Densely Connected Network in Network

Neji Kouka
MARS laboratory, ISITCom hammam Sousse, University of Sousse, Tunisia

Jawaher Ben Khalfa
MARS laboratory, ISITCom hammam Sousse, University of Sousse, Tunisia

Jalel Eddine Hajlaoui
MARS laboratory, ISITCom hammam Sousse University of Sousse, Tunisia

ABSTRACT
Recent work has shown that convolutional neural networks can be more precise, deeper and more efficient for training if they integrate shorter connections between the layers near the input and those near the output. In this paper, we adopt this observation and propose a new deep network structure called "densely connected Network in network" (DcNiN), which connects each layer of MLPconv to all other layers in the same structure in ways as the own maps of MLPconv. Characteristics for each layer are used as inputs in all subsequent layers. The interesting advantages presented by DcNiN are several. Examples include strengthening feature propagation, reducing the leakage gradient problem, reducing the number of parameters, and encouraging feature reuse. We evaluate our proposed architecture against a widely known and highly competitive database (CIFAR-10). DcNINs achieved 99.9611% accuracy on this test set.

Keywords
Convolutional Neural Networks (CNNs), Image recognition, Network in Network (NiN).

1. INTRODUCTION
With the increase in the depth of the CNN model, a new research problem emerges: as information about the input or gradient passes through many layers, it can vanish and "wash out" by the time it reaches the end of the network. Several techniques are used to solve this problem. We note among them : Stochastic depth [1], FractalNets [2], ResNets [3] and Highway Networks [4]. Stochastic depth [1], consists of shortening ResNets by randomly removing layers during formation. Highway Networks [4] and ResNets [3] by pass the signal from one layer to the next via identity connections. FractalNets [2] repeatedly combines multiple sequences of parallel layers with different number of convolutional blocks to achieve large nominal depth, while maintaining many short paths in the network. Although these different techniques vary in training procedure and network topology, they all share one main characteristic: they create short paths from the first layers to the later layers. In this article, we propose an architecture that anchors this idea in a simple communication model named "Densely Connected Network in Network". Where all the layers are connected directly to each other in order to ensure maximum information flow between the layers of the network. The advantages of the architecture are verified experimentally on CIFAR-10 classification data sets. The contributions of this work are: (i) We propose a new architecture for the DcMLPconv layers which allows to have "DcNIN" models with considerably improved performance (ii) We propose a new way to use batch normalization in the

DcNIN model to regularize and normalize them correctly and avoid overfitting during training. (iii) We present a detailed experimental study of deep model architectures that examines in depth several important aspects of DcMLPconv layers. (iv) Finally, we show that our proposed DcNIN architectures obtain interesting results on CIFAR-10 significantly improving the accuracy of DcNIN. The rest of this article is organized as follows: In Sect. 2, an overview of related works is given. Section 3 is about strategy. Experimental evaluations and comparative analysis are presented and discussed in Sect. 4. Section 5 is devoted to implementation details. The advantages and limitations of DcNIN are reported in Sect. 6. The work is concluded in the last section.

2. RELATED WORKS
To improve the performance of CNN, several techniques can be used. Among these techniques we cite: increasing the depth [5, 3, 6] and/or the width [8, 7], modifying the convolution parameters [8, 9] and reducing the size of the convolution filter [9, 10, 11], change the number of channels and feature map [11, 7]. The modification at the level of the pooling layers [12, 13, 14, 15, 16–24] and of the activation function [25, 26].

Simple linear filters are at the heart of the computations within the convolutional layer of classical CNNs. In contrast, in the lattice-based model, non-linear filters are exploited instead of classical simple linear filters such as multi-layer perceptron (MLP) [9, 16, 10,27]. Many works have exploited nonlinear filters such as the NIN model [16], DNIN [9], DrNIN [10], WDrNIN [27]. The "Network In Network" model [16], consists of several "MLPconv" layers which are stacked in a successive way. The "MLPconv" layer consists of a linear convolution layer and a two-layer MLP with a ReLU unit used as an activation function. Figure 1 illustrates the overall structure of the architecture,

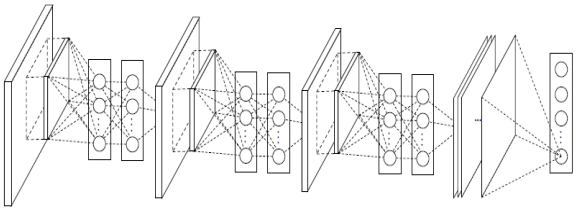

Fig 1: Network In Network (NIN)

The "Deep Network In Network (DNIN)" model [9], illustrated in Fig. 2., represents a modification of the NIN model [16]. This model consists of blocks of DMLPconv stacked in a successive way and which integrates two convolutional layers of size 3 × 3 and a nonlinear activation unit "eLU" instead of ReLU.

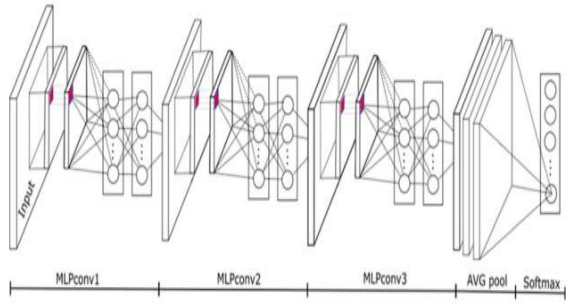

Fig.2 - Deep Network in Network

The DrNIN model proposed in [10] is a model based on DNIN [9]. It represents an improvement of the architecture of DNIN [9] by reformulating the convolutional layers of DMLPConv as residual learning functions. Figure 3 illustrates the DrNIN model composed of 3 DrMLPconv layers.

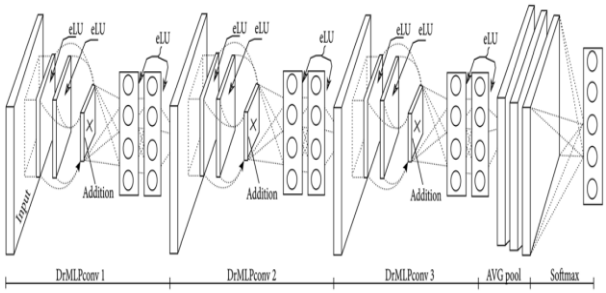

Fig. 3 - Deep residual network in network

In [27], the authors proposed the Wide deep residual networks in networks (WDrNIN) model which represents a broader model of DrNIN. In this model, the authors increased the width of the DrNINs and decreased the depth.

3. THE PROPOSED METHODS

We describe our different DrNIN model configurations for CIFAR-10. In these model configurations, the convolutional layers have 3 × 3 filters and follow a simple design rule: the layers have the same feature map size and the same number of filters; on the other hand, the exploitation of a pooling layer which is generally inserted periodically after each DcMLPconv structure. In our architecture, we perform downsampling using the maximum clustering layers of size 3 × 3 which have a stride of 2 (3 × 3/ST.2). The network ends with a global average pooling layer and a softmax layer. Figure 4 illustrates an example of the DcNIN model composed of three DcMLPconv layers.

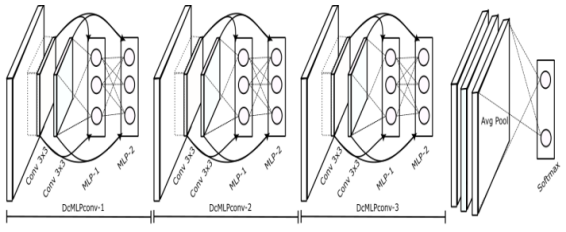

Fig.4 - DcNIN.

The overall structure of DcNIN generally consists of layer 3 DcMLPconv. Table 1 shows the overall structure of DcNIN. Additionally, it displays the output sizes after each layer used in the model. The first table describes the overall structure of DcNIN.

Table 1 : The structure of DcNIN.

Layer name	Output size
DcMLPconv-1	32 × 32
Max-pool	16 × 16
DcMLPconv-2	16 × 16
Max-pool	8 × 8
DcMLPconv-3	8 × 8
Global average pooling	1 × 1

3.1. The "DcMLPconv" structure:

The DcMLPconv structure, shown in Figure 5, consists of two convolution layers of size 3 × 3, MLP layers. These different layers are followed by a rectified linear unit (ReLU). For each layer in DcMLPconv, the feature maps of all previous layers are used as inputs, and its own feature maps are used as inputs in all subsequent layers.

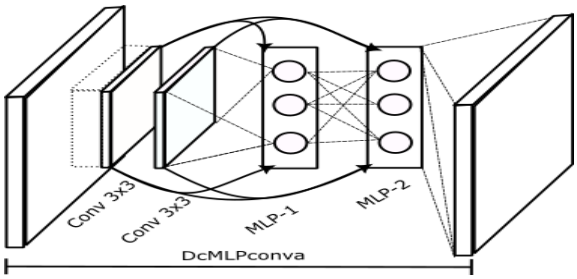

Fig 5 - DcMLPconv.

Let DcMLPconv(X) be the DcMLPconv structure, where X is a list of layers used in the structure. For example, DcMLPconv (3) denotes the base layers of DcMLPconv. DrMLPconv (3, B) denotes the DcMLPconv structure (3,) with normalization layers (B). The different structure of our DcMLPconv is presented in the table 2.

Table.2 : The configurations of DcMLPconv.

Layer	DcMLPconv (X)	
(x)	DrMLPconv (3)	DrMLPconv (3, B)
Conv-1	3 × 3 × 192/st. 1/pad 1/ReLU	3 × 3 × 192/st. 1/pad 1/ReLU/BN
Conv-2	3 × 3 × 192/st. 1/pad 1/ReLU	3 × 3 × 192/st. 1/pad 1/ReLU/BN
MLP-1	1 × 1 × 192/st. 1/pad 0/ReLU	1 × 1 × 192/st. 1/pad 0/ReLU/BN
MLP-2	1 × 1 × 192/st. 1/pad 0/ReLU	1 × 1 × 192/st. 1/pad0/ReLU/BN

3.2. Batch Normalization in DcNIN

The use of regularization represents a solution to avoid the problem of overfitting. A batch normalization [28] is already applied for DcNIN to provide a regularization effect. The batch

normalization layer was introduced in 2015 by Google researchers. The batch normalization layer was introduced in 2015 by Google researchers. It is generally located after fully connected layers or convolutional layers, and before the non-linearity. It alleviates problems with internal covariance shift in feature maps by normalizing the layer's output distribution to zero mean unit variance. Using this layer makes networks more resilient to bad initialization, moreover, it eliminates the need for using the dropout layer.

4. EXPERIMENTAL RESULTS

We evaluate our configurations on a reference data set: CIFAR-10. This database includes 60,000 RGB images of 32x32 size grouped into 10 image classes. These different images are separated between 50,000 total training images and 10,000 test images. The networks used for this database consist of three "DcMLPconv" layers. The first two "DcMLPconv" layers of all experiments are followed by a maximum pooling layer. In the following, we will refer to networks as their DcMLPconv(X) structures.

4.1. The effect of normalization layers in DcNiN

Experimental studies show that a network with batch normalization achieves higher accuracy than a network without batch normalization. The value of the increase in accuracy is about 0.36%. Table 3 represents a comparaion between the accuracy of different configurations of CIFAR-10 with a mini batch size equivalent to 128. Our results were obtained by calculating the average over 5 executions.

Table 3: Accuracies of different models on CIFAR-10.

Method	Accuracy
DcMLPconv(3)	94,49%
DcMLPconv(3,B)	94,85%

4.2. Discussions

The main objective of this work is to examine and evaluate the success of our proposed architecture in image classification and to compare the performances found with the models cited in the literature. As shown in Table 4, the DcNIN model based on DcMLPconv(3) achieved slightly better accuracy than most work that uses nonlinear filters from the literature with the CIFAR-10 dataset such as NIN[16]. , DNIN[9], DrNIN(L=5) [10] and WDrNIN [27]. Additionally, DcNIN provides classification accuracy that allows it to have a well localized location between multiple jobs. Moreover, the experimental results show that exploiting the batch normalization layer yields a useful effect in reducing the error of the classification test. Table 4 represents a comparison between the proposed model and the state of the art on the CIFAR-10 database. The results of our work are presented with a mini batch size equivalent to 128 and calculating the average of 5 executions.

Table 4 : CIFAR-10 test error. The results of our work are presented with the size of the mini batch equivalent to 128. Our results were obtained by calculating the average over 5 executions

Ref	Method	Error test(%)
(I.Goodfellow et al, 2013)	Maxout network (k=2)	9.38
(Lin M et al, 2013)	NIN	08.81
(C-Y. Lee et al, 2015)	DSN	8.22
(Alaeddine, H et al, 2021)	DNIN	07.46
(Hmidi A et al 2021)	DrNIN(L=5)	07.21
(Alaeddine H et al, 2023)	WDrNIN	06.447
Our	DcNIN	5.15
(K. He et al , 2016)	ResNet	6.43
(Zagoruyk S et al, 2017)	Wide Resnet(28,10)	3.89

5. IMPLEMENTATION DETAILS

Our models are trained using a "Root Mean Square Propagation Algorithm" with a batch size equivalent to 128 and a weight decay of 0.0001. We initialized the weights in each layer from a normal to mean random distribution with a standard deviation equivalent to 0.01. We initialized the learning rate to 0.01 and divide by 10 twice before the end. The network is trained for about 160 cycles at most on the CIFAR-10 training set in a central processing unit (CPU). The implementation is provided by the python language based on the "Keras" deep learning library to classify and recognize images.

6. ADVANTAGE AND LIMITATIONS

The proposed model provides competitive test accuracy that exceeds the accuracy of other linear filter-based models such as [9, 16, 10, 27] which allows it to rank prominently among the various works reported in the literature. DcNIN provides interesting test errors against the baseline. The importance of DcNIN also stems from its homogeneous structure which makes it well suited for implementation as an image recognition system in embedded system applications. However, DcNIN incorporates drawbacks and limitations that mainly reside in the number of convolution kernels. This negatively affects the number of parameters, computational complexity and memory.

7. CONCLUSIONS

We hope In this paper, we proposed a new convolutional network architecture, which we call Densely Connected Network in Network (DcNIN). It introduces direct connections between any two layers with the same feature map size.

DcNINs tend to produce a steady improvement in accuracy with increasing number of parameters, with no signs of performance degradation or overfitting. In multiple contexts, he has achieved cutting-edge results on CIFAR 10 compared to work based on nonlinear filters. In addition, a proposed detailed study of DcNIN is presented describing in detail the effect of different layers on improving accuracy. The results are described as acceptable compared to other architectures tested on CIFAR-10 datasets. Future work should focus on designing new versions of CNN models that can meet or exceed the level of accuracy of this proposed model requiring shorter training time with less parameter consumption.

8. REFERENCES

[1] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger. Deep networks with stochastic depth.In ECCV, 2016.

[2] G. Larsson, M. Maire, and G. Shakhnarovich. Fractalnet: Ultra-deep neural networks withoutresiduals. arXiv preprint arXiv:1605.07648, 2016.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR,2016.

[4] R. K. Srivastava, K. Greff, and J. Schmidhuber. Training very deep networks. In NIPS, 2015.

[5] Gao S, Miao Z, Zhang Q, Li Q (2019) DCRN: densely connected refinement network for objectdetection. J Phys: Conf Series., 1229, Article ID 012034

[6] Zeiler MD, Fergus R (2013) Visualizing and understanding convolutional networks.

[7] Zagoruyko S, Komodakis N (2017) Wide Residual Networks, 1605.07146, arXiv, pp 87.1–87.12 https:// doi.org/10.5244/C.30.87.

[8] Iandola F, Han S, Moskewicz M, Ashraf K, Dally W, Keutzer K (2017) SqueezeNet: alexnet-levelaccuracy with 50x fewer parameters and connected convolutional networks.

[9] Alaeddine, H., Jihene, M. Deep network in network. Neural Comput & Applic 33, 1453–1465 (2021). https://doi.org/10.1007/s00521-020-05008-0

[10] Hmidi A, Malek J (2021) Deep Residual Network in Network, Computational Intelligence and Neuroscience. Hindawi 6659083:1687–5265. https://doi.org/10.1155/2021/6659083

[11] C. Szegedy et al., "Going deeper with convolutions", CoRR, vol. abs/1409.4842, 2014, [online] Available: http://arxiv.org/abs/1409.4842.

[12] Gong Y, Wang L, Guo R, Lazebnik S (2014) Multi-scale orderless pooling of deep convolutionalactivation features. http://arxiv.org/abs/1403.1840

[13] Graham B (2014) Fractional max-pooling. https://arxiv.org/abs/1412.6071

[14] He K, Zhang X, Ren S, Sun J (2014) Spatial pyramid pooling in deep convolutional networks forvisual recognition. http://arxiv.org/abs/1406.4729.

[15] Lee C, Gallagher P and Tu Z (2015) Generalizing pooling functions in convolutional neural networks: mixed gated and tree,"https://arxiv.org/abs/1509.08985.

[16] Lin M, Chen Q, Yan S (2013) Network in network. http://arxiv.org/abs/1312.4400.

[17] Murray N, Perronnin F (2015) "Generalized max pooling," in Proceedings of the 2015 IEEEconference on computer vision and pattern recognition (CVPR), pp 2473–2480, Boston, MA USA

[18] Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In: proceedings of the 27th international conference on machine learning (ICML 2010), pp 807–814

[19] Raiko T, Valpola H, Lecun Y (2012) "Deep learning made easier by linear transformations inperceptrons," in Proceedings of the Fifteenth International Conference on Artificial Intelligence andStatistics (AISTATS12), N. D. Lawrence and M. A. Girolami, Eds., vol. 22, pp. 924–932, La Palma,Canary Islands, Spain

[20] Romero A, Ballas N, Kahou SE, Antoine C, Gatta C, Bengio Y (2014) FitNets: hints for thin deepnets

[21] Schmidhuber J (1992) Learning complex, extended sequences using the principle of historycompression. Neural Comput 4(2):234–242

[22] Simonyan K, Zisserman A (2014) Very deep convolutional networks for largescale image recognition

[23] Springenberg J, Dosovitskiy A, Brox TT, Riedmiller M (2014) Striving for simplicity: the all convolutional net. http://arxiv.org/abs/1412.6806

[24] Srivastava N, Geoffrey H, Krizhevsky A, Ilya S, Ruslan S, Dropout (2014) A simple way to prevent neural networks from overfitting. J Mach Learn Res 15:1929–1958.

[25] Chang J-R, Chen Y-S (2015) Batch-normalized maxout network in network.http://arxiv.org/abs/1511. 02583

[26] Liao Z, Carneiro G (2016) On the importance of normalisation layers in deep learning with piecewise linear activation units.

[27] Alaeddine, H., Jihene, M. Wide deep residual networks in networks. Multimed Tools Appl 82, 7889–7899 (2023). https://doi.org/10.1007/s11042-022-13696-0.

[28] S. Ioffe and C. Szegedy, Batch normalization: Accelerating deep network training byreducing internal covariate shift", CoRR, vol. abs/1502.03167, 2015.

[29] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville, and Y. Bengio. Maxout networks. In Proceedings of the 30th International Conference on Machine Learning (ICML2013), volume 28 of JMLR Proceedings, pages 1319– 1327. JMLR.org, 2013.

[30] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply supervised nets. InProceedings of AISTATS 2015, 2015