A. Detailed Structure of a Compact Network

We show a detailed structure of a compact VGGNet on CIFAR-10 dataset in Table 1. The compact model used is from the multi-pass scheme experiment ("Iter 5 Trained" from Table 1 (a) in the paper). We observe that deeper layers tend to have more channels pruned.

| Layer | Width Width* Pruned | P/F Pruned |
|-------|---------------------|------------|
| 1     | 64 22 65.6%         | 34.4%      |
| 2     | 64 62 3.1%          | 66.7%      |
| 3     | 128 83 35.2%        | 37.2%      |
| 4     | 128 119 7.0%        | 39.7%      |
| 5     | 256 193 24.6%       | 29.9%      |
| 6     | 256 168 34.4%       | 50.5%      |
| 7     | 256 85 66.8%        | 78.2%      |
| 8     | 256 40 84.4%        | 94.8%      |
| 9     | 512 32 93.8%        | 99.0%      |
| 10    | 512 32 93.8%        | 99.6%      |
| 11    | 512 32 93.8%        | 99.6%      |
| 12    | 512 32 93.8%        | 99.6%      |
| 13    | 512 32 93.8%        | 99.6%      |
| 14    | 512 32 93.8%        | 99.6%      |
| 15    | 512 32 93.8%        | 99.6%      |
| 16    | 512 38 92.6%        | 99.6%      |
| Total | 5504 1034 81.2%     | 95.6%/77.2%|

Table 1: Detailed structure of a compact VGGNet. “Width” and “Width*” denote each layer’s number of channels in the original VGGNet (test error 6.34%) and a compact VGGNet (test error 5.96%) respectively. “P/F Pruned” denotes the parameter/FLOP pruned ratio at each layer.

B. Wall-clock Time and Run-time Memory Savings

We test the wall-clock speed and memory footprint of a “70% pruned” VGGNet (from Table 1 (a) in the paper) on CIFAR-10 during inference time. The experiment is conducted using Torch [1] on a NVIDIA GeForce 1080 GPU with batch size 64. The result is shown in Table 2.

| Model            | Test Error (%) | Params Pruned |
|------------------|----------------|---------------|
| Baseline         | 6.34           | -             |
| Pruned ([2])     | 6.88           | 88.5%         |
| Pruned (ours)    | 6.20           | 88.5%         |

Table 2: Wall-clock time and run-time memory savings of a compact VGGNet.

C. Comparison with [2]

On CIFAR-10 and CIFAR-100 datasets, we compare our method with a previous channel pruning technique [2]. Unlike network slimming which prunes channels with a global pruning threshold, [2] prunes different pre-defined portion of channels at different layers. To make a comparison, we adopt the pruning criterion introduced in [2] and closely follow the per-layer pruning strategy of [2] on VGGNet [3]. The result is shown in Table 3. Compared with [2], network slimming yields significantly lower test error with a similar compression rate.

| Model            | Test Error (%) | Params Pruned |
|------------------|----------------|---------------|
| Baseline         | 26.74          | -             |
| Pruned ([2])     | 28.36          | 76.0%         |
| Pruned (ours)    | 26.52          | 75.1%         |

Table 3: Comparison between our method and [2].

References

[1] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In BigLearn, NIPS Workshop, number EPFL-CONF-192376, 2011.
[2] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf. Pruning filters for efficient convnets. arXiv preprint arXiv:1608.08710, 2016.
[3] S. Zagoruyko. 92.5% on cifar-10 in torch. https://github.com/szagoruyko/cifar.torch