Improved Imagery Throughput via Cascaded Uncertainty Pruning on U-Net++

Mingshi Li*[1], Zifu Wang†[1], and Matthew B. Blaschko‡[1]

[1]ESAT-PSI, KU Leuven, Belgium

Abstract

The extensive use of machine learning inferences in real-life earth observation and remote sensing cases has grown over recent years. Network pruning has been carefully studied in various applications to speed up the machine learning workflow, but mainstream pruning strategies often focus on specific connection significancy rather than the sample difficulty. U-Net++ as a well-versed and capable semantic segmentation deep convolutional neural network architecture, as well as its equivalents, are all facing the challenge of overconfidence, which will create barriers for a robust uncertainty-based pruning strategy to be designed. In the following study, we analyzed the efficiency of deep neural networks and semantic segmentation in satellite imagery analysis, and proposed a new tailored workflow of dynamic pruning for U-Net++ by combining the ideas of network calibration and uncertainty and defining the inference complexity of network input samples. We tested and illustrated the capability of this new workflow and delivered a successful comparative study on its effectiveness on the DeepGlobe satellite imagery road extraction dataset and how it can greatly reduce the computational cost with little performance drop.

1 Introduction

The practice of ”neural network pruning,” which comprises methodically eliminating parameters from an existing network, as introduced by LeCun et al. [9], is one of the more common strategies for lowering these resource requirements at the time of testing. In most cases, the initial network is quite comprehensive and precise, and the objective is to develop a more compact network that maintains the same level of precision. Even while pruning has been around for a very long time [1], interest in the practice has only recently exploded thanks to the development of deep neural networks in the previous ten years. Some suggest eradicating insignificant connections or weights in order to prune the neural network [7]. Other pruning techniques do not always aim for the same locations to prune [8].

Over the years, a great number of methodologies have been published to study standards that should be served as criteria for network pruning, such as importance estimation [15], or Snip based on connection sensitivity [10], some others seek clues from GAN (generative adversarial learning) [11], and some focuses on pruning pipeline organising [14]. Other works suggest applying artificial transformations and augmentations to latent feature maps [6] can also be helpful in reducing network size. Frankle and Carbin [3] state that modern complex deep feed-forward neural networks contain one or more compact and meaningful sub-networks and further facilitate the theoretical credibility of network pruning.

The definition of dynamic inference varies, there is also a growing popularity in dynamic inference which focuses on dynamic pruning metrics or dynamically re-initializing pruning strategies in an active manner [12, 13, 20]. In this paper, we define dynamic inference, and similarly dynamic pruning as : given a standard or metric with which one can evaluate the overall difficulty of the correct segmentation of an image quantitatively, the inference procedure can be built in a dynamic way that allows networks to adapt to a resource-saving mode

[*] Corresponding e-mail: mingshi.li@kuleuven.be

[†] Corresponding e-mail: zifu.wang@kuleuven.be

[‡] Corresponding e-mail: matthew.blaschko@kuleuven.be

https://doi.org/10.7557/18.6813

whenever an easier sample is served as input, which can also be referred to as dynamic inference procedure. An illustration of such segmentation difficulty is shown in Figure 1 and Figure 2, where some image samples are much easier for, in this study, U-Net++ to correctly segment roads out from the backgrounds.

2 Methodology

2.1 Network Calibration with Temperature Scaling

For classification tasks, applying a non-linear layer to the neural network's output yields a designated probability distribution across the output space that can be used to evaluate confidence. Niculescu-Mizil and Caruana [17], Gal and Ghahramani [4], Naeini et al. [16] have demonstrated, however that current deep neural networks are badly calibrated despite having higher generalization accuracy. Researchers have associated this pattern with a growth in model capacity and the general over-fitting problem [5]. Miscalibration can lead to catastrophic skewness of results when output confidence of each layer in U-Net++ cascade serves as an operand in calculations of entropy-based uncertainty levels and even further used as a metric for network optimization and pruning.

As demonstrated in studies using logistic scaling [19], classifiers predicting posterior probability often produce uncalibrated results instead of true probabilities. Inspired by this work, Guo et al. [5] showed that temperature scaling is a simple yet effective method to calibrate prediction results. The strategy is simple, using a single scalar parameter T, which represents a temperature to rescale the logits provided by our network before they get fed into the softmax function. Due to the fact that the same temperature T is applied to all classes, the relationship between classification output after calibration and uncalibrated output is monotonic. The following equation explains how temperature-scaled logits can be softened and regularized in a simple way and provide less confident probability outputs:

$$P(\hat{y}) = \frac{e^{\mathbf{z}/T}}{\sum_j e^{z_j/T}}.$$

Where T is the temperature value, z is the logit vector, z_j is the individual vector element at each output node.

2.2 Entropy as Uncertainty Metric

Entropy may serve as the most suitable option for determining the uncertainty level of classification results in segmentation tasks. For most multi-class segmentation tasks with possible occlusion or semantic area overlaps, it is often not straightforward enough to only define the uncertainty based on the top two most likely predictions. We also have to consider other labels since in most cases, especially for the scenario of multiple-stage networks such as U-Net++, the first few stages or even later ones might not be able to fully distinguish pixels from several class labels. The higher the entropy is, the more ambiguous or average the system is, thus it is an intuitive assumption that for our model to make a confident prediction would be considered as difficult. To calculate entropy scores we use the following formula :

$$\phi\left(x\right) = -\sum_{k=1}^{j} P(y_k|x)\log(P(y_k|x)).$$

Entropy score is ϕ , $P(y_k|x)$ is conditional probability of layer output given input x.

2.3 Cascaded Pruning

The architecture of U-Net++ [21] is naturally organized from small to large subnetworks : shallow to deep pattern and inference processes can be run in a cascaded order where all four layers are semi-detached (i.e. they are trained together, but can operate separately). We propose "cascaded pruning" which utilises the entropy score of the current sample to decide whether to proceed to the next layer during the inference process. If such a score threshold is reached, all downstream blocks are skipped and the network performs an early exit instead of investing a large amount of inference computation budget into the network. Otherwise, the network proceeds to the next layer.

3 Experiment and Evaluation

In our experiments, we use a four layer U-Net++ with deep supervision [21]. For our dataset we use

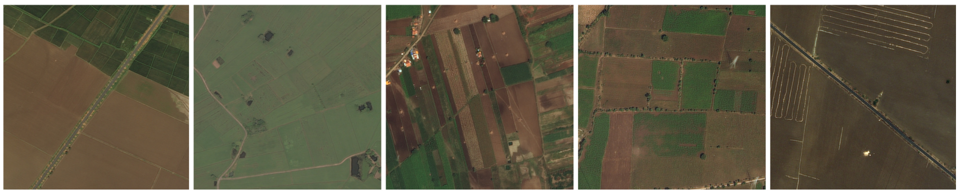


Figure 1: A demonstration of easy road extraction cases according to lowest entropy scores from the first U-Net++ layer

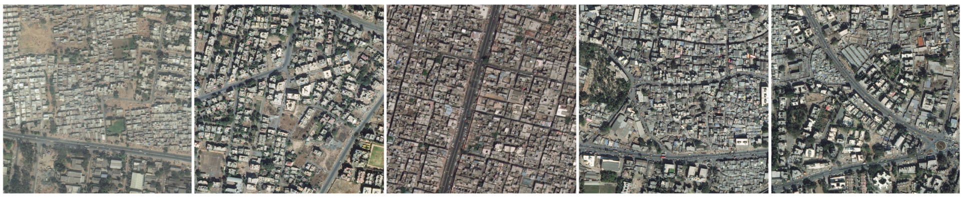


Figure 2: A demonstration of hard road extraction cases according to highest entropy scores from the first U-Net++ layer

the satellite images from CVPR18-DeepGlobe road extraction challenge [2] consisting of 6210 sample images among which 1242 samples are reserved as a test dataset. All images are compressed to a size of 256 × 256 pixels in order to fit our hardware specifications.

Multiply-accumulate operations (MAC) are used as an indicator of the computational complexity of the inference process [18]

$$FLOPs = [(C_i * K) + (C_i * K - 1)] * H_o * W_o * C_o$$

Where C,K as channel and kernel numbers, H,W as height and width.

$$MACs = \frac{FLOPs}{2}.$$

The total estimation of MACs number is used to show the computational complexity and IoU percentage is used to show the accuracy level. We take the inverse of the MACs number in GMACs to align it with IoU. We also have to do identity normalization to both numbers in advance. Weight coefficients are added to those two factors in order to customize the pruning strategy according to case-wise needs :

$$PerformanceScore = \sigma_1 * IoU + \frac{\sigma_2}{Cost}$$

$$\sigma_2 = 1 - \sigma_1.$$

σ_1 and σ_2 are the weighing factors of accuracy and computational cost impact.

4 Results and Discussion

4.1 Results on Calibration

In the calibration phase, we used the validation set to train the temperature in order to get an initial value. Although the standard of choosing the best temperature varies from case to case, in our experiment, our goal is to mitigate the mis-calibration behaviour of our network. The nature of our task, the accuracy level of satellite image segmentation calculated based off statistical collection of prediction results is challenging, yet the outcome of temperature scaling was as we expected and indeed reduced the negative effects caused by UNet++ miscalibration. Our primary metric of a successful recalibration is a relatively smaller ECE-score which means an overall lower confidence-accuracy gap between all confidence ranges.

The reliability diagrams revealed the satisfactory results of the re-calibration of UNet++ which can be seen from Figure 3 and Figure 4. Both the expected average accuracy error value (ECE) and the maximum absolute error value (MCE) were largely reduced, it is expected to be better if we can include more samples with variations. The optimal

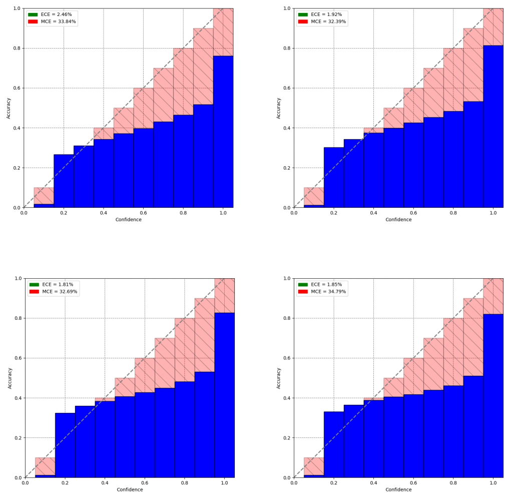


Figure 3: A collection of network reliability diagrams of L1 (TopLeft), L2 (TR), L3 (BL) and L4 (BR) **pre-calibration**, X-axis represents confidence bins, Y-axis represents accuracy levels

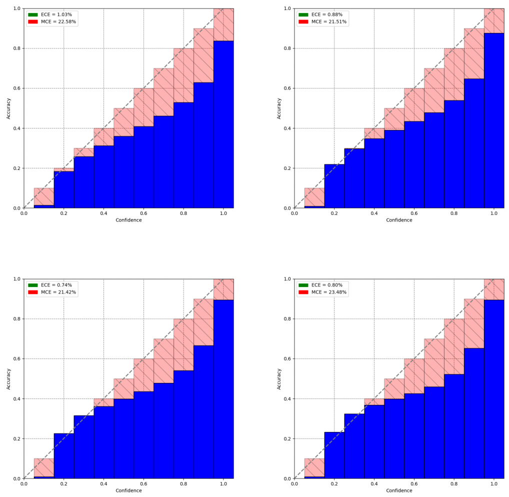


Figure 4: A collection of network reliability diagrams of L1 (TopLeft), L2 (TR), L3 (BL) and L4 (BR) **post-calibration**, X-axis represents confidence bins, Y-axis represents accuracy levels

temperature set we retrieved from the training is [**1.74, 1.67, 1.71, 1.70**]. Though the model is not perfectly aligned with the diagonal accuracy-confidence line, it is calibrated with the best temperature in order to get the lowest ECE value. We can see that the depth of the U-Net network still plays an important role and causes overconfidence in deep layers like L3 and L4.

In general, the ECE and MCE improvements are significant and successful with the trained temperature, as can be seen from Table 1. Pre-calibration results in Figure 3 shows that the network accuracy-confidence correspondence is highly skewed and will make it hard to distinguish the difficulty level of different samples at a finer scale, as compared with post-calibration results shown in Figure 4 re-calibration categorizes samples into a more evenly distributed set and helps with the downstream entropy-based thresholding.

4.2 Results on Uncertainties

We calculated the difficulties (uncertainty entropy scores) for each sample during inference. The difficulty of segmentation increases which is represented by hot colors while in other areas where network is very certain on its predictions, we get cold colors as indicated in Figure 6 (c) and (d). These heatmaps gave us very illustrative expression of which locations in the image our network feels uncertain about its prediction results. In Figure 6, those regions mainly lie on locations such as the edge of the roads and road-like objects (dirt path for example). These regions, if observed by human eyes, are also hard to be identified, thus entropy results indeed give us a good representation on how difficult specific pixels are and thanks to the temperature scaling and network calibration, we can lay our trust on the calculation of the entropy.

Architectures	ECE(pre-calibration)	ECE(post-calibration)
U-Net-L1	2.46%	1.03%
U-Net-L2	1.92%	0.88%
U-Net-L3	1.81%	0.74%
U-Net-L4	1.85%	0.80%
Architectures	MCE(pre-calibration)	MCE(post-calibration)
U-Net-L1	33.84%	22.58%
U-Net-L2	32.39%	21.51%
U-Net-L3	32.69%	21.42%
U-Net-L4	34.79%	23.48%

Table 1: ECE and MCE comparison before and after calibration

We also listed out the layer-wise output of post-calibration network in Figure 5, if we observe, for example, the binary output of first layer, we can clearly see that some areas of farmlands are also mistakened as roads (the white mist-like areas), but in the fourth layer's output these are eradicated by our network, this is due to the fact that the deeper we run into U-Net++ for inference the larger feature extraction block is thus finer features are learnt.

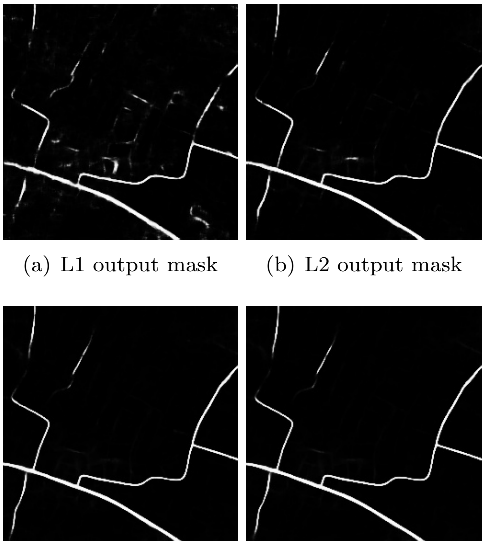
Architectures	Parameter Numbers	Computation Complexity (GMACs)
U-Net-L1	103.04k	4.03
U-Net-L2	519.11k	10.41
U-Net-L3	2.24M	20.42
U-Net-L4	9.16M	34.66

Table 2: Parameter numbers and computation complexity of all 4 stages of U-Net++

However, if we look closely into the output of layer 2, the binary mask that we get is already very similar to the actual best output that we got from layer 4. If we go even further into layer 3, we can observe almost no difference between layer 3 & layer 4, but the computational budget, as listed in Table 2, is very different in that layer 4 needs 4 times the number of parameter of layer 3, and 1.5 times the computational cost.

This is an indication that it would be optimal to stop the inference process at layer 3 and make an early exit. The experimental IoU also tells us the same story that the IoU (intersection over union) results we get (in particular for this sample we have shown as an example) from layer 3 is **0.72** and the IoU of layer 4 is **0.73**, which is a very minor improvement, while the computation resources invested is much more than this minimal accuracy improvement.

It is also worth noting that when we compare the post-calibration outputs with pre-calibration outputs, due to the overconfident nature of deep neural networks and it's early presence in all layers, oftentimes the calculated uncertainty score is not representative of the actual difficulty of the sample. See Figure 6 for the output of layer 1 from both calibrated and uncalibrated networks. Clearly, the uncalibrated network is more certain of what it produces even if it is in the very early stage of inference. This can lead to a very difficult thresholding strategy for difficulty binning, since there isn't enough margin between uncertainty score ranges to determine the difficulty of a sample. To be specific, a wider distribution of sample difficulties helps us to better distinguish the degree of difficulty of samples instead of trying to pick out narrow ranges from a rather squashed spectrum. Histograms of the general distribution of entropy-based uncertainty scores before and after calibration are illustrated in Figure 7 and Figure 8 :

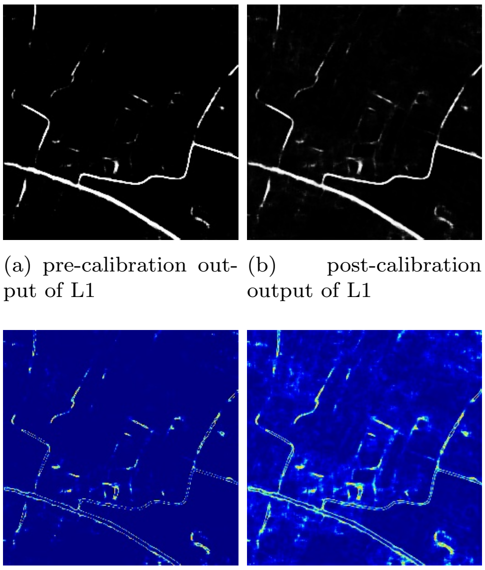

(a) L1 output mask (b) L2 output mask

(c) L3 output mask (d) L4 output mask

Figure 5: The binary output masks of four cascaded layers. As we can see, one can hardly distinguish the result of L4 from L3.

4.3 Impact on Network Performance

The method we used to measure how a threshold set can make an impact on the inference result is rather straightforward, by running an iterative threshold test on all possible threshold value based on our dataset, the values of three thresholds can be directly related to accuracy drop and speed gain. We found that the first layer early exit makes most impact on network accuracy, the deeper inference goes into the network, less impact was made on segmentation results as seen in Table 3. Our goal is to find a sweet spot of choices of three threshold values where computation complexity can be minimized while relatively high accuracy rate is preserved, which is "high accuracy-low complexity". The performance of the network can be efficiently



(a) pre-calibration output of L1 (b) post-calibration output of L1

(c) pre-calibration entropy of L1 (d) post-calibration entropy of L1

Figure 6: Outputs comparison with and without calibration. Note the mist-like uncertain regions in post-calibration output on the right, which is non-existent in pre-calibration output. It is also visible in corresponding entropy heat maps that the post-calibration network has less certain predictions.

adjusted by adopting our method, it is for user to define the most suitable threshold combination for each different application case.

5 Conclusions

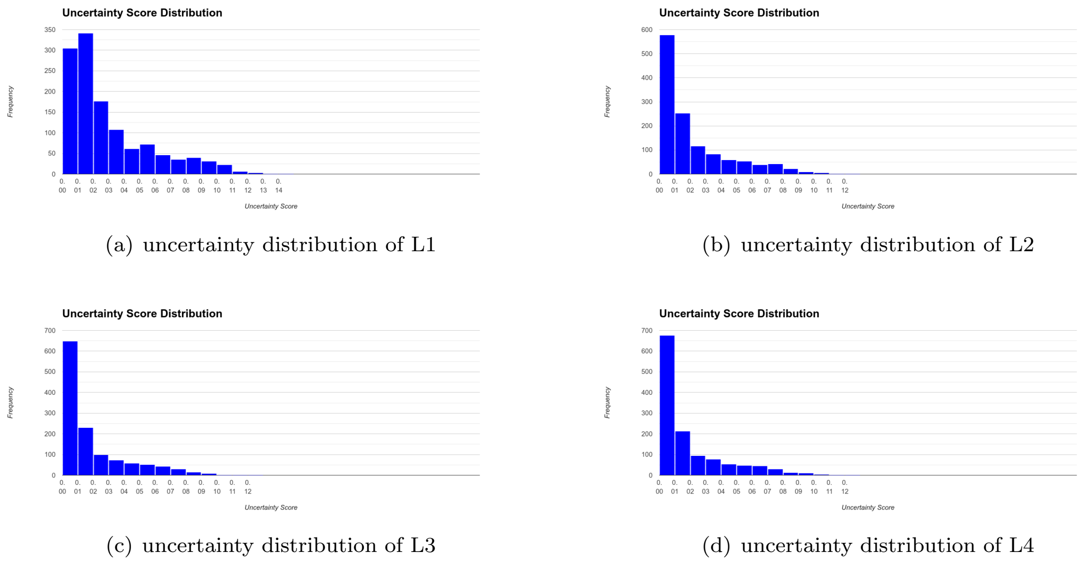
In this work, we have considered the problem of dynamic inference in the context of remote sensing. We have demonstrated an effective strategy for dynamic pruning in U-Net++ models that uses uncertainty scores to determine when early-exiting is sound. The resulting method significantly reduces inference time at negligible difference in segmentation performance. It is worth noting that, though our work shows a intuitive and simple approach to making satellite image segmentation tasks less computationally consuming, cohesive research on the relations between network calibration and entropy thresholding still needs to be done. Quantitative metric studies regarding the impact of threshold combinations and other methods of calibration could possibly provide better comparisons. Apart from satellite imagery and remote sensing applications, the cascaded uncertainty pruning is also able to be applied to other use cases, such as medical imaging.

Table 3: Some examples of speed increase (Exit at Ln represents the number of input samples of which the classification results are generated at each stage where the inference procedure of that specific sample is eventually ceased)

Exit at L1	Exit at L2	Exit at L3	IoU loss	MACs gain (%)
0	2	0	0	0.11
0	2	116	0.04	3.95
0	0	1243	0.36	41.09
0	105	22	0.59	6.64
0	105	1138	0.91	43.52
7	0	111	0.1	4.17
208	0	1035	3.39	49.01
208	723	312	6.44	65.8

Acknowledgements

This research received funding from the Research Foundation - Flanders (FWO) through project number S001421N, and the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme.



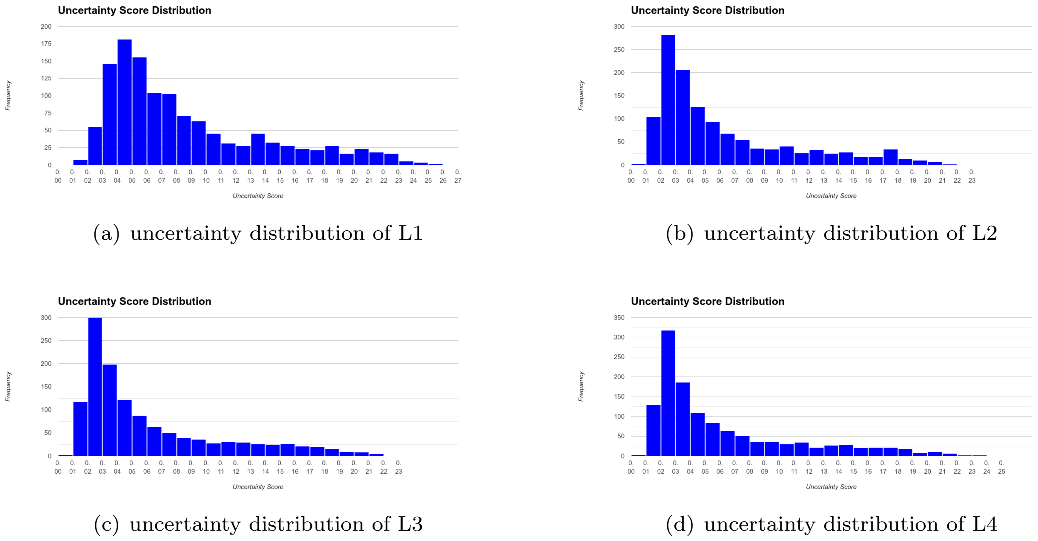
(a) uncertainty distribution of L1

(b) uncertainty distribution of L2

(c) uncertainty distribution of L3

(d) uncertainty distribution of L4

Figure 7: Pre-calibration uncertainty score distribution of four layers, X-axis is within range of {0,0.27}



(a) uncertainty distribution of L1

(b) uncertainty distribution of L2

(c) uncertainty distribution of L3

(d) uncertainty distribution of L4

Figure 8: Post-calibration uncertainty score distribution of four layers, X-axis is within range of {0,0.27}

References

[1] D. Blalock, J. J. Gonzalez Ortiz, J. Frankle, and J. Guttag. What is the state of neural network pruning? *Proceedings of machine learning and systems*, 2:129–146, 2020. doi: 10.48550/arXiv.2003.03033.

[2] I. Demir, K. Koperski, D. Lindenbaum, G. Pang, J. Huang, S. Basu, F. Hughes, D. Tuia, and R. Raskar. Deepglobe 2018: A challenge to parse the earth through satellite images. In *Proceedings of the IEEE Confer-*

ence on Computer Vision and Pattern Recognition Workshops, pages 172–181, 2018. doi: 10.1109/cvprw.2018.00031.

[3] J. Frankle and M. Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018. doi: 10.48550/arXiv.1803.03635.

[4] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016. doi: 10.48550/arXiv.1506.02142.

[5] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017. doi: 10.48550/arXiv.1706.04599.

[6] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1580–1589, 2020. doi: 10.48550/arXiv.1911.11907.

[7] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015. doi: 10.48550/arXiv.1510.00149.

[8] S. Han, J. Pool, J. Tran, and W. Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015. doi: 10.48550/arXiv.1506.02626.

[9] Y. LeCun, J. Denker, and S. Solla. Optimal brain damage. *Advances in neural information processing systems*, 2, 1989.

[10] N. Lee, T. Ajanthan, and P. H. Torr. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*, 2018. doi: 10.48550/arXiv.1810.02340.

[11] S. Lin, R. Ji, C. Yan, B. Zhang, L. Cao, Q. Ye, F. Huang, and D. Doermann. Towards optimal structured cnn pruning via generative adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2790–2799, 2019. doi: 10.48550/arXiv.1903.09291.

[12] T. Lin, S. U. Stich, L. Barba, D. Dmitriev, and M. Jaggi. Dynamic model pruning with feedback. *arXiv preprint arXiv:2006.07253*, 2020. doi: 10.48550/arXiv.2006.07253.

[13] J. Liu, Z. Xu, R. Shi, R. C. Cheung, and H. K. So. Dynamic sparse training: Find efficient sparse network from scratch with trainable masked layers. *arXiv preprint arXiv:2005.06870*, 2020. doi: 10.48550/arXiv.2005.06870.

[14] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*, 2018. doi: 10.48550/arXiv.1810.05270.

[15] P. Molchanov, A. Mallya, S. Tyree, I. Frosio, and J. Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11264–11272, 2019. doi: 10.48550/arXiv.1906.10771.

[16] M. P. Naeini, G. Cooper, and M. Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. doi: 10.1609/aaai.v29i1.9602.

[17] A. Niculescu-Mizil and R. Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632, 2005. doi: 10.1145/1102351.1102430.

[18] Nvidia. Deep learning performance documentation. `https://docs.nvidia.com/deeplearning/performance/dl-performance-convolutional/index.html`, 2022.

[19] J. Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[20] A. Zhou, Y. Ma, J. Zhu, J. Liu, Z. Zhang, K. Yuan, W. Sun, and H. Li. Learning n: m fine-grained structured sparse neural networks from scratch. *arXiv preprint arXiv:2102.04010*, 2021. doi: 10.48550/arXiv.2102.04010.

[21] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 3–11. Springer, 2018. doi: 10.48550/arXiv.1807.10165.