International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

IN-DEPTH RESIDUAL LEARNING FOR PICTURE IDENTIFICATION

Maneesh Jangir[1*], Swapnil Yogesh Chaudhari[2], Bhavishya Dwivedi[3], Surbhi Rohitbhai Patel[4], Aniket Rajendra Pagedar[5], Slok shah[6], Prof. Vivek Dave[7]

[1*]Faculty of IT & Computer Science, Parul University, Vadodara, India, 2305112120010@paruluniversity.ac.in

[2]Faculty of IT & Computer Science, Parul University, Vadodara, India, 230511212038@paruluniversity.ac.in

[3]Faculty of IT & Computer Science, Parul University, Vadodara, India, 2305112120073@paruluniversity.ac.in

[4]Faculty of IT & Computer Science, Parul University, Vadodara, India, 2305112120050@paruluniversity.ac.in

[5]Faculty of IT & Computer Science, Parul University, Vadodara, India, 2305112070010@paruluniversity.ac.in

[6]Faculty of IT & Computer Science, Parul University, Vadodara, India, 2305112070029@paruluniversity.ac.in

[7]Faculty of IT & Computer Science, Parul University, Vadodara, India, vivek.dave@paruluniversity.ac.in

*Corresponding author: Maneesh Jangir (2305112120010@paruluniversity.ac.in)

Abstract

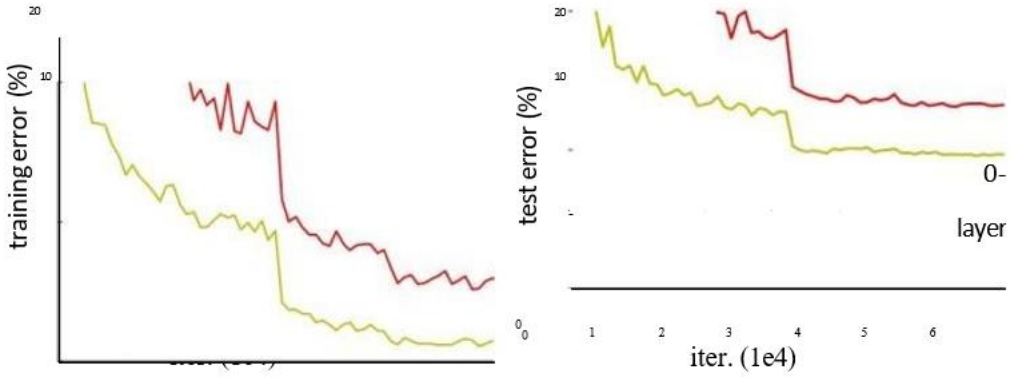
Training deeper neural networks is more challenging. To facilitate the training of networks that are significantly deeper than those previously employed, we provide a residual learning approach. As we learn, we explicitly reformulate the layers. *R*esidual functions are about the inputs from the layers, rather than mastering functions without references. We present

Residual nets achieve an error of 3.57%. With this outcome, the ILSVRC 2015 classification job won first place. Additionally, we provide a study of CIFAR-10 with layers 100 and 1000.

Many tasks involving visual identification place a premium on the depth of representation. We only get a result of our extraordinarily deep representations.

extensive empirical data demonstrating that these residual networks may be optimized

more easily and significant depth increases can improve accuracy. We assess residual networks on the ImageNet dataset that have up to 152 layers—eight times deeper than VGG nets [40]—while maintaining a lower level of complexity. On the ImageNet test set, an ensemble of these



Received: August 7, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

1. Introduction

Image categorization has seen several advances thanks to deep convolutional neural networks [22, 21] [21, 49, 39]. Low, mid, and high-level features [49] and classifiers are naturally integrated into an end-to-end multi-layer fashion by deep networks, and the depth—or several stacked layers—can enhance the "levels" of features. Recent findings [40, 43] demonstrate how important network depth is, and the top results [40, 43, 12, 16] on the difficult ImageNet dataset [35] all take advantage of “very deep” [40] models, which have a depth ranging from sixteen [40] to thirty [16]. Figure 1 is also present in many other non-trivial visual recognition tasks [7, 11, 6, 32, 27]. On CIFAR-10, training error (left) and test error (right)with "plain" networks consisting of 20 and 56 layers. The more profound network greatly benefited from very deep models.

Motivated by the importance of depth, an inquiry emerges: Is learning increasingly complex networks just a matter of adding layers? The well-known issue of vanishing/exploding gradients [14, 1, 8], which impeded convergence from the start, was a barrier to responding to this subject. Normalized initialization [23, 8, 36, 12] and intermediate normalization layers [16], on the other hand, have largely solved this issue and allowed networks with tens of layers to begin convergent for stochastic gradient descent (SGD) with backpropagation [22].

It is evident from the decline in training accuracy that not all systems are as simple to optimize. Let's look at an architecture that is shallower and it's the deeper version that has additional layers added to it. The deeper model has a solution by construction: the additional levels are identity mapping, while the remaining layers are copies of the shallower model that was learned. Given the presence of this artificial solution, training errors for a deeper model shouldn't be greater than those for a shallower model.However, tests reveal that the solutions we have available now are unable to find solutions In this paper, we introduce a deep residual learning approach to tackle the degradation issue. Rather than assuming that every few stacked layers will directly fit an ideal underlying mapping, we specifically allow these layers to fit a residual mapping. Formally, we allow the stacked nonlinear layers to represent the desired underlying mapping as H(x), and we let them map to another mapping of $F(x):= H(x) - x$. It recasts the original mapping into F(x)+x. We think that optimizing the residual mapping is less complicated than optimizing the initial, unreferenced mapping. Residual to zero then to fit an identity mapping by a stack of nonlinear layers. Similar behaviors are also observed on the CIFAR-10 set [20], indicating that our method's impacts and optimization challenges are not specific to any one dataset. We investigate models with more than 1000 layers and exhibit successfully trained models with over 100 layers on this dataset.

Using incredibly deep residual nets, we achieve great results on the ImageNet classification dataset [35]. Even though it is less complex than VGG nets, our 152-layer residual net is the deepest network that has ever been shown on ImageNet [40]. With a top-5 error rate of 3.57% on the ImageNet test set, our ensemble took first place in the 2015 ILSVRC classification competition.

Additionally, the incredibly deep representations perform exceptionally well in generalization on additional recognition tests, which leads us to.

Received: August 7, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

1.1. Related Work

The Multigrid approach, which is widely used in low-level vision and computer graphics, is responsible for solving Partial Differential Equations (PDEs). It works by reformulating the system into subproblems at many scales, each of which is in charge of the

The residual solution between a coarser and a finer scale.

Hierarchical basis pre-conditioning [44, 45], which uses variables that represent residual vectors between two scales, is an alternative to Multigrid. These solvers converge far more quickly than standard solvers that are blind to the residual nature of the solutions, as demonstrated by studies [3, 44, 45]. There has been extensive research on procedures and theories that result in shortcut connections [2, 33, 48].

Adding a linear layer connected from the network input to the output is a common first step in the training of multi-layer perceptrons (MLPs) [33, 48]. A few intermediate layers are coupled directly to auxiliary classifiers in [43, 24] to handle vanishing/exploding gradients.

Shortcut connections are used to implement the strategies for centering layer responses, gradients, and propagated errors that are proposed in the works [38, 37, 31, 46]. Two deeper branches and a shortcut branch make up the "inception" layer [43]. "Highway networks" [41, 42] provide shortcut connections with gating functions [15] along with our work.

Unlike our identity shortcuts, which are parameter-free, these gates depend on data and have parameters. The layers in highway networks indicate non-residual functions when a gated shortcut is "closed" (getting close to zero). Conversely

2. Deep Residual Learning

2.1. Residual Learning

X represents the inputs to the first of these layers. Let us consider H(x) as an underlying mapping to be fit by a few stacked layers (notnecessarily the full net). Assuming that the input and output have the same dimensions, it is equivalent to assuming that many nonlinear layers can asymptotically approximate the residual functions, or H(x) − x. This is because numerous nonlinear layers can asymptotically approximate difficult functions2. Therefore, we explicitly let stacked layers approach a residual function F(x):= H(x) − x, instead of expecting them to resemble H(x). As a result, the original function becomes F(x)+x. The simplicity of learning for each form may differ, even though they should both be able to asymptotically approximate the target functions (as expected). The unexpected phenomenon regarding the deterioration problem (Fig. 1, left) is the driving force behind this reformulation. As we covered in the introduction, a deeper model should have a training error no higher than its shallower counterpart if the additional layers can be built as identity mappings. The degradation problem implies that simulating identical mappings by several nonlinear layers may pose challenges for the solvers. If identity mappings are optimal, the solvers using the residual learning re-formulation can simply approach identity mappings by pushing the weights of the various nonlinear layers toward zero. Although identity mappings are unlikely to be optimal in real-world situations, our reformulation might aid in preconditioning the issue. The solver should find it easier to discover the perturbations concerning an identity mapping rather than learning the function as a new one if

Received: August 7, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

the optimal function is closer to an identity mapping than a zero mapping. According to our studies (Fig. 7), identity mappings appear to offer appropriate preconditioning, as the learned residual functions typically exhibit tiny responses. Maps of Identity Using Shortcuts We use residual learning in a few stacked layers at a time. In Fig. 2, a building block is displayed. Formally, a building block is considered in this work with the definition y = F(x, {Wi})+ x.

(1)

F(x)+x is the new form of the original mapping. We

Speculate that residual mapping optimization is less complicated than optimizing the original unreferenced mapping. To be discovered. For the two-layer example in Fig. 2, F = W2σ(W1x), where σ stands for ReLU [29], the biases are removed to simplify the notations. An element-wise addition and shortcut connection carry out the operation F + x. Following the addition, we adopt the second nonlinearity (i.e., σ(y); see Fig. 2Eqn. (1)'s shortcut connections don't add extra parameters or computational complexity. This is significant in our comparisons between plain and residual networks, in addition to being desirable in practice. Except for the insignificant element-wise addition, we may fairly compare plain/residual networks that concurrently have the same number of parameters, depth, width, and computational cost. In Equation (1), the dimensions of x and F have to be the same. If not (for example, while adjusting the input/output channels), we can match the dimensions by executing a linear projection via the shortcut connections:

y= F(x, {*W*i})+ *W*sx.

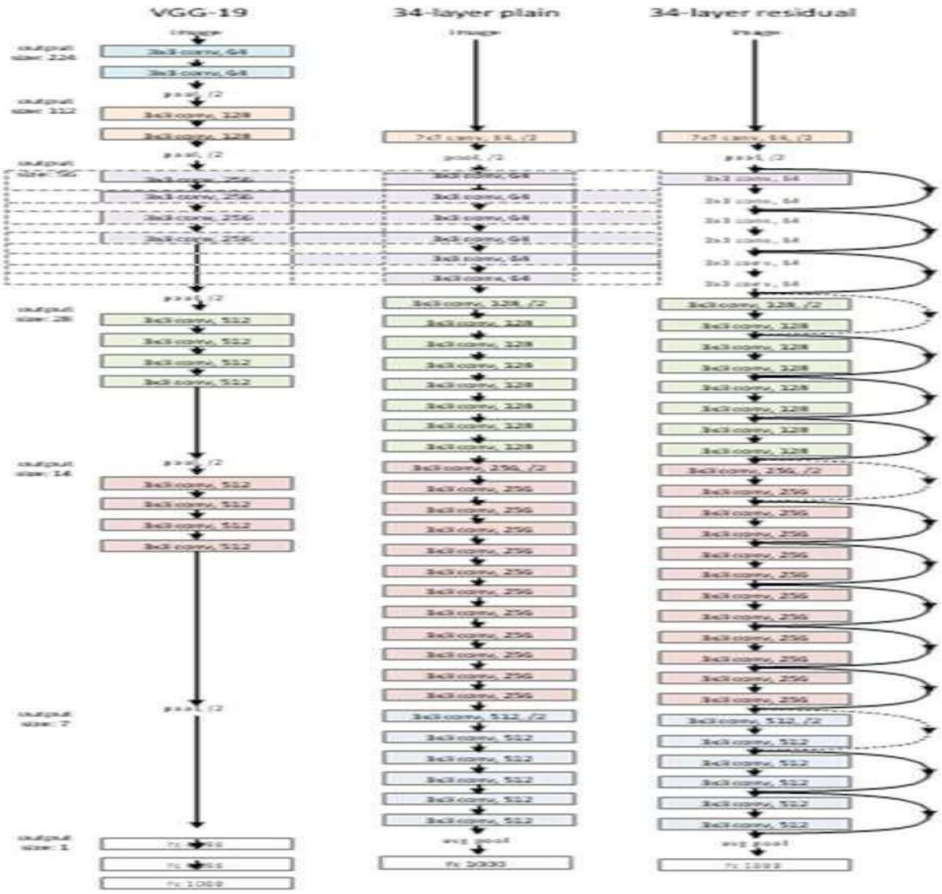
(2)

In Eqn. (1), a square matrix Was is another option. However, via our studies, we shall demonstrate that identity mapping is both affordable and sufficient to address the deterioration problem; hence, Was is only employed when the dimensions match. The residual function F has a flexible shape. While more levels are possible, the experiments in this study focus on a function F with two or three layers (Fig. 5). However, Eqn. (1) is comparable to a linear layer, y = W1x + x, for which we have not noticed any advantages if F only has one layer. Additionally, we observe that while the aforementioned notations pertain to fully- connected layers for simplicity, they are relevant to layer convolution. There are various convolutional layers that the function F (x, {Wi}) can represent. Channel by channel, two feature maps undergo element-wise addition.

Received: August 7, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Network Architectures



We have seen consistent behaviors after testing a variety of plain and residual nets. To facilitate discussion, we outline two ImageNet models as follows. Plain Network. The concept of VGG nets [40] (Fig. 3, left) is primarily responsible for our simple baselines (Fig. 3, middle).

The convolutional layers, which mostly consist of 3x3 filters, are designed with two basic principles in mind: (I) each layer has the same number of filters for an output feature map size of the same size; and (ii) if the feature map size is halved, the number of filters is doubled to maintain the time complexity per layer. We immediately carry out downsampling using convolutional layers with a stride of 2. Fig. 3 (middle) shows the total number of weighted layers, which is 34.

It is important to note that, in comparison to VGG networks [40], our model has fewer filters and is less complex (Fig. 3, left). Just 18% of VGG-19's 19.6 billion FLOPs (multiply-adds) are found in our 34-layer baseline, which contains 3.6 billion FLOPs. Figure 3 shows several ImageNet network topologies. Left: as a point of reference, the VGG- 19 models [40] (19.6 billion FLOPs). Middle: 34 parameter layer plain network with 3.6 billion FLOPs. Correct:34 parameter layers in a residual network (3.6 billion FLOPs). Dimensions are increased by the dotted shortcuts. Table 1 provides further information and other variations.

Received: August 7, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

References: -

[1] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.[2] C. M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1995.

[3] W. L. Briggs, S. F. McCormick, et al. *A Multigrid Tutorial*. Siam, 2000.

[4] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011.

[5] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zis- sermon. The PascalVisual Object Classes (VOC) Challenge. *IJCV*, pages 303–338, 2010.

[6] Girshick, R. R-CNN in Fast. In ICCV, 2015

[7] T. Darrell, J. Malik, J. Donahue, and R. Girshick. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[8] Y. Bengio and X. Glorot. being aware of how challenging deep feedforward neural network training is. AISTATS, 2010.

[9] M. Mirza, A. Courville, D. Warde-Farley, Y. Bengio, and I. J. Goodfellow. 1302.4389, Maxout networks. arXiv.

10] K. He and J. Sun. Convolutional neural networks at constrained time. In *CVPR*,2015.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014.

[12] J. Sun, S. Ren, X. Zhang, and K. He. Investigating rectifiers in depth: Outperforming humans in picture classification. ICCV, 2015.

[13] R. R. Salakhutdinov, I. Sutskever, A. Krizhevsky, N. Srivastava, and G. E. Hinton. Enhancing neural networks by stopping feature detectors from co-adapting. 2012; arXiv:1207.0580.

[14] S. Hochreiter. Research on dynamic neural networks. 1991, diploma thesis, TU Munich.

[15] J. Schmidhuber and S. Hochreiter. Long-term short-term memory. In 1997, Neural Computation, 9(8):1735–1780.

Received: August 7, 2025