Neural Networks 116 (2019) 279–287

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Differential convolutional neural network

M. Sarıgül [a,*], B.M. Ozyildirim [b], M. Avci [c]

[a] Iskenderun Technical University, Computer Engineering Department, Hatay, Turkey
[b] Cukurova University, Computer Engineering Department, Adana, Turkey
[c] Cukurova University, Biomedical Engineering Department, Adana, Turkey

HIGHLIGHTS

- A novel convolution technique named as Differential Convolution is proposed.
- The technique aims to transfer directional activation differences to the next layer.
- The technique considers how convolved features change on a feature map.
- An updated back-propagation algorithm compatible with the technique is developed.

ARTICLE INFO

Article history:
Received 16 July 2018
Received in revised form 18 April 2019
Accepted 30 April 2019
Available online 10 May 2019

Keywords:
Convolutional neural networks
Deep learning
Image classification
Convolution techniques
Pattern recognition
Machine learning

ABSTRACT

Convolutional neural networks with strong representation ability of deep structures have ever increasing popularity in many research areas. The main difference of Convolutional Neural Networks with respect to existing similar artificial neural networks is the inclusion of the convolutional part. This inclusion directly increases the performance of artificial neural networks. This fact has led to the development of many different convolutional models and techniques.

In this work, a novel convolution technique named as Differential Convolution and updated error back-propagation algorithm is proposed. The proposed technique aims to transfer feature maps containing directional activation differences to the next layer. This implementation takes the idea of how convolved features change on the feature map into consideration. In a sense, this process adapts the mathematical differentiation operation into the convolutional process. Proposed improved back propagation algorithm also considers neighborhood activation errors. This property increases the classification performance without changing the number of filters.

Four different experiment sets were performed to observe the performance and the adaptability of the differential convolution technique. In the first experiment set utilization of the differential convolution on a traditional convolutional neural network structure made a performance boost up to 55.29% for the test accuracy. In the second experiment set differential convolution adaptation raised the top1 and top5 test accuracies of AlexNet by 5.3% and 4.75% on ImageNet dataset. In the third experiment set differential convolution utilized model outperformed all compared convolutional structures. In the fourth experiment set, the Differential VGGNet model obtained by adapting proposed differential convolution technique performed 93.58% and 75.06% accuracy values for CIFAR10 and CIFAR100 datasets, respectively. The accuracy values of the Differential NIN model containing differential convolution operation were 92.44% and 72.65% for the same datasets. In these experiment sets, it was observed that the differential convolution technique outperformed both traditional convolution and other compared convolution techniques. In addition, easy adaptation of the proposed technique to different convolutional structures and its efficiency demonstrate that popular deep learning models may be improved with differential convolution.

1. Introduction

Deep learning structures have become one of the most studied fields in recent years thanks to its high performance in many different research areas (Dahl, 2015; He, Zhang, Ren, & Sun, 2016; Krizhevsky, Sutskever, & Hinton, 2012; Ravi et al., 2017; Sankar, Batri, & Partvathi, 2016; Tennakoon, Mahapatra, Ro, Sedai, & Garnavi, 2016). One of the most important properties of these structures is their representation ability largely provided by the convolution operation (LeCun et al., 1990). In the majority of the deep structures, the convolution operation is applied to extract features from the large-scaled input data. In each convolutional

* Corresponding author.
E-mail addresses: mehmet.sarigul@iste.edu.tr (M. Sarıgül), mozyildirim@cu.edu.tr (B.M. Ozyildirim), mavci@cu.edu.tr (M. Avci).

https://doi.org/10.1016/j.neunet.2019.04.025
0893-6080/

layer, there exists a number of filters, also called kernels, slid over the input images to generate a number of feature maps (Szegedy et al., 2015). A neuron in the generated feature map is connected to the previous layer's neurons within an area in the size of the corresponding filter. This area is called the receptive field of the neuron. (Lecun, Bottou, Bengio, & Haffner, 1998). Another useful property called weight sharing mechanism is to use the same filter in the whole image during convolution. This mechanism reduces the number of parameters to be trained while allowing the same feature to be searched within the whole image. Convolution operation enables to recognize visual patterns on the input data (Lecun et al., 1998).

The high performance of convolutional neural networks has led to the development of different convolution techniques. One of these techniques named as tiled convolution uses different filters for feature map neurons having close receptive fields on the input image (Ngiam, Chen, Chia, Koh, Le, & Ng, 2010). Hence, a feature map is created using a number of filters and more decisive features extracted with the same number of feature maps. Dilated convolution (Yu & Koltun, 2015), another convolution method, is based on the idea of expanding the receptive field of a filter without increasing the number of parameters. It inserts cells with zero weight values into the filters. It was shown that this may lead to a performance boost in some studies (Yu & Koltun, 2015). In another study, micro-mlp structures are used as filters and this technique is named as Network In Network (NIN) (Lin, Chen, & Yan, 2013). This allows the filters to learn more complex relations in the training process. Another well known alternative convolution technique is Inception (Szegedy et al., 2015b). In this method, filters of different sizes are contained within a single convolution layer. The convolution process can be also accompanied at the same time by the pooling process. It was shown that performance can be increased without increasing the number of parameters by using the inception module.

The main idea behind the alternative techniques is to increase the number of obtained features from a single convolution layer. Similar to this idea, in this paper, a novel convolution technique, differential convolution, is proposed. There are some novelties and advantages of the proposed method over the other approaches. While Tiled convolution increases the number of filters and keeping the number of feature maps constant, the number of feature maps is increased without increasing the number of filters in the Differential convolution. Dilated convolution attempts to increase the receptive field of a filter, on the contrary, differential convolution increases the efficiency of the filter without changing the receptive field size. The differential convolution has a different perspective than the NIN method and its implementation is much simpler. It can be used inside the inception module as traditional convolution. While Tiled CNN and Dilated CNN recommend new convolution techniques, the NIN method does not change the convolution method but the filter structure, while the inception module offers a structure where more than one technique can be used together. Therefore, the Differential CNN structure may be presented as an alternative to conventional CNN, Tiled CNN and Dilated CNN structures.

Proposed convolution technique improves the standard convolution by considering directional changes among a pixel and its neighbors. Differential convolution extracts pattern orientation of a pixel and its neighbors by calculating the additional change amounts. As in mathematical differentiation, change amount calculations take sequential changes into consideration by computing the difference among the activations of the pixels. Since images can be defined as a form of superposition of waves with different frequencies (Wallace, 1992), differential convolution generates a new feature map representing signal changes. These new feature maps lead detection of the big differences on the patterns which are valuable information for classification problems. Four pre-defined constant filters are defined to perform this operation. Each of these filters is used to calculate the differences in one of the four directions. The proposed technique applies these filters on a single feature map generated with the traditional convolution. This operation is named as differential convolution. The efficiency of the proposed technique is provided by preserving neighboring activation difference information. Additionally, neighboring activation difference is frequently used information in image processing especially for determining edges and corners. Although mathematical operations of the proposed method are simple, it requires updates on the traditional error back-propagation method. Hence, an improved error back-propagation method for differential convolution adapted structures named as accumulative error back-propagation is suggested. In the accumulative method, each neuron has received an error fed over the neighbor activations during the back-propagation.

Four different experiment sets were organized to prove the effectiveness of differential convolution. In the first experiment set, the performance of traditional CNN and performance of differential convolution adapted CNN were compared for ten different image datasets. In the second experiment set, differential convolution was utilized for popular deep structure, AlexNet, and the performance improvement of this structure was measured on ImageNet dataset. In the third experiment set, six different CNN structures were tested. Three of these structures were different traditional CNN structures and the other structures were differential convolution adapted CNN (Differential CNN), tiled convolution adapted CNN (Tiled CNN) and dilated convolution adapted CNN (Dilated CNN). Since the differential convolution is not an alternative to Inception module or NIN structure, these methods were excluded from the comparison. Differential convolution can be used in the Inception module as well as in conjunction with the NIN structure. In the fourth experiment set, differential convolution was adapted to NIN (Lin et al., 2013) and VGGNet (Simonyan & Zisserman, 2014) models. These structures are well-known and popular CNN structures. CIFAR10 and CIFAR100 datasets were utilized to measure the performance of the models in fourth experiment set. As a result of all these experiment sets, the effectiveness of the differential convolution was clearly demonstrated.

In the first experiment set, the efficiency of differential convolution was measured on 10 different datasets. since some tested datasets were unbalanced, in addition to accuracy values, precision, recall, and F1-score values were calculated. This method had increased the accuracy, precision, recall and F1 score up to 55.29%, 54.47%, 41.75%, and 56.43%, respectively. In the second experiment set, the differential convolution technique was adapted to the popular deep learning structure AlexNet and the performance increase was measured on ImageNet dataset. Differential convolution improved the top1 and top5 accuracies by 5.3% and 4.75%, respectively. In the third experiment set, the performance of the Differential CNN was compared with 5 different CNN structures for CIFAR-10 and CIFAR-100 datasets. Differential CNN performed higher performance than all compared network structures on the validation data. In the fourth experiment set differential convolution adaptation increased the accuracies of the NIN and VGGNet models up to 6.01% and 7.15%, respectively. In addition to improvements in the performance, it was also shown that the error minimization of the proposed technique is faster than that of other compared methods in the same period. These results prove the effectiveness and importance of the proposed convolution technique.

2. Convolutional neural networks

Convolutional neural networks (Lecun et al., 1998) proposed by Lecun have become one of the most successful methods in the field of pattern recognition. In these structures, locally trained filters are used for extracting visual features over the input image. Feature maps whose size has been reduced by the pooling operation are the input images of the next convolution. This process continues until deep features are extracted. After these steps, a decision is usually made by a classifier on these features (Ravi et al., 2017). A deep convolutional neural network generally refers to a structure involving convolutional layers, pooling layers, and a fully-connected network (Nielsen, 2015). In this structure, convolution operations are used for feature extraction, whereas full connected network is a classifier on these features. The fully-connected part can end up with a SoftMax output layer for classification purposes. There are many popular network structures created with these layers such as AlexNex, OverFeat, GoogLeNet, etc. One of the biggest problems is overfitting while training these structures. Some techniques that have been developed to prevent overfitting are data augmentation, DropOut and DropConnect (Deng, Dong, Socher, Li, Li, & Fei-Fei, 2009; Krizhevsky et al., 2012; Sermanet, Eigen, Zhang, Mathieu, Fergus, & LeCun, 2013; Szegedy et al., 2015).

2.1. Convolution

Convolution layer is a structure with a number of fixed-size filters that allows complex functions to be applied to the input image (Ravi et al., 2017). This process is carried out by sliding the locally trained filters over the image. Each filter has the same weight and bias values throughout the image during this process. This is called weight sharing mechanism and this mechanism provides the ability to represent the same feature on the entire image (Chandrakumar & Kathirvel, 2016; Ravi et al., 2017; Sankar et al., 2016). Local receptive field of a neuron represents the area which the neuron is connected in previous layer. Size of the receptive field is determined by the size of the filters. Let $m \times n$ and $c \times c$ be the size of the input image and the size of the kernel, i represent the image, w and b are weight and bias values of the filter, respectively. Output $o_{0,0}$ can be calculated as in Eq. (1) where f is activation function. Either ReLu or sigmoid can be used as activation function on this process (Nielsen, 2015; Sankar et al., 2016). The behavior of the ReLu activation function can also be seen in Eq. (2).

$$o_{0,0} = f\left(b + \sum_{t=0}^{c}\sum_{r=0}^{c} w_{t,r} i_{0+t,0+r}\right) \tag{1}$$

$$f(x) = \begin{cases} x & x > 0 \\ 0 & else \end{cases} \tag{2}$$

2.2. Pooling

Pooling process is applied to feature maps that have passed through convolution and activation function. This provides smaller feature maps to be generated, which are the summaries of the input feature maps. Pooling is performed by sliding a window on the image to apply the selected operation. The well-known pooling operations are maximum, average and L2 pooling. While average pooling takes the average of the input values, maximum pooling passes the maximum value throughout and L2 pooling calculates the L2 norm of the inputs (Nielsen, 2015). The biggest advantages provided by the pooling operation are reduction of the image size and extraction of the visual features independently on the image (Nielsen, 2015; Sankar et al., 2016).

2.3. Fully connected layer

After the convolution and the pooling layers, the data is transformed into a one-dimensional vector. This vector will be the input of the fully connected network. The fully-connected structure may contain one or more hidden layers. Each neuron multiplies the connection weights by the data from previous layer and adds a bias value. The calculated value passes through the activation function before being transmitted to the next layer. The calculations made by a neuron in this layer can be seen in Eq. (3).

$$fc_1 = f(b + \sum_{q=1}^{M} w_{1,q} * o_q) \tag{3}$$

where f is the activation function, w is the weight vector, o is input vector of the qth neuron and b is the bias value.

2.4. Softmax

The softmax activation function is a multi-class adapted version of the logistic regression. It is usually used in output layer for classification purposes. It is defined as in Eq. (4) (Nielsen, 2015).

$$class_j = \frac{exp(sf_j)}{\sum_q exp(sf_q)} \tag{4}$$

2.5. Data augmentation

Data augmentation is increasing number of samples, artificially. This process is carried out to prevent the overfitting situation in the neural network training. Changing the size of the image, its orientation, the lighting condition are the example operations for data augmentation. These artificial generated data are also included in the training dataset. This leads to more general feature extraction ability (Krizhevsky et al., 2012).

2.6. DropOut

DropOut is one of the techniques used for avoiding memorization. In this method, the activation of some randomly selected neurons in the network is taken as zero during training. Selected neurons are changed in each iteration of the training. The learning process becomes more reliable and the overfitting is reduced by this method (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014).

2.7. DropConnect

DropConnect is a similar technique to DropOut, developed to prevent overfitting. While the activations of the selected neurons are taken as zero in the DropOut technique, the randomly selected weight values are set to zero in the DropConnect technique. This process increases the generalization performance of the network and avoids memorizing (Wan, Zeiler, Zhang, Le Cun, & Fergus, 2013).

3. Differential convolutional neural network

In this section, the idea behind the proposed Differential Convolution technique, the details of its implementation and training algorithm are given. The CNN structures utilizing differential convolution technique are named as Differential CNNs. They consider changes among a neuron's and its neighbors' activations and apply updated back-propagation algorithm. Since the proposed technique can be considered as a plug and play module, it is applicable to all convolutional structures.

$$\begin{bmatrix} 1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Fig. 1. Fixed filters.

3.1. Differential convolution

Convolution operation, executed by sliding a number of filters over the input image is the key element of deep learning structures. It provides the extraction of the visual patterns on the input image. Hence, the more feature maps the structure generates, the more features the classifier obtains.

The idea behind the proposed convolution technique is the consideration of directional changes among a pixel and its neighbors. Differential convolution analyses pattern orientation of a pixel and its neighbors by the additional change amount calculations. As in the mathematical differentiation, sequential changes are taken into consideration by calculating the difference among the activations of the pixels. While considering the images in the form of superposition of waves with different frequencies an additional successive differential calculation must improve the pattern orientation ability of the standard convolution. This fact leads to the proposed method containing an improved convolution approach reinforced with a map of difference signal addition. These additional differential maps are produced from the feature map generated with the traditional convolution by applying predefined constant filters. The fixed filters are shown in Fig. 1. Each of these filters is used to calculate differences in one direction. As a result, additional feature maps containing signal differences for each direction are generated.

Let the feature map generated with traditional convolution be g_1 and the additional four maps be g_2, g_3, g_4 and g_5. Values of the neurons of these maps are calculated as given in (5)–(8). If the size of g_1 is $M \times N$, then size of the additional feature maps g_2, g_3, g_4 and g_5 will be $(M-1)\times N$, $M\times(N-1)$, $(M-1)\times(N-1)$ and $(M-1)\times(N-1)$, respectively. These generated additional feature maps are extended with zeros and brought to the size of the first feature map as shown in Fig. 2. Additionally, the receptive field of the filter is enlarged by one unit along two axes with this operation. Examples of the differential feature maps are given in Fig. 3.

$$g_{2,i,j} = g_{1,i,j} - g_{1,i+1,j} \tag{5}$$

$$g_{3,i,j} = g_{1,i,j} - g_{1,i,j+1} \tag{6}$$

$$g_{4,i,j} = g_{1,i,j} - g_{1,i+1,j+1} \tag{7}$$

$$g_{5,i,j} = g_{1,i+1,j} - g_{1,i,j+1} \tag{8}$$

After the first feature map produced by the traditional method, differential convolution method performs additional convolution operations with predefined constant filters. The predefined filters used in differential convolution are also useful for detecting basic structures such as edges and corners on images. Utilized filters have also been successfully used in other neural networks structures such as Cellular Neural Network (Chua & Yang, 1988).

One of the advantages of differential convolution is that increasing the depth of a single convolutional layer allows obtaining more features without increasing number of convolutional layers. The more feature maps generated, the more parameters in the next layer emerged. Although, this may seem like a disadvantage, it should be considered that a differential convolutional layer performs another convolution without the inclusion of any new trainable parameter.

3.2. Accumulative back-propagation algorithm

Differential convolution with new feature maps requires an improvement on the back propagation algorithm. During back-propagation, errors are transmitted in reverse direction to each feature map. The errors on each extra feature maps are multiplied by the corresponding fixed valued filter weights and added on corresponding errors of the first feature map. This computed error matrix is used for training the relevant filter and it is propagated backwards.

Let the error transmitted to the first map be h_1; the errors transmitted to the generated extra maps be h_2, h_3, h_4, h_5, the elements of the error matrix be E; and the size of the maps be $M \times N$. Error calculations for the relevant filter are given in (9)–(11).

$$\begin{aligned} E_{i,j} = {} & h_{1,i,j} - h_{2,i,j-1} + h_{2,i,j} - h_{3,i-1,j} + h_{3,i,j} - h_{4,i-1,j-1} \\ & + h_{4,i,j} - h_{5,i-1,j} + h_{5,i,j-1} \end{aligned} \tag{9}$$

where $i > 1, j > 1, i < M$ and $j < N$. As it is seen in Eq. (9), neurons neither in the corner nor in the edges take error feeds from all neighborhood neurons. Corner neurons of the map take error feeds from 3 neighborhood neurons as in Eq. (10).

$$E_{i,j} = \begin{cases} h_{1,i,j} + h_{2,i,j} + h_{3,i,j} + h_{4,i,j} & i = 1, j = 1 \\ h_{1,i,j} - h_{2,i,j-1} + h_{3,i,j} + h_{5,i,j-1} & i = 1, j = N \\ h_{1,i,j} + h_{2,i,j} - h_{3,i-1,j} - h_{5,i-1,j} & i = M, j = 1 \\ h_{1,i,j} - h_{2,i,j-1} - h_{3,i-1,j} - h_{4,i-1,j-1} & i = M, j = N \end{cases} \tag{10}$$

Errors to be propagated to the edge neurons of the map are calculated as in (11). Edge neurons receive error feeds from 5 neighborhood neurons.

$$E_{i,j} = \begin{cases} h_{1,i,j} - h_{2,i,j-1} + h_{2,i,j} + h_{3,i,j} + h_{4,i,j} + h_{5,i,j-1} \\ \quad i = 1, 1 < j < N \\ h_{1,i,j} - h_{2,i,j-1} + h_{2,i,j} - h_{3,i-1,j} - h_{4,i-1,j-1} - h_{5,i-1,j} \\ \quad i = M, 1 < j < N \\ h_{1,i,j} + h_{2,i,j} + h_{3,i,j} - h_{3,i-1,j} + h_{4,i,j} - h_{5,i-1,j} \\ \quad 1 < i < M, j = 1 \\ h_{1,i,j} - h_{2,i,j-1} + h_{3,i,j} - h_{3,i-1,j} - h_{4,i-1,j-1} + h_{5,i,j-1} \\ \quad 1 < i < M, j = N \end{cases} \tag{11}$$

4. Test and results

In test phase, four different experiment sets were applied to demonstrate adaptability and effectiveness of the proposed technique. In the first experiment set classification performance evaluation was aimed. In the second experiment set adaptability of the proposed technique was demonstrated. In the third experiment set proposed technique was compared with the alternative convolution techniques. In the fourth experiment set, the adaptation ability of the proposed differential convolution technique was demonstrated by hybridizing it with two models. Performance improvements were observed by considering accuracy values.

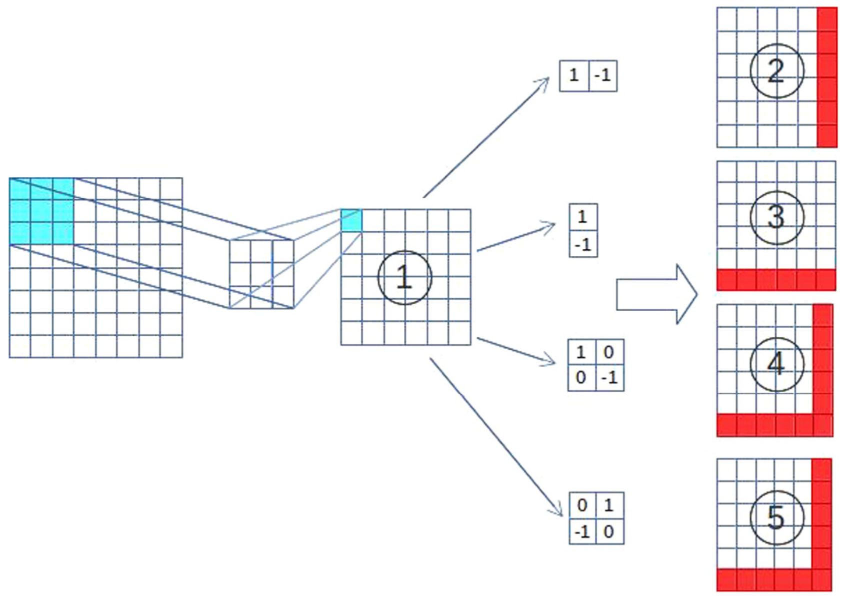

Fig. 2. Differential convolution operation.

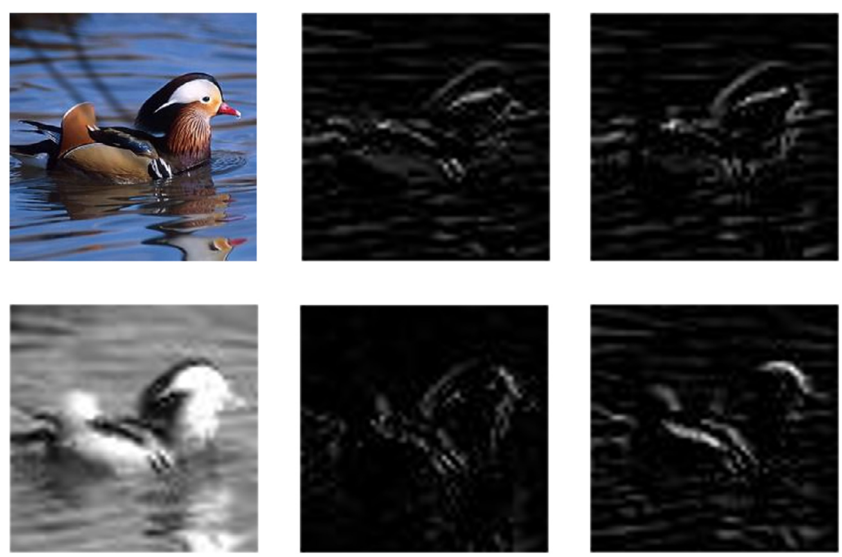

Fig. 3. Top Left: Original image; Bottom Left: Traditional feature map; Others: Differential feature maps.

4.1. Datasets

In the first experiment set, the performance of the Differential CNN is compared to the traditional CNN on 10 different datasets. Ponce bird dataset contains 600 pictures of 6 bird species (Lazebnik, Schmid, & Ponce, 2005). Ponce butterfly dataset includes images of different numbers of 7 butterfly species (Lazebnik, Schmid, & Ponce, 2004). Leeds butterfly dataset includes images of 10 butterfly species (Wang, Markert, & Everingham, 2009). Oxford flowers dataset has a total of 1360 pictures of 17 flowers species (Nilsback & Zisserman, 2006). Swedish leaf dataset contains 1125 pictures of the leaves of 15 tree species (Söderkvist, 2001). KTH-ANIMALS dataset contains 1740 images of 19 different animals (Afkham, Targhi, Eklundh, & Pronobis, 2008). Jena Flowers dataset(JF-30) includes pictures of 30 different flower species (Seeland, Rzanny, Alaqraa, Wäldchen, & Mäder, 2017). Abnormal object dataset (Saleh, Farhadi, & Elgammal, 2013) contains 626 images of 6 different objects. MSRC-v1 dataset (Winn, Criminisi, & Minka, 2005) includes 240 pictures of 8 different objects. MSRC-v2 dataset (Beck & Teboulle, 2009) contains 591 pictures categorized into 20 different classes. In the second experiment set, ILSVRC2012 version of the ImageNet dataset (Russakovsky et al., 2015) containing 1.3 million images (Krizhevsky et al., 2012) was used for measuring the performance increase of differential convolution utilized AlexNet structure. In the third experiment, CIFAR-10 and CIFAR-100 (Krizhevsky & Hinton, 2009) were used for comparing the performance of the 5 different CNN structures with the Differential CNN.

The class distribution of the datasets is given in Table 1.

4.2. Experiment details

Three different experiment sets were performed to demonstrate the effectiveness of differential convolution technique. In the first experiment set, differential convolution operation was adapted to a traditional CNN structure and the classification performance increase was measured on ten different image datasets. CNN structure with 4 convolution layers was used. Since some

Table 1
Class distribution of datasets.

Dataset name	# Of classes	# Of images
KTH-ANIMALS	19	1740
Oxford flowers	17	1360
MSRC v2	20	591
Ponce birds	6	600
Ponce butterflies	7	619
Leeds butterflies	10	832
JF-30	30	1479
Swedish leaf	15	1125
MSRCv1	8	240
Abnormal object	6	626
CIFAR-10	10	60,000
CIFAR-100	100	60,000
ImageNet	1000	1.3 M

of the datasets were unbalanced, precision, recall, and F1-score values are also calculated in addition to the accuracy value. In this experiment set, 5-fold cross-validation was utilized.

In the second experiment set, differential convolution was adapted to AlexNet, a well-known CNN structure, and the performance increase was measured on ImageNet dataset. The network was trained using mini-batches of size 128. The training process started with an initial learning rate, 1×10^{-2}. It continued until the accuracy on the training data stops increasing, and after the learning rate is lowered. This process was repeated multiple times. The learning rates used in this experiment were 1×10^{-2}, 5×10^{-3}, 1×10^{-3}, 5×10^{-4} and 1×10^{-4}, respectively. Data augmentation was also utilized to reduce the overfitting in this experiment set.

In the third experiment set, the Differential CNN was compared with 5 convolutional neural networks in different structures. All compared structures had the same number of convolutional layers. The tested Differential CNN structure contains 5 convolutional layers. The number of filters in each convolution layer was 64, 64, 128, 128, 128, respectively. The filters in the first two convolutional layers were 5×5, while the other convolutional layers contained 3×3 filters. First and fourth convolutional layers were followed by a 2×2 max-pooling layer. The first traditional CNN structure, CNN_No1, contains the same number of filters as the Differential CNN model. In other words, Differential CNN model is differential convolution equipped version of the CNN_No1 model. The second traditional CNN structure, CNN_No2, contained the same number of filters, while the filters were one unit larger for each axis. Since the differential CNN increases the receptive area by 1 unit in each axis, the second model was added to the comparison. The third traditional CNN structure, CNN_No3, had more filters to contain the same number of parameters as the Differential CNN structure for each convolutional layer. Since the differential convolution generates additional feature maps, it increases the number of parameters of the following layer. The third model with more filters to contain exactly the same number of parameters in each layer was implemented to avoid additional parameter advantage. CNN_No3 had a total of 1280 traditional convolution filters, while the Differential CNN model contained only 512 differential convolution filters. The fourth structure was a Tiled CNN (Ngiam et al., 2010) structure. The number of tiles, k, was defined as 2 for the Tiled CNN structure. Therefore, each of the two filter pairs in the structure would create only one feature map, Tiled CNN structure was given twice as many filters as differential CNN structure. The fifth structure was the Dilated CNN (Yu & Koltun, 2015) structure with the same number of filters. Dilated CNN structure utilized 2-dilated convolution. Decaying learning rate policy was used in the all tests. While the initial learning rate was 1×10^{-2}, the final learning rate was $1x10^{-4}$.

In the fourth experiment set differential convolution was adapted to NIN and VGGNet models and performance increase was measured on CIFAR10 and CIFAR100 datasets. NIN model contains 3 convolutional layers applying 3 layered mlpconv as in Lin et al. (2013). Differential convolution adaptation was applied to first layer of each mlpconv filters. VGGNet (Simonyan & Zisserman, 2014) model contains 13 convolutional layers applying 3×3 convolution operation. Batch normalization and DropOut techniques were applied to all models in this experiment set.

4.3. Evaluation metrics

Four different performance measures were utilized to evaluate the classification performance; accuracy, precision, recall, and F1-score. Accuracy is the ratio of correctly classified instances among all instances in a dataset. Accuracy value can be seen in (12) where t is the number of correctly classified instances and n is the number of instances in a dataset.

$$Acc = \frac{t}{n} \tag{12}$$

Precision is the ratio of correctly classified instances among all instances in a class. Precision value of the class c, P_c, can be seen in (13) where n_c is the total number of the instances in class c and t_c is the number of correctly classified instances in class c.

$$P_c = \frac{t_c}{n_c} \tag{13}$$

The recall is the ratio of correctly classified instances among all instances classified in the corresponding class. Recall value of class c is shown in (14) where k_c is the total number of the instances classified as belonging to class c and t_c is the number of correctly classified instances in class c.

$$R_c = \frac{t_c}{k_c} \tag{14}$$

F1-score is a combination of precision and recall measure. F1-score of class c is shown in (15) where P_c is the precision of class c and R_c is the recall of class c.

$$F_c = \frac{2 * (P_c * R_c)}{P_c + R_c} \tag{15}$$

Precision, recall and F1-score values of a classification experiment are calculated by averaging all precision, recall and F1-score values for all classes in the dataset.

Precision, recall, and F1-score can give more accurate values for datasets containing a different number of instances for different classes. Therefore, the values for these four different performance measures are given in the first experiment set. The number of instances is equal for all the classes of the datasets used for the second and third experiments, only the accuracy values are given.

4.4. Results

In the first experiment set, results showed that Differential Convolution utilized model provides better performance in any measures for all datasets. It achieved a performance increase up to 55.29% for accuracy. It had a performance boost up to 54.47% for precision. Performance increase for recall measure was up to 41.75%. It led a performance boost up to 56.43% for F1-score. The overall consideration of all the experiments showed that the average increase in accuracy, precision, recall, and F1 score was 20.91%, 20.68%, 16.19%, and 20.55%. Results of the first experiment were given in Table 2.

Table 2
First experiment set results.

	Differential convolution				Traditional convolution			
	Acc	P	R	F1	Acc	P	R	F1
MSRC-v1	78.76	78.76	79.73	78.97	61.66	61.66	67.60	64.24
Oxford flowers	69.18	69.18	70.89	70.11	58.46	58.46	60.25	59.34
MSRC-v2	49.53	49.34	53.61	51.47	34.12	33.99	38.25	35.20
Ponce birds	75.16	75.16	75.36	75.26	66.66	66.66	69.80	67.32
Ponce butterflies	84.16	83.09	85.91	84.48	75.21	74.20	78.07	75.55
Leeds butterflies	86.22	85.23	86.88	85.97	77.64	77.43	78.88	78.07
KTH-ANIMALS	45.36	45.12	49.67	46.68	29.21	28.48	35.04	29.84
Swedish leaf	91.78	91.78	91.92	91.88	90.66	90.66	91.07	90.87
JF-30	92.12	91.98	95.1	93.26	87.12	86.43	89.39	87.87
Abnormal object	43.36	43.12	54.31	45.12	36.17	35.79	49.86	37.36

Table 3
Results of Differential CNN and other CNN structures.

	CIFAR 10		CIFAR 100	
	Top 1	Top 5	Top 1	Top 5
Differential CNN	**78.53**	**98.28**	**60.97**	**85.21**
CNN No 1	74.68	97.96	55.62	81.86
CNN No 2	74.92	98.04	55.92	82.13
CNN No 3	75.27	98.15	56.60	82.47
Tiled CNN	74.82	98.06	55.73	82.05
Dilated CNN	72.14	94.86	53.25	80.27

Table 4
Results of Differential CNN and other CNN structures.

	AlexNet		AlexNet with Diff. Conv.	
	Top 1	Top 5	Top 1	Top 5
ImageNet	62.5	83.00	**65.81**	**86.94**

Table 5
Results of Differential CNN and other CNN structures.

	CIFAR 10		CIFAR 100	
	Top 1	Top 5	Top 1	Top 5
Differential VGG	**93.58**	**99.87**	**75.06**	**93.12**
Differential NIN	92.44	99.83	72.65	92.56
VGG	92.35	99.81	70.05	88.92
NIN	89.57	99.72	68.53	87.75

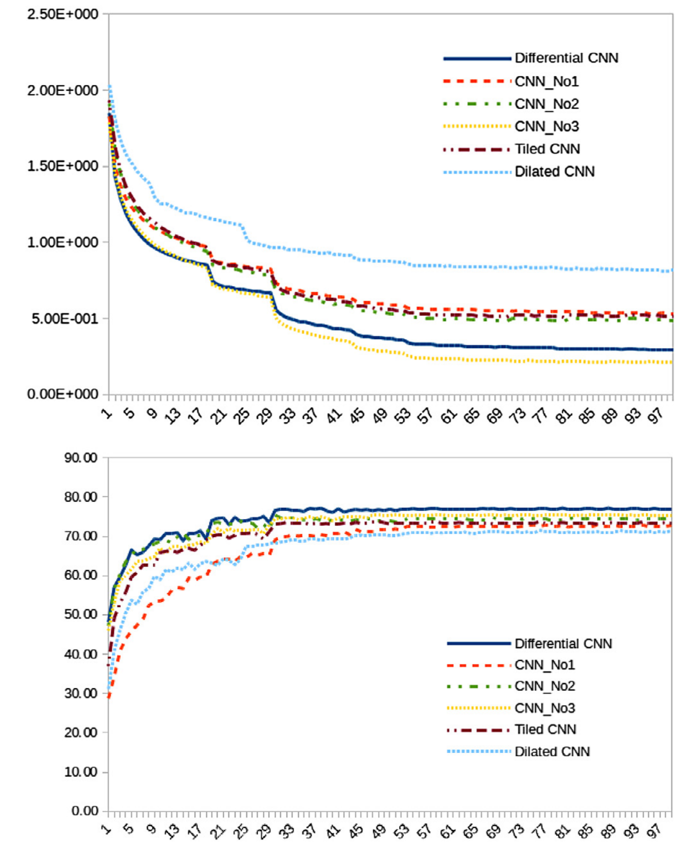

Fig. 4. CIFAR-10 results of third experiment — training error (top) and test accuracy (bottom).

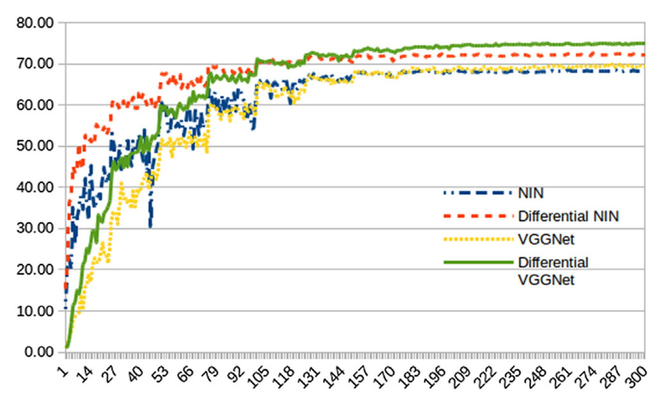

Fig. 5. CIFAR-100 test accuracies of the models in fourth experiment.

In the second experiment set, differential convolution was adapted to AlexNet structure. Alexnet is a popular deep model trained with different policies in the literature. In this experiment, two structures of which one is differential convolution adapted AlexNet and the other one is original AlexNet were trained following the same training policy and performances were measured. This utilization increased the top-1 classification accuracy by 5.3% and top-5 classification accuracy by 4.75% on ImageNet dataset. Results of the second experiment can be seen in Table 4.

In the third experiment set, differential CNN was compared with 5 different CNN structures. The differential CNN model achieved higher classification rates than all compared models. The differential CNN model showed 4.33% and 7.72% higher top-1 classification accuracy than the closest performing model for CIFAR-10 and CIFAR-100 datasets. Results of the third experiment set are given in Table 3. The training data error values and test accuracy rates of all models during the CIFAR-10 experiment are shown in Fig. 4.

In the fourth experiment set, differential convolution was adapted to NIN and VGGNet models. Differential convolution adapted NIN model had increment on the top1 classification accuracy for CIFAR-10 and CIFAR-100 datasets by 3.20% and 6.01%, respectively. Increments in the top-1 classification accuracy of the differential convolution adapted VGGNet model were 1.33% and 7.15% on CIFAR-10 and CIFAR-100 datasets, respectively. In addition, while the VGGNet model achieves more successful results than the NIN model, the Differential NIN model exceeds the performance of VGGNet model. The number of parameters of the VGGNet model is 15 million while the number of parameters of the Differential NIN model is less than 1.4 million. This can be regarded as another indicator of the effectiveness of the differential convolution operation. Results of the fourth experiment set are given in Table 5. Test accuracy rates of all models during CIFAR-100 experiment are shown in Fig. 5.

5. Conclusion

The high representation ability of the deep learning structures has made them popular. This ability is a result of convolution operation. Convolution operation is based on extracting feature maps by sliding the trainable filters on the image and calculating the activation values. These feature maps provide recognition of the various patterns on the image. The popularity of the deep learning structures led to the development of new convolutional models and techniques.

In this article, a novel convolution technique, differential convolution, and an improved back propagation algorithm are proposed. Differential convolution considers directional changes among a pixel and its neighbors. Differential convolution calculates directional change amounts among a pixel and its neighbors to extract pattern orientation of them. As in mathematical differentiation, change amount calculations consider sequential changes by computing the difference among the activations of the pixels. Since images can be defined as superposition of waves with different frequencies, differential convolution generates new feature maps representing signal changes. These maps lead to detection of the big differences on the patterns which are valuable information for classification problems. Differential convolution provides these maps by performing 4 additional convolution operations with predefined constant filters over the feature map generated by traditional convolution. Easy implementation of the differential convolution makes the technique adaptable for all the convolutional structures. Structures adapting differential convolution require an update on the error back-propagation algorithm. Therefore, an improved back propagation algorithm providing an accumulative error over the neighborhood activations is proposed. In this method, neurons additionally acquire error information via neighborhood activations. Briefly, the proposed technique utilizes neighborhood activations in both forward and backward passes. While it represents signal changes in the forward pass, it transmits error information in the backward pass.

In this paper, differential convolution technique was tested on four different experiment sets to demonstrate its performance. In the first experiment set, a traditional CNN structure was equipped with differential convolution and the performance increase was measured on 10 different image datasets. In the second experiment set performance increase of differential convolution adapted AlexNet model was measured. In the third experiment, Differential CNN was compared with different traditional CNN structures and other convolution techniques Tiled CNN, Dilated CNN. In the fourth experiment set differential convolution was adapted to NIN and VGGNet models and performance increase was measured.

In the first experiment set proposed technique increased the accuracy value up to 55.29%. In the second experiment set differential convolution improved the top1 and top5 accuracies of AlexNet by 5.3% and 4.75%, respectively. In the third experiment set, Differential CNN outperformed all compared convolutional structures. Differential CNN provided 4.33% and 7.72% higher top1 performance values than the closest performing model for CIFAR-10 and CIFAR-100. In the last experiment set, differential convolution adapted VGGNet model obtained 93.58% and 75.06% top-1 accuracy values for CIFAR10 and CIFAR100 datasets, respectively. Top-1 accuracy values of differential convolution adapted NIN model for the same datasets were 92.44% and 72.65%. These results demonstrate the effectiveness and adaptability of the differential convolution technique.

Consequently, novelties and efficiencies of the proposed technique can be summarized as follows. It preserves directional change amount among pixel activation which is valuable information to extract pattern orientation, increases the number of feature maps without increasing the number of filters, and provides error feeds through neighborhood activations. Moreover, it is easily adaptable to all CNN structures, provides an increase in performance, and faster error minimization.

References

Afkham, H. M., Targhi, A. T., Eklundh, J. O., & Pronobis, A. (2008). Joint visual vocabulary for animal classification. In *19th international conference on, Pattern recognition, 2008. ICPR 2008* (pp. 1–4). IEEE.

Beck, A., & Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, *2*(1), 183–202.

Chandrakumar, T., & Kathirvel, R. (2016). Classifying diabetic retinopathy using deep learning architecture. *International Journal of Engineering Research & Technology (IJERT)*, *5*(6), 19–24.

Chua, L. O., & Yang, L. (1988). Cellular neural networks: Applications. *IEEE Transactions on Circuits and Systems*, *35*(10), 1273–1290.

Dahl, G. E. (2015). *Deep learning approaches to problems in speech recognition, computational chemistry, and natural language text processing* (Ph.D. thesis), Graduate Department of Computer Science University of Toronto.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *IEEE conference on*, *Computer vision and pattern recognition, 2009. CVPR. 2009* (pp. 248–255). IEEE.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. (pp. 770–778).

Krizhevsky, A., & Hinton, G. (2009). *Learning multiple layers of features from tiny images*: *Tech. rep.*, Citeseer.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Lazebnik, S., Schmid, C., & Ponce, J. (2004). Semi-local affine parts for object recognition. In *British machine vision conference (BMVC'04)* (pp. 779–788). The British Machine Vision Association (BMVA).

Lazebnik, S., Schmid, C., & Ponce, J. (2005). A maximum entropy framework for part-based texture and object recognition. In *Tenth IEEE international conference on*: *Vol. 1*, *Computer vision, 2005. ICCV 2005* (pp. 832–838). IEEE.

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., et al. (1990). Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems* (pp. 396–404).

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Lin, M., Chen, Q., & Yan, S. (2013). Network in network. (pp. 1–10). arXiv preprint arXiv:13124400.

Ngiam, J., Chen, Z., Chia, D., Koh, P. W., Le, Q. V., & Ng, A. Y. (2010). Tiled convolutional neural networks. In *Advances in neural information processing systems* (pp. 1279–1287).

Nielsen, M. A. (2015). *Neural networks and deep learning*. Determination Press.

Nilsback, M. E., & Zisserman, A. (2006). A visual vocabulary for flower classification. In *IEEE Computer Society conference on*: *Vol. 2*, *Computer vision and pattern recognition* (pp. 1447–1454). IEEE.

Ravi, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., et al. (2017). Deep learning for health informatics. *IEEE Journal of Biomedical and Health Informatics*, *21*(1), 4–21.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, *115*(3), 211–252. http://dx.doi.org/10.1007/s11263-015-0816-y.

Saleh, B., Farhadi, A., & Elgammal, A. (2013). Object-centric anomaly detection by attribute-based reasoning. In *proceedings of the IEEE conference on computer vision and pattern recognition*. (pp. 787–794).

Sankar, M., Batri, K., & Partvathi, R. (2016). Earliest diabetic retinopathy classification using deep convolution neural networks. *International Journal of Advanced Engineering Technology*, *2*(1), 460–470.

Seeland, M., Rzanny, M., Alaqraa, N., Wäldchen, J., & Mäder, P. (2017). Plant species classification using flower imagesa comparative study of local feature representations. *PloS One*, *12*(2), e0170629.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. (pp. 1–16). arXiv preprint arXiv:13126229.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556.

Söderkvist, O. (2001). Computer vision classification of leaves from Swedish trees. Master's Thesis.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1–9).

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015b). *Going deeper with convolutions* (pp. 1–9). Cvpr.

Tennakoon, R., Mahapatra, D., Ro, P., Sedai, S., & Garnavi, R. (2016). Image quality classification for dr screening using convolutional neural networks. In Proceedings of the ophthalmic medical image analysis international workshop. (pp. 664–667).

Wallace, G. K. (1992). The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, *38*(1), 18–34.

Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., & Fergus, R. (2013) Regularization of neural networks using dropconnect. In *International Conference on Machine Learning*. (pp. 1058–1066).

Wang, J., Markert, K., & Everingham, M. (2009). Learning models for object recognition from natural language descriptions. (pp. 1–11).

Winn, J., Criminisi, A., & Minka, T. (2005). Object categorization by learned universal visual dictionary. In *Tenth IEEE International conference on*: *Vol. 2*, *Computer vision, 2005. ICCV 2005* (pp. 1800–1807). IEEE.

Yu, F., & Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. (pp. 1–13).