). Second network (**Net-B**) is designed to have two convolution layers with stride of 3 at the 5th and 9th residual modules. We only used two downsampling steps to make the last feature map size same as that of the Net-A. In the similar way, the number of channels is multiplied by 3 when the feature map size reduces to a third (# of channels: 16 → 48 → 144). However, this Net-B has more parameters than the Net-A. Therefore, we also created a **Net-B’** (# of channels: 16 → 44 → 120) that has a similar number of parameters as the Net-A while maintaining the architecture of the Net-B.

We used CIFAR classification datasets [8]. CIFAR contains 32×32 pixel color images consisting of 50k training images and 10k test images. CIFAR-10 is composed of 10 classes, and CIFAR-100 is composed of 100 classes. For those data, we followed the standard data augmentation method, as other studies have generally done for CIFAR dataset. The original data were padded with four pixels and randomly cropped for training; mirroring was also used. All the training settings follow the settings of the ResNet.

Table 1 shows the test accuracy of above three networks. (From now on, every results shown in this paper are average of three runs). Both the CIFAR-10 and CIFAR-100 results show that Net-A is better than both Net-B and Net-B’. Net-B shows lower performance than the Net-A although it has more parameters. This is because downsampling with stride of 3 lost a great deal of information at training stage compare to the stride of 2. This is why downsampling with stride of 3 or larger is rarely used at present. This result suggests that downsampling with smaller stride can be even better than the existing downsampling with stride of 2. If it is true that downsampling with stride of 3 is worse than downsampling with stride of 2, then downsampling with smaller stride can also be even better. The result of this section is consistent with the motivation of our study.

4 Method

4.1 Non-integer Strided Sampling

Our proposed algorithm is a simple technique that can change the feature map size without any constraints. We can understand this NSS as a simple average sampling with a non-integer stride. This stride can be larger than 1. In this case, output feature map is smaller than input feature map. The stride can also be smaller than 1. In this case, the output feature map is larger than the input feature map. The output layer of NSS is simply generated from the input layer with only one parameter, namely stride.

Consider one of the selected input layers $I \in \mathbb{R}^{h_1 \times w_1 \times m}$, where h_1, w_1, and m represent the height, width, and number of channels, respectively. Suppose the output layer of NSS is generated as $O \in \mathbb{R}^{h_2 \times w_2 \times m}$, where h_2, w_2, and m represent the height, width, and number of channels, respectively. It is clear that the input and output layers have the same number

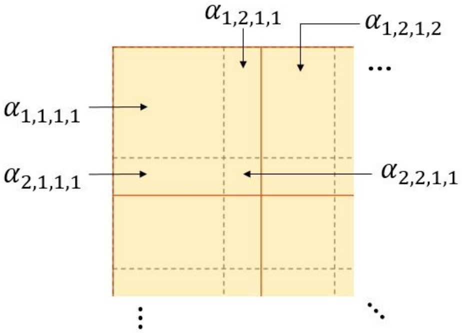


Figure 3: Magnified view of Figure 1 (upper) when the 4×4 feature map is transformed to the 3×3 feature map. Each $\alpha_{i,j,h,w}$ value indicates the area of $I_{i,j,c}$ that contributes to $O_{h,w,c}$.

of channels. Moreover, any natural numbers can be chosen for h_2 and w_2. Now, the stride of height becomes $s_h = \frac{h_1}{h_2}$, and the stride of width becomes $s_w = \frac{w_1}{w_2}$. To obtain the output feature map, each input feature map $I_c \in \mathbb{R}^{h_1 \times w_1}$ is equally divided into $h_2 \times w_2$ grid regions.

NSS operation is performed in each of $h_2 \times w_2$ regions, and the average is calculated depending on the area occupied by each pixel value. Mathematically, the NSS algorithm is described by the steps outlined below. For every $(h,w) \in \{1,\cdots,h_2\} \times \{1,\cdots,w_2\}$, $O_{h,w,c}$ is calculated from the following range of input feature map of the same channel number:

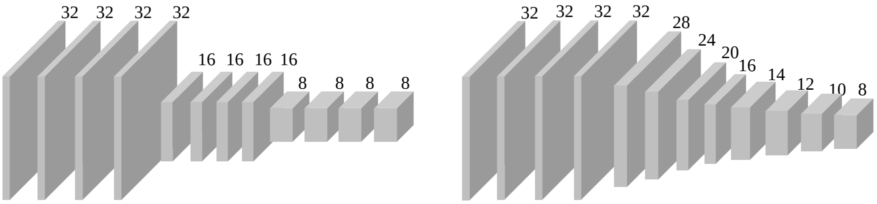
$$P_{h,w} = \{(i,j) \mid \lfloor s_h(h-1) \rfloor < i \leq \lceil s_h h \rceil \text{ and } \lfloor s_w(w-1) \rfloor < j \leq \lceil s_w w \rceil\}. \quad (1)$$

This is the expression of the corresponding region for each output point $O_{h,w,c}$ in the equally divided region of the input feature map. Let's see an example. Let $I \in \mathbb{R}^{4\times4\times C}$, and let $O \in \mathbb{R}^{3\times3\times C}$ like Figure 1 (upper). Then, for any $c \in \{1,\cdots,C\}$, equally divided region for $O_{2,2,c}$ in I becomes $P_{h,w} = \{2,3\} \times \{2,3\}$ following the eq.(1). From this specified region $P_{h,w}$, we can calculate the output feature map value. We obtain the following equation, which expresses the weighted average:

$$O_{h,w,c} = \frac{1}{s_h s_w} \sum_{(i,j)\in P_{h,w}} \alpha_{i,j,h,w} \cdot I_{i,j,c}, \quad (2)$$

where $\alpha_{i,j,h,w} \in (0,1]$ represents how much each $I_{i,j,c}$ is contained in the equally divided region for $O_{h,w,c}$. Simply, it is the weight of the $I_{i,j,c}$ pixel contributing to $O_{h,w,c}$. A simple diagram is illustrated in Figure 3 to explain the NSS algorithm.

The mathematical formulation is quite complex, but it is not difficult to understand conceptually, as it represents a simple averaging method. It calculates the weighted sum of corresponding areas and divide this by $s_h s_w$ to normalize the value. Therefore, if $s_h = 2, s_w = 2$, then it behaves like the existing 2×2 average pooling. Furthermore, if $s_h = h_1, s_w = w_1$, then it is the existing global average pooling. NSS includes several types of known sampling methods, but it is more general. One of the important thing here is that NSS does not need much computation since it just computes weighted averages of feature maps. NSS sampling steps have FLOPs about only 0.1% of overall computation.



(a) ResNet

(b) GradNet

Figure 4: Illustration of architecture of the (a) ResNet and (b) GradNet. Existing networks reduce feature map size dramatically by half. However, in GradNet, the feature map size is reduced slowly; therefore it can learn information from various sizes of feature maps.

4.2 Design of Network: GradNet

The strength of our method does not stem from the averaging algorithm; rather, it is from the architecture, which uses the middle size feature maps with smoother forms. We designed an architecture *GradNet* that reduces feature map size gradually. We distributed the downsampling stage over several steps in GradNet. A feature map of size 32 can be reduced to 24 and then 16 if we use a two-step distribution. Moreover, as stated above, there is no restriction, so other ways of making GradNet is possible. A simple version of GradNet architecture is demonstrated in Figure 4(b). The existing integer strided downsampling method (Figure 4(a)) is quite straightforward in several ways, but the feature map size of our network is reduced smoothly and looks more natural.

By using NSS, we can change any kind of network into a smoother form. The NSS layer can be applied to any position that does not make conflict with a predetermined feature map size. For example, we can apply the NSS layer to every layer in the VGG network [17] except the fully connected layers. In case of ResNet, it has an element-wise addition layer, so it can be applied only after the addition in ResNet. However, if we apply the NSS at the very bottom layer, then we can loss much information that can be extracted from high-resolution images. Therefore, from now on, NSS of GradNet always starts from the same position where the baseline network's first downsampling step occurs. In case of the 26-layer ResNet, it has 12 residual modules. Therefore, NSS can be applied to the 8 steps because 26-layer ResNet has first downsampling step after 4th residual module. At each step, the feature map size can be reduced as desired; therefore several types of GradNet can exist.

When there are n locations where NSS can be applied, we simply name each GradNet with n numbers in the order from the bottom to top, depending on how much the feature map size is reduced at each NSS layer. For the 26-layer ResNet described above, n is 8. We call these n numbers the *Sampling Parameter* of GradNet, $\mathbf{d} \in \mathbb{N}^n$. Therefore, Figure 4(b) can be labeled by $\mathbf{d} = 4\text{-}4\text{-}4\text{-}4\text{-}2\text{-}2\text{-}2\text{-}2$. Alternatively, let us express it as $\mathbf{d} = 4^4\text{-}2^4$ for the convenience when the same NSS steps are repeated. If there is a bar on the number, it implies upsampling. Therefore, $\bar{3}$ means that the feature map is increased by 3 using the NSS layer. We describe the results of an experiment that uses upsampling in Section 5.3.

5 Experiment

We conducted four experiments to verify the effectiveness of our proposed algorithm. First, with the same number of parameters, we show that our proposed architecture can improve the original networks a great deal by changing only the feature map sizes. Second, we compare our idea with the existing FMP algorithm [1]. Third, we create a classification network that has upsampling stages in the middle of the network. Finally, by applying the concept of PyramidNet [2] to our GradNet, we can additionally improve the network. Eventually, the network will no longer need to work in a step-like form.

5.1 Improvement of the baseline networks

We trained our GradNet and baseline ResNet on the CIFAR [8], SVHN [13] and ImageNet [15] datasets. We used a 38-layer ResNet with 18 residual modules. There are three baseline networks depending on how it reduces the feature map size. The first one is the original ResNet that uses convolution with stride of 2 to reduce feature map size. The second and third baselines use 2×2 max pooling and 2×2 average pooling, respectively. The baseline networks setting is same for the experiments in Section 5.2.

As stated above, GradNet can be generated from any kind of network. There are no rules, so any form of network architecture can be generated. In this work, we used three GradNets with the following characteristics: the first network downsamples equally over four times, which can be expressed as $\mathbf{d} = [6\text{-}0\text{-}0]^4$; the second downsamples equally over six times, which can be expressed as $\mathbf{d} = [4\text{-}0]^6$; and the third gradually changes every feature map through the networks, which is $\mathbf{d} = 2^{12}$. These three architectures change feature map sizes more gradually than existing networks.

Accuracy results are shown in Table 2. There are some interesting observations from these results. The proposed GradNet outperforms baseline networks by a large margin. Especially, GradNet $\mathbf{d} = [4\text{-}0]^6$ shows 5.22% higher performance than the original ResNet in C100. In addition, GradNet $\mathbf{d} = [6\text{-}0\text{-}0]^4$ shows 4.16% higher performance than ResNet in C10. This is a remarkable result, as we only changed the size of the feature map while using the same number of parameters. From these results, we can see that smoothly changing the feature map size using the NSS layer is helpful for the learning. In GradNet, each convolution layer can learn various types of information.

When data augmentation was applied, the difference becomes smaller. However, our GradNet still outperforms original networks in all the cases. This is a interesting result that the performance difference is related to the data augmentation. Data augmentation helps the network to observe various types of feature maps that are translated and mirrored. Therefore, the role of data augmentation and the proposed method overlap considerably in terms of enhancing the diversity of feature maps that the network can observe. Data augmentation can compensate for the abrupt change in feature map size induced by downsampling with stride of 2. Therefore, the additional benefit of the GradNet is less in this case compared to the case without data augmentation.

At first, we expected that the gradual change could be effective in learning knowledge at several scales. However, what we can observe is that the $\mathbf{d} = [6\text{-}0\text{-}0]^4$ and $\mathbf{d} = [4\text{-}0]^6$ networks show higher performance than the $\mathbf{d} = 2^{12}$ network. From this result, we realize that the network needs a reasonable amount of steps to learn certain information from the feature map of a specific size. Reducing the size of the feature map quickly before learning some intermediate-level information can become an obstacle for training. Therefore, to ex-

Model	Downsampling	C10	C10+	C100	C100+	SVHN
ResNet	Conv, stride 2	85.32	92.05	54.78	67.63	95.76
ResNet	2×2 Max pool	86.39	91.87	56.57	67.78	95.70
ResNet	2×2 Ave pool	87.02	92.16	58.18	68.10	95.94
GradNet	[6-0-0]^{4}	**88.45**	**92.73**	59.79	68.45	96.24
GradNet	[4-0]^{6}	88.44	92.41	**60.00**	**68.47**	**96.33**
GradNet	[2]^{12}	88.31	92.30	59.80	67.57	96.17

Table 2: Top-1 accuracy on CIFAR [8] and SVHN [13] datasets. We used the 38-layer ResNet as the baseline. The sampling param of GradNet denotes how much the feature map sizes are reduced at each of the 18 residual modules. C means CIFAR, and '+' sign represents standard data augmentation is used. In all the cases, GradNet outperforms original networks.

Model	Top-1	Top-5
ResNet	68.80	88.45
GradNet	**69.90**	**89.24**

Table 3: Top-1 and Top-5 accuracy on ImageNet [15] datasets. We used the 34-layer ResNet as the baseline. GradNet is made to distribute all the downsampling steps of baseline over two steps.

tract important knowledge effectively, it is necessary to have relaxation stages in the middle; following this, we can learn various types of information by reducing the feature map size gradually and smoothly.

ImageNet [15]: CIFAR [8], and SVHN [13] are small scale datasets that have small images. To show the robustness of our method, we've also compared our result on well-known large scale classification dataset *ImageNet*. The result is described in Table 3. In this experiment, random cropping with 224 × 224 and mirroring are used for the data augmentation. Even with data augmentation, we can observe significant improvement. As ImageNet has larger images, there is larger amount of decrease in feature map size when using usual 2x2 downsampling. So, NSS would be more helpful since GradNet makes it possible to observe feature maps of various intermediate sizes. For this experiment, GradNet is designed to distribute the downsampling of baseline networks over two steps.

Plain network: We've also conducted experiment on the plain network VGG [17] that has a different structure from ResNet. Experimental results on VGG [17] is described in Table 4. GradNet also outperforms original VGG network by a large margin on all the datasets. Our algorithm also works for plain network. For this experiment, GradNet is designed to distribute the downsampling of baseline networks over two steps.

From above several experiments, we confirm that our method is robust to various settings.

Model	C10	C10+	C100	C100+	SVHN
VGG	81.83	91.08	47.79	66.19	94.81
GradNet	84.80	91.81	53.32	68.63	**95.28**

Table 4: Top-1 accuracy on CIFAR and SVHN datasets. We used a plain network VGG instead of ResNet to show that our algorithm works well for general other network. GradNet is designed to distribute the downsampling of baseline networks over two steps

5.2 Comparison with FMP

FMP is the most well-known method in which it is possible to reduce the feature map size with ratio between 1 and 2. We compared our method with FMP under the same conditions. We used 38-layer ResNet as a baseline as above. Table 5 shows the classification result of two networks using FMP and NSS. Our GradNet shows better performance than FMP network in both CIFAR-10 and CIFAR-100.

Model	Downsampling	CIFAR-10	CIFAR-100
FMP	$[4\text{-}0]^6$	90.66	65.21
FMP	$[6\text{-}0\text{-}0]^4$	91.59	67.17
GradNet	$[4\text{-}0]^6$	92.41	**68.47**
GradNet	$[6\text{-}0\text{-}0]^4$	**92.73**	68.45

Table 5: Top-1 accuracy on CIFAR datasets with data augmentation. FMP and GradNet are compared with same network architecture and sampling parameters.

Moreover, we try to upsample middle feature maps in the network by using the NSS layer in Section 5.3. This is one thing that FMP cannot do because it is limited to downsampling with ratio between 1 and 2. FMP seems free to change the feature map size, but unlike our proposed method, it is not completely free.

5.3 Upsampling in the Classification Task

All the known classification networks only use downsampling steps. Upsampling or unpooling is mostly used for segmentation tasks [8, 12, 14]. As we stated above, NSS has another strength that upsampling is also possible to any size. This is why the name of our proposed idea is NSS, not 'non-integer strided downsampling.'

We designed several GradNet architectures that can tell us about the effect of upsampling in classification tasks. Because larger feature map size is difficult to train, usually complex network that has bigger capacity is required. So, we used WRN-28-4 which has four times as many channels than original ResNet. Using the WRN-28-4 as a baseline network, we designed two types of GradNet with upsampling. This is a new type of classification network. The result is shown in Table 6. The GradNet with $\mathbf{d} = \bar{4}^2\text{-}6^2\text{-}5^4$ outperforms any other networks for both datasets. This improvement of the performance is caused by the fact that it observes even more diverse set of feature maps. Looking at a feature map larger than size 32 can be helpful to learn a new kind of knowledge in the CNN. This is a very surprising result, and it opens various new potentials for upsampling in CNN architectures.

Model	Downsampling	CIFAR-10	CIFAR-100
WRN-28-4	Conv, stride 2	87.48	63.83
GradNet	$[8\text{-}0]^2\text{-}[4\text{-}0]^2$	91.64	68.34
GradNet	$4^4\text{-}2^4$	91.61	69.18
GradNet	$\bar{4}^4\text{-}10^4$	91.42	69.26
GradNet	$\bar{4}^2\text{-}6^2\text{-}5^4$	**91.89**	**69.84**

Table 6: Top-1 accuracy on CIFAR datasets without data augmentation. The bar on the sampling param of GradNet means that the size of the feature map is increased by that amount. For example, $\bar{4}$ means the feature map size is increased by 4

Model	CIFAR-10	CIFAR-100
ResNet	84.87	52.43
PyramidNet	85.54	54.18
GradNet	85.84	55.05
PyramGradNet	**86.39**	**55.69**

Table 7: Top-1 accuracy on CIFAR datasets without data augmentation. We used the 26-layer ResNet as the baseline. PyramidNet uses "*add*, α=48", GradNet uses sampling param of 2^{12} from the bottom, and PyramGradNet uses the both settings together.

5.4 Pyramidal Gradual Network

GradNet aims to have gradual change in the feature map size because dramatic change is not helpful for learning various types of information. However, this simple architecture is not completely smooth because its number of channels does not change gradually. By combining GradNet and Pyramidnet, we can create a completely smooth network. We designed a experiment that show the effect of combination of two networks. We call the combined version of the two networks as *PyramGradNet*.

The result is shown in Table 7. As we expected, GradNet and PyramidNet shows higher accuracy than ResNet, and PyramGradNet outperforms others. As a result, GradNet and PyramidNet assist each other, since each has own specific advantages, while they both aim to accomplish similar goals. Therefore, this result gives potentials of continuous form of CNN.

Although PyramidNet has similar flavor, our work is different from PyramidNet as we focus on different hyper parameter. PyramidNet is about the number of channels but with usual 2×2 downsampling. Our work controls spatial resolutions of the feature maps.

6 Conclusion

CNN has a stereotype that we can only reduce or increase the feature map size with an integer ratio. In this work, we designed a new sampling method that does not have any restrictions when changing the feature map size.

We conducted several experiments to show the advantages of our network, GradNet. We compared this with several other baselines, and there were surprising improvements with the same number of parameters. We also showed that it outperforms the existing method FMP. In addition, we create a new type of classification network that uses upsampling layers by employing NSS. Moreover, by applying the idea of PyramidNet, we achieved a more powerful, and continuous form of network. This is a novel observation and important step for solving various problems associated within the CNN structure.

Acknowledgements

This work was supported by the ICT R&D program of MSIP/IITP, [2016-0-00563, Research on Adaptive Machine Learning Technology Development for Intelligent Autonomous Digital Companion] and the Industrial Convergence Core Technology Development Program(No. 10063172) funded by MOTIE, Korea.

References

[1] Benjamin Graham. Fractional max-pooling. *arXiv preprint arXiv:1412.6071*, 2014.

[2] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016.

[5] Seunghoon Hong, Hyeonwoo Noh, and Bohyung Han. Decoupled deep neural network for semi-supervised semantic segmentation. In *Advances in Neural Information Processing Systems*, pages 1495–1503, 2015.

[6] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European Conference on Computer Vision*, pages 646–661. Springer, 2016.

[7] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[8] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[10] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. In *ICLR*, 2017.

[11] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[12] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[13] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5, 2011.

[14] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.

[15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[16] Pierre Sermanet, David Eigen, Xiang Zhang, MichaÃńl Mathieu, Robert Fergus, and Yann Lecun. *Overfeat: Integrated recognition, localization and detection using convolutional networks*. 2014.

[17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[18] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[19] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017.

[20] Andreas Veit, Michael J Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. In *Advances in Neural Information Processing Systems*, pages 550–558, 2016.

[21] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016.

[22] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.