BasisNet: Two-stage Model Synthesis for Efficient Inference Supplementary Material

Mingda Zhang[1] Chun-Te Chu[2] Andrey Zhmoginov[2] Andrew Howard[2] Brendan Jou[2]
Yukun Zhu[2] Li Zhang[2] Rebecca Hwa[1] Adriana Kovashka[1]

[1]Department of Computer Science, University of Pittsburgh [2]Google Research

{mzhang,hwa,kovashka}@cs.pitt.edu {ctchu,azhmogin,howarda,bjou,yukun,zhl}@google.com

A. Detailed model architecture of BasisNet

Here we describe the details about the proposed BasisNet, including the lightweight model and basis models.

A.1. Lightweight model

For BasisNet-MV2, the lightweight model follows the architecture described in Table 2 of [8], and we use multiplier of 0.5 and input image resolution of 128. The lightweight model has a computation overhead of 30.3M MAdds and a model size of 1.2M parameters.

For BasisNet-MV3, we use MobileNetV3-small for our lightweight model as described in Table 2 of [5], and we use multiplier of 1.0 and input resolution of 128 or 224 for different experiments. The model size for the lightweight model is 2.5M parameters regardless of input image resolutions. With 128×128 image, the lightweight model has 19.9M MAdds computation overhead, and with 224×224 image the computation overhead is 56.5M MAdds.

As described in Sec. 3.1, the lightweight model has two tasks, one for initial classification prediction and the other for combination coefficients prediction. The first task is similar with any regular classification task, and can be formally described as:

$$\hat{y} = \text{LM}(f(x); W_{\text{LM}}) \tag{1}$$

Note that the two tasks share all but the final classification head, thus the extra computation for predicting the combination coefficients is negligible.

A.2. Detailed architectures for different experiments

Here we describe the detail about models in different experiments. Unless stated otherwise, we use the following settings as default for BasisNet-MV2 and BasisNet-MV3:

- For MobileNetV2 experiments, the first-stage lightweight model is MobileNetV2 with 0.5x multiplier and input image resolution of 128 (MV2, 0.5x128) and the second stage has 8 basis models of MobileNetV2 1.0x with image resolution of 224 (MV2, 1.0x224). Basis models share parameters in layers from L1 to L10 and final classification layer, and differ in parameters in L11 to L17.
- For MobileNetV3, the lightweight model is MobileNetV3-small with 1.0x multiplier and input image resolution of 128 (MV3-small, 1.0x128). The second stage has 16 basis models of MobileNetV3-large with 1.0x multiplier and resolution of 224 (MV3-large, 1.0x224), and they share parameters in first 7 and last 2 layers, and differ in parameters in L8 to L15.

Comparison with MobileNets (Sec. 4.2) We use BasisNet-MV2 with 8 bases and the lightweight model is MV2 (0.5x128). Each basis model is a MV2 (1.0x224) and they only differ in parameters from L11-17. The basis models dropout rate is $1/8$.

For BasisNet-MV3, we use 16 bases each of a MV3-large (1.0x224), and the lightweight model is MV3-small (1.0x128). All basis models share parameters except for in layers L8-15. The basis model dropout rate is $1/16$.

The effect of regularization for proper training (Sec. 4.3) We use the same model architectures for BasisNet-MV2 and BasisNet-MV3 with Sec. 4.2.

Number of bases in basis models (Sec. 4.4) We use BasisNet-MV3 with different number of basis models, but each is a MV3-large (1.0x224). The lightweight model is MV3-small (1.0x224) and all basis models share parameters except for layers L11-15. For BasisNet with no more than 8 bases we use basis model dropout rate of $1/8$ and for all others (16 to 128 bases) we use a basis model dropout rate of $1/16$.

Comparison with CondConv (Sec. 4.5) For BasisNet-MV3, we use 16 basis models each of a MV3-large (1.0x224), and the lightweight model is MV3-small (1.0x224). All basis models share parameters except for layers L11-15. The basis model dropout rate is $1/16$.

Early stop to reduce average inference cost (Sec. 4.6) We use the same BasisNet-MV3 model as in Sec. 4.5.

B. Implementations and training hyperparameters

Our project is implemented with TensorFlow [1]. Following [8] and [5], we train all models using synchronous training setup on 8x8 TPU Pod, and we use standard RMSProp optimizer with both decay and momentum set to 0.9. The initial learning rate is set to 0.006 and linearly warms up within the first 20 epochs. The learning rate decays every 6 epochs for BasisNet-MV2 (4.5 epochs for BasisNet-MV3) by a factor of 0.99. The total batch size is 16384 (i.e. 128 images per chip). For stabilizing the training, as described in Section 3.3 we keep $\epsilon = 1$ for the first 10K training steps then linearly decays to 0 in the next 40K steps. We also used gradients clipping with clip norm of 0.1 for BasisNet-MV3. In general, all BasisNet and reference baseline models are trained for 400K steps. We set the L2 weight decay to 1e-5, and used the data augmentation policy for ImageNet from AutoAugment [3]. We choose the checkpoint from [11] as our EfficientNet-b2 teacher model for distillation, and for BasisNet-MV3 both lightweight model and all basis models are trained with teacher supervision. For BasisNet-MV2, we only distill the basis models but use the groundtruths labels without label smoothing for training the lightweight model. For basis models dropout, we use dropout rate of $1/8$ for all BasisNets with no more than 8 bases, and use $1/16$ for the rest which has 16 or more bases. Following [5], we also use exponential moving average with decay 0.9999 and set the dropout keep probability to 0.8.

C. Comparison with other efficient networks

In Table 1 of our main paper, we show a comparison table with recent efficient neural networks on ImageNet classification benchmark. For baselines we directly use the statistics from the corresponding original papers, even though the training procedures could be very different. Some common tricks in literature include knowledge distillation♦, training with extra data♥, applying custom data augmentation♠, or using AutoML-based learned training recipes (hyperparameters)♣. Different models may choose subsets of these tricks in their training procedure. For example, [11] use 3.5B weakly labeled images as extra data and use knowledge distillation to iteratively train better student models. CondConv [12] use AutoAugment [3] and mixup [13] as custom data augmentation. [10] reported in a concurrent work that combining AutoAugment and knowledge distillation can have even stronger performance boost, because soft-labels from knowledge distillation helps alleviating label misalignment during aggressive data augmentation. In FBNetV3 [4] the training hyperparameters are treated as components in the search space and are obtained from AutoML-based joint architecture-recipe search. OFA [2] use the largest model as teacher to perform knowledge distillation to improve the smaller models. Notably, in our main paper, unless stated otherwise, we *always reported the statistics from our re-implementations*, thus the comparison in our ablation studies are fair, but some results might be inconsistent with this table. It is also worth mentioning that even though we did not explicitly use extra data for training BasisNet, the teacher model checkpoint that we used for knowledge distillation is from noisy student training [11], thus our model may indirectly benefit from the extra data (thus noted by ♡). However, we also experimented with 1.4x MobileNet-V2 as teacher model (which is not exposed to extra data) to train BasisNet-MV2, and verified that the main conclusion still holds.

D. More quantitative experiments

D.1. Detailed comparison with MobileNets (Sec. 4.2)

In Table 1, we show original data of Fig. 3 of the main paper, so readers can get the exact accuracy numbers more easily. Specifically, we show the model performance with different regularizations at 4 different image resolutions `{128, 160, 192, 224}` in the last four columns. We compare the data augmentation (Preprocess, *regular* represents the Inception preprocess as in [8, 5], and *AA* represents AutoAugment from [3]), distillation with different teachers (MV2 1.4x represents MobileNetV2 with 1.4x multiplier, EfN-b2 represents EfficientNet-b2 model from [11]), and basis model dropout.

We experimented with different teacher network to distill the BasisNet. Note that the MobileNetV2 1.4x teacher we used is from [8] and has accuracy of 74.9%, and our BasisNet achieves even higher accuracy of 75.4% than the teacher. We also experimented different variations of EfficientNet (b0, b2, b4, b7) and find that models trained with EfficientNet-b2 has the best performance, and using even better teacher network does not bring performance gain to the BasisNet. We suspect this is related to the gap between teacher and student network as reported in [6].

D.2. Detailed experiments for number of bases in basis models (Sec. 4.4)

In Table 2, we present the original data for Fig. 4 of the main paper, so readers can get the exact accuracy numbers more easily. Notably, we find that BasisNet-MV3 with

Table 1. Detailed comparison of BasisNet-MV2 with MobileNetV2.

Model	**Preprocess**	**Distillation**	**# Bases (BMD)**	**128**	**160**	**192**	**224**
MobileNetV2	regular	None	N/A	66.6	69.5	71.5	72.9
MobileNetV2	AA	None	N/A	67.8	70.7	72.7	73.7
MobileNetV2	AA	MV2 1.4x	N/A	68.8	71.4	72.4	73.1
MobileNetV2	AA	EfN-b2	N/A	69.8	72.6	73.8	74.9
BasisNet-MV2	regular	None	8 (0)	68.6	71.4	73.3	74.7
BasisNet-MV2	AA	None	8 (0)	70.4	72.8	74.6	75.6
BasisNet-MV2	regular	EfN-b2	8 (0)	71.8	74.8	76.2	77.2
BasisNet-MV2	regular	None	8 (1/8)	69.1	71.9	73.7	75.0
BasisNet-MV2	AA	None	8 (1/8)	70.9	73.2	75.1	75.9
BasisNet-MV2	AA	MV2 1.4x	8 (1/8)	72.3	73.8	74.7	75.4
BasisNet-MV2	AA	EfN-b2	8 (1/8)	73.5	75.9	77.0	78.1

Table 2. Detailed comparison of BasisNets with different number of bases.

Model	#MAdds(M)	#Params(M)	Acc.(%)
MV3 (1.0x224)	217	5.45	77.7
MV3 (1.25x224)	356	8.22	79.7
MV3 (1.5x224)	489	11.3	80.6
MV3 (2.0x128)	276	19.1	79.2
MV3 (2.5x128)	435	29.0	80.4
#Bases=1	273	8.07	77.7
#Bases=2	274	9.19	78.0
#Bases=4	277	11.4	78.8
#Bases=8	281	15.9	79.6
#Bases=16	**290**	**24.9**	**80.3**
#Bases=32	308	42.8	80.5
#Bases=64	344	78.6	80.7
#Bases=128	416	150.3	80.9

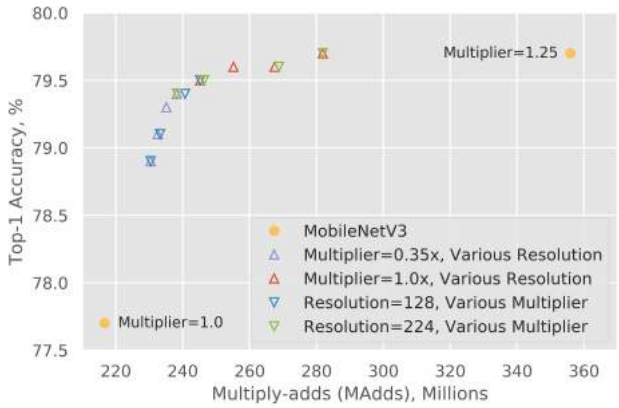

Figure 1. BasisNet-MV3 with lightweight model of different sizes (#MAdds).

16 bases is a good balance between model accuracy and computation budget, achieving 80.3% top-1 accuracy with $290M$ Madds. This table also shows that BasisNet technique optimizes MAdds at the expense of model size.

D.3. Model synthesis with varying sized lightweight model

We studied the performance of BasisNet with lightweight model of different size. Here the size is measured by the Multiply-adds (MAdds) as we pay more attention to the inference cost. We experimented with a BasisNet-MV3 of MV3-large (1.0x224) with 8 bases. The lightweight model is MV3-small, and we experimented with two hyperparameters, i.e. the input image resolution to lightweight model (`{128, 160, 192, 224}`) and the multiplier (`{0.35, 0.5, 0.75, 1.0}`). As shown in Figure 1, even an extremely efficient lightweight model (MV3-small (0.35x128), computation overhead of 13.8M Madds) can lead to a performance boost from 77.7% to 78.9% (+1.2%). This experiment shows that resolution and multiplier can have an equivalent effect as reported in [7] and a lightweight model with a smaller computation overhead can bring most of the performance gain. Thus it might be more beneficial to scale the model multiplier and resolution coordinately [9].

D.4. Model synthesis with early termination

To better understand the capability of the lightweight model and the synthesized specialist, we split the 50K validation images into multiple buckets according to the sorted highest probability, and show the accuracy of different models for each bucket in Figure 2. Specifically, we show the accuracy within each bucket by the lightweight model, synthesized specialist model and a reference MobileNetV3 baseline. We observe that for at least one third of images where lightweight model has high prediction confidence, the accuracy gaps between these three models are negligible ($< 1\%$). The BasisNet has clear advantage over MobileNet in all buckets, especially for more difficult (low confidence) cases.

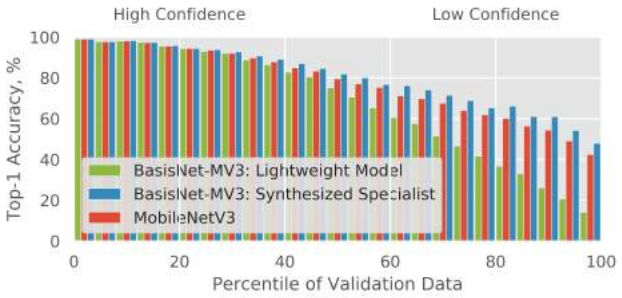

Figure 2. Prediction accuracy is comparable for more confident predictions (e.g. top 40%), and the synthesized specialist consistently outperforms regular MobileNet in all buckets.

E. More qualitative visualizations

E.1. Top categories handled by different basis models

In Figure 3 we show several most strongly activated categories for four different basis models on ImageNet validation set. Specifically we trained BasisNet-MV3 with 16 bases, and checked the mean weights at the last non-sharing layer (L15) and show the categories that have the highest mean weights. It is clear that the lightweight model captures the fine-grained visual similarity, for example the base 2 seems to handle the fluffy dogs while the base 14 is more about short-haired dogs. Another example is for base 13 that a clear grid pattern can be found in the images, but semantically these categories are loosely related.

E.2. Combination coefficients for visually similar categories

In Figure 4 we show 10 categories regarding different types of cars and the mean predicted combination coefficients for these categories in all layers. Obviously the lightweight model assigns similar coefficients for various cars, implying the effectiveness of the lightweight model. For example, we see that in Layer 14 almost all cars are relying on base 8, and in L15 all cars use a combination of base 3 and base 6. Quantitatively BasisNet over these 10 categories have an accuracy of 76.6%, but a corresponding regular MobileNetV3 has only 73.2%.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016. 2

[2] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. *International Conference on Learning Representations (ICLR)*, 2020. 2

[3] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[4] Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Bichen Wu, Zijian He, Zhen Wei, Kan Chen, Yuandong Tian, Matthew Yu, Peter Vajda, et al. Fbnetv3: Joint architecture-recipe search using neural acquisition function. *arXiv preprint arXiv:2006.02049*, 2020. 2

[5] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 1, 2

[6] Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 2

[7] Mark Sandler, Jonathan Baccash, Andrey Zhmoginov, and Andrew Howard. Non-discriminative data or weak model? on the relative importance of data and model resolution. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019. 3

[8] 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 (CVPR)*, 2018. 1, 2

[9] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019. 3

[10] Longhui Wei, An Xiao, Lingxi Xie, Xin Chen, Xiaopeng Zhang, and Qi Tian. Circumventing outliers of autoaugment with knowledge distillation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[11] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[12] Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2

[13] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2

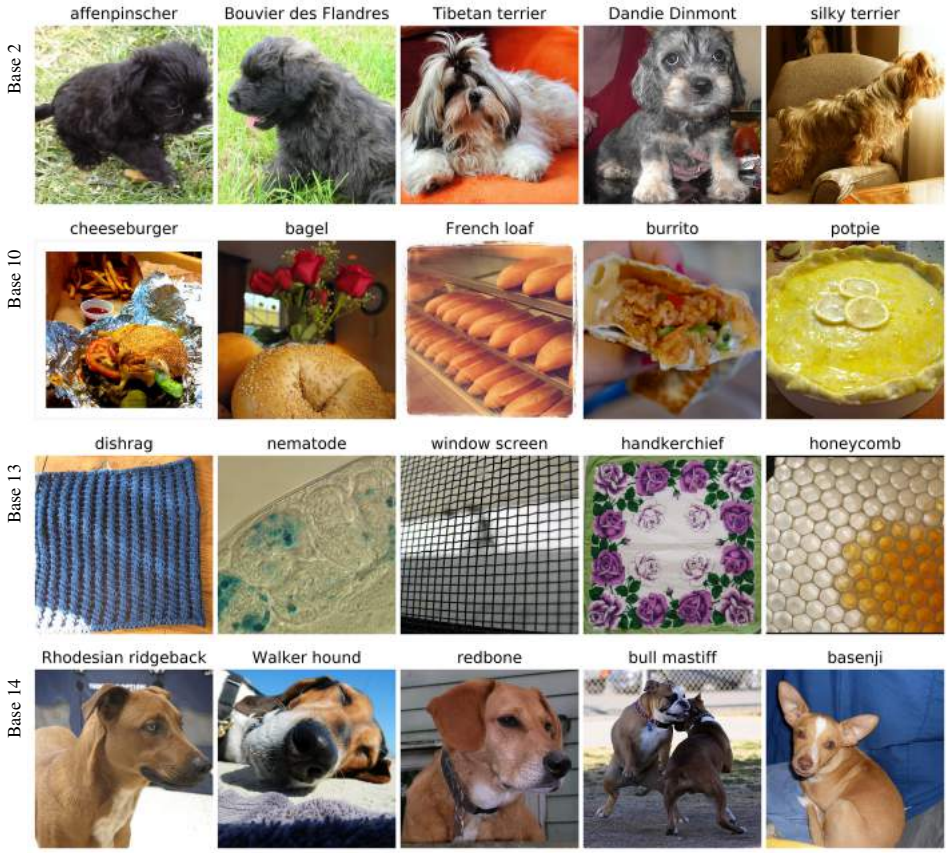

Figure 3. Categories with highest mean coefficients for different basis models.

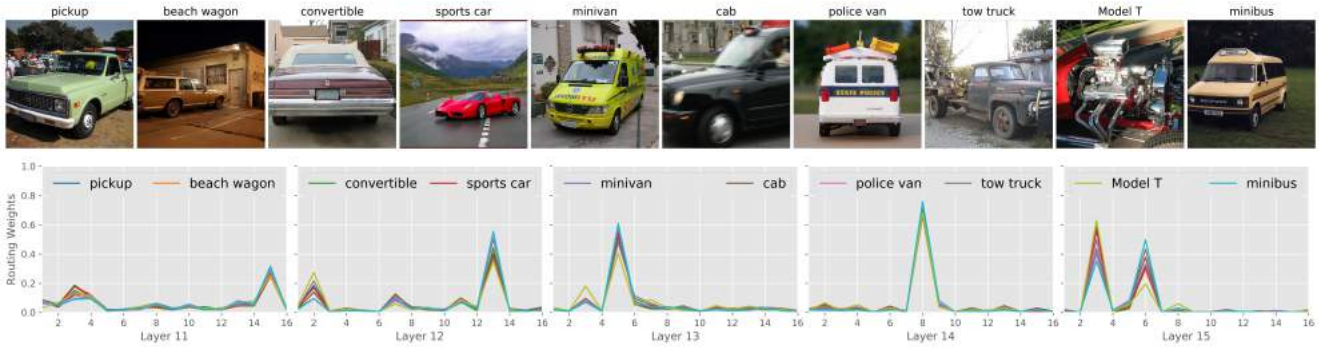

Figure 4. Visualization of predicted combination coefficients for similar categories over all layers.