2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

PyramidEnsemble: Joining Large and Small Models

Adrian Köring
Chair of Computational Intelligence
Otto-von-Guericke-University
Magdeburg, Germany
adrian.koering@ovgu.de

Christoph Steup
Chair of Computational Intelligence
Otto-von-Guericke-University
Magdeburg, Germany
steup@ovgu.de

***Abstract*—In this paper, we aim to improve segmentation performance and uncertainty calibration within a fixed computational budget. We propose PyramidEnsembles, which contain members ranging from small over medium and large, to overcome one major problem in applying neural networks to the automotive domain: the trade-off between model performance and overconfidence in the uncertainty predictions. PyramidEnsembles use multiple models of different sizes from the same family in order to combine their strengths: good segmentation performance and well-calibrated uncertainties. We focus our experiments on EfficientNet-based segmentation models applied to the Cityscapes dataset, which is widely used in the field of autonomous driving. We evaluate single models, uniform ensembles (one architecture repeated) and PyramidEnsembles (combination of different model capacities) composed of the EfficientNet model family. Our evaluations show that within the same computational budget, PyramidEnsembles can outperform a single model in terms of segmentation performance while providing better calibrated uncertainties. Scaling over different computational budgets shows that this performance gap increases further. Different uniform ensembles offer a comparable segmentation or uncertainty calibration performance: 3 copies of the EfficientNet-B3 achieve an IoU of 0.7195 while an ensemble of 7 EfficientNet-B0 models yields an Expected Calibration Error (ECE) of 0.0667. One PyramidEnsemble containing an instance of EfficientNet-B0 through B3 is a close second on either metric at 0.7188 IoU and 0.0698 ECE and offers a better trade-off between segmentation performance and uncertainty calibration in this computational budget.**

***Index Terms*—Uncertainty Estimation, Semantic Segmentation, Ensembles, EfficientNet, Cityscapes**

I. Introduction

In the field of robotics and automated driving, image classification and segmentation are employed to extract information from camera images. Besides model performance on this primary task, the trustworthiness of these predictions is equally important. Uncertainty estimation aims to quantify the model's confidence in its prediction and helps downstream processes to adjust their trust into the provided scene information.

A key difficulty in this field is that larger neural networks (NNs) typically perform well on the primary task, but tend to be overconfident in their predictions. Smaller models yield less confident predictions at the price of overall subpar classification/segmentation performance. However, their confidence better matches the actual accuracy of the associated predictions, resulting in a better calibrated uncertainty estimate [1].

Ensembles query multiple models and combine their predictions. This machine learning technique has long been used to improve model performance. In addition, they are known to produce better calibrated uncertainty estimates. However, ensembles are often considered unsuitable for real-time applications due to their increased computational complexity. Over time, several approaches have been developed to decrease the computational budget by exploiting the structure of the used models.

In this paper, we propose PyramidEnsembles as a new model selection method that includes models with different capacities. We group models ranging from small to large and aim to combine their respective strengths: segmentation performance paired with well-calibrated uncertainties.

To evaluate the suitability of our approach for the automated driving scenario, we compare single models, uniform ensembles and PyramidEnsembles within the same computational budget. This shows PyramidEnsembles to strike a better balance in terms of segmentation performance and uncertainty calibration on the Cityscapes Dataset [2].

II. Related Work

Theoretical derivations of model uncertainty begin with a Bayesian approach: By determining the posterior distribution of model parameters, we obtain the probability associated with each prediction. However, the actual distribution is often computationally intractable [3].

Monte Carlo Dropout is used to approximate Bayesian inference in neural networks [4]. Dropout randomly masks a fraction of model weights and creates slightly different models in each run. By repeatedly sampling from exponentially many subnetworks, we can approximate the posterior distribution of the model. In practice, the estimate converges after about 50 iterations. However, this considerably increases the test-time computational complexity of the model [4].

Ensembles combine multiple models trained in different ways: for example, by bagging, boosting, with different initialization, hyperparameters, or architectures. This increases the diversity of models compared to Dropout, so that a few models are sufficient to obtain a good approximation of the output distribution [4]. Furthermore, ensembles help mitigate the overconfidence exhibited by individual models, making the predicted softmax score a more reliable estimate of the model's

This work was partially funded by the German Ministry for Education and Research (BMBF) under project AULA-KI with grant number 01IS22061A.

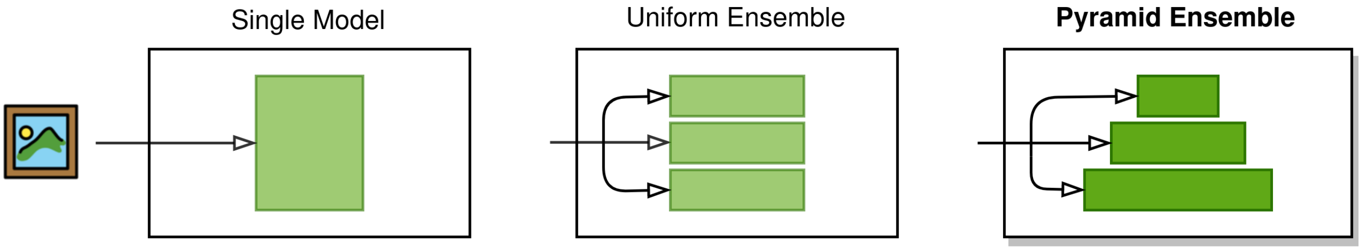

Fig. 1. Uniform Ensembles use equally sized models to improve segmentation performance and uncertainty calibration over a single model. PyramidEnsembles combine members with varying model capacity to join their strength: good calibration in small models and better segmentation performance with larger ones.

(un-)certainty. However, storing multiple models increases memory consumption and cache pressure.

While ensembles work well with fewer than 50 forward passes, they are still more expensive than a single model. This motivates recent work on more efficient ensembles: BatchEnsemble [5] decomposes the parameters into one weight shared among the ensemble members and a binary mask that selects different elements from the shared weight for each member. This saves both persistent and random-access memory, but does not substantially reduce the computational budget compared to standard ensembles [5].

PackedEnsembles [6] use grouped convolutions to combine the ensemble members into a single CNN. Using the same number of groups in each convolutional layer separates data flow between ensemble members. This formulation better utilizes device parallelization and reduces the total number of parameters, yielding a less computationally costly ensemble.

Temporal ensembles like [7] propose an optimization for image streams – an application often limited by computational cost. They unravel the ensemble over time, with each frame being processed by another ensemble member. For static scenes, this would return the original ensemble output once each member participated. For dynamic scenes, an optical flow module is added to track the image patches through time. Along a sequence of patches, the ensemble output can be aggregated as usual. While this formulation reduces the computational complexity per step, it adds latency to the uncertainty estimation. This is undesirable because, intuitively, the uncertainty estimate is most useful for the first detection – which should also be the most uncertain. When the object is re-identified in successive frames, the models' confidence can increase. Neither of these properties can be observed here, because inference with a single model per time step incentivizes to choose a comparably large and therefore overconfident one. This initial prediction can then be balanced out as more members contribute. However, beyond the longer latency until a reliable uncertainty estimate is available, the intermediate values behave counter intuitively, as the initial predictions are associated with overconfident uncertainties rather than a cautious start.

Our work resembles [8] more closely. Their results suggest that within the same computational budget, uniform ensembles can outperform a single model. We build upon this and additionally investigate the ensembles' predictions in terms of uncertainty calibration. Instead of uniform ensembles formed by models of equal capacity, we consider the more generic case of ensembles with members of varying capacity.

III. Methodology

Motivated by smaller models producing better calibrated predictions, while larger models have an advantage in terms of task-performance, we propose to join CNNs with different model capacities (from small to large – resembling a pyramid) into one ensemble (see Fig. 1). This combination allows each member to play to its respective strength and yields an improved model in both performance on the primary task and calibration of uncertainty estimates.

Combining only small models shows diminishing returns after a few ensemble members. While their joint effort produces very well calibrated uncertainties, the task-performance reaches an upper bound [9]. Coalescing larger models improves task-performance at the cost of less accurate uncertainty estimates – due to fewer models fitting into the same computational budget.

Our PyramidEnsemble approach is facilitated by families of classification CNNs, which are used as encoders in segmentation models. A family of models shares the same sequence of layers (architecture) but allows to derive models of varying capacities by scaling a base model. Many recent works like MobileNets [10], EfficientNet [11] or ConvNeXt [12] propose models in such a way. Each family defines a number of models ranging from small to large. PyramidEnsembles are created on the basis of such families by including several models with the same architecture (e.g. MobileNet or EfficienNet) but different capacities (e.g. EfficientNet-B0 and EfficientNet-B3). This allows to include relatively large models, which can contribute their segmentation performance and, on the other hand, a couple of smaller models, which contribute better calibrated uncertainties. Including more and smaller models in the ensemble benefits uncertainty calibration. Overall, PyramidEnsembles should provide a better trade-off compared to uniform ensembles and outperform singular models within the same compute budget.

The compositional architecture of PyramidEnsembles gives rise to an additional degree of freedom over uniform ensembles: by including multiple members of varying capacity, exponentially many ensembles can be formed. Let F be a

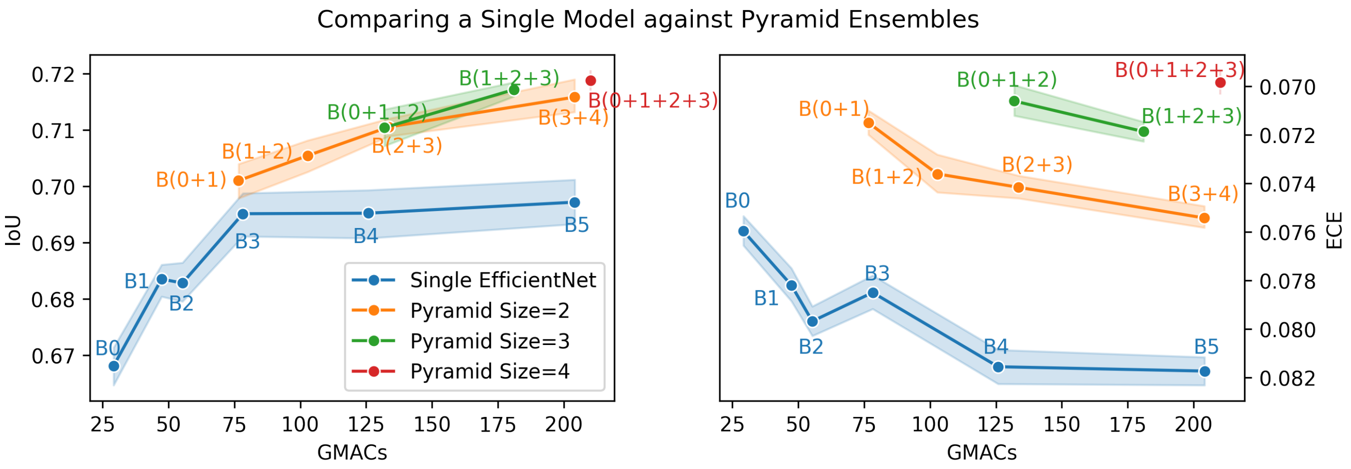

Fig. 2. Our main experiment compares a single EfficientNets-B0 through B5 against PyramidEnsembles build from smaller Efficient-Nets. While ECE is to be minimized, we flip the y-axis s.t. both metrics are improving upwards. IoU is comparable between different ensemble compositions, ECE favors large ensembles with smaller models. Single models are outperformed by any equally sized ensemble in both regards.

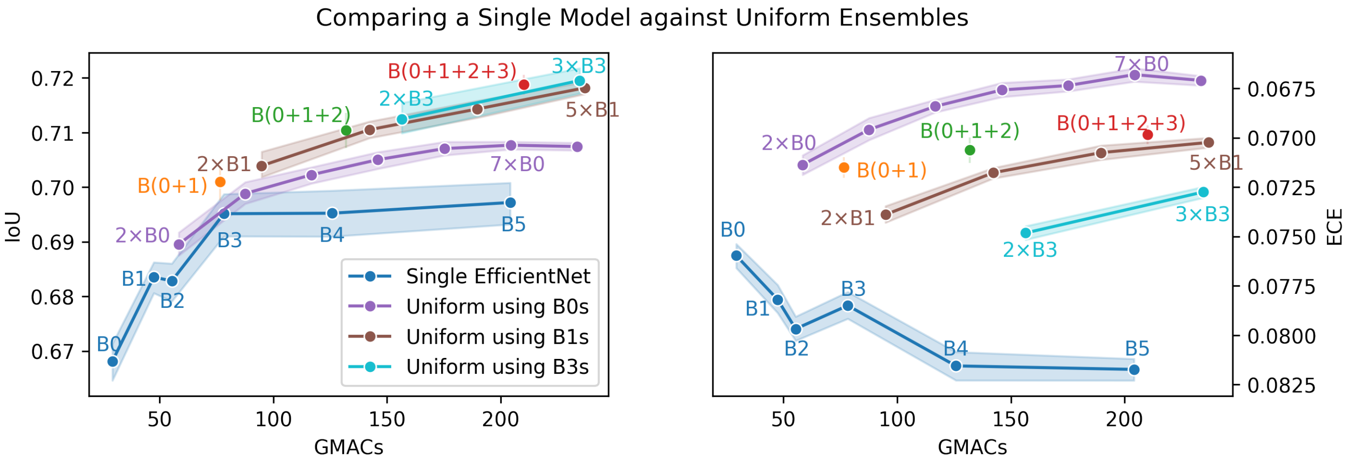

Fig. 3. We further compare against Uniform Ensembles. We vary the number of ensemble members for three EfficientNet encoders which shows comparable segmentation results outperforming the single models, too. The segmentation performance of B0-Ensembles is limited but with good calibration results. Ensembles of larger models can improve segmentation results at the cost of worse uncertainty calibration, causing Uniform Ensembles to not scale as well as PyramidEnsembles. Our best performing Pyramid B(0+1+2+3) is included for reference. It shows comparable segmentation performance with superior uncertainty calibration results. (ECE's y-axis is flipped again s.t. both metrics are improving upwards.)

set of encoders, with each member f^i varying in capacity. Encoders in F are not limited to originate from the same family, but this simplification is considered here. Ensembles are a tuple constructed from one or more f^i without limits on diversity or repetition. Uniform ensembles contain one model architecture f^i z times: $e_u = (f_0^i, \ \ldots, f_z^i)$. PyramidEnsembles, on the other hand, contain models of various sizes: $u_p = (f_0^i, f_1^j, \ \ldots, f_z^k)$ with the model capacity $C^i = \mathrm{GMAC}(f^i)$ varying (without loss in generality: increasing) between members $C^i \leq C^j \leq \cdots \leq C^k$. Compared to uniform ensembles, this causes PyramidEnsembles to create a much larger landscape of possible ensembles: $E = \{(f_0^i, \ \ldots, f_z^j)\} \subseteq F \times F \times \cdots \times F$. An example of such a combination is a PyramidEnsemble of the largest and the smallest available encoder, skipping all medium-sized variants in-between.

This increased hyperparameter space makes evaluating the performance of all feasible PyramidEnsembles more difficult. However, our results suggest that selecting members consecutively, and maximizing ensemble size, performs favorably compared to including fewer but larger models in both aspects: segmentation performance and uncertainty estimation quality. Therefore, we will limit ourselves to using consecutive family members $(f^i, f^{i+1}, \ \ldots, f^z)$ with $C^i < C^{i+1}$ s.t. $\nexists \ f^k \in F$ with $C^i < C^k < C^{i+1}$.

$$\hat{\boldsymbol{y}} = \frac{1}{z} \sum_{i \in e} f^i(\boldsymbol{x}) \tag{1}$$

$$\hat{\boldsymbol{u}} = 1 - \mathrm{softmax}(\hat{\boldsymbol{y}}) \tag{2}$$

The ensemble output $\hat{\boldsymbol{y}}$ of an ensemble $e \in E$ is the average of each member model prediction (see Eq. 1). As metric of the prediction's uncertainty estimation $\hat{\boldsymbol{u}}$, we use the complementary output probability (see Eq. 2).

A second key aspect of our proposed method is the comparison of models within one compute budget. We argue that at a comparable level of Multiply-Accumulates (MACs) an ensemble of smaller models can outperform a single model. This effect is especially pronounced for large models, probably due to overfitting, as shown in [8].

Furthermore, an ensemble can be beneficial for latency critical systems, since preliminary results are available earlier. When processing members in parallel or sequentially starting with the smallest, the prediction from the smallest model will be available sooner than from the single (ensemble-sized) model. Additionally, the prediction from the smallest member will already provide a relatively good uncertainty calibration. Hence, it might be beneficial to allow for incremental output from the ensemble in order to provide follow-up components an uncertain, but maybe faster prediction. Later models can contribute better segmentation results and finalize the associated uncertainty estimate.

IV. Evaluation

We test our approach on a semantic segmentation task using EfficientNet [11] as model family. Their model scaling approach creates a wide range of model capacities, from which we select the range of EfficientNet-B0 through B5 for our experiments. Inference with larger models like B6 and B7 drops to 15 images per second or less even on Nvidia-A100s and hence will be out of reach for embedded systems operating near real-time. On top of the EfficientNet encoder, we deploy LR-ASPP as a segmentation decoder [13].

Since only the encoder differs between different architectures of our evaluated models, we will refer to our complete segmentation model in the following only by the name of its encoder. Furthermore, figures may skip the EfficientNet prefix and use B0 through B5 as a shorthand. PyramidEnsembles indicate their members with B$(i + ... + j)$ for an ensemble containing one model each. Uniform Ensembles have the shorthand of $z \times$ Bi for an ensemble containing z copies of our EfficientNet-Bi based segmentation model.

We evaluate our proposal on the urban scene-parsing dataset Cityscapes [2]. As training set, we utilize the finely annotated images at a resolution of 512×1024 pixels. We train for 15k steps and optimize the model with AdamW starting at a learning rate of 1×10^{-3} and a weight decay of 0.02. The learning rate is reduced following a cosine decay down to 1×10^{-7} during training.

$$IoU = \frac{T_p}{T_p + F_p + F_n} \tag{3}$$

$$ECE = \sum_{i=0}^{N} \text{bin}_i |\text{certainty}_i - \text{accuracy}_i| \tag{4}$$

We compute two metrics in order to determine good and well-calibrated classifiers simultaneously: For segmentation performance, we calculate the Jaccard Index / Intersection-over-Union (IoU, see Eq. 3). IoU compares the amount of correctly classified pixels T_p with the sum of false positively F_p and negatively F_m classified pixels. Uncertainty calibration is measured by the Expected Calibration Error (ECE, see [14] and Eq. 4 following, [15]). ECE bins predictions according to their certainties (we use $N = 15$ bins) and compares the accuracy in each bin with the average certainty. For example, at a certainty level of around 0.8 approximately 80 % of predictions should be correct and the absolute difference to this correctness represents the calibration error. Furthermore, this value is scaled by the fraction of data points bin_i in each interval. Computing the absolute error penalizes over- and underconfidence equally. The detriment of deviation in either direction may be application specific, but in our case both cases are considered important.

ECE requires ground truth annotations, but the Cityscapes test annotations are not public. Consequently, we report results as mean over the validation split which could bias results since the validation set was also used to establish hyperparameters. However, only training of individual models was considered during parameter selection. Ensembles were created without additional cross-check against the validation set. Therefore, uniform and pyramid ensembles are treated equally and comparing single models with PyramidEnsembles should favor the baseline.

Our first experiment compares PyramidEnsembles against single EfficientNets across different compute budgets (from B0 through B5). We plot segmentation performance (IoU) and uncertainty calibration (ECE) relative to their computation cost – measured as Giga-MACs (GMACs). The results (Fig. 2) show that segmentation performance of a single model increases together with the compute budget, but at diminishing returns. Simultaneously, ECE deteriorates as the model size grows.

PyramidEnsembles outperform the single model across several budgets and in both IoU and ECE, with the margin increasing for larger compute budgets – matching observations in [8]. Ensembles containing larger models (e.g., B(2+3)) still yield better segmentation performance at the cost of worse uncertainty calibration. However, pairing more but smaller models achieves comparable segmentation performance (for example B(0+1+2) vs. B(2+3)) with better uncertainty estimates on the same budget. Following this trend, B(0+1+2+3) combines the four smallest encoders and outperforms all other PyramidEnsembles and single models in terms of IoU and ECE, while featuring a compute budget comparable to EfficientNet-B5.

In addition to comparing single models with PyramidEnsembles, we repeat our experiments with uniform ensembles. Fig. 3 shows these runs grouped by encoder type and enumerating different ensemble sizes ranging from two through the extent of comparable model capacities. The key difference with uniform ensembles is their intricate balance:

while ensembles of EfficientNet-B0s improve in task performance and yield very well calibrated uncertainties, both suffer from diminishing returns as the ensemble grows. One step up, ensembles of B1s improve segmentation results with diminished calibration performance, but appear to be on-par with PyramidEnsembles overall. However, EfficientNet-B3s can only improve segmentation by a small amount, while their ECE deteriorates considerably. Selected results in Table I show two separate uniform ensembles yield best results, while one PyramidEnsemble comes in second in both metrics and represents a good compromise.

Furthermore, we include Monte Carlo Dropout (MCD) in our evaluation. Combining 7 of our smallest models already matches the compute budget of the largest model at around 200 GMACs which we focus our evaluation on. In literature, however, 50 iterations are suggested for good estimates. We therefore limit the encoder to a deterministic forward pass and only apply test-time dropout on the feature embeddings. This corresponds to only sampling the decoder weights. Additionally, these models were trained with dropout, unlike our single and ensembles models. Under MCD, the test-time compute budget increases from 125.8 GMACs to 168.6 for our EfficientNet-B4 based model and grows from 204.1 to 251.6 GMACs for the EfficientNet-B5 based one. Compared to single models, the MCD EfficientNet-B4 outperforms both the single EfficientNet-B4 and -B5 in both IoU and ECE at a lower GMAC count. On the other hand, both MCD variants trail uniform and PyramidEnsembles with even lower GMAC counts by a wide margin.

Considering this trade-off between the three metrics, IoU, ECE and GMACs we visualize the Pareto optimal models in Figure 4. We can see that single models are only competitive for low compute budgets, where no ensembles can be formed from the given set of base models. Larger models and ones including MCD are outperformed by ensembles. Uniform ensembles and especially the various ensembles based on EfficientNet-B0 represent an optimal area in the regime of lower IoU performance but superior uncertainty calibration, visible in the string of uniform models in the center. Towards the area of increased IoU, PyramidEnsembles begin to dominate and outperform all but one uniform ensemble.

In summary, we argue that PyramidEnsembles scale better since growing compute budges are matched with improvements in both segmentation performance and uncertainty calibration (for example, by using B(0+1), B(0+1+2) and B(0+1+2+3)). An obvious downside of PyramidEnsembles is the need for model families and the resulting limited number of ensemble options. This is counterbalanced by the diminishing return of increasing the ensemble size for uniform ensembles. Consequently, we consider PyramidEnsembles the superior choice if a model family is available and accurate uncertainty prediction is necessary.

Efficient-Net(s)	IoU ↑ ±σ	ECE ↓ ±σ	GMACs ↓
B0	0.668 ± 0.0061	0.076 ± 0.0011	29.2
B1	0.683 ± 0.0050	0.0782 ± 0.0012	47.4
B2	0.682 ± 0.0061	0.0797 ± 0.0011	55.3
B(0+1)	0.701 ± 0.0052	0.0715 ± 0.0014	76.6
B3	0.695 ± 0.0066	0.0785 ± 0.0012	78.3
B4	0.695 ± 0.0075	0.0815 ± 0.0013	125.8
B(0+1+2)	0.710 ± 0.0056	0.0706 ± 0.0012	131.9
50 × B4 (MC)	0.6982 ± 0.004	0.0787 ± 0.0008	168.6
B5	0.697 ± 0.0067	0.0817 ± 0.001	204.1
7 × B0	0.7077 ± 0.0012	**0.0667** ± 0.0006	204.4
B(0+1+2+3)	**0.7188** ± 0.0033	**0.0698** ± 0.0008	210.2
3 × B3	**0.7195** ± 0.004	0.0728 ± 0.0005	234.7
5 × B1	0.7182 ± 0.002	0.0702 ± 0.0004	237.1
50 × B5 (MC)	0.6989 ± 0.0048	0.0796 ± 0.0009	251.6

TABLE I

SELECTED RESULTS FOR SINGLE MODELS, UNIFORM AND PYRAMID ENSEMBLES GROUPED BY COMPUTE BUDGET WITH FOCUS ON THE 200 GMACS GROUP. IOU AND ECE ARE AVERAGED OVER 11 RUNS AND REPORTED WITH STANDARD DEVIATION σ.

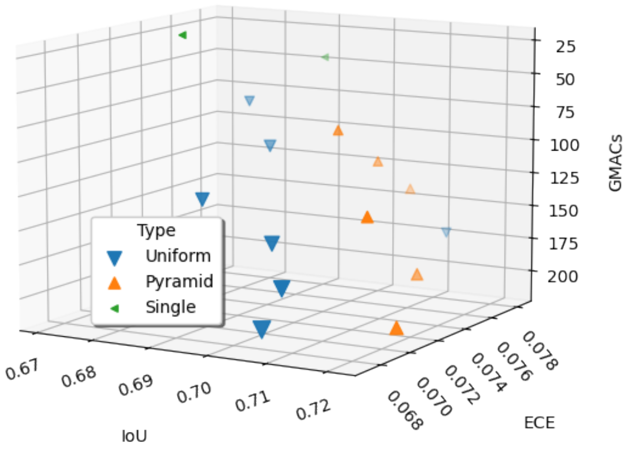

Fig. 4. Overview of the trade-offs of GMACs, IoU and ECE for all non-dominated ensembles. The types indicate the ensemble type, whereas the size of the markers indicate the size of the ensemble.

V. Conclusion

We propose PyramidEnsembles, a new ensemble method, which combines members of varying model capacity from one neural network family. We compare them with individual models, Monte Carlo dropout and uniform ensembles within the same computational budget. Our results indicate that PyramidEnsembles outperform single larger models and Monte Carlo dropout in terms of segmentation (IoU) and uncertainty calibration performance (ECE). Furthermore, while some uniform models yield comparable segmentation performance, they suffer from diminishing returns when adding further models. Creating uniform ensembles from larger models increases segmentation performance, but causes calibration performance to decline. This shows PyramidEnsembles to combine the strength of small models (better calibrated uncertainties) with the segmentation performance of larger models. Furthermore, they offer better scaling behavior by improving both metrics as the computational budget increases.

Future work needs to investigate more generic Pyramid-Ensemble strategies including repetitions of member models. Additionally, more model families should be investigated with a wider range of compute budgets. The increase in hyper-parameter space may need more sophisticated approaches to the search for optimal hyperparameters, like Evolutionary Algorithms. Finally, evaluating PyramidEnsembles uncertainty estimation in cross- and out-of-domain settings is necessary in order to apply it in real-world scenarios.

References

[1] J. Z. Liu, S. Padhy, J. Ren, Z. Lin, Y. Wen, G. Jerfel, Z. Nado, J. Snoek, D. Tran, and B. Lakshminarayanan, "A simple approach to improve single-model deep uncertainty via distance-awareness," *CoRR*, vol. abs/2205.00403, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2205.00403

[2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[4] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. W. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *CoRR*, vol. abs/2011.06225, 2020. [Online]. Available: https://arxiv.org/abs/2011.06225

[5] Y. Wen, D. Tran, and J. Ba, "Batchensemble: An alternative approach to efficient ensemble and lifelong learning," *CoRR*, vol. abs/2002.06715, 2020. [Online]. Available: https://arxiv.org/abs/2002.06715

[6] O. Laurent, A. Lafage, E. Tartaglione, G. Daniel, J. Martinez, A. Bursuc, and G. Franchi, "Packed ensembles for efficient uncertainty estimation," in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. [Online]. Available: https://openreview.net/pdf?id=XXTyv1zD9zD

[7] P. Huang, W. T. Hsu, C. Chiu, T. Wu, and M. Sun, "Efficient uncertainty estimation for semantic segmentation in videos," *CoRR*, vol. abs/1807.11037, 2018. [Online]. Available: http://arxiv.org/abs/1807.11037

[8] D. Kondratyuk, M. Tan, M. Brown, and B. Gong, "When ensembling smaller models is more efficient than single large models," *CoRR*, vol. abs/2005.00570, 2020. [Online]. Available: https://arxiv.org/abs/2005.00570

[9] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Mach. Learn.*, vol. 51, no. 2, pp. 181–207, 2003. [Online]. Available: https://doi.org/10.1023/A:1022859003006

[10] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

[11] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *CoRR*, vol. abs/1905.11946, 2019. [Online]. Available: http://arxiv.org/abs/1905.11946

[12] Z. Liu, H. Mao, C. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," *CoRR*, vol. abs/2201.03545, 2022. [Online]. Available: https://arxiv.org/abs/2201.03545

[13] A. Howard, M. Sandler, G. Chu, L. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for mobilenetv3," *CoRR*, vol. abs/1905.02244, 2019. [Online]. Available: http://arxiv.org/abs/1905.02244

[14] M. P. Naeini, G. F. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using bayesian binning," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, B. Bonet and S. Koenig, Eds. AAAI Press, 2015, pp. 2901–2907.

[15] A. Kumar, P. Liang, and T. Ma, "Verified uncertainty calibration," 2020.