ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 6, No. 1, 1378-1383 (2021)
www.astesj.com
Special Issue on Multidisciplinary Sciences and Engineering

ASTES Journal
ISSN: 2415-6698

A-MnasNet and Image Classification on NXP Bluebox 2.0

Prasham Shah*, Mohamed El-Sharkawy

IoT Collaboratory at IUPUI, Department of Electrical and Computer Engineering, Purdue School of Engineering and Technology, Indianapolis, 46202, USA

ARTICLE INFO

Article history:
Received: 28 October, 2020
Accepted: 08 February, 2021
Online: 28 February, 2021

Keywords:
Deep Learning
A-MnasNet
NXP Bluebox 2.0
Cifar-10

ABSTRACT

Computer Vision is a domain which deals with the challenge of enabling technology with vision capabilities. This goal is accomplished with the use of Convolutional Neural Networks. They are the backbone of implementing vision applications on embedded systems. They are complex but highly efficient in extracting features, thus, enabling embedded systems to perform computer vision applications. After AlexNet won the ImageNet Large Scale Visual Recognition Challenge in 2012, there was a drastic increase in research on Convolutional Neural Networks. The convolutional neural networks were made deeper and wider, in order to make them more efficient. They were able to extract features efficiently, but the computational complexity and the computational cost of those networks also increased. It became very challenging to deploy such networks on embedded hardware. Since embedded systems have limited resources like power, speed and computational capabilities, researchers got more inclined towards the goal of making convolutional neural networks more compact, with efficiency of extracting features similar to that of the novel architectures. This research has a similar goal of proposing a convolutional neural network with enhanced efficiency and further using it for a vision application like Image Classification on NXP Bluebox 2.0, an autonomous driving platform by NXP Semiconductors. This paper gives an insight on the Design Space Exploration technique used to propose A-MnasNet (Augmented MnasNet) architecture, with enhanced capabilities, from MnasNet architecture. Furthermore, it explains the implementation of A-MnasNet on Bluebox 2.0 for Image Classification.

1 Introduction

Computer Vision is becoming an essential application in this modern world. With advances in technology autonomous cars, drones and UAVs, robots etc have been enabled with vision capabilities. These technologies use Convolutional Neural Networks (CNNs) to process the images or video input. They are used for applications like Image Classification, Object Detection and Semantic Segmentation etc. CNNs are a part of Deep Learning, which is a subset of Machine Learning (ML). Due to the advances in AI, ML capabilities increased and this enabled a whole new field of Deep Learning. This field deals with creating, optimizing and implementing algorithms which enables technology to become self-reliant and gain human level precision. The prime goal is automation of these technologies in a way that they are able to operate perfectly without any human intervention.

Convolutional Neural Networks are used in developing healthcare technologies, which have almost human level precision and are used to save lives in hospitals. Medical imaging devices which, detect even minute particles, are used for diagnosis of various diseases. Autonomous cars which will make roads safer and will reduce fatal life-threatening accidents, Autonomous Drones which will revolutionize the logistics and will deliver packages more efficiently, UAVs which will aid militaries with surveillance and help prevent wars, smart cameras which will be able to recognize people and track their activities to reduce crimes. Smart imaging of atmosphere for weather prediction and warnings for natural calamities like tsunami, tornadoes etc. These are a few examples of how CNNs are making a major impact in our lives.

These technologies require computational power, speed, accuracy and precision. In order to make them efficient, the algorithms have to be fast having low latency, working efficiently on low power, being more accurate and consuming less memory. CNNs have so many layers so as the architectures become deeper and wider, giving more accuracy, their computational cost increases. These CNN models have to be more compact and should work as efficiently as the novel architectures. With new research and advent of new algorithms, now it is possible to make CNNs more efficient having

*Corresponding Author: Prasham Shah, Email: pashah@purdue.edu

competitive size and accuracy. This research aims to contribute towards this same goal, making CNNs compact in terms of model size and increasing its accuracy so that they can be used for such applications[1].

A-MnasNet [2] is derived by Design Space Exploration of MnasNet. New algorithms were implemented for more efficiency with competitive size and accuracy. Furthermore, it is deployed on NXP Bluebox 2.0 for Image Classification [3]. RTMaps Studio is used to deploy the model on the hardware by establishing a TCP/IP connection.

In this paper, section 2 discusses MnasNet architecture. Section 3 introduces A-MnasNet [2] and discusses the new algorithms implemented to achieve better results. Section 4 demonstrates deployment of A-MnasNet [2] on NXP Bluebox 2.0 for Image classification [3]. Section 5 states the hardware and software requirements to conduct this research. Section 6 illuminates the results that demonstrate the model comparison, model size and accuracy trade-offs and deployment results. Section 7 is conclusion which gives the gist of this research.

2 Prior Work

This section gives an insight on MnasNet [4] architecture. It explains the features of this architecture and what can be improved in order to make it more efficient.

2.1 MnasNet Architecture

MnasNet [4] was introduced in 2019 by Google Brain Team with a goal of proposing an approach of neural architecture search which would optimize the overall accuracy of the architecture and reduce the latency of the model on mobile devices. MnasNet outperformed all existing novel architectures on ImageNet dataset with enhanced accuracy. This architecture extracts features by using depthwise separable convolutional layers with 3*3 and 5*5 filter dimensions and mobile inverted bottleneck layers. It has ReLU [5] activation layers which introduces non-linearity to the network. It uses Batch normalization to adjust and scale the output of activation layers. Dropout regularization is used to reduce the overfitting in the network.

Figure 1 shows MnasNet architecture and illuminates its convolutional layers. This architecture was designed for ImageNet dataset so the input dimensions were set to 224*224*3. It had a validation accuracy of 75.2% and outperformed all other existing state-of-the-art architectures. For this research, MnasNet was used with CIFAR-10 [6] dataset so the input dimensions were modified to 32*32*3. Table 1 shows various layers of the architecture.

An important aspect to observe about MnasNet is that it only explores the channel dimensions of the feature maps throughout its architecture. The convolutional layers fail to explore the spatial feature scale in the network. The activation layers used in the architecture have become outdated and today there are better algorithms which work more efficiently then the used activations.

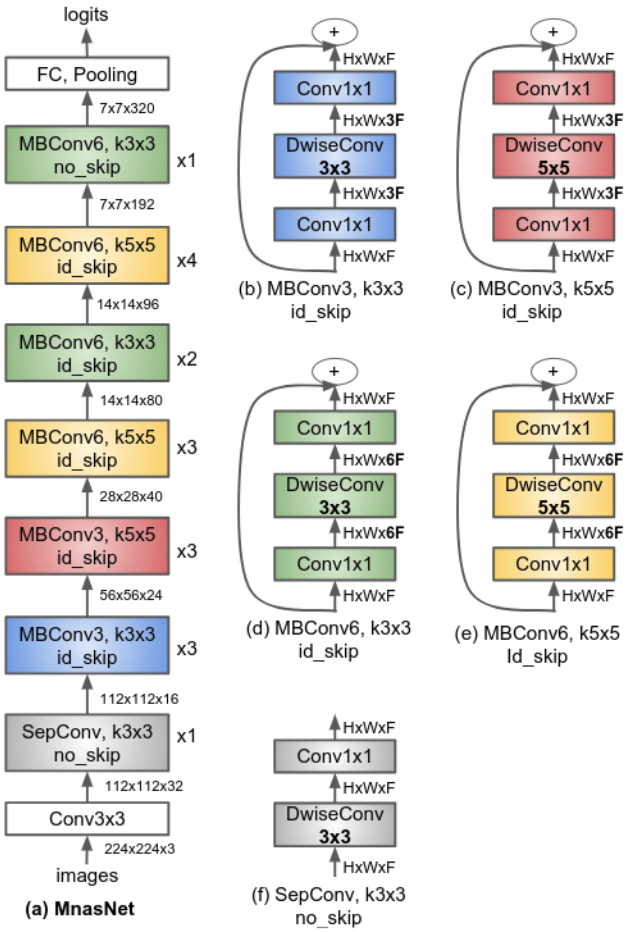

Figure 1: MnasNet architecture

Table 1: MnasNet Architecture

MnasNet Architecture					
Layers	Convolutions	t	c	n	s
$32^2 \times 3$	Conv2d 3×3	-	32	1	1
$112^2 \times 32$	SepConv 3×3	1	16	1	2
$112^2 \times 16$	MBConv3 3×3	3	24	3	2
$56^2 \times 24$	MBConv3 5×5	3	40	3	2
$28^2 \times 40$	MBConv6 5×5	6	80	3	2
$14^2 \times 80$	MBConv6 3×3	6	96	2	1
$14^2 \times 96$	MBConv6 5×5	6	192	4	1
$7^2 \times 192$	MBConv6 3×3	6	320	1	1
$7^2 \times 320$	FC,Pooling			10	

t: expansion factor, c: number of output channels, n: number of blocks and s: stride

3 A-MnasNet Architecture

This section gives an insight on the A-MnasNet [2] CNN architecture and its features.

3.1 Convolution Layers

Harmonious Bottleneck Layers [7] were introduced in the architecture. This special convolutional layers extract features across the spatial and channel dimensions. They are more efficient then the depthwise separable convolutional layers. Harmonious layers consist of spatial contraction-expansion keeping the number of channels constant and channel expansion-contraction keeping the spatial dimensions constant. This down-scaling reduces the number of parameters of the model. Hence, the model size is reduced. Since it extracts features more efficiently, the overall model accuracy is enhanced.

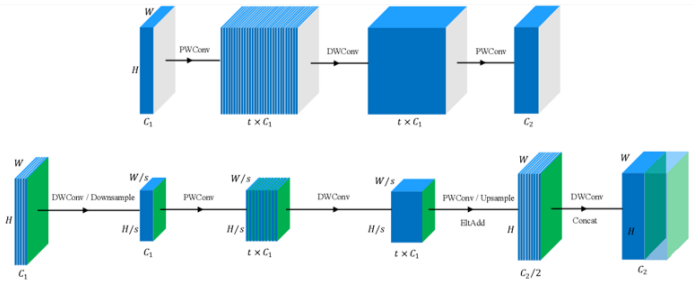

Figure 2: Comparison of Depthwise Separable Convolution Layer and Harmonious Bottleneck Layer.[7]

A comparison between Harmonious bottleneck layers [7] and depthwise separable layers is demonstrated in Figure 2. Here, Input size and output size is represented by H and W respectively. C1 and C2 denote the input and output feature channels where K is the size of the filter and s is the stride value.
The total cost of depthwise separable convolution is

$$(H \times W \times C1 \times K \times K) + (H \times W \times C1 \times C2) \quad (1)$$

The total cost of harmonious bottleneck layer is

$$B/s^2 + (H/s \times W/s \times C1 + H \times W \times C2) \times K^2 \quad (2)$$

where, B is the computational cost of the blocks inserted between the spatial contraction and expansion operations.

Dowmsampling the channel dimensions and extracting features from the spatial dimensions results in reduced spatial size of wide feature maps. The parameters of the model are reduced by a factor of the stride value. As a result the model size reduces and the accuracy of the model increases.

Hence, these layers were added to enhance the performance of the model. Table 2 shows various layers of A-MnasNet [2].

Table 2: A-MnasNet Architecture

A-MnasNet Architecture					
Layers	Convolutions	t	c	n	s
$32^2 \times 3$	Conv2d 3×3	-	32	1	1
$112^2 \times 32$	SepConv 3×3	1	16	1	2
$112^2 \times 16$	MBConv3 3×3	3	24	3	2
$112^2 \times 24$	Harmonious Bottleneck	2	36	1	1
$56^2 \times 36$	MBConv3 5×5	3	40	3	2
$112^2 \times 40$	Harmonious Bottleneck	2	72	1	2
$28^2 \times 72$	MBConv6 5×5	6	80	3	2
$112^2 \times 80$	Harmonious Bottleneck	2	96	4	2
$14^2 \times 96$	MBConv6 3×3	6	96	2	1
$14^2 \times 96$	MBConv6 5×5	6	192	4	1
$7^2 \times 192$	MBConv6 3×3	6	320	1	1
$7^2 \times 320$	FC,Pooling			10	

t: expansion factor, c: number of output channels, n: number of blocks and s: stride

3.2 Learning Rate Annealing or Scheduling

This parameter defines the rate at which the network learns the parameters. It sets the speed of learning features. It a large value is set then the speed of training would increase but with that, the accuracy would decrease as the efficiency of learning more features would reduce. This parameter is tuned throughout the training process in a descending manner. The learning rates are scheduled so that the step size in each iteration gives out the best output for efficient training. There are various methods to set learning rates. Some them have been demonstrated in Figure 3.

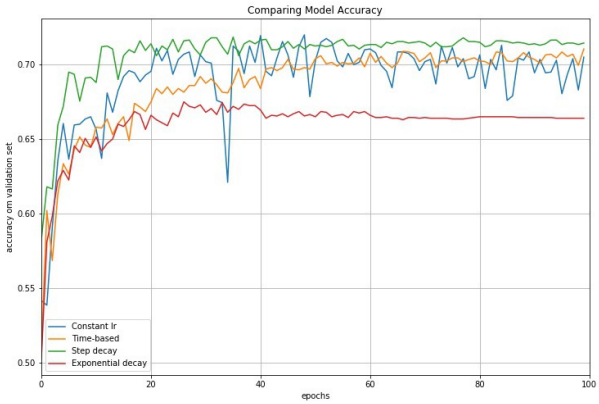

Figure 3: Comparision of different LR scheduling methods.

3.3 Optimization

Optimization is an essential aspect of enhancing the efficiency of the CNN models. Basically, it is a method of correcting the calculated error. It might sound simple but it just as complicated. There are various optimizers which are used to do this optimization. There are constant learning rate algorithms and adaptive learning rate algorithms. For this research, Stochastic Gradient Descent optimizer

[8] was used because of its enhanced performance on optimizing A-MnasNet [2]. It was used with varying values of learning rates during training with a momentum set to 0.9.

3.4 Data Augmentation

This step is used to preprocess the dataset. Various transformations are applied on the images. The purpose of data augmentation is the enhance the accuracy of the model. In other words, it enables the CNN model to learn features more efficiently. AutoAugment [9] is implemented to preprocess the dataset. It uses reinforcement learning to automatically select the transformations that are most efficient for the CNN model. It defines two parameters for the process. First is the transformation and the other is the magnitude of applying it. These two together are defined as a Policy. A Policy consists of sub-policies which have those two parameters. Different sub-policies are selected by the controller which best suit the training. This process is demonstrated in Figure 4. It was evident that by using AutoAugment [9] the accuracy of A-MnasNet increased to 96.89% from 92.97%.

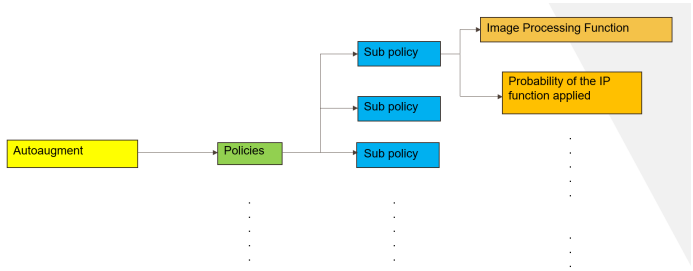

Figure 4: AutoAugment

4 Image Classification on NXP Bluebox 2.0

A-MnasNet [2] was deployed on NXP Bluebox 2.0 for Image Classification [3]. BlueBox 2.0 by NXP is a real-time embedded system which serves with the required performance, functional safety and automotive reliability to develop the self-driving cars. It is a ASIL-B and ASIL-D compliant hardware system which provides a platform to create autonomous applications such as ADAS systems, driver assistance systems.

There are three processing units in Bluebox 2.0. There is a vision processor, S32V234, a radar processor, S32R274, and a compute processor, LS2048A. This enables to develop autonomous applications. The vision processor and the compute processors are used to perform Image Classification [3] using A-MnasNet [2]. Figure 5 shows the methodology to deploy A-MnasNet [2] on Bluebox 2.0 for Image Classification.

After training A-MnasNet [2], the model is imported in the RTMaps Studio using its python component. A TCP/IP connection is used for data transmission between RTMaps Design Studio and NXP Bluebox 2.0. The model was successfully implemented on the hardware and was able to predict the object in the input images accurately.

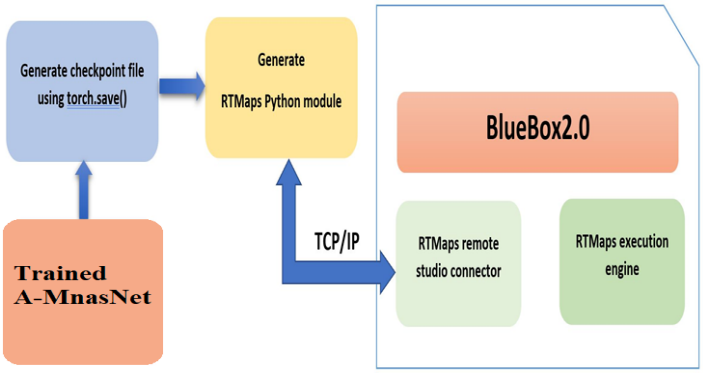

Figure 5: Deployment of A-MnasNet on NXP BlueBox 2.0

RTMaps Studio was used for real-time implementation of A-MnasNet[2] on NXP Bluebox 2.0. It is an asynchronous high performance platform which has an advantage of an efficient and easy-to-use framework for fast and robust developments. It provides a modular toolkit for multimodal applications like ADAS, Autonomous vehicles, Robotics, UGVs, UAVs, etc. It provides a platform to develop, test, validate, benchmark and execute applications.

Figure 6: RTMaps connection with Bluebox 2.0

Figure 6 shows the role of RTMaps in the implementation. It provides an interface between the hardware and the embedded platform. It uses a TCP/IP connection by assigning a static IP address to enable data transfer between them.

5 Hardware and Software used

- NXP Bluebox 2.0
- Aorus Geforce RTX 2080Ti GPU
- Python version 3.6.7.
- Pytorch version 1.0.

- Spyder version 3.6.
- RTMaps Studio
- Tera Term
- Livelossplot

6 Results

MnasNet [4] achieved 80.8% accuracy with CIFAR-10 [6] dataset achieving 12.7 MB model size without data augmentation. New convolutional blocks resulted in an increase of accuracy to 92.97%. Furthermore, AutoAugment [9] was implemented for enhancing overall accuracy which increased from 92.97% to 96.89%. The final model is of 11.6 MB and a validation accuracy of 96.89%.

Implementation of new algorithms resulted in enhanced accuracy. It is demonstrated in Table 3. The overall accuracy increases by 16.09% with a slight decrease in model size.

Table 3: Comparison with baseline architecture

Comparison of models		
Architecture	Model Accuracy	Model size (in MB)
MnasNet	80.8%	12.7
A-MnasNet	96.89%	11.6

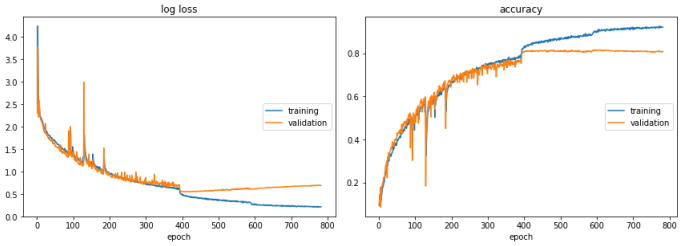

Figure 7: MnasNet (Baseline Architecture)

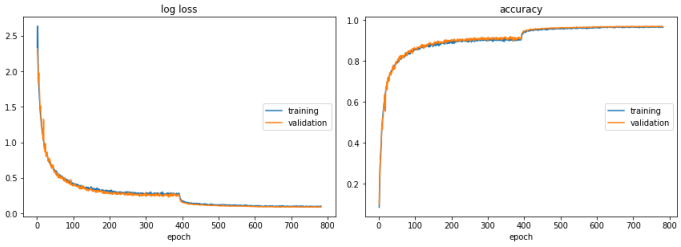

Figure 8: A-MnasNet (Proposed Architecture)

Figure 7 and figure 8 demonstrate the training curves. It is evident that by implementing new algorithms resulted in reduced overfitting and better accuracy.

For resource constrained hardware, desired model size and accuracy can be obtained by tuning width multiplier. Table 4 demonstrates the idea of having multiple size and accuracy by scaling width multiplier.

Table 4: varying model size by width multiplier

Scaling A-MnasNet with width multiplier		
Width Multiplier	Model Accuracy	Model size (in MB)
1.4	97.16%	22
1.0	96.89%	11.6
0.75	96.64%	6.8
0.5	95.74%	3.3
0.35	93.36%	1.8

Figure 9 shows the console output for Image Classification on NXP Bluebox 2.0. The model predicts that the object in the input image is a car. The window in the left corner shows the console output in Tera Term.

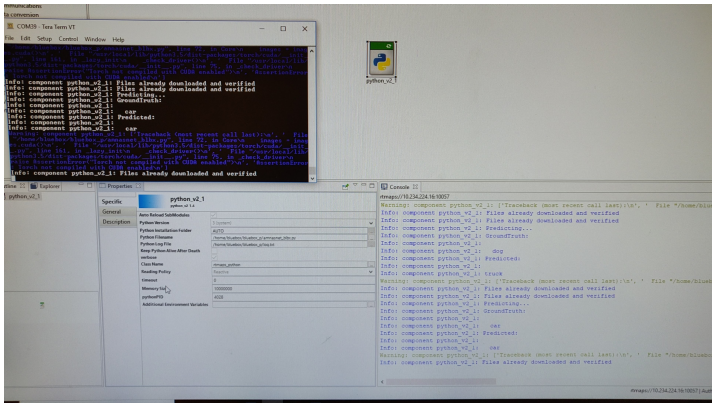

Figure 9: Image Classification using A-MnasNet

7 Conclusion

Design Space Exploration of MnasNet [4] architecture resulted in enhanced accuracy and a decrease in model size. The proposed architecture, A-MnasNet [2], has an accuracy of 96.89% and a size of 11.6 MB. It surpasses MnasNet [4] in terms of model size and accuracy. New convolutional layers were added to make A-MnasNet [2] more efficient in extracting features. AutoAugment[9] was used to preprocess the dataset. The model was scaled with width multiplier to get various trade-offs between model size and accuracy. The architectures were trained and tested on Cifar-10 [6] dataset. Furthermore, they were implemented on NXP Bluebox 2.0 for Image Classification. Results show that the new architecture successfully detects the object and successfully performs Image Classification.

Conflict of Interest The authors declare no conflict of interest.

References

[1] P. Shah, "DESIGN SPACE EXPLORATION OF CONVOLUTIONAL NEURAL NETWORKS FOR IMAGE CLASSIFICATION," 2020, doi:https://doi.org/10.25394/PGS.13350125.v1.

[2] P. Shah, M. El-Sharkawy, "A-MnasNet: Augmented MnasNet for Computer Vision," in 2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS), volume 2020-Augus, 1044–1047, IEEE, 2020, doi: 10.1109/MWSCAS48704.2020.9184619.

[3] C. Wang, Y. Xi, "Convolutional Neural Network for Image Classification," Johns Hopkins University, 2015.

[4] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, Q. V. Le, "MnasNet: Platform-Aware Neural Architecture Search for Mobile," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2815–2823, IEEE, 2019, doi:10.1109/CVPR.2019.00293.

[5] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," arXiv, 2018.

[6] "CIFAR-10 and CIFAR-100 datasets," doi:https://www.cs.toronto.edu/{\~{}}kriz/cifar.html.

[7] D. Li, A. Zhou, A. Yao, "HBONet: Harmonious Bottleneck on Two Orthogonal Dimensions," in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 3315–3324, IEEE, 2019, doi:10.1109/ICCV.2019.00341.

[8] S. Ruder, "An overview of gradient descent optimization algorithms," arXiv, 2016.

[9] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, Q. V. Le, "AutoAugment: Learning Augmentation Policies from Data," arXiv, 2018.