

Bias Loss for Mobile Neural Networks

Lusine Abrahamyan^{1,3}

lusine.abrahamyan@vub.be

Valentin Ziatchin²

valentin.ziatchin@picsart.com

Yiming Chen^{1,3}

cyiming@etrovub.be

Nikos Deligiannis^{1,3}

ndeligia@etrovub.be

¹ETRO Department, Vrije Universiteit Brussel (VUB), Pleinlaan 2, B-1050 Brussels, Belgium

²PicsArt Inc., San Francisco, USA

³imec, Kapeldreef 75, B-3001 Leuven, Belgium

Abstract

Compact convolutional neural networks (CNNs) have witnessed exceptional improvements in performance in recent years. However, they still fail to provide the same predictive power as CNNs with a large number of parameters. The diverse and even abundant features captured by the layers is an important characteristic of these successful CNNs. However, differences in this characteristic between large CNNs and their compact counterparts have rarely been investigated. In compact CNNs, due to the limited number of parameters, abundant features are unlikely to be obtained, and feature diversity becomes an essential characteristic. Diverse features present in the activation maps derived from a data point during model inference may indicate the presence of a set of unique descriptors necessary to distinguish between objects of different classes. In contrast, data points with low feature diversity may not provide a sufficient amount of unique descriptors to make a valid prediction; we refer to them as random predictions. Random predictions can negatively impact the optimization process and harm the final performance. This paper proposes addressing the problem raised by random predictions by reshaping the standard cross-entropy to make it biased toward data points with a limited number of unique descriptive features. Our novel Bias Loss focuses the training on a set of valuable data points and prevents the vast number of samples with poor learning features from misleading the optimization process. Furthermore, to show the importance of diversity, we present a family of SkipblockNet models whose architectures are brought to boost the number of unique descriptors in the last layers. Experiments conducted on benchmark datasets demonstrate the superiority of the proposed loss function over the cross-entropy loss. Moreover, our SkipblockNet-M can achieve 1% higher classification accuracy than MobileNetV3 Large with similar computational

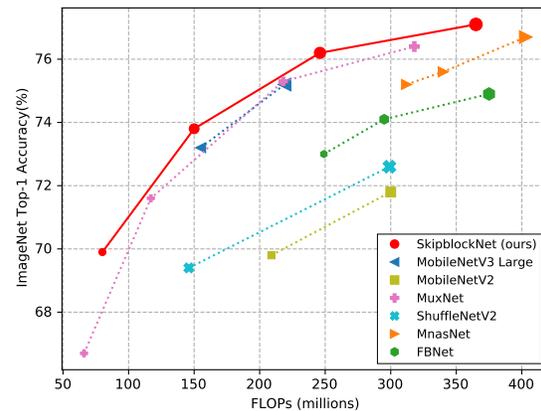


Figure 1. Accuracy v.s. FLOPs on ImageNet. Our SkipblockNet model trained with the proposed bias loss outperforms previous well-performing compact neural networks trained with the cross-entropy loss.

cost on the ImageNet ILSVRC-2012 classification dataset. The code is available on the link - https://github.com/lusinlu/biasloss_skipblocknet.

1. Introduction

Deep CNNs have shown superior performance on numerous computer vision tasks, such as classification, semantic segmentation, and object detection. Typically, models with high predictive power contain a large number of parameters and require a substantial amount of floating point operations (FLOPs); for example, Inception-v3 [45] has approximately 24M parameters and requires 6GFLOPs to process an image with a spatial size of 299×299 pixels. With the advent of AI applications in mobile devices, sev-

eral studies have focused on developing high-performance CNNs for resource-constrained settings. Several studies have focused on compressing existing high-performance pretrained models. The compression of the models can be achieved by performing quantization [51, 25, 56, 38, 29, 55], pruning [14, 13, 10, 16], or knowledge distillation [17, 4]. Typically, the downside of these methods is an inevitable degradation of performance.

Another research line has focused on designing compact neural networks and architectural units [34, 12, 49, 53, 6, 48]. For example, Xception [6] introduced a cost-efficient replacement for the conventional convolution, namely, depthwise separable convolution. ShuffleNet [53] replaced convolutional layers with a combination of pointwise group convolution with channel shuffle operation. The authors of EfficientNet [48] proposed a scaling method that uniformly scales a model’s width, depth, and resolution using a set of fixed scaling coefficients. However, a significant performance improvement in these methods is mostly connected with an increase in the number of parameters [49, 12]. The solution to this problem can be the design of a task-specific objective function. The advantage of designing an objective function over the creation of a new architecture is that the former approach can improve the accuracy of a model without increasing the number of parameters. In general, the preferred loss function for classification is the cross-entropy; however, there exist studies indicating that other objectives can outperform the standard cross-entropy loss [46, 52, 32]. The authors of [46] proposed to compute cross-entropy with the weighted mixture of targets from the uniform distribution. In scenarios where the class imbalance problem exists, [32] proposed to down-weight the loss assigned to well-classified examples. In [39], the authors proposed a meta-learning reweighting algorithm in order to tackle the problem of label noise in the dataset. Although these objectives achieve great performance boost, they target specific problems related mostly to the dataset and do not consider differences between the optimization of compact neural networks and their large counterparts. Diverse and even abundant information in the feature maps of high-performance CNNs often guarantees a comprehensive understanding of the input data. In compact CNN, due to the small numbers of parameters, the amount of extracted features will be smaller, and may not be sufficient to describe the object to be classified. For certain data points, these features may lack unique descriptors required to distinguish between the objects of different classes. As a result, in the absence of a sufficient amount of unique descriptors, the model cannot produce a valid prediction. We refer to these as random predictions that contribute no useful learning signal to the optimization process.

To address this problem, we design *Bias Loss*, a new loss that weights each data point’s contribution in proportion to

the diversity of features it provides. As a simple measure of diversity, we take the signal’s variance, which describes how far the feature maps’ values are spread from the average. Based on the variance, we design a nonlinear function, whose values serve as weights for the cross-entropy. This way, we let data points with diverse features have a higher impact on the optimization process and reduce a mislead caused by random predictions.

To further realize bias loss’s full potential, we propose the SkipblockNet architecture to address the problem of a lack of extracted features in the last layer. Specifically, we design lightweight intermediate blocks to straightforwardly transfer the low-level features from the first layers to the lasts using skip connections. The usage of the proposed blocks will increase the number of data points with a large number of unique descriptors. Experimental results showed that the proposed *Bias Loss* is able to boost the performance of the existing mobile models, such as MobileNetV3 Large [18] (+0.5%), ShuffleNetV2 0.5× [35] (+0.6%), SqueezeNet [23] (+1%). Moreover, SkipblockNet can surpass state-of-the-art compact neural networks such as MobileNetV3, on numerous tasks with fast inference on mobile devices.

To summarize, our contributions are three-fold: (1) we design a loss function to reduce the mislead in the optimization caused by random predictions in compact CNNs; (2) we propose an efficient neural architecture to increase the number of data points with a large number of unique descriptive features; (3) our model achieves state-of-the-art performance on the ImageNet classification task under resource constrained settings.

2. Related Work

Many strategies have been proposed for designing compact, computationally efficient, and high-performance CNNs. Bellow, we present two major categories of solutions: the design of mobile architectures and task-oriented objective functions.

2.1. Mobile Architectures

Several CNN architectures have been developed for resource constraint settings [20, 19, 35, 23, 12, 34]. Among them, the MobileNet [19, 42, 18] and ShuffleNet [53, 35] families stand out due to their high performance achieved with fewer FLOPs. MobileNetV2 [42] introduced inverted residual blocks to improve the performance over MobileNetV1 [19]. Furthermore, MobileNetV3 [18] utilized NAS (Neural Architecture Search) technology [47, 50, 37] resulting in achieving higher performance with fewer FLOPs. ShuffleNet [53] introduced the channel shuffle operation to boost the flow of the information within channel groups. ShuffleNetV2 [35] further improved the actual speed on hardware. Despite the high performance

achieved with very few FLOPs, the importance of maintaining unique descriptive features in the last layers of the network has never been well exploited. To that end, we propose SkipblockNet, an architecture that is designed to increase the number of unique descriptive features in the last layers and reduce the number of random predictions. SkipblockNet shares many similarities with the previous high-performance CNNs, in particular, the inverted residual blocks used in MobileNetV3 [18] and the concept of skip connections utilized in U-Net [40]. We emphasize that our simple modifications achieve superior results not due to innovation in design but due to the combination of the network with our novel loss. In this way, we can benefit from the developed loss the most.

2.2. Objective Functions

The most common choice for the objective function in many tasks is the cross-entropy. However, various studies have indicated that the design of the loss function, aimed to tackle a specific problem, can have significant benefits [32, 39, 46, 57, 26, 22]. Lin *et al.* [32] proposed to reshape the standard cross-entropy to address the problem of foreground-background class imbalance encountered during the training of an object detector. The mechanism of label smoothing [46] suggests using "soft" targets in the cross-entropy calculation. These "soft" targets are a weighted mixture of original targets with the uniform distribution over labels. This technique helps preventing the network from becoming over-confident in numerous tasks like image classification, language translation, and speech recognition. Various studies have attempted to address the obstacle caused by noisy labels [39, 54]. In [39] the authors introduce a variation of the weighted cross-entropy, where weights are being learned by the multi-layer perceptron. The focus of these works has primarily been to optimize the performance of models with a large number of parameters. On the contrary, our loss is designed to tackle the problem arising because of the lack of parameters in compact models, namely the problem of possible mislead in an optimization process caused by random predictions.

3. Bias Loss

We design the *Bias Loss* to address the resource-constrained classification scenario in which there can be a mislead during the optimization process of deep convolutional neural networks [30, 43, 15] caused by random predictions. We advocate that, in compact neural networks, data points failing to provide a sufficient amount of unique features that can describe the object force the model to produce random predictions, that is, predictions made in the absence of feature diversity.

As a simple metric of diversity in all of our experiments, we adopt the signal variance, which can indicate how far the

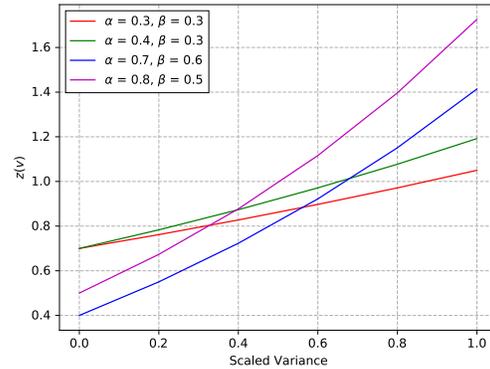


Figure 2. The proposed nonlinear function $z(v)$ given in (6), where v is the scaled variance. The function comprises two hyperparameters α , and β . An increase of β reduces the impact of low variance data points on the cumulative loss. α controls the influence of the high variance data points.

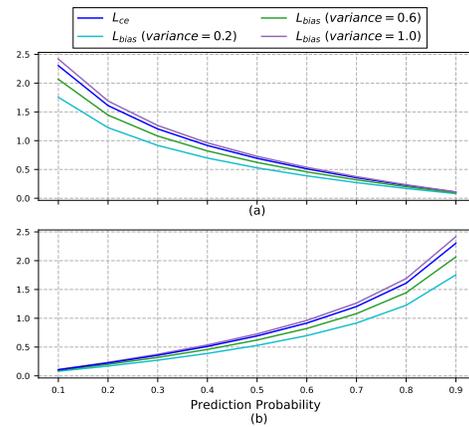


Figure 3. Loss v.s. prediction probability (the output of the softmax) for (a) correct predictions and (b) incorrect predictions. L_{ce} denotes the cross-entropy and the α, β hyperparameters in the L_{bias} loss are equal to 0.3.

feature maps' values are spread out from the average. The intuition behind this choice is that the higher the variance, the higher the chances of obtaining a large number of unique features. For the variance calculations, the feature maps of the last convolutional layer (before the pooling and dropout operations) are used. This helps avoiding distortions in the results and estimating better the learning signal that a data point provides. Let $T \in \mathbb{R}^{b \times c \times h \times w}$ be the output of the convolutional layer, where b is a batch size, c is a number of input channels, and h and w are the height and width of the tensor. Prior to the variance calculations, T is unfolded into a two-dimensional array $t \in \mathbb{R}^{b \times n}$, where $n = c \times h \times w$. The variance of the feature maps of the i th data point in the

batch is equal to

$$v_i = \frac{\sum_{j=1}^n (t_j - \mu)^2}{n - 1}, \quad (1)$$

where

$$\mu = \frac{\sum_{j=1}^n t_j}{n}. \quad (2)$$

In addition, the variance is scaled to the range $[0, 1]$ for further use in the loss function, that is,

$$v_i = \frac{(v_i - \min)}{(\max - \min)}, \quad (3)$$

where, at each iteration, \max and \min is the maximum and minimum values of the activations in the batch of feature maps. This is performed to ensure that outliers in the variance values will not lead to large changes in the loss and will not make the model unstable.

Furthermore, we propose to inject this knowledge about the absence of the unique descriptive features into the optimization process, and to this end, we present the new loss function, namely the *Bias Loss*. The bias loss is a dynamically scaled cross-entropy loss, where the scale decays as the variance of data point decreases.

Let $X \in \mathbb{R}^{c \times h \times w}$ be the feature space, where c is a number of input channels and h, w are the height and width of the input data, and $Y = \{1, \dots, k\}$ be the label space, where k is the number of classes. In a standard scenario, we are given a dataset $D = (x_i, y_i)_{i=1}^N$, where each $(x_i, y_i) \in X \times Y$, and a neural network $f(x; \theta)$, where θ denotes the model parameters. Conventionally, the training aims at learning a model by minimizing the expected loss for the training set. In general, the cross-entropy loss for a classification problem is

$$L_{ce} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^k y_{ij} \log f_j(x_i; \theta), \quad (4)$$

where we consider that the output layer of the neural network is a softmax. In order to calibrate the contribution of each data point into the cumulative loss, we propose to add a nonlinear scaling function, which aims at creating a bias between the data points with low and high variance. The bias loss is defined as

$$L_{bias} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^k z(v_i) y_{ij} \log f_j(x_i; \theta), \quad (5)$$

$$z(v_i) = \exp(v_i * \alpha) - \beta, \quad (6)$$

where α and β are tunable contribution parameters and v is the scaled variance of the output of the convolutional layer. The bias function is visualized for several values of α and β

in Figure 3. We notice two properties of the bias function: (i) when the variance is low, the function values reach their minimum, $(1 - \beta)$, and the impact of these data points is down-weighted. As the variance increases, the $z(v)$'s values, together with the influence of the data point, exponentially increase. (ii) The parameter α smoothly adjusts the rate of the impact of high variance examples. With the increase of α , the impact of high variance data points also increases. In addition, Figure 2 presents the values of the bias loss depending on the variance and the prediction score. The loss is down-weighted mainly for low confidence and low variance data points for both correct and incorrect predictions. Furthermore, it is up-weighted for the high confidence and high variance incorrect predictions, as learning from this kind of data points with a large number of unique features can have a positive impact on the optimization process. Our empirical results suggest that selecting $\alpha = 0.3, \beta = 0.3$ leads to the best performance.

Intuitively, the proposed function helps focusing the learning on examples that can provide a large number of unique features and reducing the possible mislead in the optimization process caused by random predictions.

4. SkipblockNet Mobile Architectures

We also introduce a new computational block and a new CNN architecture to further increase the gain in performance obtained via the bias loss. The presented block can be easily integrated into existing architectures and boost the information flow toward the last layers, without additional effort.

4.1. Skip Block

The idea of the skip block is to deliver the low-level features directly from the first layers to the last ones. The block's design is motivated by the U-Net [40] architecture, where, in an autoencoder style architecture, the outputs of layers with the same spatial dimensions in the encoder and decoder are connected via skip connections. Generally, in classification networks, the layers' spatial sizes gradually decrease, making it impossible to use skip connections straightforwardly. To address this limitation, we propose an intermediate block, which is brought to connect layers with different spatial sizes and enrich the last layers with the low-level features extracted from the first layers. As shown in Figure 4, the skip block consists of a pooling operation combined with convolutions. First, in order to keep the key features and reduce the spatial sizes, we apply an adaptive average pooling, followed by three convolutional layers. Batch normalization (BN) [24] and ReLU nonlinearity [1] are applied after each convolutional layer except for the last one where ReLU is not used. The choice of the adaptive average pooling is motivated by the fact that it takes all features into account, making it possible for the

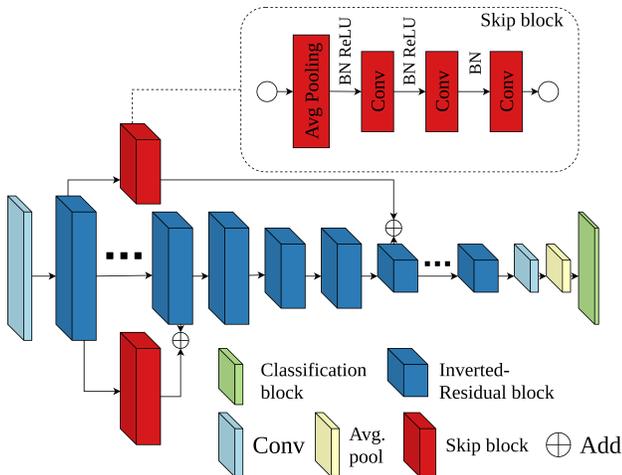


Figure 4. Overview of the SkipblockNet architecture. On top of the inverted residuals [18], SkipblockNet uses skip blocks to transfer high-level features of the first block to the last layers. The network design is deliberately simple, which allows concentrating on the contribution of the novel bias loss, which boosts the performance by focusing the training on a set of data points with a rich learning signal.

skip block to process all input values. Concerning the convolutional layers’ parameters, the setup proposed in MobileNetV3 for the inverted residual blocks is used.

4.2. SkipblockNet

Since our primary goal is to boost the number of unique descriptive features in compact neural networks, while mitigating computational complexity, we propose a SkipblockNet architecture that deploys skip blocks. Due to its superior performance as a design baseline, we follow the architecture of MobileNetV3 [18]. SkipblockNet (Figure 4) consists of the stack of inverted residual and the classification blocks of MobileNetV3, and includes our novel skip block. The first layer is a convolution with 16 filters followed by 15 inverted residual blocks. Two skip blocks are inserted after the first inverted residual block (Figure 4) with the purpose of transferring the information to the sixth and tenth inverted residual blocks. After the skip and inverted residual blocks, a convolutional layer and global average pooling are applied before the final classification block, which consist of dropout and fully connected layers. Similar to MobileNetV3, we use hard-swish nonlinear functions due to their efficiency. As can be seen in Table 2, the latency of SkipblockNet on mobile devices is on par with that of MobileNetV3. Although the described architecture can already guarantee high performance and low latency, there can be situations where a faster model or higher accuracy may be required. In order to provide a fully customizable network, we integrate the width multiplier, presented in the

inverted residual block, into the skip block so as to control the number of the channels in each layer. By manipulating the width multiplier, the width of the entire network can be changed. That will lead to changes in the model size and computational cost, as well as changes in performance. In general, increase of the multiplier will lead to increase in the performance and latency, and vice versa. The presented architectures give a basic design for reference, and for further improvement, AutoML methods [2, 9, 27] can be used to tune the skip blocks and boost the performance.

5. Experiments

We present empirical results to demonstrate the efficiency of the novel bias loss and the proposed family of SkipblockNet models. We report results on three tasks: image classification, object detection, and transfer learning. All experiments were performed on a single machine with 2 GeForce RTX 2080 Ti GPUs. Further, during trainings, activation maps with outliers produce very high variances. In turn, these high variances will lead to high values of the bias function and make the training unstable. To avoid this effect, in all experiments, we clamp the output of the bias function to the range of [0.5, 1.5].

5.1. ImageNet Classification

We set experiments on ImageNet [41] and compare the achieved accuracies versus various measures of resource usage such as FLOPs and latency.

Training Setup: ImageNet is a large-scale image classification dataset with over 1.2M training and 50K validation images belonging to 1000 classes. For experiments on ImageNet, we follow most of the settings used in EfficientNet [48]: the RMSProp optimizer with a decay of 0.9 and a momentum of 0.9; a batch norm momentum of 0.99; a weight decay of $1e - 5$; and an initial learning rate of $1e - 6$ that increased to 0.032 in the initial 3 epochs [11] and then decays by 0.97 every 2.4 epochs. Furthermore, we adopt Inception preprocessing with an image size of 224×224 pixels [44], a batch size of 512, and complement the training with an exponential moving average with a decay rate of 0.99995. The reported results are single-crop performance evaluations on the ImageNet validation set. The aforementioned setting is adopted in order to perform a fair comparison, as most of the state-of-the-art architectures [49, 12, 34, 18] that we are comparing with are using the same setup.

Results: Table 1 shows the performance of the SkipblockNet family models in relation to several modern resource-constraint network architectures. The networks are grouped into four levels of computational complexity: 50 – 100, 100 – 200, 200 – 300, and 300 – 400 million FLOPs. We compare them in terms of accuracy, number of parameters and computational complex-

Table 1. Comparison of state-of-the-art resource constraint neural networks over accuracy, FLOPs, and number of parameters. The results are grouped into sections by FLOPs for better visualization.

Model	FLOPs	Parameters	Top-1 Acc. (%)	Top-5 Acc. (%)
MobileNetV2 0.5× [42]	97M	2.0M	65.4	86.4
MUXNet-xs [34]	66M	1.8M	66.7	86.8
MobileNetV3 Small 1.0× [18]	66M	2.9M	67.4	-
SkipblockNet-XS (with bias loss)	81M	2.3M	69.9	88.9
ShuffleNetV2 1.0×	146M	2.3M	69.4	88.9
MUXNet-s [34]	117M	2.4M	71.6	90.3
ChamNet-C [7]	212M	3.4M	71.6	-
MobileNetV3 large 0.75×	155M	4.0M	73.3	-
SkipblockNet-S (with bias loss)	152M	3.6M	73.8	91.4
FBNet-A [50]	249M	4.3M	73.0	-
MobileNetV3 Large 1.0×	219M	5.4M	75.2	-
MUXNet-m	218M	3.4M	75.3	92.5
GhostNet 1.3×	226M	7.3M	75.7	92.7
MixNet-S [49]	256M	4.1M	75.8	92.8
SkipblockNet-M (with bias loss)	246M	5.5M	76.2	92.8
ProxylessNAS [5]	320M	4.1	74.6	92.2
MnasNet-A2 [47]	340M	4.8M	75.6	92.7
EfficientNet-B0 [48]	390M	5.3	76.3	93.2
MUXNet-l	318M	4.0M	76.6	93.2
MobileNetV3 large 1.25×	356M	7.5M	76.6	-
SkipblockNet-L (with bias loss)	364M	7.1M	77.1	93.4

Table 2. Top-1 accuracy v.s. latency on Google Pixel family phones (Pixel-*n* denotes a Google Pixel-*n* phone). All latencies are in ms and are measured using a single core with a batch size of one. The top-1 accuracy is calculated on ImageNet.

Model	Top-1 (%)	Pixel 4	Pixel 3
SkipblockNet-M	76.2	27	42
GhostNet 1.3×	75.7	27	41
MnasNet-A2	75.6	21	39
MobileNetV3 1.0×	75.2	26	38
MobileNetV2 1.0×	71.8	27	38

ity (FLOPs). Overall, our family of SkipblockNet models (SkipblockNet-XS, SkipblockNet-S, SkipblockNet-M, SkipblockNet-L) trained with the bias loss outperforms other competitors at the different computational complexity levels. Specifically, SkipblockNet-M achieves 76.2% accuracy with 246 MFLOPs, which is higher by 1% compared with the MobileNetV3 Large [18] and by 0.4% compared with MixNet-S [49]. Figure 1 and Figure 5 visualize the trade-off obtained by SkipblockNet and previous compact neural networks.

Inference Speed: We measure the inference speed of the SkipblockNet-M on Google Pixel phones using the PyTorch V1.6 Mobile framework [36]. We use a single core in all our measurements. Table 2 reports the latencies of the SkipblockNet along with the other state-of-the-art compact neural networks. The results suggest that Skipblock-

Table 3. Ablation study for different techniques. The baseline is MobileNetV3 1.0× and the combination of the baseline with the skip blocks is SkipblockNet-M.

Top-1 (%)	baseline	skip block	bias loss
75.2	✓		
75.7	✓		✓
75.5	✓	✓	
76.2	✓	✓	✓

Net can achieve 1% higher accuracy than the MobileNetV3 with computational overhead higher only by 1ms on Google Pixel 4.

Impact of Different Components on the Performance: To investigate the importance of the different techniques used in SkipblockNet, we conduct a series of experiments on the ImageNet dataset, shown in Table 3. We first consider the MobileNetV3, a baseline architecture for our SkipblockNet, and trained it with the Bias Loss. As shown in Table 3, the bias loss can increase the accuracy of MobileNetV3 by 0.5%, compared with the training with cross-entropy. To evaluate the impact of skip blocks, we examined the performance of the baseline MobileNetV3 with the SkipblockNet-M (which is the MobileNetV3 architecture plus skip blocks), both trained with the cross-entropy. The results indicate that a gain of 0.3% can be obtained only by using the skip blocks. Moreover, by enriching the last

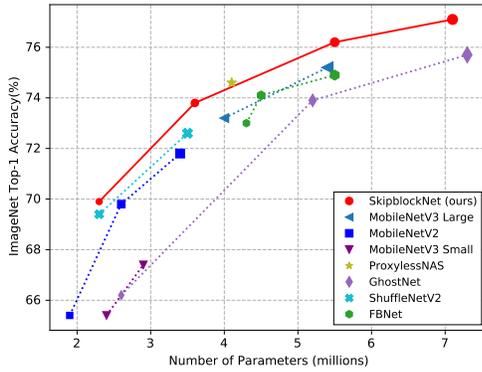


Figure 5. Top-1 classification accuracy v.s. number of parameters on ImageNet for various compact CNNs. Our SkipblockNet is trained with the proposed bias loss.

layers with the low-level information of the first layers, we can increase the number of data points with high variance and make the boost in the performance related to the usage of the bias loss even higher (i.e., an increase of 0.5% in the case of MobileNetV3 and 0.7% for SkipblockNet-M). Moreover, in order to show the skip blocks’ advantage over a simple increase of the depth multiplier, we trained MobileNetV3 1.05× with the 247M FLOPs and 5.9M parameters and compare it with SkipblockNet-M (246M FLOPs, 5.5M parameters). When trained with the cross-entropy, SkipblockNet-M achieves 75.5% accuracy, while MobileNetV3 1.05× achieves 75.3%.

5.2. Classification with Bias Loss

To verify the effectiveness of the proposed bias loss, we apply it on several resource constraint neural networks and conduct experiments using the CIFAR-100 [31] classification dataset. The CIFAR-100 dataset [31] consists of 60,000 images from 100 classes. The dataset is divided into 50,000 training and 10,000 testing images. For training on CIFAR-100, we use an SGD optimizer with a momentum equal to 0.9 and a weight decay of $5e - 4$. The initial learning rate is set to $1e - 1$ and then decays at the epochs 60, 120, 160 at a rate of 0.2. For data augmentation, images are randomly flipped horizontally and rotated between the angles $[-15, 15]$. Table 4 reports the accuracy of neural networks trained with cross-entropy, focal loss [32] and bias loss. The results show that models trained with bias loss systematically outperform models trained with cross-entropy and focal loss by about 1% and 0.5%, respectively. The results indicate that our loss can boost the performance regardless of the architecture. In particular, when compared with the cross-entropy, for ShuffleNetV2 [35] 0.5×, the accuracy is increased by 1.5%, for SqueezeNet [23] by 1%,

Table 4. Comparison of compact CNNs accuracies trained on CIFAR-100 with the bias loss and cross-entropy.

Model	Params	Top-1 (%)	Top-1 (%)	Top-1 (%)
		CE loss	Focal loss	Bias loss
ShuffleNetV2 0.5×	1.4M	69.5	69.8	71
MobileNetV2 0.75×	2.6M	68	68.2	68.6
NASNet-A ($N = 4$)	5.3M	77.2	77.5	78
SqueezeNet	1.25M	69.4	69.8	70.4
DenseNet ($k = 12$)	7M	78.9	79.5	79.9

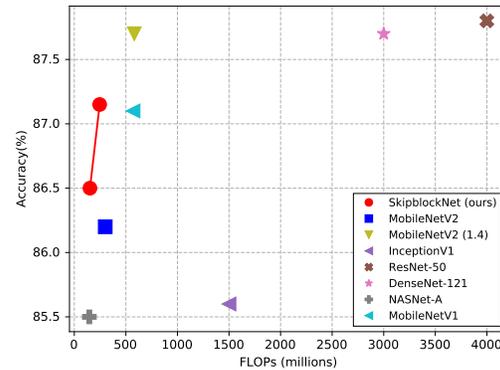


Figure 6. Transfer learning performance. Trade-off between top-1 accuracy and number of FLOPs.

and for MobileNetV2 0.75× [42] by 0.6%.

5.3. Transfer Learning

We have also evaluated our SkipblockNet on the transfer learning task using the Food101 [3] dataset. Food-101 consists of 75, 750 training and 25, 250 testing images from 101 different classes. Figure 6 compares the accuracy against FLOPs for our models and the list of other neural networks. Each SkipblockNet model is first trained from scratch on ImageNet and all weights are fine-tuned on the Food101 dataset using a setup similar to [28]. The accuracy and FLOPs results for the rest of the models are taken from [28]. The results show that our SkipblockNets significantly outperform previous compact neural networks and have accuracy on par with the models with a large number of parameters. Specifically, SkipblockNet-M achieves 0.95% higher accuracy, than MobileNetV2 [42], with 1.2× higher efficiency.

5.4. Object Detection

To evaluate the generalization ability of SkipblockNet, we conduct object detection experiments on the PASCAL VOC detection benchmark [8]. We use the PASCAL VOC 2012 trainval split as training data and report the mean Aver-

Table 5. Average/Max/Min variances of the output of the n -th and last convolutional layers for different models, where BL and CE indicates trainings with the bias loss and cross-entropy, respectively.

Model	5th layer			20th layer			last layer		
	avg.	max	min	avg.	max	min	avg.	max	min
SkipblockNet-M (CE)	1.7	2.4	1.6	0.6	1.2	0.1	0.09	0.2	0.04
SkipblockNet-M (BL)	2.	2.5	1.7	1.	1.6	0.2	0.15	0.2	0.09
ShuffleNet (CE)	1.2	1.6	0.9	0.3	0.5	0.02	0.02	0.07	0.01
ShuffleNet (BL)	1.4	1.7	0.9	0.4	0.7	0.03	0.04	0.1	0.02
MobileNetV3 1.0× (CE)	1.7	2.3	1.6	0.4	1.	0.06	0.05	0.09	0.01
MobileNetV3 1.0× (BL)	1.9	2.4	1.9	0.7	1.5	0.1	0.09	0.1	0.03
Inception V3 (CE)	3.3	5.9	1.9	5.2	9.3	2.4	0.7	3.6	0.2
DenseNet (CE)	3	6.1	1.9	4.1	7.2	1.4	0.7	2.4	0.2

Table 6. The performance for PASCAL VOC2007 Detection.

Model	Parameters	FLOPs	mAP (%)
VGG + SSD	26.2M	31B	77.2
MobileNet + SSD	9.4M	1.6B	67.5
MobileNetV2 + SSD	8.9M	1.4B	73.1
SkipblockNet-S + SSD	9.4M	1.4B	73.6

age Precision (mAP) on the test split. Our experiments use the Single Shot Detector (SSD) [33] as a detection framework and SkipblockNet as the feature extraction backbone. To set up additional layers, we follow the procedure described in MobileNetV2 [42]. We train all the models with the SGD optimizer for 200 epochs, with a batch size of 42, an input image size of $300 \times 300 \times 3$, and an initial learning rate of 0.01 with cosine annealing. Table 6 reports the mAP achieved with the SkipblockNet compared with other models. Under similar resource usage, SkipblockNet-S + SSD achieves 0.5% higher mAP than MobileNetV2 [42].

5.5. Analysis of the Variance in the Neural Networks

To show the role of the variance in the CNNs and the impact that the bias loss and the skip blocks can have on it, we conduct experiments on a range of well-known architectures. We examine the distribution of the values in convolutional layers in the networks with a large number of parameters like Inception V3 [45] and DensNet169 [21] and in compact ones. The purpose of the experiment is to compare the variance in the large and compact models and quantify the boost in the variance that the bias loss and skip blocks can provide. We took models pre-trained on ImageNet and examined the average, maximum and minimum values of the variance within the different layers. The results presented in Table 1 indicate that: (1) the variance in large models is significantly higher than that in compact models. Hence, large models can extract a decent amount of descriptive features for almost all samples, and the proposed strat-

egy of reweighting will not boost their performance. (2) The bias loss can increase the variance throughout the model. (3) Skip blocks enrich later layers of a model with low-level features, thereby increasing the variance (SkipblockNet-M (CE) vs MobileNetV3 1.0×(CE)). The increase in the variance leads to the boost in the number of up-weighted data points meaning that, in the case of training with the bias loss, the optimizer will benefit from learning from more useful data points. Hence, the combination of skip blocks with the bias loss can bring a higher gain in accuracy.

6. Conclusion

In this paper, we proposed the *Bias Loss*, a novel loss function designed to improve the performance of the compact CNNs by reducing a mislead during the optimization process caused by the data points with poor learning features. Furthermore, we presented a family of SkipblockNet models whose architectures are brought to reduce the number of data points with poor learning features. Our extensive experiments, conducted on benchmark datasets and models, illustrate that the proposed loss is able to boost the performance of existing compact CNNs. Moreover, our SkipblockNet-M achieves significantly better accuracy and efficiency than all the latest compact CNNs on the ImageNet classification task.

References

- [1] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018. 4
- [2] James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, pages 2546–2554, 2011. 5
- [3] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014. 7

- [4] Cristian Buciluă, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 535–541, 2006. 2
- [5] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. In *International Conference on Learning Representations*, 2018. 6
- [6] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1251–1258, 2017. 2
- [7] Xiaoliang Dai, Peizhao Zhang, Bichen Wu, Hongxu Yin, Fei Sun, Yanghan Wang, Marat Dukhan, Yunqing Hu, Yiming Wu, Yangqing Jia, et al. Chamnet: Towards efficient network design through platform-aware model adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11398–11407, 2019. 6
- [8] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of Computer Vision*, 88(2):303–338, 2010. 7
- [9] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems*, pages 2962–2970, 2015. 5
- [10] Ariel Gordon, Elad Eban, Ofir Nachum, Bo Chen, Hao Wu, Tien-Ju Yang, and Edward Choi. Morphnet: Fast & simple resource-constrained structure learning of deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1586–1595, 2018. 2
- [11] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large mini-batch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 5
- [12] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1580–1589, 2020. 2, 5
- [13] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *arXiv preprint arXiv:1510.00149*, 2015. 2
- [14] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1135–1143. Curran Associates, Inc., 2015. 2
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 3
- [16] Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. Amc: Automl for model compression and acceleration on mobile devices. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2
- [17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *Stat*, 1050:9, 2015. 2
- [18] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1314–1324, 2019. 2, 3, 5, 6
- [19] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2
- [20] Gao Huang, Shichen Liu, Laurens Van der Maaten, and Kilian Q Weinberger. Condensenet: An efficient densenet using learned group convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2752–2761, 2018. 2
- [21] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017. 8
- [22] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in Statistics*, pages 492–518. Springer, 1992. 3
- [23] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 2, 7
- [24] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015. 4
- [25] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018. 2
- [26] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016. 3
- [27] Kirthevasan Kandasamy, Willie Neiswanger, Jeff Schneider, Barnabas Poczos, and Eric P Xing. Neural architecture search with bayesian optimisation and optimal transport. In *Advances in Neural Information Processing Systems*, pages 2016–2025, 2018. 5
- [28] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2661–2671, 2019. 7
- [29] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018. 2

- [30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. 3
- [31] A. S. Krizhevsky. Learning multiple layers of features from tiny images. *Technical Report*, 2009. 7
- [32] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017. 2, 3, 7
- [33] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016. 8
- [34] Zhichao Lu, Kalyanmoy Deb, and Vishnu Naresh Boddeti. Muxconv: Information multiplexing in convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12044–12053, 2020. 2, 5, 6
- [35] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 116–131, 2018. 2, 7
- [36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 6
- [37] Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. Efficient neural architecture search via parameters sharing. In *International Conference on Machine Learning*, pages 4095–4104, 2018. 2
- [38] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, pages 525–542. Springer, 2016. 2
- [39] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pages 4334–4343, 2018. 2, 3
- [40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015. 3, 4
- [41] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 5
- [42] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. 2, 6, 7, 8
- [43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2015. 3
- [44] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. 2017. 5
- [45] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 1, 8
- [46] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 2, 3
- [47] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2820–2828, 2019. 2, 6
- [48] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114, 2019. 2, 5, 6
- [49] Mingxing Tan and Quoc V Le. Mixconv: Mixed depthwise convolutional kernels. 2019. 2, 5, 6
- [50] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10734–10742, 2019. 2, 6
- [51] Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. Quantized convolutional neural networks for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4820–4828, 2016. 2
- [52] Lingxi Xie, Jingdong Wang, Zhen Wei, Meng Wang, and Qi Tian. Disturblabel: Regularizing cnn on the loss layer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4753–4762, 2016. 2
- [53] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018. 2
- [54] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems*, pages 8778–8788, 2018. 3
- [55] Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen. Incremental network quantization: Towards lossless cnns with low-precision weights. 2017. 2

- [56] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016. [2](#)
- [57] Xinge Zhu, Hui Zhou, Ceyuan Yang, Jianping Shi, and Dahua Lin. Penalizing top performers: Conservative loss for semantic segmentation adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 568–583, 2018. [3](#)