Multimedia Tools and Applications (2023) 82:7889–7899
https://doi.org/10.1007/s11042-022-13696-0

TRACK 2: MEDICAL APPLICATIONS OF MULTIMEDIA

Wide deep residual networks in networks

Hmidi Alaeddine[1] · Malek Jihene[1,2]

Received: 12 May 2022 / Revised: 14 July 2022 / Accepted: 15 August 2022 /
Published online: 20 August 2022

Abstract

The Deep Residual Network in Network (DrNIN) model [18] is an important extension of the convolutional neural network (CNN). They have proven capable of scaling up to dozens of layers. This model exploits a nonlinear function, to replace linear filter, for the convolution represented in the layers of multilayer perceptron (MLP) [23]. Increasing the depth of DrNIN can contribute to improved classification and detection accuracy. However, training the deep model becomes more difficult, the training time slows down, and a problem of decreasing feature reuse arises. To address these issues, in this paper, we conduct a detailed experimental study on the architecture of DrMLPconv blocks, based on which we present a new model that represents a wider model of DrNIN. In this model, we increase the width of the DrNINs and decrease the depth. We call the result module (WDrNIN). On the CIFAR-10 dataset, we will provide an experimental study showing that WDrNIN models can gain accuracy through increased width. Moreover, we demonstrate that even a single WDrNIN outperforms all network-based models in MLPconv network models in accuracy and efficiency with an accuracy equivalent to 93.553% for WDrNIN-4-2.

Keywords Deep network in network · Convolution neural network · CIFAR-10

Abbreviations

CNN	Convolutional Neural Network
CPU	Central Processing Unit
GPU	Graphics Processing Unit
RAM	Random Access Memory
DDR	Double Data Rate

✉ Hmidi Alaeddine
alaeddine.hmidi@fsm.rnu.tn

Malek Jihene
jihenemalek14@gmail.com

[1] Faculty of Sciences of Monastir, Laboratory of Electronics and Microelectronics, LR99ES30, Monastir University, 5000 Monastir, Tunisia

[2] Higher Institute of Applied Sciences and Technology of Sousse, Sousse University, 4000 Sousse, Tunisia

NIN	Network in Network
DNIN	Deep Network in Network
DrNIN	Deep Residual Network in Network
MLP	Multilayer perceptron
DMLPconv	Deep MLPconv
DrMLPconv	Deep Residual MLPconv
ReLU	Rectified Linear Unit
eLU	Exponential Linear Unit

1 Introduction

In residual networks, every fraction of a percent improvement in accuracy costs almost double the number of layers, and thus as a natural consequence of this large increase in network depth during training. The network will develop a problem of decreasing feature reuse, which makes network training very slow. The decrease in feature reuse during forward propagation refers to the problem of vanishing gradients in the forward direction. Residual networks solve the problem of degradation through connection jumps or shortcuts, by short-circuiting the shallow layers to the deeper layers. The Deep Residual Network in Network (DrNIN) architecture, above all other Deep Network in Network (DNIN) architectures, exhibited great accuracy improvements and convergence behavior superior to the competition. Increasing the depth of the DNIN [1] unexpectedly caused a problem in degradation of accuracy. The proposed solution was to reformulate the convolutional layers as residual learning functions. Therefore, so far, the study of DNINs has mainly focused on the order and number of layers inside a DMLPconv block and the depth of the networks. The main disadvantage of residual networks lies in the high number of layers where the depth of residual deep networks can evolve up to thousands of layers and up to thousands of layers for deep network in network residual networks. To solve this problem, a work published in 2016 [40] show that width has a greater effect than depth. Convinced by this principle, we propose to expand the DrMLPConv blocks of DrNIN [18] in order to have a more precise classification. In this paper, we address the challenges of improving classification by introducing a broader and more efficient DrNIN [18] architecture for computer vision which takes its name from DrNIN's paper [18] with the famous "WRN" [40]. The benefits of this work have been validated experimentally in the CIFAR-10 classification challenges. The contributions of this work are:

(i) We propose a new layer architecture WDrMLPConv which represents the core of Wide Deep Residual Networks in Networks (WDrNIN) architecture with improved performance.
(ii) We present a detailed experimental study that thoroughly examines several important aspects of WDrMLPconv blocks.
(iii) We present a detailed experimental study of multi-width deep model architectures that broadly examines several important aspects of WDrMLPconv layers.
(iv) Finally, we show that our proposed WDrNIN models achieve interesting results on CIFAR-10 significantly improving accuracy and learning speed.

The rest of this article is organized as follows: Section 2 provides an overview of related work. Section 3 bear the strategy. Experimental results are presented in section 4. Evaluations and

comparative analyzes are presented and discussed in the section 5. Section 6 is dedicated to implementation details. Advantage and limitations of WDrNIN are presented in section 7. The work is concluded in section 8.

2 Related works

Over the years, various techniques have been applied to improve the value of accuracy, and this is evident in the work that followed Alexnet until the publication of ResNet. Generally, deep networks have highlighted their success in many works in the post-2015 period [1–4, 6, 7, 10, 11, 16, 17, 19–24, 26, 28, 31–34, 37, 41]. These solutions are represented in the application of modifications at the level of the convolution layer such as increasing the depth [10, 17, 41] and/or the width [21, 40], modifying the type, the parameters [8, 9] and reducing filter size [1, 18, 38], changing the number of channels and feature map [38, 40]. The modification at the level of the layers of pooling [12, 14, 15, 25, 28–36, 39] and of the activation function [5, 27]. In classical CNNs, simple linear filters represent the beating hearts of computations inside the convolutional layer. On the other hand, in the model based on the network in network, the nonlinear filters are exploited instead of simple classical linear filters like the multilayer perceptron (MLP) [1, 28, 30]. Various works have exploited this type of nonlinear filters such as the NIN model [28], DNIN [1], DrNIN [18]. NIN [28] consists of several layers MLPconv, stacked in succession, which integrates a linear convolution layer and two MLP layers with a ReLU activation function. A global average pooling layer is used instead of the fully connected layers that are traditionally used in CNNs. The calculation performed by the MLP conversion layer is as follows:

$$f^{1}_{i,j,k_1} = \max\left(\omega^{1T}_{k1} x_{i,j} + b_{k_n}, 0\right)$$
$$f^{n}_{i,j,n} = \max\left(\omega^{nT}_{k_n} f^{n-1}_{i,j} + b_{k_n}, 0\right)$$

Knowing that:

- **(i, j)** represents the pixel index in the feature map,
- $\mathbf{x_{ij}}$ designates the input patch centered at the location (i, j).
- **k** is used to index the channels of the feature map and n denotes the number of layers.

Figure 1 illustrates the overall structure of the NIN [28] architecture.

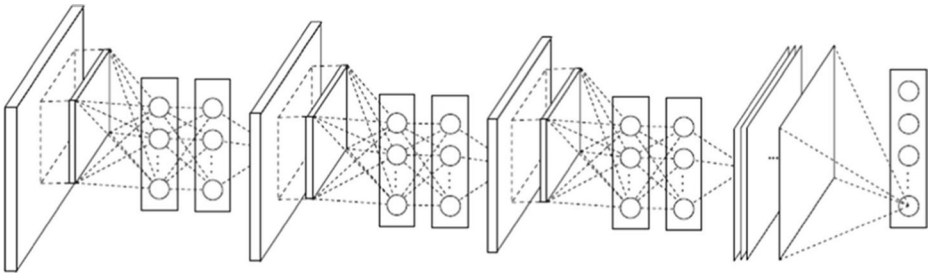

Fig. 1 Network in network

A modification of the NIN model [28] is presented in the Deep Network In Network (DNIN) model [1]. This model consists of the stacked DMLPconv block which integrates two convolutional layers of size 3 × 3 and a nonlinear activation unit "eLU" instead of ReLU. In this architecture, the eLU function [9] is used instead of ReLU [36] to alleviate the problem of leakage gradients and accelerate the learning speed. The DNIN [1] is shown in Fig. 2.

In [18], the authors proposed the DrNIN model based on DNIN [1]. It represents an improvement of the architecture of DNIN [1] by applying the residual learning framework to the different MLPconv layers, and reformulating the convolutional layers of DMLPConv as residual learning functions. Figure 3 illustrates the DrNIN model composed of three DrMLPconv layers.

The first model that applied the residual function in CNNs is the ResNet model [17]. In this model published in 2015, a residual block is proposed to facilitate the formation of very deep networks. In this model, the average global pooling layer is followed by a Softmax layer is exploited as the classification layer. Note that the residual block with identity mapping described in formula-represented (1). x_{l+1} and x_l are the input and output of the l^{th} unit in the network, F is a residual function and w_l are parameters of the block.

$$x_{(l+1)} = x_l + F(x_k, W_l) \tag{1}$$

In 2016, Wide ResNet [40] showed that using a wider Residual Block represents an effective solution to improve the accuracy than deepen residual networks.

From this work, we considered these approaches to improve the model of DrNIN [18] in order to obtain a better performance where one can widen the DrMLPconv blocks with a widening factor k which scales the width of the l blocks WDrMLPconv.

3 Proposed model

Compared to the original DrMLPconv architecture of the DrNIN model [18], a stretching is applied to the DrMLPconv layers with a stretching factor k. The new layer is named "wide deep residual MLPconv (WDrMLPconv)". The original "base" block of DrMLPconv is shown in Fig. 4a. The new WDrMLPconv block is shown in Fig. 4.

Figure 4a presents the structure of DrMLPconv is based on a residual block [17], two multilayer perception layers and two convolution layers of size 3 × 3, MLP layers. These different layers are followed by an eLU activation. Figure 4b presents a basic wide residual block architecture integrating two consecutives wide 3 × 3 convolution layers with forward

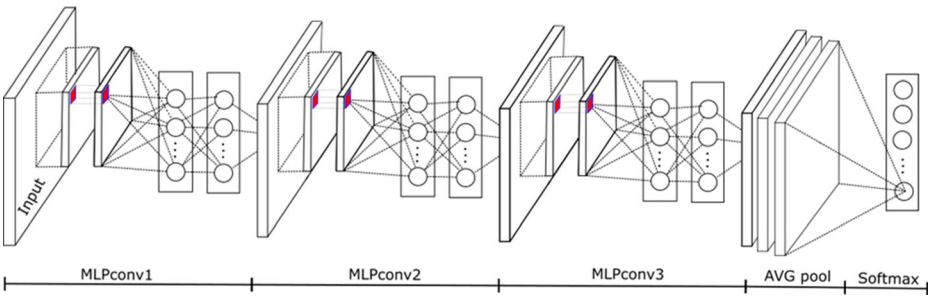

Fig. 2 Deep Network in Network

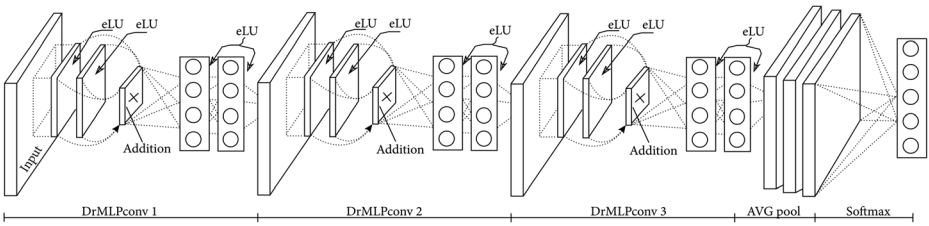

Fig. 3 Deep residual network in network

addition and return to main flow. In Fig. 4c, a bottleneck architecture that integrates two 1 × 1 convolution layers and one 3 × 3 convolution layer. The first layer Conv 1 × 1 decrease the dimension of the feature size, the Conv 3 × 3 layer reduces it, then Conv 1 × 1 increases it before adding and returning to the main stream. This configuration is exploited to make the WDrMLPConv block even thinner. Figure 4d presents a basic WDrMLPConv architecture integrating a dropout layer between two consecutive 3 × 3 convolution layers with batch normalization and ReLU before adding and returning to the main stream.

The WDrNIN model only uses 3 × 3 filters because they are filters with a very small receptive field: 3 × 3 (which is the smallest size to capture the notion of left/right, top/bottom, center). In the rest of the article, the following notation: WDrNIN-L-k is used. For example, for a network with 20 layers and widening factor k = 3. The notation would be WDrNIN-20-3. The new basic structure of DrMLPconv is based on a residual block [17], a multilayer perception (with a depth of two layers) which is described as a complex nonlinear filter. Note that basic DMLPconv, as shown in Fig. 4a, consists of two convolution layers of size 3 × 3, MLP layers. These different layers are followed by an eLU activation. Let WDrMLPConv(X)

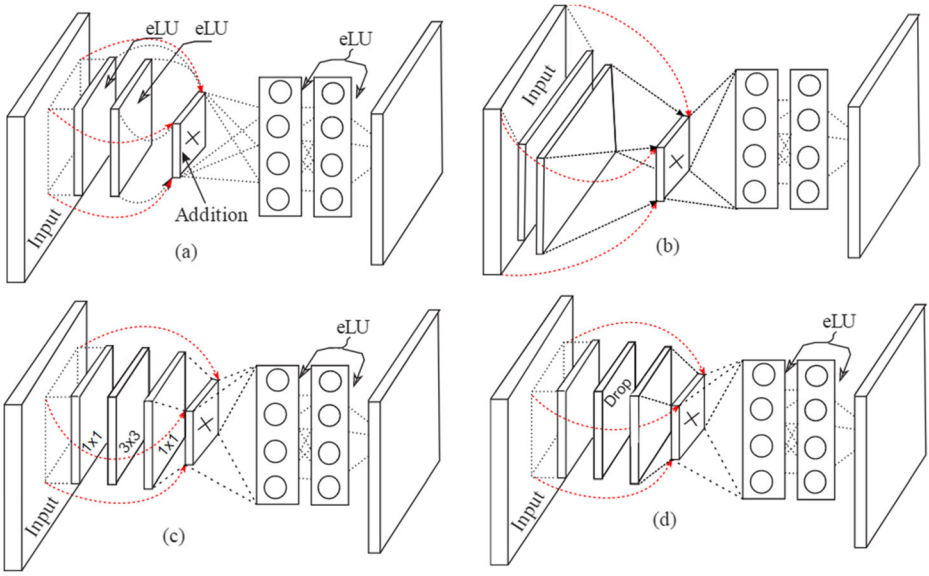

Fig. 4 **a** A schematic example of "basic" DrMLPconv layer, **b** a schematic example of "basic" Wide DrMLPconv. **c** an example schematic of a bottleneck, **d** an example schematic of a "basic" Dropout-DrMLPconv layer

be the WDrMLPconv layer, where X denotes a list of layers used in the WDrMLPconv(x) structure. For example, for WDrMLPConv(3, 3, D) denotes the basic structure WDrMLPconv integrating a residual block applied to two convolution layers of size 3 × 3 with the two MLP layers with a dropout between the two 1 × 1 convolution layers. For WDrMLPConv(3, 3), it is the same structure but without using the dropout layers. The WDrMLPConv(1,3,1) denotes a structure with two 1 × 1 layers that embeds a 3 × 3 layer between them. All configurations of the WDrMLPconv layer are equipped with the eLU nonlinearity [13]. The different WDrMLPconv structure adopted in this work is shown in Table 1.

The general structure consists of L groups of **WDrMLPConv$_x$** with x belonging to {1,2,3,4,5} ends with an average pooling layer and a final classification layer. The overall average filter size of the grouping layer depends on the depth factor "L". Table 2 summarizes the sizes of these overall average clustering layers. The introduced enlargement factor k represents the width of the model and changes from one network to another. The number of convolution kernels for each convolution group and block is described in Table 2.

As mentioned earlier, the WDrNIN network admits two factors: Deepening factor l and widening factor k. Note that a network is said to be "wide" only if k is greater than 1. Hence when k = 1, WDrMLPconv has the same width as DrMLPconv. Let **WDrNIN-l-k** be the notation used. To describe the WDrNIN model with its depth and width parameters. The general structure is made up of l WDrMLPconv blocks. These blocks are followed by an average pooling and a final classification layer. Figure 5 shows the structure of WRN-3-1 wide residual networks.

4 Experimental results

We evaluate our different configurations on a reference data set: CIFAR-10. In the following, we describe our results and analyze the performance.

4.1 Type of convolutions, number of convolutions and width of a residual block

We start by reporting the results using the WDrNIN models with l = 3, 4 and 5 and k = 2 for WDrMLPConv(1,3,1) and WDrMLPConv(3,3) and WDrMLPConv(3, 3, D). The accuracy of the test is calculated as the average of 2 runs. The time per training epoch is also calculated as an average of 2 runs.

Table 3 presents the error test (%) on CIFAR-10 with a factor k = 2 and the two different block types WDrMLPConv (1.3.1) and WDrMLPConv (3.3).

Table 1 The configurations of WDrMLPConv

Layer	WDrMLPConv (X)	
(X)	WDrMLPconv (3, 3)	DrMLPconv (3,3, D)
Conv-1	3x3x192 / st. 1/ pad 1 / eLU	3x3x192 / st. 1/ pad 1/ eLU / BN
Dropout		×
Conv-2	3x3x192 / st. 1/ pad 1 / eLU	3x3x192 / st. 1/ pad 1 / eLU / BN
MLP-1	1x1x160 / st. 1 / pad 0 / eLU	1x1×160 / st. 1 / pad 0 / eLU/ BN
MLP-2	1x1×192 / st. 1 / pad 0 / eLU	1x1×192/ st. 1 / pad 0 / eLU / BN

Table 2 The number of kernels for WDrMLPconv

Layers	Conv 1x1	Conv 3 × 3	MLP-1	MLP-2
Number	96	192	160	192

In this table, we can notice that WDrMLPConv(1,3,1) offers an accuracy of 08.09 and consumes 42.09 s for one training epoch. WDrMLPConv (3.3). WDrMLPConv (3.3) generates an error of 08.09 and consumes 42.09 s for a learning period. WDrMLPConv (3.3) consumes 58.6 s for a training period and offers 07.62% as test error.

In the following, we limit our work on WDrNIN with WDrMLPConv (3.3) in order to be consistent with other techniques and methods as well. We test and analyze the block deepening factor l to see its performance effect. Table 4 shows test accuracy (%) on CIFAR-10 with k=2 and WDrMLPConv(3.3) with various l. Note that the number of parameters increases linearly with the depth factor "l". In addition to all that above, we also test and analyze the effect of the enlargement factor k.

It is observed that WDrNIN with $l = 4$ and $k = 2$ - WDrMLPConv(3,3) was found to be the best compared to the same network using l different from 4. The WDrNIN model with WDrMLPConv(3,3) is the fastest in terms of time (time, s) which measures a training epoch compared to another model which exploits the same parameters l and k with WDrMLPConv(3,3). We note that our results were obtained with a batch size equivalent to 128 in all our experiments.

4.2 Dropout in residual blocks

As enlargement increases do does the number of parameters. Although the networks already have a batch normalization layer that provides a stabilization effect, adding a Dropout layer [36] after the eLU layer in each residual block is done, as shown in Fig. 4d, in order to prevent networks from over-fitting. Overall, Dropout [36] is described as a regularization technique to reduce overfitting in neural networks. This avoids complex co-fittings on the training sample data.

We trained the models with the Dropout layer inserted into a residual block between the convolutions. The dropout probability values were 0.5. Exploiting the dropout layer brings improvements in test accuracy. It leads to an increase in test accuracy on CIFAR-10 ranging from 0.027% up to 0.043% for an average of 2 runs with the WDrNIN-4-2 models shown in Table 5.

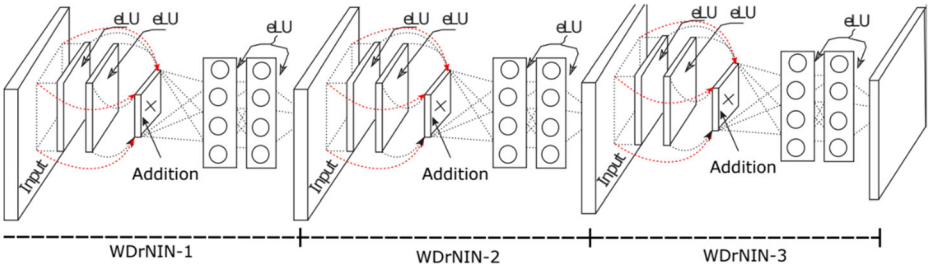

Fig. 5 Structure of WDrNIN-3-1 wide residual networks

Table 3 Test error (%) on CIFAR-10 with k = 2 and the two different types of blocks. The average of 2 runs. Time (time,s) measures an epoch of training

Block type	Depth	Time,s	CIFAR-10
WDrMLPConv ***(1,3,1)***	3	42.09	08.09
WDrMLPConv ***(3,3)***		58.6	07.62

5 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 in the literature. It is recalled that the proposed model was trained by performing transfer learning. As shown in Table 6, the WRN-4-2 model achieved slightly better accuracy than most literature studies with the original dataset. Using the Dropout layer, the WRN-4-2 model was better in terms of accuracy. The experimental studies showed that the WDrNIN models, which offer the best classification performance in the exploited literature, could be the best performance in all works of plant disease classification. A comparison with the results of different studies and works is presented in Table 6.

The experimental results demonstrate the effectiveness of the proposed contribution. Moreover, they show that the WDrNIN-4-2 with the dropout layer offers better results, in terms of classification accuracy, than the various other models.

6 Implementation details

All models used in this study were compiled with CPU support. All experimental studies were conducted in Google cloud environment on Linux operating system running on Dell Intel Core i5- 2450 M 2.50 GHz processor and 6 GB DDR4–2400 RAM. All codes are made with python algorithm based on "TensorFlow" deep learning framework to classify and recognize images, which is an open source deep neural network library written in Python language. All the

Table 4 Test precision (%) on CIFAR-10 with k = 2. and WDrMLPConv(3.3) with various l

L	CIFAR-10
3	07. 62
4	06. 49
5	06. 73

Table 5 The effect of the Dropout in WDrNIN. (The average of 2 runs)

Block type	Depth	K	Dropout	Error test
WDrMLPConv(3,3)	3	2		07.62
		2	×	07.593
	4	2		06.49
		2	×	06.447
	5	2		06.73
		2	×	06.692

Table 6 CIFAR-10 Test Accuracy

Ref.	Method	Error test (%)
[28]	NIN	08.81
[1]	DNIN	07.46
[18]	DrNIN (L=5)	07.21
Our	WDrNIN	06.447

experiments are carried out is executed on a total of 160 epochs, During the training we exploited the stochastic gradient descent with a Momentum equivalent to 0.9 in all the experiments carried out. The basic learning rate is equivalent to 0.005 and decreases by a factor of 10 every 10 epochs. The weight loss is equivalent to 0.0005.

7 Advantage and future works

The WDrNIN models used offer an interesting test accuracy which is located at the head of the various works reported in the literature. The importance of the exploited WDrNIN model also lies in its repetitive and homogeneous structure which make it very suitable and compatible for integration into embedded system applications. In future work, it is planned to expand and artificially augment the CIFAR-10 dataset by increasing the number of classes. This will contribute to the development of models and architecture capable of achieving more precise and interesting precisions. By publishing these models and architectures in embedded electronics and mobile applications, experts, researchers and the visually impaired will be able to discover and classify the images and make useful and necessary decisions.

8 Conclusion

Classification based on deep learning [23] has become popular in the field of image processing. In this work, a WDrNIN deep learning model is proposed for image classification and detection. In addition, a study is presented on the width of WDrNIN networks as well as on the use of dropout in these different architectures. WDrNIN was compared to widely known deep learning models used in image detection and classification from the CIFAR-10 dataset. The WDrNIN-4-2 models with/without the dropout layer were found to be the most accurate in terms of accuracy compared to other widely known CNN models. The WDrNIN-4-2 model with the suppression layer achieved an accuracy of 93.553% in the CIFAR-10 dataset, while the WDrNIN-4-2 model achieved an accuracy of 93.51%. Additionally, when analyzing the model training times by epoch, it was found that the WDrNIN-4-2 model was the fastest of the CIFAR-10 set, but the accuracy was lower than that of the other WDrNIN models with different depth and width. Additionally, WDrNIN models have achieved well-localized accuracy in the literature for CIFAR-10,

Acknowledgments This work was supported by the Electronics and Microelectronics Laboratory.

Data availability The data used to support the findings of this study are included within the article.

Declarations

Conflict of interest The authors declare that there are no conflicts of interest regarding the publication of this article.

References

1. Alaeddine H, Jihene M (2020) Deep network in network. Neural Comput Applic 134
2. Alom MdZ, Hasan M, Yakopcic C, Taha T, Asari V (2018) Recurrent residual convolutional neural network based on U-Net (R2U-Net) for medical image segmentation
3. Bengio Y, Glorot X (2010) Understanding the difficulty of training deep feedforward neural networks. Proceed AISTATS 9:249–256
4. Chan T, Jia K, Gao S, Lu J, Zeng Z and Ma Y (2014) "PCANet: a simple deep learning baseline for image classification?" , http://arxiv.org/abs/1404.3606.
5. Chang J-R, Chen Y-S (2015) Batch-normalized maxout network in network. http://arxiv.org/abs/1511.02583
6. Chen L-C, Papandreou G, Kokkinos I et al (2018) DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans Patt Anal Machine Intell 40(4):834
7. Chen T, Goodfellow I, Shlens J (2016) "Net2net: accelerating learning via knowledge transfer," http://arxiv.org/abs/1511.05641.
8. Ciresan D, Meier U, Schmidhuber J (2012) "Multi-column deep neural networks for image classification," http://arxiv.org/abs/1202.2745.
9. Clevert D-A, Unterthiner T, Hochreiter S (2016) Fast and accurate deep network learning by exponential linear units (ELUs) comments: published as a conference paper at ICLR 2016 subjects—learning (cs.LG), 2016
10. Gao S, Miao Z, Zhang Q, Li Q (2019) DCRN: densely connected refinement network for object detection. J Phys: Conf Series., 1229, Article ID 012034
11. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. J Mach Learn Res 9
12. Gong Y, Wang L, Guo R, Lazebnik S (2014) Multi-scale orderless pooling of deep convolutional activation features. http://arxiv.org/abs/1403.1840
13. Goodfellow IJ, Warde-Farley D, Mirza M, Courville AC, Bengio Y (2013) Maxout networks. https://arxiv.org/abs/1302.4389
14. Graham B (2014) Fractional max-pooling. https://arxiv.org/abs/1412.6071
15. He K, Zhang X, Ren S, Sun J (2014) Spatial pyramid pooling in deep convolutional networks for visual recognition. http://arxiv.org/abs/1406.4729
16. He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: surpassing human-level performance on ImageNet classification
17. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition
18. 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
19. Huang G, Sun Y, Liu Z, Sedra D, Weinberger KQ (2016) Deep networks with stochastic depth. Comp Vision-ECCV 2016 9908:646–661
20. Huang G, Liu Z, Weinberger KQ, Van Der Maaten L (2017) Densely connected convolutional networks
21. Iandola F, Han S, Moskewicz M, Ashraf K, Dally W, Keutzer K (2017) SqueezeNet: alexnet-level accuracy with 50x fewer parameters and connected convolutional networks
22. Ioffe S, Szegedy C (2015) "Batch normalization: accelerating deep network training by reducing internal covariate shift", http://arxiv.org/abs/1502.03167.
23. Johnson JM, Khoshgoftaar TM (2019) Survey on deep learning with class imbalance. J Big Data 6:27. https://doi.org/10.1186/s40537-019-0192-5
24. LeCun Y, Bottou L, Orr GB, Muller K-R (1998) "Efficient backprop," in Neural Networks: Tricks of the Trade, Springer, Berlin, Germany,
25. 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.
26. Lee C-Y, Xie S, Gallagher P, Zhang Z, Tu Z(2014) "Deeply-supervised nets," , http://arxiv.org/abs/1409.5185.

27. Liao Z, Carneiro G (2016) On the importance of normalisation layers in deep learning with piecewise linear activation units
28. Lin M, Chen Q, Yan S (2013) Network in network. http://arxiv.org/abs/1312.4400
29. Murray N, Perronnin F (2015) "Generalized max pooling," in Proceedings of the 2015 IEEE conference on computer vision and pattern recognition (CVPR), pp 2473–2480, Boston, MA USA
30. 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
31. Raiko T, Valpola H, Lecun Y (2012) "Deep learning made easier by linear transformations in perceptrons," in Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS-12), N. D. Lawrence and M. A. Girolami, Eds., vol. 22, pp. 924–932, La Palma, Canary Islands, Spain
32. Romero A, Ballas N, Kahou SE, Antoine C, Gatta C, Bengio Y (2014) FitNets: hints for thin deep nets
33. Schmidhuber J (1992) Learning complex, extended sequences using the principle of history compression. Neural Comput 4(2):234–242
34. Simonyan K, Zisserman A (2014) Very deep convolutional networks for largescale image recognition
35. Springenberg J, Dosovitskiy A, Brox TT, Riedmiller M (2014) Striving for simplicity: the all convolutional net. http://arxiv.org/abs/1412.6806
36. 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
37. Sutskever I, Martens J, Dahl GE, and Hinton GE(2013) "On the importance of initialization and momentum in deep learning," in Proceedings of the 30th International Conference on Machine Learning (ICML-13), S. Dasgupta and D. Mcallester, Eds., vol. 28, pp. 1139–1147, JMLR Workshop and Conference Proceedings, New Brunswick, NJ, USA
38. Szegedy C (2015) "Going deeper with convolutions," in Proceedingsof the 2015 IEEE conference on computer vision and pattern recognition (CVPR), pp 1–9, Boston, MA USA
39. Yoo D, Park S, Lee J, Kweon I (2015) "Multi-scale pyramid pooling for deep convolutional representation," in Proceedings of the 2015 IEEE conference on computer vision and pattern recognition (CVPR), pp 1–5, Boston, MA USA
40. Zagoruyko S, Komodakis N (2017) Wide Residual Networks, 1605.07146, arXiv, pp 87.1–87.12 https://doi.org/10.5244/C.30.87.
41. Zeiler MD, Fergus R (2013) Visualizing and understanding convolutional networks