HBONext: HBONet with Flipped Inverted Residual

Sanket Ramesh Joshi and Mohamed El-Sharkawy
Department of Electrical and Computer Engineering
IoT Collaboratory lab, Purdue School of Engineering and Technology, IUPUI
joshisr@iu.edu, melshark@iupui.edu

***Abstract*—The top-performing deep CNN (DCNN) architectures are presented every year based on their compatibility and performance ability on the embedded edge applications, significantly for image classification. There are many obstacles in making these neural network architectures hardware friendly due to the limited memory, lesser computational resources, and the energy requirements of these devices. The addition of Bottleneck modules has further helped this classification problem, which explores the channel interdependencies, using either depthwise or groupwise convolutional features. The classical inverted residual block, a well-known design methodology, has now gained more attention due to its growing popularity in portable applications. This paper presents a mutated version of Harmonious Bottlenecks (DHbneck) with a Flipped version of Inverted Residual (FIR), which outperforms the existing HBONet architecture by giving the best accuracy value and the miniaturized model size. This FIR block performs identity mapping and spatial transformation at its higher dimensions, unlike the existing concept of inverted residual. The devised architecture is tested and validated using CIFAR-10 public dataset. The baseline HBONet architecture has an accuracy of 80.97% when tested on CIFAR-10 dataset and the model's size is 22 MB. In contrast, the proposed architecture HBONext has an improved validation accuracy of 88.30% with a model reduction to a size of 7.66 MB.**

***Index Terms*—Convolution Neural Networks (CNN), Harmonious Bottleneck (HBO), Flipped Inverted Residual (FIR), CIFAR-10.**

I. Introduction

ImageNet classification challenge that started in 2012 has been a driving force to develop today's best performing CNN architectures, which are growing deeper and becoming more efficient with every passing year [1], [2]. With the increasing quest of improving these architectures for higher accuracy yield, many sophisticated mathematical approaches are under development but at the cost of higher computational and storage requirements. Many novel studies have emerged to build further light-weight CNN models that are compatible and feasible for actual real-time implementation and to investigate this critical problem.

The basic building modules of any convolutional neural networks (CNNs) are convolutional and pooling layers. In comparison, the newly developed light architectures contain depthwise separable convolutions that have phenomenally improved accuracy performance. These commonly used convolutional techniques deal with the input's spatial dimensions and depth dimension [3], [4]. The MobileNets family of models uses standard convolution for the first layer and build other layers using depthwise separable convolution, which is a mix of pointwise and depthwise convolutions [5], [6]. However, both the convolution methods give similar results, just that the depthwise separable convolution is much faster and so helpful. Residual block is the next breakthrough, the skip-connection block that learns residual function for the layer input and, in practice, used as bottleneck blocks as they are less computationally intensive [8]. The purpose here is to make residual blocks thinner to increase the depth, reduce the total number of parameters, and bring down complex matrix multiplications. MobileNetv2 [9] is built using the inverted residual concept with linear bottlenecks, which enhances information flow in the representation space, and helps achieve better accuracy results than its predecessors. To further study the inverted residual, we investigate the sandglass approach that deals with the high-dimensional residuals that transmit more gradients back to enhance the network training in a better fashion [4]. MobileNets [4], [5], [7], principles focus more on the channel transformation but neglects the orthogonal dimensions, and so with the introduction of HBONets this dimension has been explored which helped for model accuracy gains [3]. HBONet is a study that presents a unique arrangement of spatial and channel transformations in a bilaterally symmetrical fashion with a shared benefit that yields improved performance than MobileNetv2. The goal is to build the best possible light-weight image classification architecture with a prospect of its real-time implementation on the embedded edge. This work presents a combination of the concept of HBONets embedded with a sandglass approach for residual (FIR) and proposes a new architecture called HBONext.

The paper composition is as follows: The literature history is discussed in Section II, and the novel proposed architectural design is described in Section III. The steps for training and validating the architecture are described in Section IV and Section V presents the detailed results to understand the contribution of various parameters. The conclusion and prospects are found in Section VI.

II. Background

From the last few years, several variants of neural network architectures have emerged, especially targeting mobile edge devices. This section briefly reviews the already discovered light models, and exclusively mentions transformational

This is the author's manuscript of the article published in final edited form as:

Joshi, S. R., & El-Sharkawy, M. (2021). HBONext: HBONet with Flipped Inverted Residual. 2021 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS), 1–5. https://doi.org/10.1109/DTS52014.2021.9498121

methods with spatial dimensions and gives an overview of the inverted residual technique.

SqueezeNet [10], [11] is one of the first light models that use smaller convolution layers with a 1x1 convolutional squeeze layer followed by two parallel 3x3 convolutions expand layer but with fewer parameters than AlexNet [1], [10]. Then comes MobileNetv1 [5], which replaces expensive layers with a cheaper depthwise separable convolution to further improve computational efficacy. MobileNetv2 [9] is exactly the opposite of SqueezeNet. It first expands and then reduces the number of channels; it also introduces an inverted residual block for further parameter reduction compared to MobileNetv1. MnasNet-A1 [12] has an expansion, a depthwise, and a bottleneck layer, with a residual connection to the previous bottleneck. It also additionally introduces a squeeze and excitation method, which compresses the channel vector first and then tries to restore only important features. There are various types of convolutions that are used to extract important characteristics from an input image. In the Harmonious Bottleneck method, depthwise separable convolutions are used to focus both on the spatial and channel dimensions. This process is comprised of two sections first down-sampling the spatial dimension keeping the channels constant (H/s x W/s x C_1) and then expanding the channels (H/s x W/s x t x C_1), whereas up-sampling the spatial dimensions with channel reduction by half (H x W x C_2/2), and further concatenating with the input's partial channels (H x W x C_2) or its pooled version. Fig 1. represents the Operation Bottleneck block of HBONet, in which (H x W) is the height and weight of the input/output feature, t is the channel expansion factor, s is a stride, C_1 is the input channel, and C_2 is the output channels of the block. Thus, the final calculated value of this module is:

$$B/s^2 + (H/s \times W/s \times C_1 + H \times W \times C_2) \times K \times K \quad (1)$$

Here, K denotes the kernel size, and B signifies the calculated value of the inserted blocks between the two operating sections. This implementation, if adopted in any CNN's architectures, produce lighter models and gives impressive accuracy. Due to the limitations of lower dimensionality at the start of inverted residual blocks, which is expected to hinder the efficient capture of useful information due to channel compression, we adopt the technique used in sandglass block, which has a wider architecture and is expected to better gradient confusion as per the recent study [3], [4]. In this work, we successfully integrate these two ideas and explore the design space modifications.

The non-linearity activation function ReLU6 was used in the original implementation of HBONet. Since it is able to concentrate on the positive values only, in order to also preserve the negative values with minimal cost of computation, the use ELU activation is considered with the HBONext version. It has alpha, which is a positive constant value typically used between 0.1 and 0.3. ELU helps generate more accurate results by converging to zero faster but cannot overcome exploding gradient problems [13], [14]. This is represented as:

$$\begin{aligned} f(x) &= x, && forx > 0 \\ &= \alpha(e^x - 1), && forx \leq 0 \end{aligned} \quad (2)$$

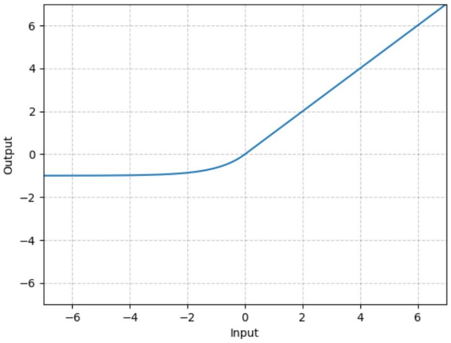

Fig. 2: ELU non-linear activation [14].

This block is used in the DHBneck block of HBONext, which has significantly boosted the model accuracy along with the Cosine Annealing learning rate scheduling strategy [18]. The plan is to also implement this architecture for image classification using CIFAR-10 [15] on embedded edge Bluebox by NXP using RTMaps [19]. This process will validate the modified architecture HBONext for the achieved accuracy to implement it any framework supported embedded hardware.

III. PROPOSED HBONEXT ARCHITECTURE

This section presents the derived harmonious bottleneck structure (DHbneck). It replaces the existing inverted residual block of the baseline with a flipped inverted residual (FIR), thus producing a new light-weight architecture called HBONext, as seen in Table I. below:

In this table, t denotes the expansion factor, c is the channel output, n is the number of times the block repeats, and s is the stride value.

Here are few fundamental approaches followed for its implementation.

- The non-Linear activation function is carefully replaced from Relu6 with ELU in place.
- An element-wise skip connection is added, which helps in overcoming the vanishing gradient problems. Also, the FIR block's modification proceeds as Dwise-Pwise-Pwise-Dwise to achieve a sandglass approach, unlike the one used in the baseline architecture.
- Reconsidering the bottleneck module for its spatial dimensions and channel dimensions significantly to help further reduce the model size.

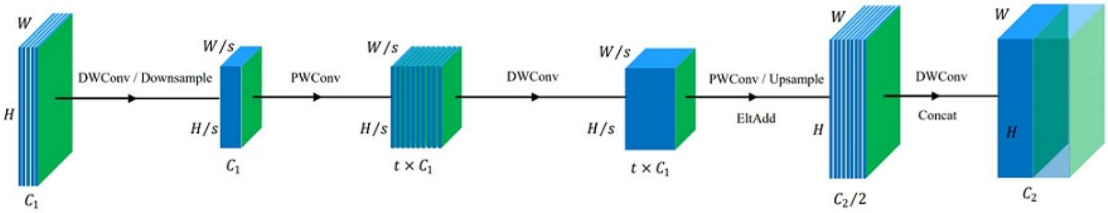

Fig. 1: Operation Bottleneck block of HBONet [1].

TABLE I: HBONext Architecture

Input size	Operator	t	c	n	s
$32^2 \times 3$	conv2d 3 ×3	-	64	1	2
$16^2 \times 64$	FIR block	2	32	1	1
$16^2 \times 32$	DHbneck	2	16	2	1
$16^2 \times 16$	DHbneck	2	32	4	2
$8^2 \times 32$	DHbneck	2	64	4	2
$4^2 \times 64$	DHbneck	2	96	4	2
$2^2 \times 96$	DHbneck	2	128	2	1
$2^2 \times 128$	DHbneck	1	256	2	1
$2^2 \times 256$	conv2d 1 x 1	1	512	1	2
$2^2 \times 512$	FIR block	2	256	2	2
$1^2 \times 256$	FIR block	2	128	1	1
$1^2 \times 128$	FIR block	1	10	1	1
$1^2 \times 10$	conv2d 1 x 1	-	1024	1	1
$1^2 \times 1024$	avgpool 7 x 7	-	-	1	-
$1^2 \times 1024$	FC Layer	-	k	-	-

*t, c, n, s are elaborated below

A. Derived Harmonious Bottleneck

A simple change is made by choosing the kernel of 3 x 3 and the ELU activation function. The block in Fig.3 gives the detailed operational view of the HBONext bottleneck structure, and the FIR skip connection with the appropriate selection of stride value. Stacking smaller convolutional layers is easier than stacking bigger ones. It also seeks to adjust the outcome, built on the hypothesis that a positive outcome would occur. In this implementation we make use of 3 × 3 as a kernel size since the odd-sized filters symmetrically split the previous layer pixels along the output pixel and this size also help for lesser parameter goal unlike that in the original work with kernel size of 5 × 5.

B. Flipped Inverted Residual Block (FIR)

The sandglass block in Fig.4(b) is mainly developed to protect more feature information while transitioning from the lower layer to top layers so that residual connection is placed to connect higher dimension features. It consists of light-weight 3 x 3 depthwise convolutions applied onto the higher dimensions to extract rich spatial information.

IV. Training Setup

The main idea was to train our HBONext model on the CIFAR-10 dataset for image classification purposes. CIFAR-10 is a set of 60,000 images (32 x 32) classified into ten categories that is widely used in deep learning and machine vision applications. This public dataset is further split into two parts for testing and evaluation to better understand the model's efficiency depending on its accuracy parameter.

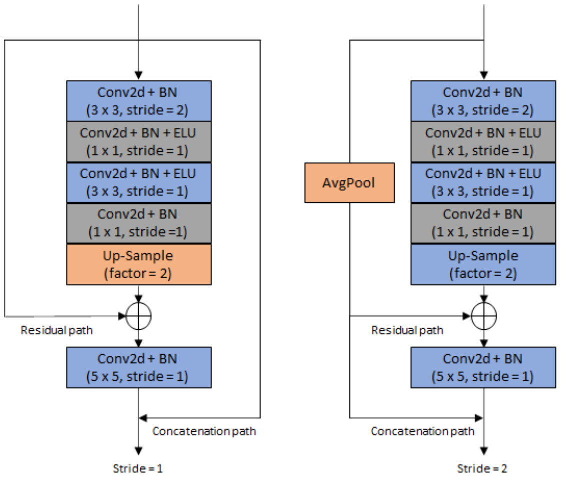

Fig. 3: Harmonious Bottleneck module with different strides

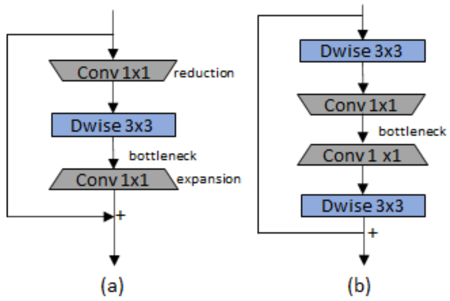

Fig. 4: Residual blocks (a) traditional bottleneck arrangement, (b) sandglass block with bottleneck arrangement

A. Hardware and Software Used

The complete model training was accomplished using Google Colab environment, a simple to use platform which provides free access to any available GPUs by the Google servers. The preliminary results are also generated using NVIDIA GeForce GTX 1080Ti GPU. The PyTorch based packages like Livelossplot, torchsummaryX were used for graphical representation and calculations of the number to parameters. We implemented our proposed model with the

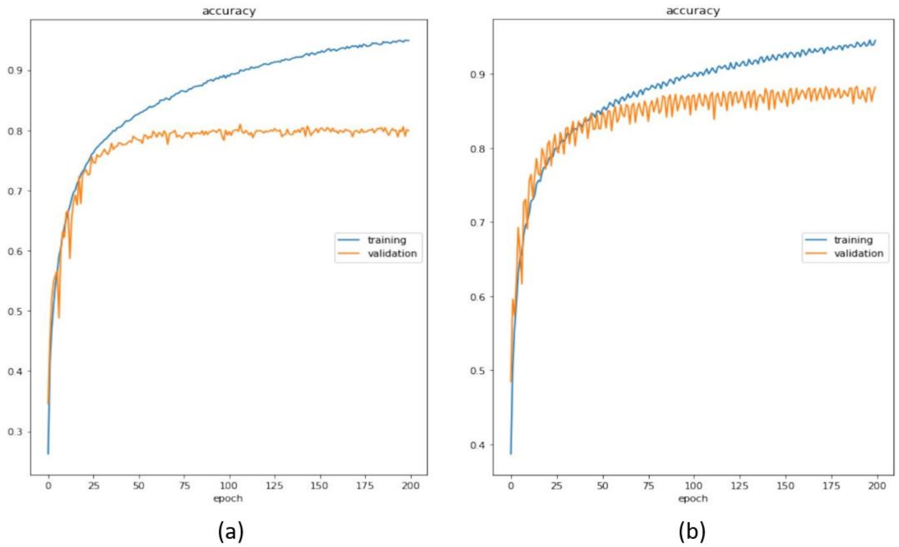

Fig. 5: Accuracy vs the number of epochs: (a) HBONet (baseline), (b) HBONext (proposed architecture)

standard values of width multiplier like 1, 0.75, 0.5, and 0.25. The purpose of using different widths is to make the network thinner at each layer consistently. The entire model is trained using the Stochastic Gradient Descent (SGD) optimizer, with the momentum fixed to 0.9, the weight decay adjusted to 4e-5, and Nesterov included. A batch size of 128 is used in the model, and a learning rate of 0.01. Also, setting the learning rate using a cosine annealing scheduler has helped us achieve competitive results.

V. RESULTS

The proposed model, HBONext, trained from scratch using CIFAR-10 increased the accuracy gain by 9.06% with a model size reduction of about 65.18% compared to the baseline model, HBONet. The integration of harmonious bottlenecks with the FIR strategy has helped in achieving the below-mentioned results. The training graphs on the CIFAR-10 is visualized using the Pytorch platform and the Livelossplot package, as seen in Fig.5, which is a graph of Accuracy versus the number of iterations for a width value of 1.0. The proposed model HBONext has an 88.30% accuracy for validation on CIFAR-10 and with a model size of 7.66 MB, comparable to its reference model's accuracy of 80.97% and initial model size of 22 MB.

TABLE II: Width Multiplier Variants

Width Value	Accuracy	Model's size
HBONext(1.5)	89.60%	16.08 MB
HBONet (1.5)	82.75%	48.34 MB
HBONext(1.0)	**88.30%**	**7.66 MB**
HBONet (1.0)	80.97%	22.00 MB
HBONext(0.75)	**87.70%**	**4.67 MB**
HBONet (0.75)	79.93%	13.80 MB
HBONext(0.50)	85.30%	2.48 MB
HBONet (0.50)	76.25%	7.04 MB
HBONext(0.25)	79.80%	1.07 MB
HBONet (0.25)	71.22%	2.65 MB

TABLE II summarizes the model variants based on their width multiplier values. This model was successfully trained using the Google Colab environment on CIFAR-10 with a specific width multiplier value. Its corresponding accuracy and model size values are carefully noted to spot the differences. The width multiplier values of 1.5, and 0.75, significantly improved the results, and its respective accuracy and model size is highlighted in this table.

VI. Conclusion

This work demonstrates an image classification competency using the CIFAR-10 dataset with the proposed HBONext architecture. HBONext is a derived version of its primitive block harmonious bottleneck and a mutated version called Flipped Inverted Residual (FIR) block. When trained on CIFAR-10 using the Pytorch framework, the model gives an improved validation accuracy of 88.30% with a model reduction to a size of 7.66 MB. The study presents in detail the comparison of our model with respect to the different values of width multiplier trained with the optimizing techniques and the cosine annealing scheduling methods for learning. With the different values of width multipliers, the lighter models are achieved that can be easily implemented on any embedded vision application. The future scope involves use of different techniques like data augmentation, and learning rate scheduling meachnism to improve performance metric and finally deploying this proposed model on the embedded edge hardware to test its real-time application for image classification purposes.

References

[1] Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1097-1105.

[2] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3), 211-252.

[3] Li, D., Zhou, A., Yao, A. (2019). Hbonet: Harmonious bottleneck on two orthogonal dimensions. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 3316-3325).

[4] Daquan, Z., Hou, Q., Chen, Y., Feng, J., Yan, S. (2020). Rethinking bottleneck structure for efficient mobile network design. arXiv preprint arXiv:2007.02269.

[5] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

[6] Hollemans, M. (2017, June 14), Google's MobileNets on the iPhone. [Online]. Available: https://Machinethink.Net/Blog/Googles-Mobile-Net-Architecture-on-Iphone/.

[7] Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., ... Adam, H. (2019). Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 1314-1324).

[8] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[9] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).

[10] Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size. arXiv preprint arXiv:1602.07360.

[11] Duggal, J. K., El-Sharkawy, M. (2019, September). Shallow SqueezeNext: An Efficient Shallow DNN. In 2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES) (pp. 1-6). IEEE.

[12] Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., Le, Q. V. (2019). Mnasnet: Platform-aware neural architecture search for mobile. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2820-2828).

[13] Nwankpa, C., Ijomah, W., Gachagan, A., Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. arXiv preprint arXiv:1811.03378.

[14] ELU — PyTorch 1.7.1 documentation. (2019). [Online]. Available: https://pytorch.org/docs/stable/generated/torch.nn.ELU

[15] "CIFAR-10 and CIFAR-100 datasets." https://www.cs.toronto.edu/ kriz/cifar.html (accessed Nov. 28, 2020).

[16] Hollemans, M. (2020, April 8). New mobile neural network architectures. [Online]. Available: https://machinethink.net/blog/mobile-architectures/

[17] Brownlee, J. (2019, July 5). Convolutional Neural Network Model Innovations for Image Classification. Machine Learning Mastery. [Online]. Available: https://machinelearningmastery.com/review-of-architectural-innovations-for-convolutional-neural-networks-for-image-classification/

[18] Lau, S. (2018, June 20). Learning Rate Schedules and Adaptive Learning Rate Methods for Deep Learning. Medium. [Online]. Available: https://towardsdatascience.com/learning-rate-schedules-and-adaptive-learning-rate-methods-for-deep-learning-2c8f433990d1

[19] Venkitachalam, S., Manghat, S. K., Gaikwad, A. S., Ravi, N., Bhamidi, S. B. S., El-Sharkawy, M. (2018). Realtime Applications with RTMaps and Bluebox 2.0. In Proceedings on the International Conference on Artificial Intelligence (ICAI) (pp. 137-140). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).