

A New Approach to Image Classification by Convolutional Neural Network

Shamim Ibne Shahid¹, Md. Shahjahan²

¹Khulna University of Engineering & Technology, Bangladesh

²Khulna University of Engineering & Technology, Bangladesh
titu2297@yahoo.com, jahan@eee.kuet.ac.bd

Abstract— The backpropagation gradient through a convolutional neural network to train the weights of the filters(kernel) is the same as a regular deep network which is to calculate the partial derivative of the error function at the output of fully connected layer with respect to these weights. In this paper we introduce the Fixed Kernel Convolutional Neural Network that skips the training process of the filters between the convolutional layers and exploits the idea of a predefined kernel remaining fixed throughout the training process and a simple feedforward layer that can help acquiring invariances among images having similar kind of patterns at different locations. We compare the performance of the proposed architecture with few other classifying techniques which it outperforms using the Caltech-256 database.

Keywords— Averagepooling, Maxpooling, Feedforward network, Target vector.

I. INTRODUCTION

Convolution neural networks(CNN) are designed in a way so it can perform classification/processing of data that are in the form of a matrix such as a colored image for example which consists of three 2D arrays of identical size. The basic idea behind a CNN which is known as convolution is to use a kernel consisting of an array of weights that is slid across a matrix data outputting another matrix whose elements are the summations of dot products of these weights with the corresponding elements of the superimposed part of the first matrix. This step is always followed by subsampling and might be repeated several times until the fully connected layer. For a while now it has been believed that the use of neural network is most likely to transcend human capability for doing things like pattern recognition, image classification, self-driving etc. Although that sounds tangible, there is still a lot of improvements to be done. Until now, we have seen a few theories that have shown remarkable success in recognizing visual image. The first concept of Convolutional Neural Network was presented by Yann LeCun et al. in [1] in the 1980s. which was composed of two kinds of layers, respectively called convolutional layers and subsampling layers. However, many years after that, there was still not a major breakthrough of CNN. One of the main reason was that it was not possible to get ideal result for large size images.

In 2006, Geoffrey Hinton and Salakhutdinov published an article in "Science" [2], which opened the new

door to a deep learning era. Hinton suggested that with the neural network with multiple hidden layers, it is possible to improve the accuracy of classification and prediction by improving different degrees of abstract representation of the original data. Since then artificial neural network has got huge attention. The network we present in this paper possesses a structure that uses the benefits of multilayered feedforward network [3] along with the convolution of image matrix and unlike the conventional CNN structure where several filters have to be learned by the network, we used only one fixed filter. In section 2, we discuss the related works, in section 3, we describe the overall structure of the network structure, in section 4 and 5 the training and testing procedures are described respectively. In section 6, we compare the result obtained by this network with some other classifying methodologies. Finally, we discuss about the lackings and future possibilities of the presented network.

II. RELATED WORK

There are many classification systems based on multiple modules including feature extraction, segmentation and other image processing techniques. They are used for commercial purposes. The main objective of image processing is to reduce the size of data. Combined with CNN structure such works include the article of Hubel and Wiesel [4] on the structure of the visual cortex in a cat's brain. Even though it has inspired numerous architectures of artificial neural networks designed specifically for visual data processing, it was quite complicated to mimic the process on image recognition. Convolutional neural networks proposed by Yann LeCun [5] presents a structure embracing the idea of various types of neurons organized within one network, each one of them having its specific function in image processing. In [6], [7], [8], utilizing multi-level features in CNNs through skip-connections has been found to be effective for various vision tasks. But however, it is not appropriate in dealing with microscopic images. Convolutional networks are capable of processing the image data with minimum or no preprocessing. There are other notable network architecture innovations which have yielded competitive results. The Network in Network (NIN) [9] structure includes micro multi-layer perceptron into the filters of convolutional layers to extract more complicated features. In Deeply Supervised Network (DSN) [10], internal layers are directly supervised by auxiliary classifiers, which can strengthen the gradients received by earlier layers. In [11], Deeply-Fused Nets (DFNs) were proposed to improve

information flow by combining intermediate layers of different base networks. The augmentation of networks with pathways that minimize reconstruction losses was also shown to improve image classification models [12]. Highway Networks [13] were amongst the first architectures that provided a means to effectively train end-to-end networks with more than 100 layers using bypassing paths along with gating units. Highway Networks with hundreds of layers can be optimized without difficulty. Recent variations of ResNets [14] show that many layers contribute very little and can in fact be randomly dropped during training. This makes the state of ResNets similar to recurrent neural network. FractalNets [15] repeatedly combine several parallel layer sequences with different number of convolutional blocks to obtain a large nominal depth, while maintaining many short paths in the network.

III. BRIEF DESCRIPTION OF THE PROPOSED ARCHITECTURE

The overall system flow is shown in Fig. 1. It begins with taking RGB images as input which are all resized equally into (200×200) .

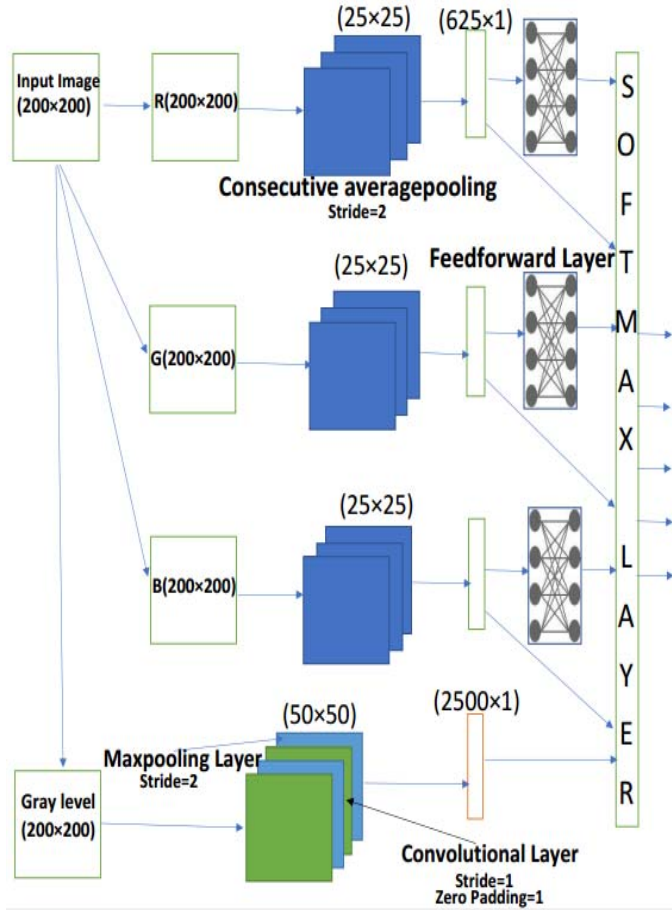


Fig. 1. Proposed Fixed Kernel CNN Structure

Each color component matrix of the RGB image is separated while at the same time it is converted into a gray level image. The component matrices are made to separately go through an

average pooling layer. Here we average every 2×2 submatrices of the image component which reduces the matrix size into half of its previous value. For example, if we subsample a 200×200 matrix this way, the output layer will yield a matrix of the size 100×100 . This process is repeated two more times until the size of the image component reduces to 25×25 . The same procedure is followed for all the three-color components of the image. Meanwhile, we apply convolutional layer on the gray level image. For this process the convolution kernel applied here is well suited for extracting the shape of the image. In order to reduce the size of the image here we use a maxpooling layer after each convolution which outputs the maximum value for every 2×2 element of the gray level image. A consecutive convolution and maxpooling operation is applied two times on this network and as a result a 50×50 matrix yields for every input image. Now each 25×25 matrix derived from the three color-component matrices and the 50×50 from the gray level image are resized into vertical matrices of the size 625×1 and 2500×1 respectively. The reason why we have to form these vertical matrices is that a fully connected layer can not accept 2D input. The final three vertical matrices from the RGB components are used as input of the feedforward network as shown in Fig. 1. We fix a target vector of 25 elements for each of the feedforward network whose values are dependent on the output of the last average pooling layer (it will be discussed in the next section). The outputs of every feedforward network are piled up followed by their inputs one after another as shown in the following Fig. 1 to form a vertical matrix which is further integrated with the 2500×1 matrix yielded from the gray level image and fed into the SoftMax layer.

Here in the network structure, we used only one predefined kernel for convoluting on the gray scale image. The filter is so designed that it prioritizes the edge of the object and ignores the background. In a conventional CNN, different kernels are trained for convoluting in different convolutional layers. In the Fixed Kernel CNN, there are two convolutional layers each of which uses the same fixed kernel for convolution. In this way when a gray scale image is repeatedly convoluted and pooled, the effects of different kind of shapes present in the image more or less tend to shift towards the center which is convenient to acquiring invariances. The patterns of the color components are also taken into consideration after reducing their sizes by repeated average pooling and passing them through a feedforward network and a SoftMax layer after rasterizing. The network structure is favorable to detecting similar attributes of a particular class while remaining flexible to their differences.

IV. TRAINING

The network keeps taking input images from the training sample folders and after being resized equally the color component matrices are sent to a consecutive average pooling layers. The outputs of the last pooling layers are resized into a vertical matrix for all the color component matrices of the training sample which then acts as input for both the feedforward and SoftMax layer. Each of the feedforward layer

consists of 10 hidden neurons. Here, we use tansig function for the hidden layer and pureline function for the output layer neurons. The target values for the output layer of feedforward network are fixed as a 25×1 vector formed by taking the maximum value of each 5×5 submatrix of the output from the last average pooling layer. The feedforward network uses the following error function for learning: -

$$E = (1/2) \sum_{j=1}^{25} (d_j - y_j)^2$$

where, y_j = Output signal of j-th neuron ;
 d_j = j-th element of the target vector;

Now once the feedforward networks are trained, their input values are again forwarded through the network which contributes to the input matrix of the SoftMax layer.

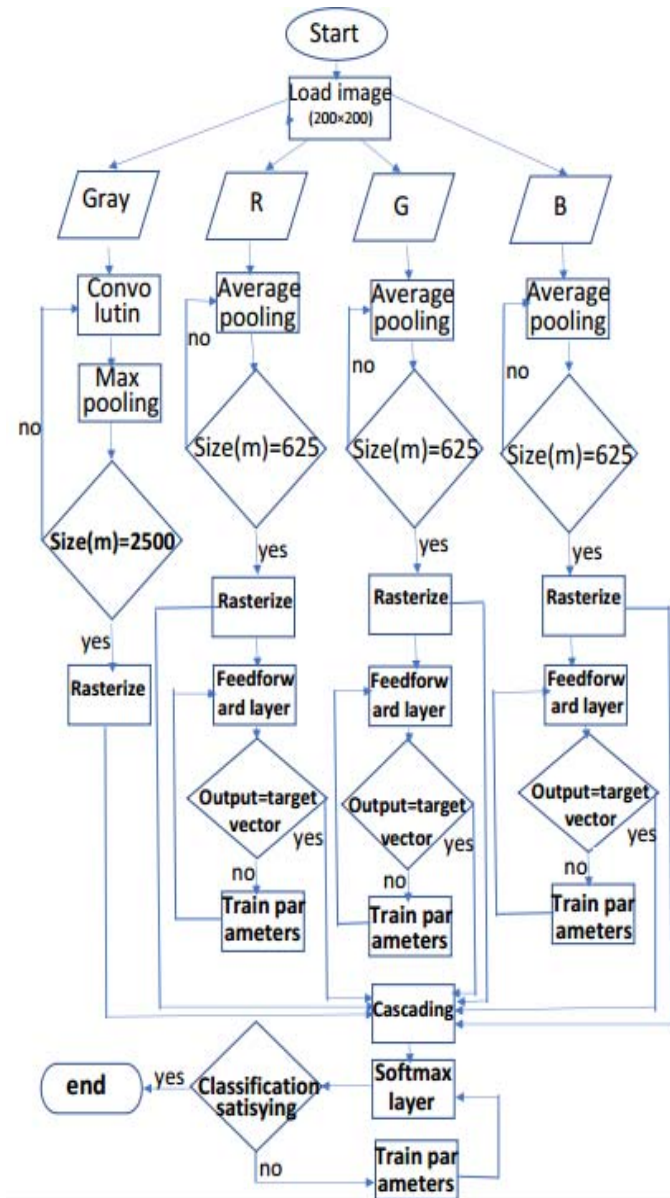


Fig. 2. Training Flow Chart

For convoluting the gray level image, the following predefined kernel is used.

$$K = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

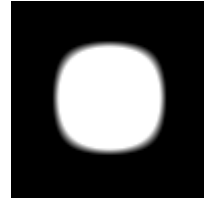


Fig. 3 (a) .

Fig. 3 (b) .

The filter is shown as image on the right side after its size was changed into 100×100. Each convolution is followed by a maxpooling layer whose function is the same as the average pooling layer except that here we have to take the maximum value for each 2×2 submatrix of the previous layer output. The input matrix for the SoftMax layer is thereby set by cascading vertically the outputs of the feedforward layers followed by their inputs and also the result which is obtained from repeated convolution and maxpooling operation on the gray level image. The target vector for this layer is set as a vertical probability matrix by keeping the element at the position which specifies the type of the input image as 1 while all other elements as 0. The training process of the network finishes once the SoftMax layer is trained against these target values that leads to satisfactory classification of the training image.

V. TESTING

For testing purpose, an image is processed in the same way as it was used to train the network. The color components after being separately made to pass through the consecutive average pooling layers are given as input to the feedforward layers whose output values help constituting the final input of the SoftMax layer. At the same time the gray level values of the image is also passed through the convolution and maxpooling layer in order. Now, we simply pile up all these values as shown in Fig. 1 and pass it through the trained SoftMax network. If we train the network with five classes, the output of the SoftMax layer would be a vertical matrix of five elements each of which representing the probability of the image to fall under the class specified by the row number of that element. In this network, we classified the image according to the class for which the image shows maximum probability.

VI. EXPERIMENT & RESULT











For experimenting the performance of the network, we used the Caltech-256 Database [16]. Here, out of the 256 type images, we selected 5 folders that had comparatively more number of items. After removing the black and white images, the first half portion(serial wise) of each folder was used for

training while the second half portion for testing purposes. The data characteristics so evolved are shown below:

Table 1. Data Characteristics

	Type	Training	Testing
Class 1	Motorbike	394	393
Class 2	Leopard	95	95
Class 3	Airplane	400	400
Class 4	Human face	218	217
Class 5	Watch	101	100
Total		1208	1205

Table 2. Sample images of the classes

	Training image	Testing image
Class 1		
Class 2		
Class 3		
Class 4		
Class 5		

After the network was trained by images of the following 5 classes, we let it classify a mixed folder containing a total of 1205 images. The %Precision (true positive/total image identified in a class) and %Recall (true positive/total image of that class in test data) shown by the Fixed Kernel CNN structure and a conventional convolutional neural network [17] with three input channels after having it trained with the same data mentioned in Table 1, are given below:

Table 3. Network Performance

	Fixed Kernel CNN		Conventional CNN	
	%Precision	%Recall	%Precision	%Recall
Class 1	99.22	97.45	97.20	97.20
Class 2	98.95	100	96.84	96.84
Class 3	98.26	99.25	97.70	95.75
Class 4	99.52	97.23	97.19	93.58
Class 5	87.85	94	69.82	81
%Avg.	96.76	97.58	91.75	92.87

It is worth mentioning that the network performance is generally superior with increasing number of training samples. It also depends on what kind of images are used for training the network. If the training samples of a particular class contain variety then the network will be able to recognize different looking images of that class but at the same time it can also contribute to more misclassification owing to the fact that the various training samples of that class might share common attributes with other classes. For example, let us consider few misclassifications made by the network:



Fig. 4. Few misclassified images

Here the first image (a) was classified as motorbike since both type of images contain round figures and white background. In case of the second image (b), it was classified as watch as in the training folder, a lot of watches had color that resembles exactly as the background of this image. The third

image (c) was classified as airplane because of the similarity of the shirt color with a number of airplane images used for training that had the background of the sky in it.

We also calculated overall %accuracy (fraction of the total image on test data set that are rightly classified) of this network, the mentioned conventional CNN structure, support vector machine classifier [18] using bags of key points features [19], k-nearest neighbor classifier [20] and patternet [21] for the given data set in Table 1 and found them to be as:

Methodology	%Accuracy
Proposed Fixed kernel CNN	98
Conventional CNN	94.68
SVM classifier	92
KNN classifier	83.15
Patternet	82.65

VII. CONCLUSION

The network structure we present here accounts only the color component and shape which is why it is more suitable to classify images of simple visual objects. Unlike many deep CNN structures which can take into account multiple features by training various filters at each convolutional layer, the Fixed Kernel CNN method is not apparently favorable to classify images which require lots of complex feature to be grasped by the network until it is efficiently integrated with large number of parameters that can be derived from different image processing/segmentation techniques. However, the network is particularly useful for programming a relatively much faster classifier that involves less mathematics throughout the process.

REFERENCES

- [1] Lecun, Y. "Generalization and Network Design Strategies." Connectionism in perspective, 1989.
- [2] Hinton G E; Salakhutdinov R R. "Reducing the dimensionality of data with neural networks".Science, 2006.
- [3] Simon Haykin, Neural Network A Comprehensive Foundation, Ontario, Canada, CA:Mcmaster University.

- [4] D.H Hubel and T.N Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex." J. Physiology, vol. 160, no. 1, pp 160-154, 1962.
- [5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", in Proc. of the IEEE, vol. 86, no. 11, pp. 2278-2324, nov 1998.
- [6] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In CVPR, 2015.
- [7] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015.
- [8] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun. Pedestrian detection with unsupervised multi-stage feature learning. In CVPR, 2013.
- [9] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang and Z. Tu, "Deeply-supervised nets". In AISTATS, 2015.
- [10] M. Lin, Q. Chen, and S. Yan, "Network In Network". In ICLR, 2014.
- [11] J. Wang, Z. Wei, T. Zhang, "Deeply-fused nets". In ICML, 2016.
- [12] Y. Zhang, K. Lee, "Augmentation supervised neural networks with unsupervised objectives for large scale classification". In ICML, 2016.
- [13] R. K. Srivastava, K. Greff, and J. Schmidhuber. Training very deep networks. In NIPS, 2015.
- [14] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger. Deep networks with stochastic depth. In ECCV, 2016.
- [15] G. Larsson, M. Maire, and G. Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. arXiv preprint arXiv:1605.07648.
- [16] Caltech-256 website, [online]. Available: http://www.vision.caltech.edu/Image_Datasets/Caltech256/
- [17] Train neural network for deep learning, website, [online], available: <https://www.mathworks.com/help/nnet/ref/trainnetwork.html>
- [18] Train an image category classifier-MATLAB, website, [online], available: <https://www.mathworks.com/help/vision/ref/trainimagecategoryclassifier.html>
- [19] Csurka, G., C. R. Dance, L. Fan, J. Willamowski, and C. Bray Visual Categorization with Bag of Keypoints, Workshop on Statistical Learning in Computer Vision, ECCV 1 (1-22), 1-2.
- [20] K-nearest neighbor classification – MATLAB, website, [online], available: <https://www.mathworks.com/help/stats/classificationknn-class.html>
- [21] Pattern recognition network – MATLAB, website, [online], available: <https://www.mathworks.com/help/nnet/ref/patternnet.html>