

Biologically inspired deep residual networks

Prathibha Varghese, Arockia Selva Saroja

Department of Electronics and Communication, Noorul Islam Centre for Higher Education, Tamil Nadu, India

Article Info

Article history:

Received Sep 10, 2022

Revised Dec 10, 2022

Accepted Dec 21, 2022

Keywords:

Convolutional neural networks

Deep learning

Hexagonal pixel

Residual

Square pixel

ABSTRACT

Many difficult computer vision issues have been effectively tackled by deep neural networks. Not only that but it was discovered that traditional residual neural networks (ResNet) captures features with high generalizability, rendering it a cutting-edge convolutional neural network (CNN). The images classified by the authors of this research introduce a deep residual neural network that is biologically inspired introduces hexagonal convolutions along the skip connection. With the competitive training techniques, the effectiveness of several ResNet variations using square and hexagonal convolution is assessed. Using the hex-convolution on skip connection, we designed a family of ResNet architecture, hexagonal residual neural network (HexResNet), which achieves the highest testing accuracy of 94.02%, and 55.71% on Canadian Institute For Advanced Research (CIFAR)-10 and TinyImageNet, respectively. We demonstrate that the suggested method improves vanilla ResNet architectures' baseline image classification accuracy on the CIFAR-10 dataset, and a similar effect was seen on the TinyImageNet dataset. For Tiny- ImageNet and CIFAR-10, we saw an average increase in accuracy of 1.46% and 0.48% in the baseline Top-1 accuracy, respectively. The generalized performance of advancements was reported for the suggested bioinspired deep residual networks. This represents an area that might be explored more extensively in the future to enhance all the discriminative power of image classification systems.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Prathibha Varghese

Department of Electronics and Communication Engineering, Noorul Islam Centre for Higher Education

Kumaracoil, Thuckalay-629180, Tamil Nadu, India

Email: prathibha@sngce.ac.in

1. INTRODUCTION

Deep convolutional neural networks have greatly advanced computer vision [1]-[4]. Networks with a greater number of nodes are better able to capture the subtleties of high-dimensional visual data that lack linearity. The following factors, however, cause network performance to decline as network depth increases:

- Vanishing gradient: gradients (partial derivatives) are calculated with the aid of the chain rule in backpropagation. Throughout this training, gradients typically decrease at an exponential rate optimized using the activation function being used, meaning that the network's gradient gets less and smaller [5], [6].
- Harder optimization: it has been discovered that an increase in the number of layers in neural networks corresponds to an increase in training errors [7].

The problem of vanishing gradients is directly addressed by the design of such a residual network (ResNet) He *et al.* [1] which includes several skip connections in addition to the basic convolution layers. Contrary to conventional convolution layers, these connections make it simple for gradients to propagate backward

without degrading. Additionally, ResNet is frequently employed as the primary feature extractor in numerous computer vision applications due to skipping connections' inherent benefits and its generalization capacity [8]–[10].

This paper presents hexagonal residual networks (Hex-ResNet), a hybrid design with biological inspiration that enhances deep residual network generalization with hexagonal convolutional filters. We demonstrate that adding small hexagonal filters along a few skip links can improve the ResNet architecture's base performance. The main strength of our strategy is how well it combines the benefits that each of the separate (square and hexagonal) tessellations has to offer. We implement Hoozeboom *et al.* [10] which proposes effective procedures for hexagonal convolution by utilizing an comparable total of two square convolutions, in contrast to Steppa and Holch [11] which operates directly on a hexagonal lattice. By applying adequate computational support based on square tessellations, we can increase performance in this way. In comparison to several existing ResNet models, our suggested design has increased testing and validation accuracy based on Top-1 and Top-5 accuracy. Also, demonstrated how these improved concerts were achieved lacking appreciable computational rise. We provide the following summary of our efforts and findings:

- We improved the baseline picture classification accuracy of the vanilla ResNet by adding hex convolutions with a couple of skip connections.
- By conducting extensive tests on the benchmark datasets CIFAR-10 and TinyImageNet, we verified the effectiveness of the recently suggested Hex-ResNet architecture.
- We demonstrate that, across various ResNet settings, improving the accuracy of image categorization from scratch by adding hex convolutions to the skipped connection paths.

The rest of this paper is organized as: section 2 discusses all the studies that are linked to our methodology, and a survey of the associated literature is provided. Section 3 first covers the fundamentals of traditional ResNet topologies, skip connections, and then our suggested Hex-ResNet architecture in depth. We provide the experimental findings and training methods for our suggested architecture in section 4 utilizing the CIFAR-10 and TinyImageNet datasets. In section 6, we draw a conclusion based on our observations.

2. RELATED WORKS

2.1. Residual networks

Deep residual networks are a perfect network to serve as the backbone feature extractor for different computer vision tasks since they can circumvent the vanishing gradient problem using skip connections [1], [9], [12], [13]. Li and He [14] presented a convex k-method employing various area parameter altering criteria and offered an enhanced ResNet via changeable shortcut connections Wightman *et al.* [15] for numerous ResNet configurations in the Timm open-source toolbox, pre-trained models and shared competitive training parameters were made available. The study of Schlosser *et al.* [16] added pre-activation ResNets by rearranging the building block's components to enhance the signal propagation path. All of the well-known efforts that have ResNet as their primary feature extractor are best suited for data that is defined on a square lattice [15], [16].

2.2. Hexagonal convolution operations

Let \mathbf{S} be the equivalent image described on square lattice tessellations, and \mathbf{H} be the input image data representation on hexagonal lattice tessellations. We denote the hexagonal kernels by \mathbf{K}^l , where l indicates its size. We have assumed that the kernel weights are one for mathematical demonstration simplicity. But we employ the trainable kernel weights in the final implementation. As shown in Figure 1, we generate comparable rectangular kernels ($\mathbf{K}_{r1}^1 \in \mathbb{R}^{2 \times 3}$ and $\mathbf{K}_{r2}^1 \in \mathbb{R}^{3 \times 1}$) corresponding to \mathbf{K}^1 . Remember that a size one hexagonal kernel will have two similar rectangular shaped kernels. Similar to this a hexagonal kernel of size l will have $l + 1$ comparable rectangular kernels. Convolutions with these kernels \mathbf{K}_{r1}^1 and \mathbf{K}_{r2}^1 can be now simply done with efficient PyTorch routines. However, we must suitably pad \mathbf{S} in three distinct ways to produce hexagonal convolutions using rectangular kernels, We must properly pad \mathbf{S} in three different ways as shown in Figure 1. Let \mathbf{S}_1 , \mathbf{S}_2 , \mathbf{S}_3 be each of the three padded variations of \mathbf{S} . When rectangular kernels are convoluted mathematically \mathbf{K}_{r1}^1 and \mathbf{K}_{r2}^1 with \mathbf{S}_1 , \mathbf{S}_2 , \mathbf{S}_3 can be designed as:

$$\mathbf{P}_1 = \mathbf{S}_1 *_{(1,2)} \mathbf{K}_{r1}^1 \quad (1)$$

$$\mathbf{P}_2 = \mathbf{S}_2 *_{(1,2)} \mathbf{K}_{r1}^1 \quad (2)$$

$$\mathbf{P}_3 = \mathbf{S}_3 *_{(1,1)} \mathbf{K}_{r2}^1 \quad (3)$$

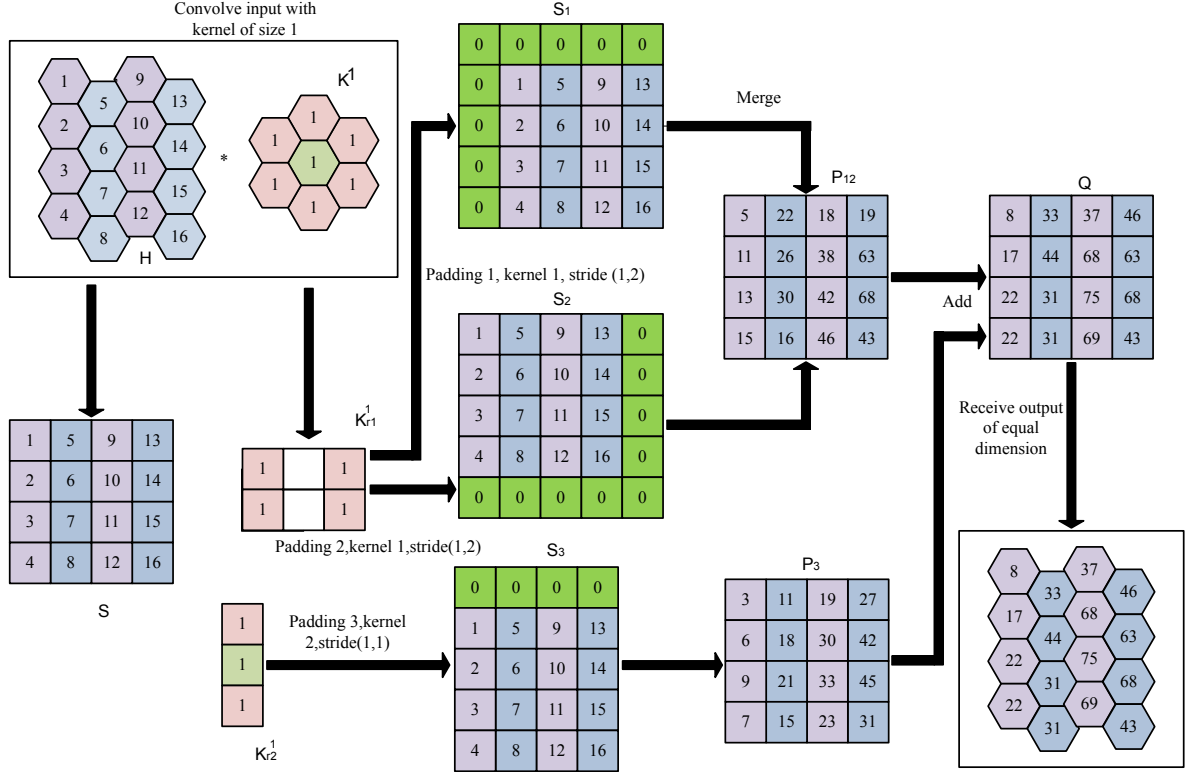


Figure 1. Implementation of hexagonal convolution [13] for hexagonal lattice image

where $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$ indicate the convolution outcomes with the kernels \mathbf{K}_{r1}^1 and \mathbf{K}_{r2}^1 . The convolution operator $*_{(x,y)}$ denotes with stride of x and y units along the rows and columns, respectively. The following step is to integrate \mathbf{P}_1 and \mathbf{P}_2 by selecting the alternate columns as shown in Figure 1. Mathematically we represent the merge operation as:

$$\mathbf{P}_{12} = \text{MERGE}(\mathbf{P}_1, \mathbf{P}_2) \quad (4)$$

The square equivalent of hexagonal convolution is obtained by one final addition operation as:

$$\mathbf{Q} = \mathbf{P}_{12} \oplus \mathbf{P}_3 \quad (5)$$

where \oplus denotes the element-wise addition operation. The output \mathbf{Q} if more processing is required, is reorganized into a hexagonal lattice as shown in Figure 1.

2.3. Hex-ResNet

The proposed Hex-ResNet is shown in Figure 2. The skip connections used projection shortcuts with hexagonal convolution and trained as mentioned by He *et al.* [1]. The 34-layer Hex-ResNet is with the integration of square convolution and hexagonal convolution is analysed. Different variants of Hex-ResNet is developed similar to ResNet 34 architecture.

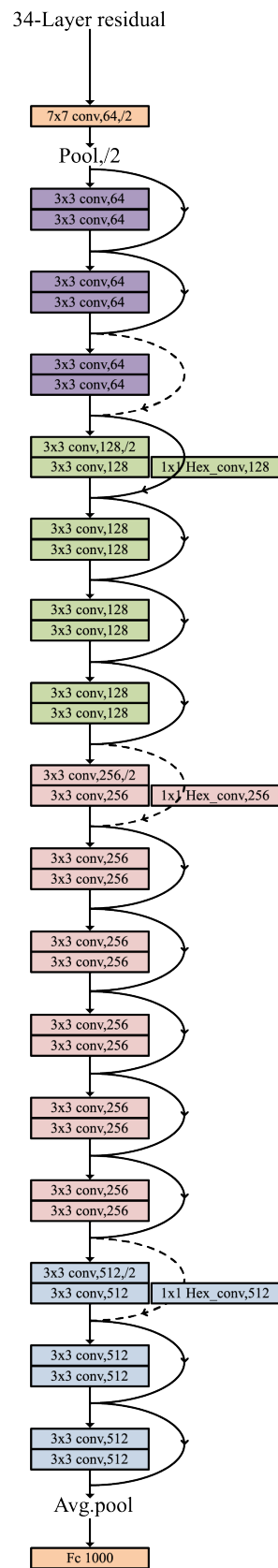


Figure 2. Proposed Hex-residual network architecture (readers are requested to zoom in to view the details)

3. EXPERIMENTAL RESULTS

In this section, We go into detail about our CIFAR-10 experiment outcomes. We also employ the TinyImageNet dataset for showing the efficacy of our method. Following He *et al.* [1], We use Top-1 and Top-5 accuracy as the performance metrics.

3.1. Software and hardware

In order to compute hexagonal solutions, we use the PyTorch-based tool HexagDly [11]. The input image hexagonal framework is preserved over its output side thanks to a combination of precise padding and a striding strategy. It uses Google Colab Pro+, a platform for web applications. This contains several integrated packages that let us train our model on powerful GPUs. The majority of the Tesla V100-SXM2-16 GB and A100-SXM4-40GB GPUs with compute capacities of 7.0” and 8.0” respectively were used throughout our studies.

3.2. Datasets

3.2.1. CIFAR-10

The CIFAR-10 benchmark dataset is the industry standard for classifying images. The proposed architecture is tested on this dataset of 600,000 images organized into 10 categories. The test set is made up of 10,000 images, while the training set is made up of an initial 50,000 photos. Every training image was given a 4-pixel padding on all sides, and a 32-pixel crop was then randomly selected from either the padding image or even the padding image’s horizontal flip. The dataset used in the test was not expanded. Both the train and validation subsets of the augmented train dataset, totaling 5,000 pictures, were created Figure 3 displays samples of both training in Figure 3(a) and evaluation images from several categories in the CIFAR-10 dataset in Figure 3(b). The optimum solution was discovered using stochastic gradient descent with the following parameters: learning rate=0.1, momentum=0.90, and weight decay=0.001. The optimization criterion utilized was cross-entropy loss. After that, we took the divided learning rate and trained the model for 182 epochs with 32k, 48k, as well as 64k iterations, respectively.

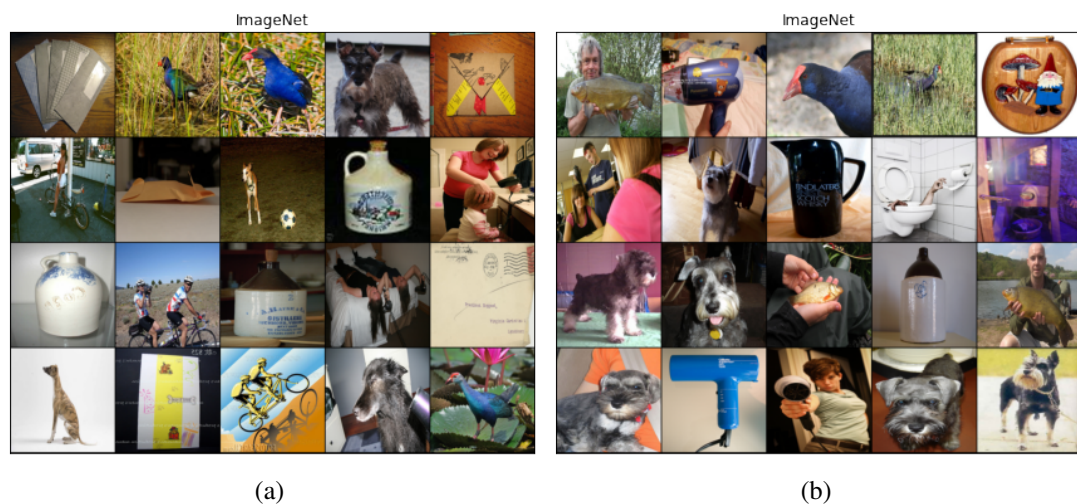


Figure 3. Sample images from CIFAR-10: (a) training dataset and (b) testing dataset

3.2.2. TinyImageNet

The tinyImageNet dataset, which contains 10,000 training pictures as well as 1,000 validation pictures, was used to train this Hex-ResNet architecture. Each of its 200 classes contains 500 training images that are each 64×64 pixels in size Figure 4 displays TinyImageNet’s initial training in Figure 4(a) and testing images in Figure 4(b). Each subset was then classified as two sets: a train set as well as a validation set, each with an 80:20 ratio. The datasets were enhanced using a variety of approaches, including a) center crop and padding; b) rotation; c) scaling; d) shearing; e) translation; f) horizontal flip; and g) vertical flip. Stochastic gradient descent is the optimizer that is employed.

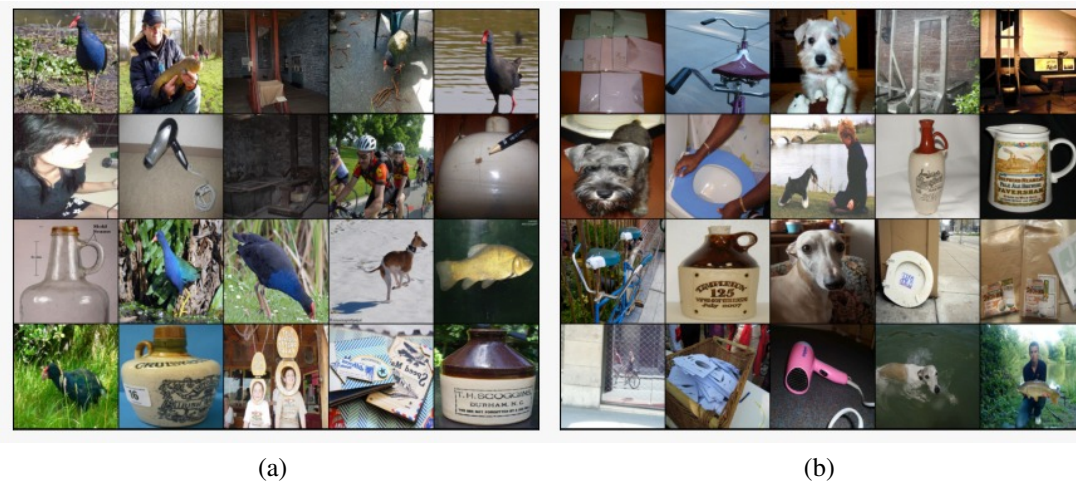


Figure 4. Sample images from ImageNet 2012: (a) training dataset and (b) testing dataset

3.3. Quantitative analysis

The quantitative analysis using the CIFAR-10 and Tiny-ImageNet datasets is depicted in this section. Table 1 shows the Top-1 and Top-5 accuracy and error percentage for CIFAR-10 for various ResNet configurations. The performance of our approach against the baseline residual architectures on CIFAR-10 dataset [17]–[19] is analysed. As observed, our method outperforms all others in terms of validation accuracy, Top-1 and Top-5 accuracy, and error rates except for Hex-ResNet 32. (Hex-ResNet 20, 44, 56). In addition, Hex-ResNet 32, 44, as well as 56 exhibits superior validation accuracy with reduced Top-1 error, accordingly, whenever contrasted to their ResNet equivalents that share the same quantity of layers. Additionally, in contrast to ResNet model variant 20 and higher ResNet model 56, Hex-ResNet reduces the Top-1 error by 0.9% and 0.2%, respectively. Figures 5 (a) to (d) depicts the validation loss vs epochs for both the base ResNet and Hex-ResNet configurations. Note that Hex-ResNet performs better than ResNet in terms of convergence speed and validation loss. This clearly shows that when compared to their respective ResNet equivalents, the feature representation created by the proposed architecture has superior generalization ability and faster convergence.

Table 1. Error rates and accuracy percentage on CIFAR-10. The best scores are indicated by using bold font. (H) indicates the HexResNet configuration

| Model | Parameters | Validation Acc % | Top1 acc % | Top 1 error % | Top 5 acc % | Top 5 error % | Testing accuracy % |
|----------------|------------|---------------------|---------------|------------------|----------------|------------------|-----------------------|
| ResNet - 20 | 272474 | 91.92% | 91.44% | 8.56% | 99.63% | 0.37% | 91.64% |
| ResNet - 20(H) | 287130 | 92.12% | 92.36% | 7.64% | 99.67% | 0.33% | 92.12% |
| ResNet - 32 | 466906 | 92.24% | 92.03% | 7.97% | 99.74% | 0.26% | 92.55% |
| ResNet - 32(H) | 481114 | 92.54% | 92.65% | 7.35% | 99.83% | 0.17% | 93.14% |
| ResNet - 44 | 661338 | 91.64% | 92.14% | 7.86% | 99.76% | 0.24% | 92.83% |
| ResNet - 44(H) | 675098 | 92.92% | 92.94% | 7.06% | 99.92% | 0.08% | 93.27% |
| ResNet - 56 | 855770 | 92.97% | 92.16% | 7.84% | 99.79% | 0.21% | 93.07% |
| ResNet - 56(H) | 869082 | 93.14% | 93.25% | 6.75% | 99.96% | 0.04% | 94.02% |

We present the quantitative analysis correspond to TinyImageNet dataset [20]–[22] in Table 2. We report only the validation scores here due to the unavailability of test set annotations for this dataset. The results in Table 2 show that proposed Hex-ResNet configurations has lesser validation error compared to their respective ResNet models. Additionally, all Hex-ResNet variants outperform traditional ResNets and have a reduced error rate. This demonstrates that the suggested Hex-ResNet architecture is more effectively addressing the vanishing gradient problem and has a high accuracy percentage. Second, the validation accuracy of the Hex-ResNet 20, 32, 44, 56, and 110 layers was higher than that of their traditional ResNet counterparts. One of the important comparisons that show how effective hex residual learning is on incredibly deep systems is this one. Additionally, Hex-ResNet brought down the error rate by 1.3% compared to the classical ResNet variants. The advantages of hex residual learning, particularly for deeper network systems, are evident from this comparison.

Lastly, Figures 6(a) to (d) describes the variation of validation loss with respect to epochs for various ResNet configurations. As can be observed, the Hex-ResNet architecture converges more quickly than the traditional ResNet architecture. As a result, Hex-earliest ResNet's stages of convergence are faster and more precise. Additionally, Hex-ResNet validation loss variation stability is substantially higher than that of its equivalent ResNet equivalents.

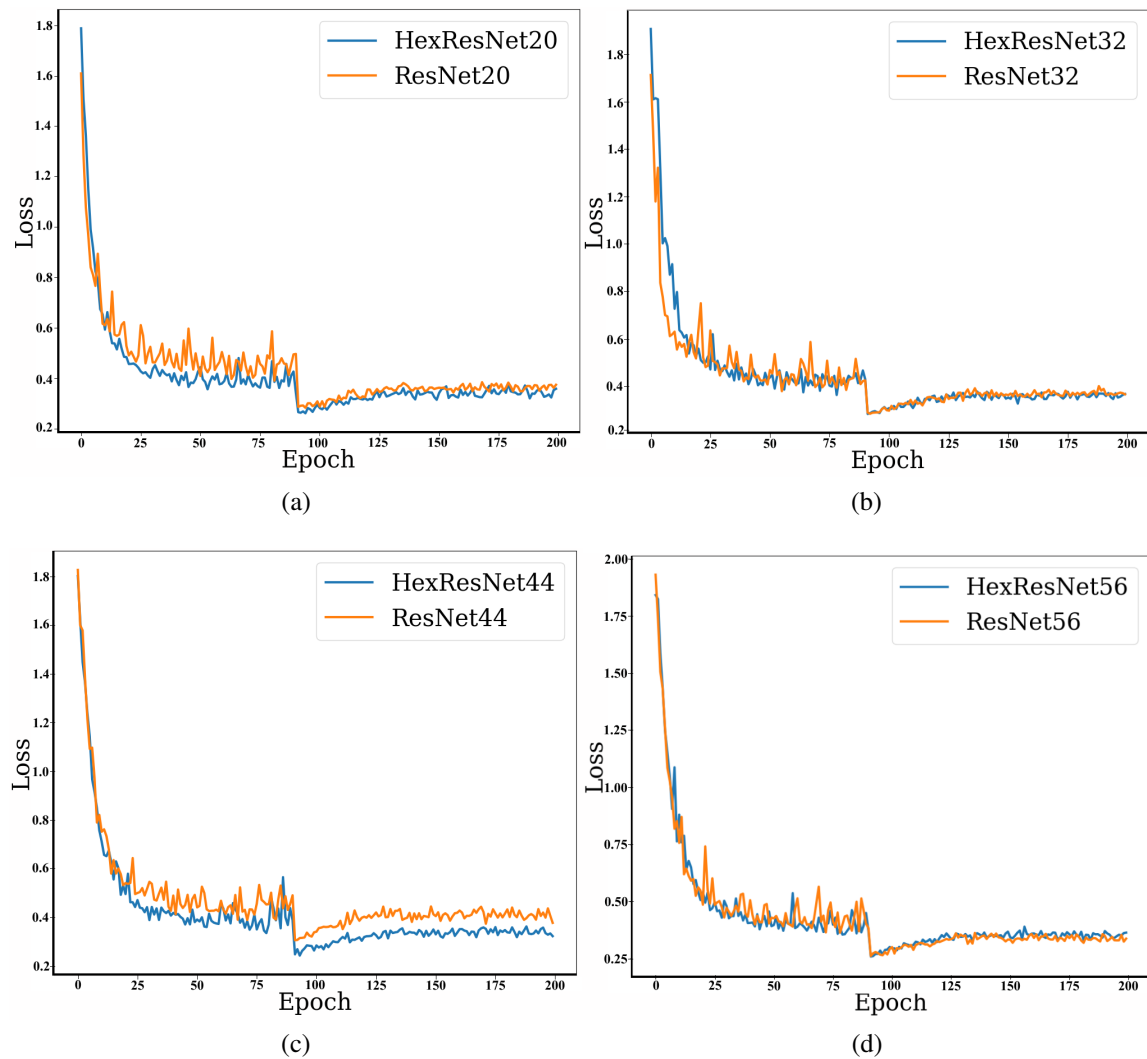


Figure 5. Validation loss variation with respect to epochs for CIFAR-10 dataset. ResNet vs HexResNet: (a) 20 layers, (b) 32 layers, (c) 44 layers, and (d) 56 layers

Table 2. Error rates and accuracy percentage on TinyImageNet. (H) indicates the HexResNet configuration

| Model | Parameters | Validation Accuracy (%) | Error Rate (%) |
|----------------|-----------------|-------------------------|----------------|
| ResNet - 20 | 2,84,824 | 48.05 | 51.95 |
| ResNet - 20(H) | 2,99,480 | 49.51 | 50.49 |
| ResNet - 32 | 4,79,256 | 52.38 | 47.62 |
| ResNet - 32(H) | 4,93,464 | 52.73 | 47.27 |
| ResNet - 44 | 6,73,688 | 53.65 | 46.35 |
| ResNet - 44(H) | 6,87,448 | 54.43 | 45.57 |
| ResNet - 56 | 8,68,120 | 55.01 | 44.99 |
| ResNet - 56(H) | 8,81,432 | 55.71 | 44.29 |

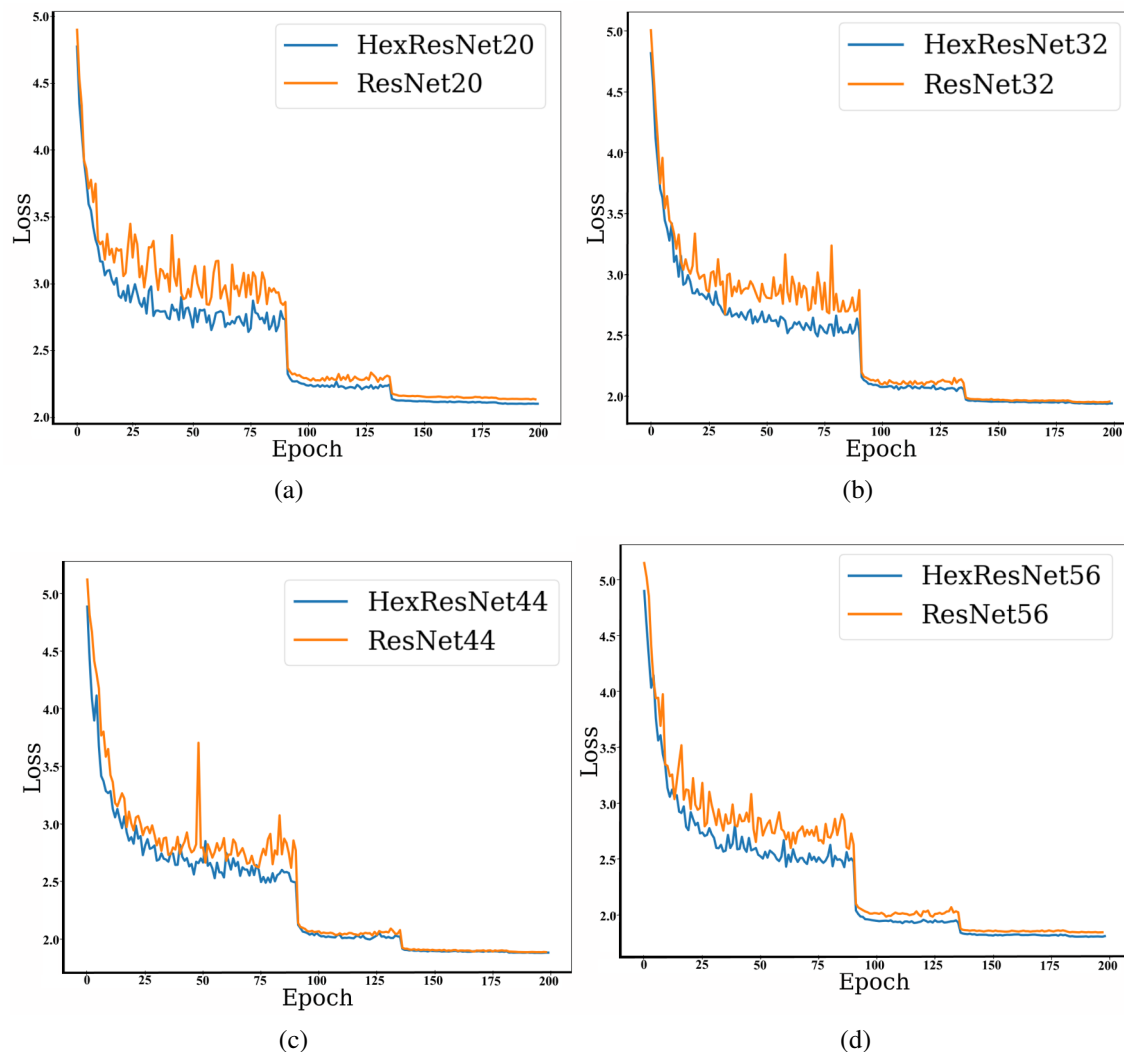


Figure 6. Validation loss variation with respect to epochs for TinyImageNet dataset. ResNet vs HexResNet: (a) 20 layers, (b) 32 layers, (c) 44 layers, and (d) 56 layers

Compared to ResNet built on pure square tessellations, our technique is less susceptible to noisy input. The fact that it is visible from both tables is most significant. Table 2 gives the error rates and parameter comparison on Tiny ImageNet dataset. Most importantly, it can be seen from both the Table 2 in comparison to the amount of parameters in their respective ResNet counterparts, the number of extra parameters brought about by hexagonal convolutions is negligibly small.

Table 3 summarizes the best results obtained on CIFAR-10 dataset and Table 4 summarizes the best results obtained on TinyImageNet dataset using different deep learning models. Results of all the models trained on scratch is shown in Table 4, shows that the best model is HexResNet network achieving 44.29% lowest error percentage.

Table 4. Summary of different state-of-art-architecture performance on TinyImageNet dataset

| Model | # Parameters | #Error% |
|-----------------------------|--------------|--------------|
| InceptionNet [20] | 8.3M | 56.9 |
| ResNet [11] | 11.28M | 53.1 |
| VGG-19 [24] | 40.2M | 49.78 |
| VGG-16 [25] | 36.7M | 48.09 |
| HexResNet (proposed) | 8.8M | 44.29 |

Table 3. Summary of different state of art architecture performance on CIFAR 10 dataset

| Model | # Parameters | #Error (%) |
|----------------------------|--------------|-------------|
| Maxout [15] | >6M | 9.38 |
| Network in Network [16] | ~1M | 8.81 |
| DenseNet [17] | 2.5M | 8.39 |
| Highway Network [23, 24] | 1.25M | 8.80 |
| ResNet [11, 25] | 0.46M | 7.51 |
| HexResNet(proposed) | 0.48M | 6.73 |

4. CONCLUSION

In this research work, we proposed biologically inspired hybrid residual network architecture HexResNet which combines the advantages offered by both square and hexagonal tessellations. We have shown that using hexagonal convolutions can help us advancing the performance of baseline ResNet architectures on both CIFAR-10 and TinyImageNet datasets. From the experimental results, we could show that our approach has better generalisation ability as well as improved convergence properties over the classical ResNet without increasing significant computational overhead due to hexagonal convolutions. Extension to other computer vision applications is a potential future direction of our work.




REFERENCES

- [1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2016, vol. 2016-December, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.
- [3] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision—ECCV 2014: 13th European Conference*, Sep. 2014, pp. 818–833, doi: 10.1007/978-3-319-10590-1_53.
- [4] M. Z. Alom et al., "The history began from AlexNet: a comprehensive survey on deep learning approaches," *arXiv preprint arXiv:1803.01164*, Mar. 2018, doi: 10.48550/arXiv.1803.01164.
- [5] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, Mar. 1994, doi: 10.1109/72.279181.
- [6] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Journal of Machine Learning Research*, 2010, vol. 9, pp. 249–256.
- [7] M. S. Ebrahimi and H. K. Abadi, "Study of residual networks for image recognition," in *Intelligent Computing - Proceedings of the 2021 Computing Conference*, 2021, pp. 754–763, doi: 10.1007/978-3-030-80126-7_53.
- [8] Z. Wu, C. Shen, and A. van den Hengel, "Wider or deeper: revisiting the resnet model for visual recognition," *Pattern Recognition*, vol. 90, pp. 119–133, Jun. 2019, doi: 10.1016/j.patcog.2019.01.006.
- [9] M. Farooq and A. Hafeez, "COVID-ResNet: a deep learning framework for screening of COVID19 from radiographs," *arXiv preprint arXiv:2003.14395*, Mar. 2020, doi: 10.48550/arXiv.2003.14395.
- [10] E. Hoogeboom, J. W. T. Peters, T. S. Cohen, and M. Welling, "HexaConv," *arXiv preprint arXiv:1803.02108*, Mar. 2018.
- [11] C. Steppa and T. L. Holch, "HexagDLy—processing hexagonally sampled data with CNNs in PyTorch," *[SoftwareX]*, vol. 9, pp. 193–198, Jan. 2019, doi: 10.1016/j.softx.2019.02.010.
- [12] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Computer Vision—ECCV 2016: 14th European Conference*, Oct. 2016, pp. 630–645, doi: 10.1007/978-3-319-46493-0_38.
- [13] M. Erdmann, J. Glombitza, and D. Walz, "A deep learning-based reconstruction of cosmic ray-induced air showers," *Astroparticle Physics*, vol. 97, pp. 46–53, Jan. 2018, doi: 10.1016/j.astropartphys.2017.10.006.
- [14] B. Li and Y. He, "An improved ResNet based on the adjustable shortcut connections," *IEEE Access*, vol. 6, pp. 18967–18974, 2018, doi: 10.1109/ACCESS.2018.2814605.
- [15] R. Wightman, H. Touvron, and H. Jégou, "ResNet strikes back: an improved training procedure in timm," *arXiv preprint arXiv:2110.00476*, Oct. 2021.
- [16] T. Schlosser, F. Beuth, and D. Koworko, "Biologically inspired hexagonal deep learning for hexagonal image generation," in *Proceedings - International Conference on Image Processing, ICIP*, Oct. 2020, vol. 2020-October, pp. 848–852, doi: 10.1109/ICIP40778.2020.9190995.
- [17] T. Schlosser, M. Friedrich, and D. Koworko, "Hexagonal image processing in the context of machine learning: Conception of a biologically inspired hexagonal deep learning framework," in *Proceedings - 18th IEEE International Conference on Machine Learning and Applications, ICMLA 2019*, Dec. 2019, pp. 1866–1873, doi: 10.1109/ICMLA.2019.00300.
- [18] J. Luo, W. Zhang, J. Su, and F. Xiang, "Hexagonal convolutional neural networks for hexagonal grids," *IEEE Access*, vol. 7, pp. 142738–142749, 2019, doi: 10.1109/ACCESS.2019.2944766.
- [19] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," pp. 1–60, 2009, doi: 10.1.1.222.9220.
- [20] E. Huynh, "Vision Transformers in 2022: An update on tiny ImageNet," *arXiv preprint arXiv:2205.10660*, May 2022.
- [21] H. Benbrahim and A. Behloul, "Fine-tuned Xception for image classification on tiny ImageNet," in *2021 Proceedings of the International Conference on Artificial Intelligence for Cyber Security Systems and Privacy, AI-CSP 2021*, Nov. 2021, pp. 1–4, doi: 10.1109/AI-CSP52968.2021.9671150.
- [22] Z. Abai and N. Rajmalwar, "DenseNet models for tiny ImageNet classification," *arXiv preprint arXiv:1904.10429*, Apr. 2019.




- [23] H. Jung, M. K. Choi, J. Jung, J. H. Lee, S. Kwon, and W. Y. Jung, "ResNet-based vehicle classification and localization in traffic surveillance systems," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Jul. 2017, vol. 2017-July, pp. 61–67, doi: 10.1109/CVPRW.2017.129.
- [24] T. Gong and H. Niu, "An implementation of ResNet on the classification of RGB-D images," in *Benchmarking, Measuring, and Optimizing: Second BenchCouncil International Symposium, Bench 2019*, vol. 12093 LNCS, USA: Springer International Publishing, 2020, pp. 149–155.
- [25] M. B. Nourian and M. R. Aahmadzadeh, "Image de-noising with virtual hexagonal image structure," in *1st Iranian Conference on Pattern Recognition and Image Analysis, PRIA 2013*, Mar. 2013, pp. 1–5, doi: 10.1109/PRIA.2013.6528440.

BIOGRAPHIES OF AUTHORS



Prathibha Varghese    received her B.Tech. Degree in Electronics and Communication Engineering from M.G University, India, in 2005, and M.Tech. Degree in VLSI and Embedded systems from CUSAT, India, in 2010. She is pursuing Ph.D. in Deep Neural Networks at Noorul Islam Centre for Higher Education (NICHE), Tamil Nadu. Her research interests are primarily at the intersection of deep learning and computer vision applications. Specifically, very keen to build generalizable deep neural network architecture models from very limited training data. She published many research articles in both Scopus and Sci-indexed journals. She can be contacted at email: prathibha@sngce.ac.in.



Dr. G. Arockia Selva Saroja    received her B.E degree in Electronics and Communication Engineering from Manonmaniam Sundaranar University, India, in 1997 and M.E degree in Communication Systems from Madurai Kamaraj University, India, in 1998 and Ph. D Degree from Noorul Islam Centre for Higher Education, India, in 2017. She is currently working as an Associate Professor in the Department of Electronics and Communication Engineering. She has been working in the institution since 1997 and has 15 years of research experience. Her research includes medical image processing, computer vision, machine learning and wireless networks. She can be contacted at email: gassaroja@gmail.com.