

# Convolution layer with nonlinear kernel of square of subtraction for dark-direction-free recognition of images

Hyoseob Kim<sup>1</sup>, Hojun Yoo<sup>2</sup>, Jung Lyul Lee<sup>3</sup>, Seoungho Lee<sup>4</sup>

<sup>1,2,4</sup>Department of Civil and Environmental Engineering, Kookmin University, 77 Jeongneung-Ro, Seongbuk-Gu, Seoul, 02707, Korea

<sup>2</sup>GeoSystem Research Corporation, Gunpo-si, Gyeonggi-do, Korea

<sup>3</sup>Graduate School of Water Resources, Sungkyunkwan University, Suwon-si 16319, Korea

<sup>2</sup>Corresponding author

**E-mail:** <sup>1</sup>[hkim@kookmin.ac.kr](mailto:hkim@kookmin.ac.kr), <sup>2</sup>[yoojhj@geosr.com](mailto:yoojhj@geosr.com), <sup>3</sup>[jlllee6359@hanmail.net](mailto:jlllee6359@hanmail.net), <sup>4</sup>[stock@kookmin.ac.kr](mailto:stock@kookmin.ac.kr)

Received 19 June 2020; received in revised form 19 July 2020; accepted 31 July 2020  
DOI <https://doi.org/10.21595/mme.2020.21552>



Copyright © 2020 Hyoseob Kim, et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Abstract.** A nonlinear kernel with a bias is proposed here in the convolutional neural network. Negative square of subtraction between input image pixel numbers and the kernel coefficients are convolved to conform new feature map through the convolution layer in convolutional neural network. The operation is nonlinear from the input pixel point of view, as well as from the kernel weight coefficient point of view. Maximum-pooling may follow the feature map, and the results are finally fully connected to the output nodes of the network. While using gradient descent method to train relevant coefficients and biases, the gradient of the square of subtraction term appears in the whole gradient over each kernel coefficient. The new subtraction kernel is applied to two sets of images, and shows better performance than the existing linear convolution kernel. Each coefficient of the nonlinear subtraction kernel has quite image-equivalent meaning on top of pure mathematical number. The subtraction kernel works equally for a given black and white image set and its reversed version or for a given gray image set and its reversed version. This attribute becomes important when patterns are mixed with light color and dark color, or mixed with background color, and still both sides are equally important.

**Keywords:** nonlinear kernel, convolutional neural network (CNN), subtraction neural network (SNN), dark-direction-free.

## 1. Introduction

The Convolutional Neural Network (CNN) using linear kernel convolution, a partial group of the artificial neural network, has been widely used for classification problems as well as detection [1], recognition [2], or segmentation problems from images [3-6]. Various sophisticated branches of the linear kernel CNN have appeared through deepening, modification of architecture, or combining pooling since its debut.

Nonlinearity of the CNN could be reviewed from both visual neural recognition and mathematical point of views. Inherent nonlinearity of visual recognition has been reported in physiology, which is related to the reception level [7, 8]. New trials also include involving nonlinear kernel in the inception level in neural networks. Typical feedforward networks include nonlinearities with the activation function in the convolution layer, and pooling method in pooling layer. Mathematical nonlinearity exists in the existing linear kernel neural network system through nonlinear activation function. We often want the final classification answers as 1 or 0, resembling binary system, and thus nonlinear transformation of interim information into numbers between 1 and 0 is carried out by using nonlinear activation function. Treatment of the nonlinearity with the nonlinear activation function is distinguished from that within the receptive field, and is not dealt with here.

Various activation functions are used equation such as ReLU function [9], LReLU function [10], PReLU function [11], maxout and Probout function [12]. Among them, the activation function ReLU has become popular [9, 13]. Data argumentation was used to reduce overfitting

[14, 15]. Inherent nonlinearity in classification problem has often used not only common activation functions like sigmoid or hyperbolic tangent function, but also other nonlinear function.

Pooling is selectively used during forward pass, which could be thought as another form of nonlinear operation, probably to widen acceptable allowance of small modification of patterns. Rarely, minimum pooling, max-min pooling have been used for specific purposes [16]. Max pooling was added to DCNN [17, 18]. Max pooling, average pooling have commonly followed the convolutional layer, but with some problems. Alternatives were proposed, including Lp pooling [19], stochastic and fractional max pooling [20], and mixed pooling [21].

Other trials exist to incorporate nonlinearity except the activation function or pooling. A group of researchers used a nonlinear kernel residual network in their work to classify handwritten character recognition, which may be a specific branch of the convolutional neural networks [22]. They focused to skip some levels and used the rectified linear function in their residual function. The object of the process includes retaining gradients of previous steps, and its performance is under investigation.

Lin et al. [23] proposed “network in network” method by using multilayered perceptions, and argued its strong points to classify patterns with wide-range sizes, but their trial could be classified in inserting nonlinearity at activation function level. Zhai et al. [24] proposed double convolution operation to reduce space requirements.

Despite above several approaches within various levels in the neural network to improve accuracy is classification problem by adopting nonlinear pooling process, activation function architectural shape, the inception point of the convolution, which is the most important part of the whole network has remained linear except [25].

Zoumpourlis et al. [25] proposed a nonlinear convolution kernel as a quadratic form using the Volterra kernels within the inception level to resemble the visual recognition nonlinearity [26]. Their kernel involves multiplication of two input variables and augmented number of weight coefficients. Their quadratic functions are applied to the input pixel numbers, and computation became very heavy.

One of the important objectives of a neural network for pattern recognition is to classify or identify an image, as yes or no answers. The neural network should have reasonably accurate coefficients enough to distinguish a new image, and training process often makes use of multiple number of images. Basic concept should be if a new image resembles one or many other images used for training, then the new image is classified into the yes group. It has been believed that convolution of the kernel and the input pixel numbers can represent the resemblance degree between images.

From the logical point of view defining resemblance is not straightforward. A user should have clear intention whether he/she would allow translation, rotation, scaling, symmetry, color reverse, background inclusion, multiplicity, inclination due to three-dimensional viewing, or minor shape deformation. User’s intention affects logical and following mathematical formation of kernel or the neural network architecture.

Essential features of the convolutional approach are first using small zone convolutional kernels. A kernel confines a small patch within a wide two-dimensional plane image, evaluate the weights of a local pattern within the kernel size. Small-sized kernels are expected to distinguish dominant local pattern by being applied on to moving patches of interest in two-dimensional way, which cannot be achieved by pure one-dimensional arrangement of image information.

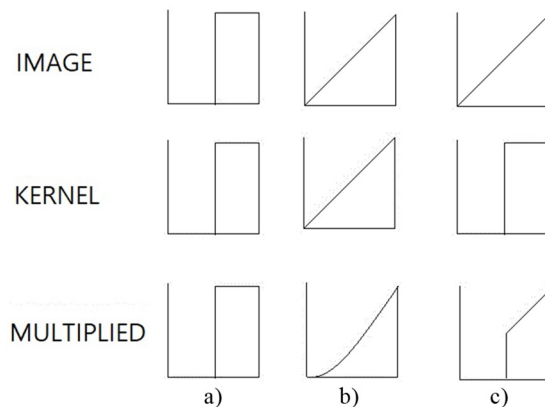
A convolutional kernel has two-dimensional array coefficients, each of them are then multiplied to the color numbers of image pixel matrix. These could be either 0 to 1 numbers for black and white images. Strictly speaking arbitrary two different numbers could be assigned to black and white colors, respectively. 0 to 255 8-bit gray scale numbers could be assigned to a gray image pixels, or three sets of numbers for red, green, and blue will be assigned to a full color image pixels. Convolved results for black and white images or gray images are often followed by maximum-pooling (max-pooling) process, the object of which is another problem.

Basic concept of the convolution is that the linear convolution, in other words summation of

multiplications between the image pixel numbers and kernel coefficients produces larger resultant summation, if the kernel is well trained to distinguish the right image. Original mathematical convolution includes a reversing and shifting of the multiplying function, and integration of the multiplied function. In that sense, the CNN would mean convolution-like procedure. The CNN has been successfully used for many situations. However, there still remains whether any nonlinear summation method would work for specific images only or most images.

Fully connected layer will have weight coefficients representing the relative contribution amounts of each node to the final output. Outputs of the images for training are compared to true answers, and usually minimizing the total errors is the target of training. One principle is that the two sets of weights: weight coefficients of the convolution layer, and the fully connected layer should work as positive way, if a larger output number represents the true side.

The question is whether the kernel we found from the convolution between the kernel and a given image would give the maximum value. Let's consider a one dimensional function,  $y = 0$  between 0 and 0.5, and 1 between 0.5 and 1, see Fig. 1(a). Ignoring reversing and shifting of the multiplying function, we want to find the multiplying function, the integration of which is limited to 0.5. We find the best kernel function of exactly the same function as the original function, and its convolutional integration is 0.5, and its shape is the same as the original function or the kernel function. This result is representing cases of common black and white images. In this case the information is either 0 or 1, which is locally constant, and the kernel may have constant value within the local zone. This complies with the convolution concept well that if a kernel is similar to the original shape, then the convolution is largest. However, when multiple number of training images are dealt with during the training process, some portion of all images may have black color, and the other portion may have white color at a pixel. Then, finally trained optimum coefficient of the kernel at the pixel may have a real numbers between 0 and 1.



**Fig. 1.** Effects of kernel resembling original functions on one-dimensional coordinate:  
 a) black and white image; b) and c) gray image

Now, when we have the original function as  $y = x$  between 0 and 1 like a gray image, and assign a kernel function as the same as the original function, the convolutional integration is  $1/3$  in case (b). When we select another kernel function as 0 between 0 and 0.5, and 1 between 0.5 and 1, convolutional sum is  $3/8$  in case (c), which is slightly larger than case (b). The convolving function, or kernel function with exactly the same shape as original function does not produce the maximum value, or an extreme value. The resembling kernel function simply tends to produce relatively large convolutional integration. If we want to find the optimum kernel function with an upper limit integration value to produce the largest convolutional integration, the kernel found will be a delta function positioned at the largest independent variable place. If we call convolution as multiplication of coefficients and input pixel numbers, this operation could be called as linear with respect to the input pixel number which is regarded as independent variable. This example

implies that application of the linear convolution may not be the unique way for classification of gray images.

It is considered that a nonlinear kernel could overcome the gray-direction-dependent solutions, either by resembling the nonlinearity of the visual reception operations in nature, or by taking rational reasoning for more efficient mathematical convergence.

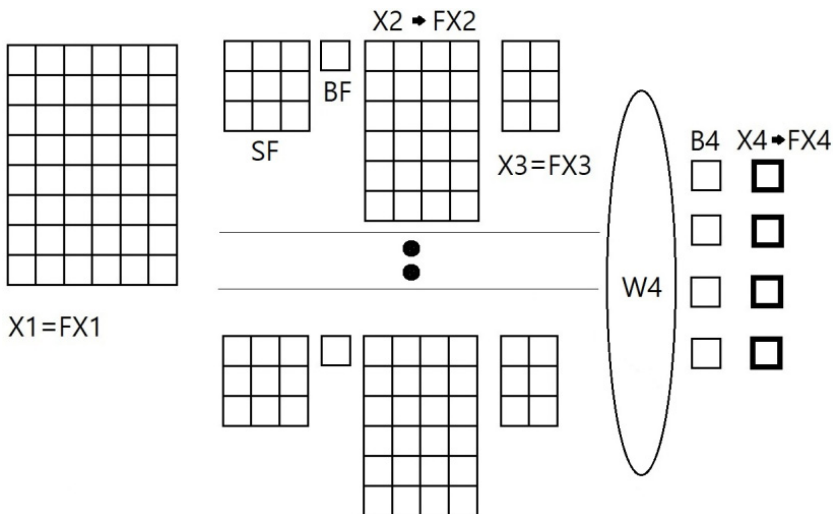
## 2. Proposing kernel with square of subtraction

Ideal kernel should account for the similarity between images; images used for training and the images to be examined. We propose a kernel, the coefficient of which are directly subtracted from the local number of the pixel in common two-dimensional input image patches. Let's consider a typical neural network architecture with one convolution layer with the new kernel and a pooling layer, see Fig. 2. We call it "Subtraction Neural Network" (SNN) afterwards.

An important attribute to use the feature map with the fully connected layer is that a better kernel coefficients should lead to larger feature map element, so that the weights applied to the feature map elements play a positive role to produce larger output. Therefore, we propose a kernel that the subtraction of between the input pixel number and the kernel coefficient is squared, its sign is reversed, and a bias is added. The feature map elements  $X2$  are computed from the following Eq. (1):

$$X2_{KF,I2,J2} = BF_{KF} - \sum_{IF} \sum_{JF} (FX1_{KF,I1*,J1*} - SF_{KF,IF,JF})^2, \quad (1)$$

where  $X2$  is the feature map,  $FX1$  is the transferred input two-dimensional image pixel number matrix by a linear function,  $SF$  is the kernel coefficient matrix for subtraction,  $BF$  is the bias of each kernel,  $KF$  is the kernel serial index,  $IF2$ ,  $JF2$  are matrix element indices of a two-dimensional kernel,  $I1*$ ,  $J1*$  are two-dimensional matrix element indices of the input image pixel number matrix to match a moving kernel. It resembles the least square method, but with a negative sign here. Eq. (1) is quadratic, and thus nonlinear to both  $FX1$  and  $SF$ .



**Fig. 2.** SNN architecture to be considered:  $X1$  – input image pixel matrix;  $FX1$  – transferred  $X1$  by linear function;  $SF$  – subtraction kernel coefficient matrix;  $BF$  – bias of subtraction kernel;  $X2$  – feature map;  $FX2$  – transferred  $X2$  with sigmoid function;  $X3$  – feature map after pooling;  $FX3$  – transferred  $X3$  by linear function;  $W4$  – weights between fully connected layer and the output layer;  $B4$  – bias between the fully connected layer and the output;  $X4$  – output before activation function;  $FX4$  – output after sigmoid activation function

A pooling layer with arbitrary size of pooling filter often follows the above convolution layer, and the present architecture has one. Training of the subtracting kernel coefficients requires partial derivatives of the total error  $E_t$  with respect to each kernel coefficient  $SF$  and kernel bias  $BF$ , that is:

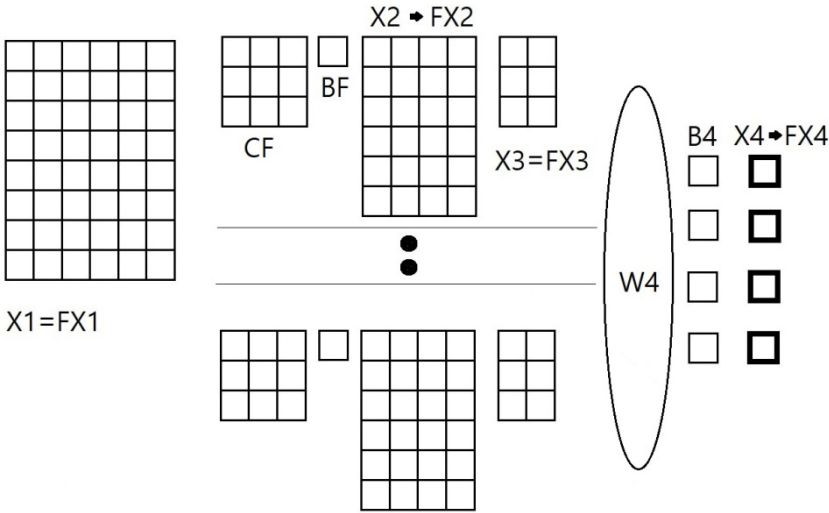
$$\frac{\partial E_t}{\partial SF_{KF,IF,JF}} = f \left( \frac{X2_{KF,I2,J2}}{\partial SF_{KF,IF,JF}} \right), \quad (2)$$

$$\frac{\partial E_t}{\partial BF_{KF}} = f \left( \frac{X2_{KF,I2,J2}}{\partial BF_{KF}} \right). \quad (3)$$

The partial derivative of the total error in Eq. (2) is obtained from Eq. (1), and the derivative of  $X2$  over the kernel coefficient  $SF$  includes the kernel coefficient  $SF$  itself as well as the  $FX1$ , that is:

$$\frac{\partial X2_{KF,I2,J2}}{\partial SF_{KF,IF,JF}} = 2(FX1_{KF,I1*,J1*} - SF_{KF,IF,JF}). \quad (4)$$

It would be meaningful to compare the present proposal of nonlinear form of SNN with an existing common linear convolutional neural network. We consider a linear CNN with the same architecture as that of the above proposed nonlinear SNN of Fig. 2 except the kernel part, see Fig. 3.



**Fig. 3.** CNN architecture to be compared: CF means linear convolutional kernel; X1 – input image pixel matrix; FX1 – transferred X1 by linear function; CF – linear convolution kernel coefficient matrix; BF – bias of linear kernel; X2 – feature map; FX2 – transferred X2 with sigmoid function; X3 – feature map after pooling; FX3 – transferred X3 by linear function; W4 – weights between fully connected layer and the output layer; B4 – bias between the fully connected layer and the output; X4 – output before activation function; FX4 – output after sigmoid activation function

When we use an existing linear convolutional kernel in the convolutional neural network, the feature map elements are computed from convolution between the kernel elements and the input image pixel number matrix elements as follows:

$$X2_{KF,I2,J2} = BF_{KF} + \sum_{IF} \sum_{JF} CF_{KF,IF,JF} FX1_{KF,I1*,J1*}. \quad (5)$$

The derivatives of  $X2$  with respect to  $CF$  for the convolutional kernel are also different from Eq. (4) for the subtraction kernel, that is:

$$\frac{\partial X2_{KF,I2,J2}}{\partial CF_{KF,IF,JF}} = FX1_{KF,I1*,J1*} \quad (6)$$

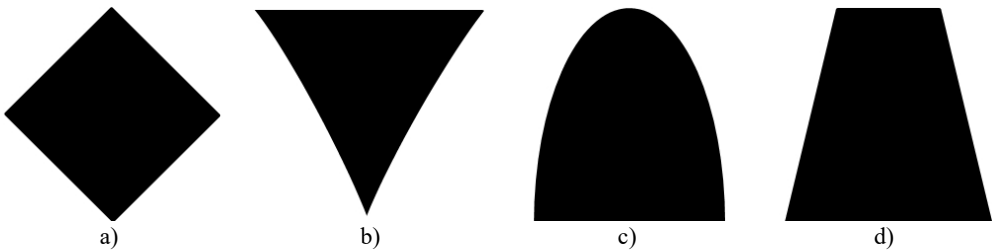
Eq. (4) of the nonlinear subtraction kernel contains the same variable  $FX1$  as Eq. (6) but twice larger. Another difference between the new subtraction kernel and the conventional linear convolutional kernel is that Eq. (4) involves  $SF$  term itself like an independent variable, which represent inherent nonlinearity of the present subtraction kernel. It is expected that the training time for a SNN should be similar to that of the corresponding linear CNN with the same architecture, if the number of epoch iteration is retained. If convergence criteria is limiting the total error allowance, training time becomes dependent on the performance of a network. Comparison is needed to examine the training efficiency of the two networks, nonlinear and linear ones.

### 3. Application of the present nonlinear subtraction approach

The nonlinear subtraction kernel approach proposed here is examined against two sets of images: 4 black and white images, and 4 gray images.

#### 3.1. Black and white images

Four simple geometries are chosen as a classification problem. All four shapes are symmetric with respect to vertical line. Their widths and heights are the same, their positions are the same, and are at centers of the whole images. Pixel matrices of all four input images have size of  $100 \times 100$ .



**Fig. 4.** Four black and white images used for comparison of present nonlinear SNN and existing linear convolutional neural network: a) square, b) parabolic on both sides, c) half-elliptic; d) trapezoidal

Number of kernels is chosen as 4, kernel size is  $3 \times 3$ , and filter size of pooling is  $1 \times 1$  which is the same as no pooling as default. Zero padding is not used during computation process from the input matrix to the feature map. Learning rate of 0.0005 is used.

Training of the SNN shows decrease of the total error as training number increases, see Fig. 5.

The training of the present nonlinear subtraction kernel shows faster decrease of the total error than the existing linear kernel as the iteration number continues. This is a clear indication that the present nonlinear kernel performs better than the existing linear kernel for the given image data sets, taking account that both networks have the same network architecture.

Trained SNN can be examined to any new image. Image (a) of Fig. 4 is chosen for checking the ability to distinguish the four original figures of Fig. 4 as self-assessment. Because the image was the given information for training of the SNN, the ideal expectation of the application of Fig. 4(a) is close to 1.0. The application results depend on the iteration number, and the accuracy increases as the iteration number increases, see Fig. 6.

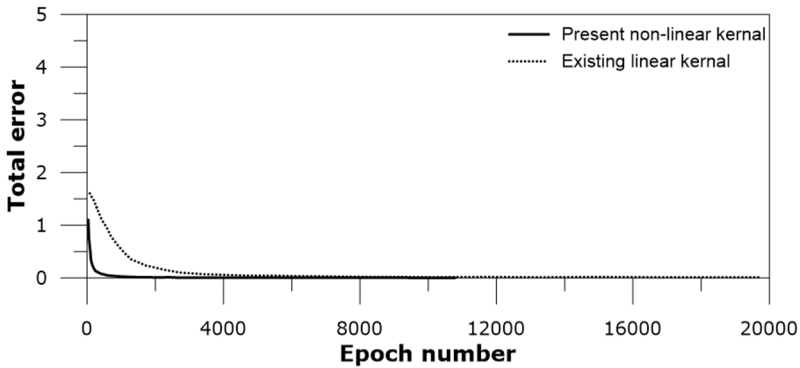


Fig. 5. Total error versus iteration number for black and white images: real line – present nonlinear kernel; dotted line – existing linear kernel

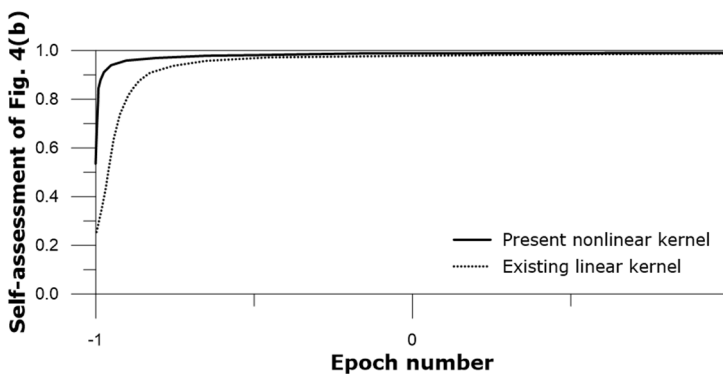


Fig. 6. Output of application of image Fig. 4(a): real line – present nonlinear kernel; dotted line – existing linear kernel

Both existing linear and present nonlinear kernel networks produce asymptotic approach toward 1.0 after many training iterations. The present nonlinear kernel network approaches 1.0 earlier than the existing linear kernel network. This difference also demonstrates the effectiveness of the present nonlinear kernel. Other self-assessments on Figs. 4(b, c, and d) show similar trends to Fig. 4(a).

Optimum kernel size has attracted interest to many neural network modelers. The optimum kernel size may be dependent to many relevant factors like image pattern, existence of background, or color style, and their relationship has not yet been clearly arranged. Although the number of black and white images is not large, it would be meaningful to see the effect of the kernel size by applying neural networks to these images. Pooling process is not included in these tests, in other words,  $1 \times 1$  pooling is applied to these tests. Tested kernel sizes are all square, i.e.  $1 \times 1$ ,  $3 \times 3$ ,  $5 \times 5$ , and  $7 \times 7$ . The total errors after 20000 training epoch iterations for several kernel sizes of the present nonlinear SNN and the existing linear neural network are shown in Fig. 7.

$3 \times 3$  kernels exhibited relatively best results for both the present nonlinear SNN and the existing linear neural network compared to other sized kernels. Optimum kernel size may depend on various parameters like pattern type, micro-pattern frequency, or image edge shape, and is not further investigated here. Important thing is that  $3 \times 3$  kernels for the present nonlinear neural network is better than that of the linear neural network work, and all other sized kernels have the same trend.

Pooling is another nonlinear process, and often adopted after convolution process. Tests on the max-pooling filter size are carried out to see their effect. Retaining  $3 \times 3$  kernel size, square pooling filter sizes of  $1 \times 1$  (means no pooling),  $2 \times 2$ , and  $7 \times 7$  are examined after both the present nonlinear

subtraction kernel convolution and the existing linear kernel convolution, see Fig. 8.

No pooling case produces the smallest total error after 20000 training iterations for both the present nonlinear kernel network and the existing linear kernel network.  $2 \times 2$  pooling filter shows the next best results to no pooling. This result may be related to the images of Fig. 4, and could be thought as specific result, rather than general one. All pooling filter sizes of the subtraction nonlinear kernel network are better than the existing linear kernel network. The present examination results on the pooling filter sizes are based on a limited number of images, and finding optimum pooling filter size for other images may vary.

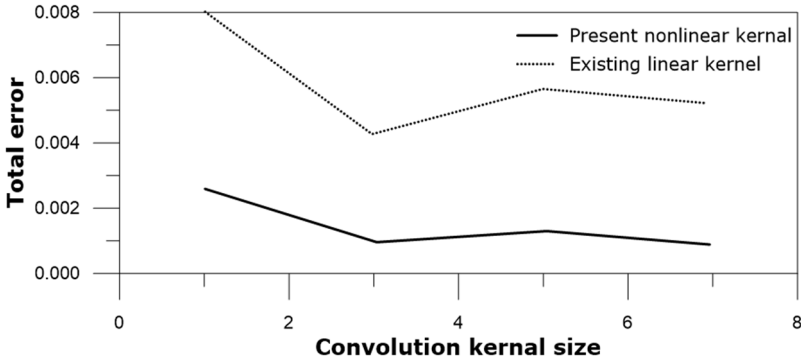


Fig. 7. Total error versus kernel size: real line – present nonlinear kernel; dotted line – existing linear kernel

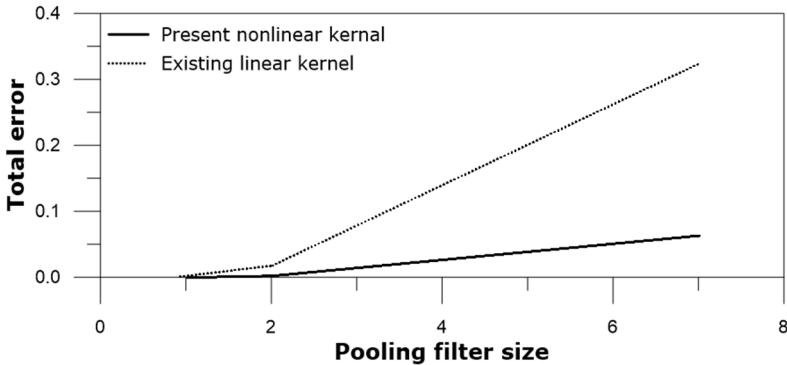


Fig. 8. Total error versus pooling size: real line – present nonlinear kernel; dotted line – existing linear kernel

### 3.2. Gray images

Gray images have different characteristics from black and white images from mathematical point of view. Gray image pixel is expressed as an intermediate real number between 0 and 1 or their multiples, respectively, while black and white image pixel is expressed by bimodal manner like binary numbers. Behavior of convolution kernel in the CNN may depend on image type.

Selected gray images are all square in shape filled with vertical gray stripes, see Fig. 9. As the first step to examine the present nonlinear kernel, it would be meaningful to test simple patterns. Vertical stripes mean that they are virtually one-dimensional, and examination results are expected to provide intuitive results. Background color is white. Image sizes are  $20 \times 20$ , while the sizes of gray parts are all  $10 \times 10$ .

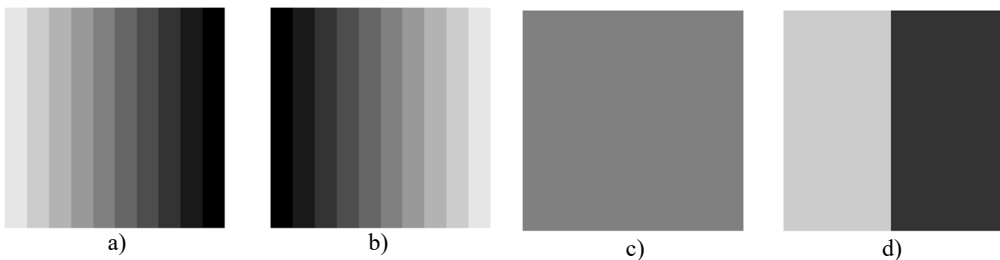
Fig. 9(a) image is vertical stripes getting darker in the right side, Fig. 9(b) image is the horizontally reversed image of Fig. 9(a), Fig. 9(c) is plain gray of 50 % black, and Fig. 9(d) is combination of two wide vertical gray stripes of 20 % and 80 % black.



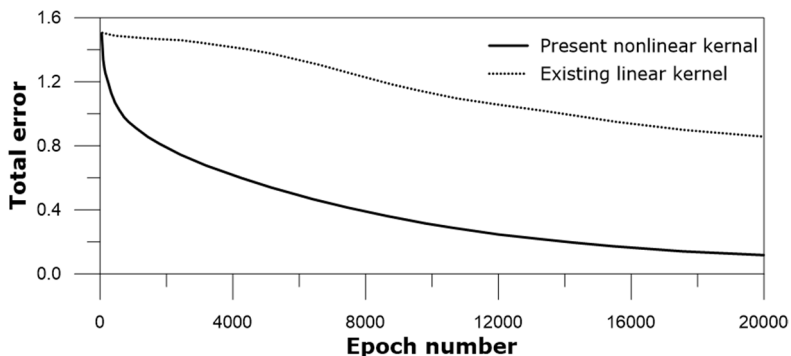
Number of kernels chosen for tests is 4, kernel size is  $3 \times 3$ , and pooling filter size is  $1 \times 1$  (no pooling). Zero padding is not used during computation process from the input matrix to the feature map. Learning rate of 0.0005 is used.

The total errors during training procedure decrease as the epoch iteration number increases, see Fig. 10. The training accelerate optimization of convolution kernel coefficients effectively for the present nonlinear kernel neural network, while it seems that the speed of training existing linear kernel network is not high. Considering the fact that the same neural network architecture is for both networks, this result may support the usefulness of the present nonlinear kernel, although the present test results are based on only four specific gray images.

Trained nonlinear subtraction kernel CNN can be examined to any new image. Image of Fig. 9(a) is chosen for checking the ability to distinguish the four original figures of Fig. 9 as self-assessment. Because the image was the given information for training of the subtraction kernel neural network, the ideal expectation of the application of Fig. 9(a) is close to 1.0. The application results show that the accuracy increases with the epoch iteration number, and the outputs eventually converge to 1.0, see Fig. 11.



**Fig. 9.** Four gray images to be examined with present SNN and CNN: a) darker right, b) darker left, c) half-gray; d) two gray tones, i.e. 20 % and 80 % black color



**Fig. 10.** Total error versus iteration number: real line – present nonlinear kernel; dotted line – existing linear kernel

Both existing linear and present nonlinear kernel networks produce asymptotic approach toward 1.0 after many training iteration. The present nonlinear kernel network approaches 1.0 earlier than the existing linear kernel network. This difference also demonstrates the effectiveness of the present nonlinear kernel. Other self-assessments on Figs. 9(b), (c), and (d) show similar trends to Fig. 9(a).

We look at the effect of the kernel size for the present nonlinear kernel and the existing linear kernel on the four test gray images. It would still be meaningful to see the effect of the kernel size by applying neural networks on our limited number of gray images. Pooling process is not included in these tests, in other words,  $1 \times 1$  pooling is applied to these tests. Tested kernel sizes are all square, i.e.  $1 \times 1$ ,  $3 \times 3$ ,  $5 \times 5$ , and  $7 \times 7$ . The total errors after 20000 training epoch iterations of the present nonlinear SNN and the existing linear neural network are shown in Fig. 12.

5×5 kernel exhibits best results, and 3×3 kernel is the next for the present nonlinear subtraction kernel convolutional neural networks. On the other hand kernel size does not give much different total error for the existing linear kernel convolutional neural network. Optimum kernel size may depend on various parameters like pattern type, micro-pattern frequency, or image edge shape, and thus the present results could not be generalized for other images. It should be noted that 5×5 and 3×3 kernels for the present nonlinear neural network is better than that of the linear neural network work. The results may imply that overall behavior of the present nonlinear kernel convolution is promising compared to the existing linear kernel convolution.

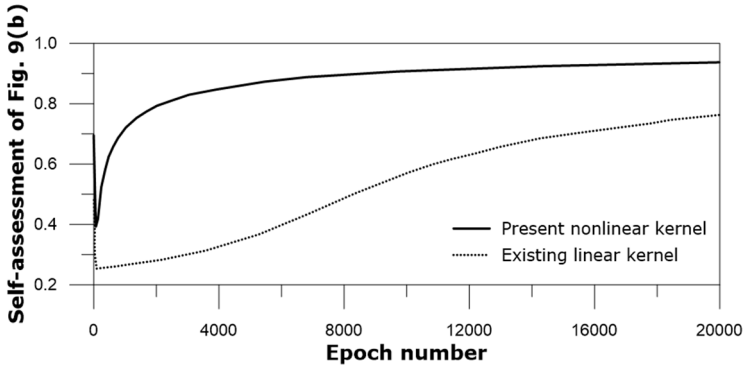


Fig. 11. Output of application of image Fig. 9(a): real line – present nonlinear kernel; dotted line – existing linear kernel

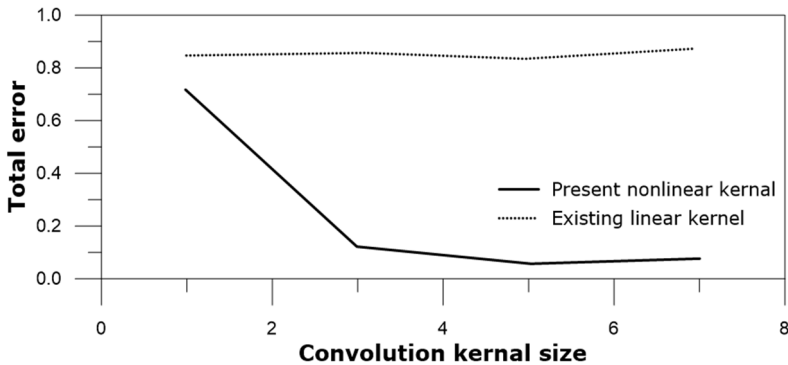


Fig. 12. Total error versus kernel size: real line – present nonlinear kernel; dotted line – existing linear kernel

Pooling is another nonlinear process, and often adopted after convolution process. Tests on the max-pooling filter size are carried out to see their effects on the total error. Square pooling filter sizes of 1×1 (means no pooling), 2×2, 3×3, and 6×6 are examined after both the present nonlinear subtraction kernel convolution and the existing linear kernel convolution, while the convolution kernel size is kept 3×3, see Fig. 13.

1×1 pooling case, in other words, no pooling case produced the smallest total error after 20000 training iterations for both the present nonlinear kernel network and the existing linear kernel network. 2×2 pooling filter shows the next best results to no pooling. It should be noted that this result is obtained from the four images of Fig. 9, and could be thought as specific problem, rather than general one. 3×3 pooling filter sizes of the subtraction nonlinear kernel network are better than the existing linear kernel network. The total errors of 6×6 pooling filter networks for both the nonlinear kernel and the existing linear kernel show slow decrease, meaning poor performance. At least this large pooling filter size like 6×6 is not desirable for given gray images.

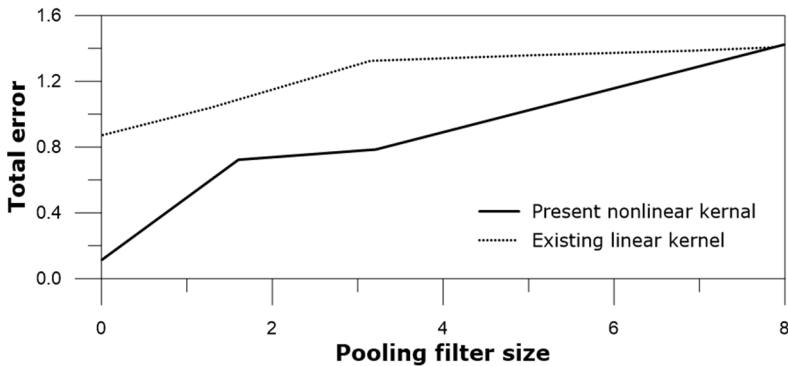


Fig. 13. Total error versus pooling size: real line – present nonlinear kernel; dotted line – existing linear kernel

#### 4. Conclusions

The concept of finding a linear kernel which results in the largest integration for given black and white image matches well with the existing linear kernel convolutional neural network, but not for gray images. A kernel composed of a bias and negative summation of square of subtraction between the input pixel number and the kernel coefficient itself within the moving input pixel patch for convolution is proposed here. The square of subtraction represent the intensity of similarity between the kernel coefficient matrix elements and the image pixel number matrix elements as expressed in the summation equation. It resembles the least square method, but with a negative sign here and biases are added. Training of the neural network to obtain optimum kernel coefficients requires a derivative with respect to the kernel coefficient includes the input pixel number and the kernel coefficient itself, and makes the formation nonlinear.

The present nonlinear kernel formulation is examined to two sets of images: four black and white images, and four gray images. The network training results are compared to those of corresponding CNN with the same architecture and parameters except the kernel. Application results show that the present nonlinear subtraction kernel is more efficient than the existing linear convolutional kernel, although the tests are limited to two sets of images: black and white images and gray images. After all, the nonlinear kernel of the present network produced fast decrease of the total error, and its intention is well explained in the kernel equation.

It is believed that the present nonlinear kernel is based on sounder background to obtain maximum outputs. Further examinations against various images and conditions are needed to enhance reliability of the presently proposed nonlinear subtraction convolution kernel of the convolutional neural network. It seems that application areas of the present nonlinear subtraction kernel CNN could be expanded to full color images as well.

#### Acknowledgements

This work is a result of the research project, “Practical Technologies for Coastal Erosion Control and Countermeasure” funded by the Ministry of Oceans and Fisheries, Korea.

#### References

- [1] Ren Y., Zhu J., Li J., Luo Y. Conditional generative moment-matching networks. *Advances in Neural Information Processing Systems*, 2016, p. 2928-2936.
- [2] Wang M., and Deng W. Deep face recognition. *ArXiv:1804.06655*, 2018.
- [3] Moeskops P., Wolterink J. M., van der Velden B. H. M., Gilhuijs K. G. A., Leiner T., Viergever M. A., Išgum I. Deep learning for multi-task medical image segmentation in multiple modalities. *MICCAI*, 2016, p. 478-486.

- [4] **Bernal J., Kushibar K., Cabezas M., Valverde S., Oliver A., Lladó X.** Quantitative analysis of patch-based fully convolutional neural networks for tissue segmentation on brain magnetic resonance imaging. *Journal of IEEE*, Vol. 7, 2019, p. 89986-90002.
- [5] **Cun Y. L., Boser B., Denker J. S., Howard R. E., Hubbard W., Jackel L. D., Henderson D.** Handwritten Digit Recognition with a Back-Propagation Network. *Advances in Neural Information Processing Systems 2*, Morgan Kaufmann Publishers Inc., San Francisco, USA, 1990, p. 396-404.
- [6] **Fukushima K.** Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, Vol. 36, Issue 4, 1980, p. 193-202.
- [7] **Szulborski R. G., Palmer L. A.** The two-dimensional spatial structure of nonlinear subunits in the receptive fields of complex cells. *Vision Research*, Vol. 30, Issue 2, 1990, p. 249-254.
- [8] **Niell C. M., Stryker M. P.** Highly selective receptive fields in mouse visual cortex. *Journal of Neuroscience*, Vol. 28, Issue 30, 2008, p. 7520-7536.
- [9] **Nair V., Hinton G. E.** Rectified linear units improve restricted Boltzmann machines. *Proceedings of the 27th International Conference on Machine Learning*, 2010.
- [10] **Maas A. L., Hannun A. Y., Ng A. Y.** Rectifier nonlinearities improve neural network acoustic models. *Proceedings of the 30th International Conference Machine Learning*, 2013.
- [11] **He K., Zhang X., Ren S., Sun J.** Deep residual learning for image recognition. *ArXiv 1512.03385*, 2015.
- [12] **Goodfellow I. J., Warde Farley D., Mirza M., Courville A. C., Bengio Y.** Maxout networks. *International Conference on Machine Learning*, 2013.
- [13] **Jarrett K., Kavukcuoglu K., Cun Y. L.** What is the best multi-stage architecture for object recognition? *IEEE International Conference on Computer Vision*, 2013.
- [14] **Ciresan D. C., Meier U., Masci J., Maria G. L., Schmidhuber J.** Flexible, high performance convolutional neural networks for image classification. *Proceedings of the International Joint Conference on Artificial Intelligence*, Vol. 1, 2011, p. 1237-1242.
- [15] **Ciresan D., Meier U., Schmidhuber J.** Multi-column deep neural networks for image classification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.
- [16] **Blot M., Cord M., Thome N.** Max-min convolutional neural network for image classification. *ArXiv:1610.07882*, 2016.
- [17] **Ranzato M. A., Huang F. J., Boureau Y., Cun Y. L.** Unsupervised learning of invariant feature hierarchies with applications to object recognition. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
- [18] **Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Rabinovich A.** Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.
- [19] **Hyvarinen A., Koster U.** Complex cell pooling and the statistics of natural images. *Network: Computation in Neural Systems*, Vol. 18, Issue 2, 2007, p. 81-100.
- [20] **Turaga S. C., Murray J. F., Jain V., Roth F., Helmstaedter M., Briggman K., Seung H. S.** Convolutional networks can learn to generate affinity graphs for image segmentation. *Neural Computation*, Vol. 22, Issue 2, 2010, p. 511-538.
- [21] **Zeiler M. D., Fergus R.** Stochastic pooling for regularization of deep convolutional neural networks. *ArXiv: 1301.3557*, 2013.
- [22] **Rao Z., Zeng C., Wu M., Wang Z., Zhao N., Liu M., Wan X.** Research on a handwritten character recognition algorithm based on an extended nonlinear kernel residual network. *KSII Transactions on Internet and Information Systems*, Vol. 12, Issue 1, 2018, p. 413-435.
- [23] **Lin M., Chen Q., Yan S.** Network in network. *International Conference on Learning Representations*, 2014.
- [24] **Zhai S., Cheng Y., Lu W., Zhang Z.** Doubly convolutional neural networks. *Conference on Neural Information Processing Systems*, 2016.
- [25] **Zoumpourlis G., Dومانoglou A., Vretos N., Daras P.** Non-linear convolution kernels for CNN-based learning. *Computer Vision and Pattern Recognition*, *arXiv: 1708.07038*, 2017.
- [26] **Volterra V.** *Theory of Functionals and of Integral and Integro-Differential Equations*. Dover Publications, 2005.



**Hyo-Seob Kim** received his Ph.D. from Liverpool University in 1993. He is a Professor of Coastal and Harbor engineering at Kookmin University. His main interests are sediment transport, AI and wind power generation. He has conducted a number of R&D research projects, and now conduct research applying AI techniques to also coastal Marine. His major research and development history include development of Marine and environmental prediction system: MEPS, 2008-2013, Coastal Erosion Control: Stage I: KMIST, 2014-2018.



**Hojun Yoo** completed his master's degree in Kookmin University in 2013, and is currently working at the Geosystems Research Consultant, also conducting a Ph.D. program at Kookmin University. His main interests include coastal marine modeling, sediment movement, AI techniques, and marine debris. He has participated in a number of research projects, and now conduct research applying AI techniques to also coastal Marine.



**Jung Lyul Lee** received his Ph.D. from Florida University. He is a Professor of coastal environment engineering at Sungkyunkwan University. His main interests are sediment transport, coastal environment, coastal facilities. He has conducted a number of R&D research projects, and now conduct research applying AI techniques to also coastal Marine. He is working on research projects and efforts to reduce coastal erosion in the east coast of Korea. His major research and development history include Coastal erosion control: Stage I: KMIST, 2014-2018, "Practical Technologies for Coastal Erosion Control and Countermeasure" funded by the Ministry of Oceans and Fisheries, Korea, 2019~Now.



**Seungho Lee** completed his master's degree in Kookmin University in 2018, and is currently conducting a Ph.D. program at Kookmin University. His main interests include coastal marine modeling, AI techniques. He has participated in a number of research projects, and now conduct research applying AI techniques to also coastal Marine.