Simplified EfficientNet for Resource Constrained Environment

Harsha Vardhini Vasu
hxv190005@utdallas.edu

Abstract

In this paper, we use Resnet with inverted residual blocks and squeeze-and-excite (SE) blocks to perform Image Classification. The model architecture presented in this paper is a slightly modified version of the standard ImageNet structure. The proposed model without the SE block achieves 71.18% test accuracy, and the model with the SE block achieves 72.96% test accuracy. The computational complexity of the model without SE block was computed in terms of MACs and the number of filters used.

1. Introduction

Nowadays neural Networks are being used in almost all the applications we use. It is preferable to use an application that is both accurate and responds fast, but it is not possible. There is a trade off between how accurate a model can perform and how much time it takes to respond. We can't generalize that accuracy is more important than performance or vise versa. For example, Any application used in the medical field has to be very accurate as the lives of people will depend on it, and few minutes of delay can be acceptable at the cost of improved accuracy. But in the case of real-time applications like the vision in self-driving cars, detecting objects on the street has to be performed in seconds. Accuray can be traded for performance time in this case.

This paper uses resnet similar to ImageNet for image classification. Resnet gives better accuracy than the network architectures that are linear at the cost of increasing execution time and memory. This issue has been handled by introducing SE blocks in the inverted residual block. Section 2 of this paper talks about the state-of-the-art models for the task of image classification, Section 3 describes the structure of the proposed model, Section 4 shows the achieved results and Section 5 shows the computational complexity of the blocks used in the network.

2. Related Work

The network in this paper was heavily inspired by SENet [1], MobileNetV2 [2], MnasNet [3], CondConv [4], MobileNetV3 [5] and EfficientNet [6].

Squeeze-and-Excitation Networks [1] focuses on relationships between channels, unlike most of the CNN models. It squeezes the information into (1 x 1 x C) tensor, thus losing spatial information, and then it does the excitation operation to fully capture channel-wise dependencies. Adding this block shows significant improvement in accuracy in existing state-of-the-art models at the cost of increasing the number of computations.

MobileNetV2 [2] used inverted residuals with a bottleneck to reduce the number of operations and memory required to achieve the state-of-the-art model's accuracy, making this model more suitable for mobiles and other resource-constrained environments.

MnasNet [3] proposed an automated mobile neural architecture search (MNAS) approach as it is challenging to perfectly balance the trade-off between the accuracy and latency given the wide range of network architectures that have to be considered. The proposed approach uses squeeze and excite operations in the inverted residual blocks similar to MobilenetV2 and considers accuracy and model latency as parameters to find the best CNN model for the problem.

MobileNetV3 [5] used a combination of layers from MobileNetV2 and MnasNet as building blocks and included modified swish nonlinearities and SE blocks to improve the model. However, they replaced the sigmoid in both these operations with hard-sigmoid, as sigmoid is inefficient to compute and hard to maintain accuracy.

EfficientNet [6] proposed a compound scaling method that can scale up a ConvNet to improve both accuracy and efficiency without a trade-off. The network is scaled by balancing all three dimensions of a network (width, depth, input resolution) using a set of fixed scaling coefficients instead of arbitrary scales.

In Designing Network Design Spaces [7], authors propose a new method for network architecture search, which has the advantages of both manual design and NAS. Instead of focusing on designing a single network, they focused on finding a design space (parameterized set of model architectures). This approach is five times faster than EfficientNet models.

3. Design

The structure of the model proposed in this paper is very similar to the standard ImageNet model. Figure 1 shows the abstract skeleton of the ImageNet model. In the proposed model, stride by 1 operation is used instead of the stride by 2 operation in the first 2 residual blocks shown in Figure 1, as the image size used in this experiment (3 x 64 x 64) has only 1/4 times the number of rows and columns of the ImageNet images (3 x 256 x 256).

The network's encoder portion has a standard 3x3 convolutions (Tail) followed by five residual blocks (Body). The decoder portion of the network is an average pooling layer followed by a linear layer.

Figure 2 shows the three structures tested for the residual block. Figure 2a is a standard inverted residual block with a bottleneck, Figure 2b is a standard inverted residual block with a bottleneck and a squeeze and excite (SE) Block.

The proposed architecture differs from EfficientNet in a few ways. This design uses only 3x3 filters, whereas EfficientNet uses both 3x3 and 5x5 filters. Similarly, many parameters like the number of times a layer is repeated, stride in block1 and block2, Input resolution 3x56x56, number of head filters is 16 are all reduced to have a smaller version of the EfficientNet model.

For implementations based on the SE enhanced building block, the internal rank reduction ratio R = 4.

For implementations based on SE and conditional convolution enhanced building blocks, the number of experts M = 4.

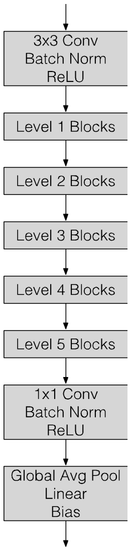

Figure 1: Network structure; the linear layer output dimension and bias dimension is the number of classes

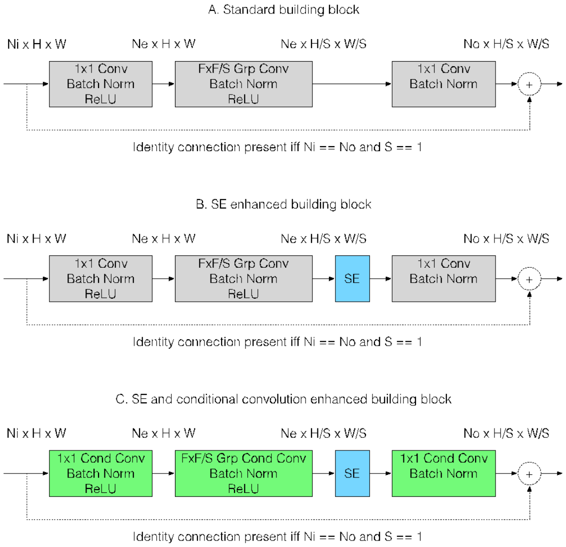

Figure 2: [A] Standard building block, [B] SE enhanced building block and [C] SE and conditional convolution enhanced building block

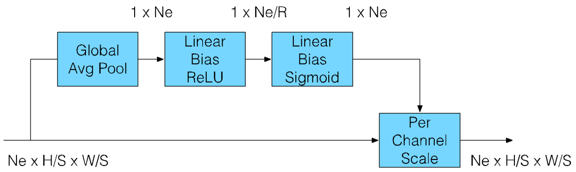

Figure 3: Squeeze and excite uses a learned input dependent per channel weighting to re weight input feature maps with internal rank reduction R

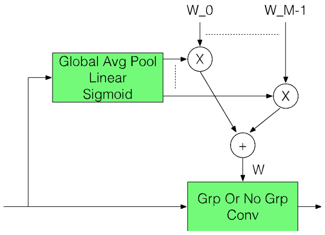

Figure 4: Conditional convolution uses 3D / 4D convolution weight tensors W_m from M experts combined to a single 3D / 4D weight tensor W via a learned input dependent weighted sum with fully grouped convolution / not fully grouped convolution

Re-peat	Input NixWxH	Operation
1	3x56x56	Conv (3x3/1), Batch Norm, ReLU
1	16x56x56	Block (Ne=4Ni, F=3, S=1, ID=True)
1	16x56x56	Block (Ne=4Ni, F=3, S=1, ID=False)
1	24x56x56	Block (Ne=4Ni, F=3, S=1, ID=True)
1	24x56x56	Block (Ne=4Ni, F=3, S=2, ID=False)
2	40x28x28	Block (Ne=4Ni, F=3, S=1, ID=True)
1	40x28x28	Block (Ne=4Ni, F=3, S=2, ID=False)
3	80x14x14	Block (Ne=4Ni, F=3, S=1, ID=True)
1	80x14x14	Block (Ne=4Ni, F=3, S=2, ID=False)
4	160x7x7	Block (Ne=4Ni, F=3, S=1, ID=True)
1	160x7x7	Block (Ne=4Ni, F=3, S=1, ID=False)
1	320x7x7	Conv (1x1/1), Batch Norm, ReLU
1	1280x7x7	Global Avg Pool, Linear, Bias
1	1x100	Output

Table 1: Network specification; block is either a [A] standard building block, [B] SE enhanced building block or [C] conditional convolution enhanced building block

4. Training

Table 2 includes a summary of all training hyper parameters. Note that this is a ~ generic ImageNet training routine such as you would find in RegNetX/Y [7]. Training routines that use more complex data augmentation, additional data, different train and test resolutions, more epochs, … can achieve higher accuracies.

Table 3 includes final training results and figure 5 shows a plot of the per epoch accuracy and loss curves.

Parameter	Value
Batch Size	256
Number of Epochs	55 (5 + 50)
Learning rate - initial	0.01
Learning rate - final	0.001
Data Resize dimension	64
Data Crop dimension	56

Table 2: Training hyper parameters

Block	Training time	Accuracy
Standard	10hr 2min	71.18%
SE enhanced	5hrs 50min	72.96%

Table 3: Training final results

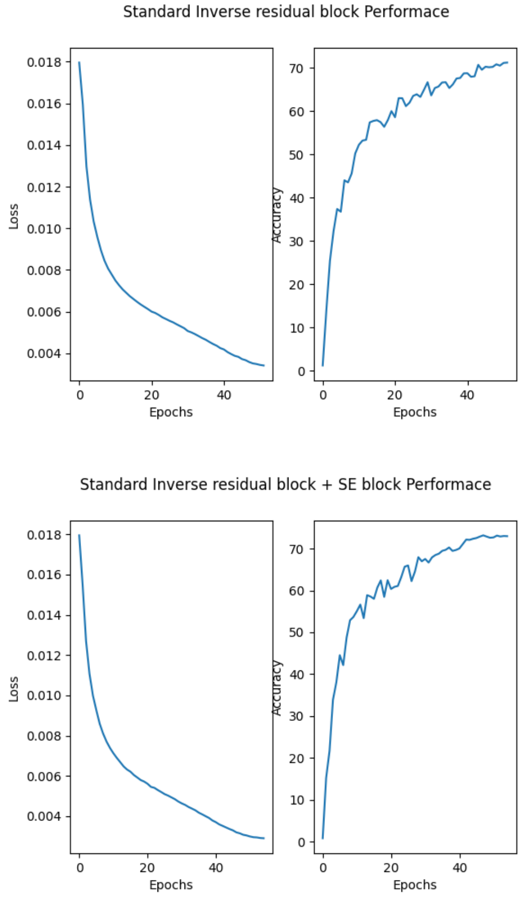

Figure 5: Training per epoch accuracy and loss curves

5. Implementation

Table 4 shows per operation MACs and number of filter coefficients for the stem convolution, convolutions in all standard blocks (taking into account repeats) and the head convolution and matrix multiplication, along with their sum for the full network.

Operation	Rep	MAC	Filter Cx
Conv (3x3/1), Batch Norm, ReLU	1	1354752	16
Block (Ne=4Ni, F=3, S=1, ID=True)	1	122028032	144

Block (Ne=4Ni, F=3, S=1, ID=False)	1	123633664	152
Block (Ne=4Ni, F=3, S=1, ID=True)	1	274563072	216
Block (Ne=4Ni, F=3, S=2, ID=False)	1	75264000	232
Block (Ne=4Ni, F=3, S=1, ID=True)	2	381337600	720
Block (Ne=4Ni, F=3, S=2, ID=False)	1	52684800	400
Block (Ne=4Ni, F=3, S=1, ID=True)	3	572006400	2160
Block (Ne=4Ni, F=3, S=2, ID=False)	1	52684800	800
Block (Ne=4Ni, F=3, S=1, ID=True)	4	762675200	5760
Block (Ne=4Ni, F=3, S=1, ID=False)	1	195686400	1600
Conv (1x1/1), Batch Norm, ReLU	1	20070400	1280
Global Avg Pool, Linear, Bias	1	6272000	100
Total	–	2640261120	13580

Table 4: Per operation and total MAC and filter coefficient counts for all trainable operations

6. Conclusion

In this paper we used a simplified version of EfficientNet for image classification tasks. We introduced SE block in the inverted residual block to trade-off improvement in accuracy at the cost of slight increase in computational cost, and we were able to improve the accuracy from 71.18% to 72.96%.

The proposed model can be improved by using filters with larger filter size to increase receptive field size, by increasing the reduction factor in SE block to learn dependencies between the channels, by increasing the number of reparations of the blocks. Can try using Conditional convolution along with SE in the inverted residual block.

References

[1] J. Hu et. al., "Squeeze-and-excitation networks," arXiv:1709.01507, 2017.

[2] M. Sandler et. al., "MobileNetV2: inverted residuals and linear bottlenecks," axXiv:1801.04381, 2018.

[3] M. Tan et. al., "MnasNet: platform-aware neural architecture search for mobile," arXiv:1807.11626, 2018.

[4] B. Yang et. al., "CondConv: conditionally parameterized convolutions for efficient inference," arXiv:1904.04971, 2019.

[5] A. Howard et. al., "Searching for MobileNetV3," arXiv:1905.02244, 2019.

[6] M. Tan and Q. Le, "EfficientNet: rethinking model scaling for convolutional neural networks," arXiv:1905.11946, 2019.

[7] I. Radosavovic et. al., "Designing network design spaces," arXiv:2003.13678, 2020.