Class Based Thresholding in Early Exit Semantic Segmentation Networks

Alperen Görmez and Erdem Koyuncu, *Senior Member, IEEE*

***Abstract*—We consider semantic segmentation of images using deep neural networks. To reduce the computational cost, we incorporate the idea of early exit, where different pixels can be classified earlier in different layers of the network. In this context, existing work utilizes a common threshold to determine the class confidences for early exit purposes. In this work, we propose Class Based Thresholding (CBT) for semantic segmentation. CBT assigns different threshold values to each class, so that the computation can be terminated sooner for pixels belonging to easy-to-predict classes. CBT does not require hyperparameter tuning; in fact, the threshold values are automatically determined by exploiting the naturally-occurring neural collapse phenomenon. We show the effectiveness of CBT on Cityscapes, ADE20K and COCO-Stuff-10K datasets using both convolutional neural networks and vision transformers. CBT can reduce the computational cost by up to 23% compared to the previous state-of-the-art early exit semantic segmentation models, while preserving the mean intersection over union (mIoU) performance.**

***Index Terms*—Early exit, neural collapse, segmentation.**

I. Introduction

As deep learning advances rapidly, new, larger models are frequently introduced, leading to improved performance [1]–[3]. However, larger models, while capable of learning complex patterns, come with higher inference costs. In the era of decentralized computing on edge devices (e.g., IoT), minimizing the inference cost of large models becomes crucial for deployment on resource-constrained devices [4], [5].

To reduce the inference cost without compromising performance, early exit networks are proposed [6], [7]. Early exit networks capitalize on the heterogeneity of real world data. Since not all data samples have the same "difficulty", "easy" data samples can be allowed to exit the model early to save computation [6]–[10]. Early exit networks have been studied in conjunction with network pruning [11], [12]. They also have close ties with the phenomenon of *neural collapse* [8], [13].

The neural collapse phenomenon states that as one travels deeper in a neural network, the intermediate representations become more disentangled, forming distinct clusters at the last layer, which makes classification easier [13]. Recent works expand on this phenomenon and show that clusters begin to form even at earlier layers [8], [14], resulting in a so-called *cascading collapse*. In the supervised setting, each cluster corresponds to a class where the model is trained on, and the mean of the cluster is referred to as simply a *class mean*.

Submission date: January 24, 2024. This work was supported in parts by the Army Research Lab (#W911NF2120272), the Army Research Office (#W911NF2410049), and the National Science Foundation (CNS-2148182).

The authors are with the Department of Electrical and Computer Engineering, University of Illinois Chicago, Chicago, IL 60607 USA (e-mails: {agorme2, ekoyuncu}@uic.edu).

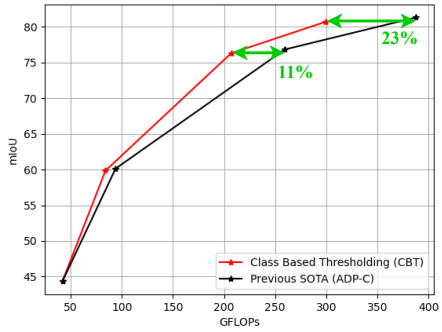

Fig. 1. Comparison of CBT with the previous state-of-the-art on the Cityscapes dataset for HRNetV2-W48 model.

In [8], the authors propose an early exit mechanism utilizing the neural collapse phenomenon, outperforming various existing schemes. Specifically, a representation that is sufficiently close to a class mean at any given layer can be allowed an early exit, without significant penalty in classification performance. However, the idea of using the nearest class mean decision rule is not immediately applicable to the task of semantic segmentation since one now needs to perform pixel-wise classification. In fact, in the image classification task, there is one input and it belongs to one class. Therefore, the representations of the images with the same label can be averaged, and class means can be calculated. The intermediate layer outputs will be close to only one class mean, and a meaningful prediction can be performed based on the distances to the class means [8]. On the contrary, in semantic segmentation, one input has many pixels, each of which belong to different classes. One image has one representation at each layer, and components of a representation corresponds to many pixels. Hence, class means cannot be immediately calculated from the representations for individual pixels for the task of semantic segmentation.

Having too many pixels in an image results in the curse of dimensionality, which presents an additional complication. In fact, even if the class means could be obtained, using the nearest class mean decision for the pixels would be too costly since there are thousands of pixels in an image. These make it infeasible to calculate the class means for the pixels using existing algorithms (e.g. [8]). Nevertheless, utilizing the neural collapse phenomenon for semantic segmentation would be particularly useful because the amount of computation can be reduced significantly for the state-of-the-art semantic segmentation models [15]–[23].

We propose "Class Based Thresholding (CBT)", a novel algorithm that reduces the computational cost while preserving the model performance for the semantic segmentation task. Leveraging the neural collapse phenomenon, CBT calculates

This article has been accepted for publication in IEEE Signal Processing Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LSP.2024.3386110

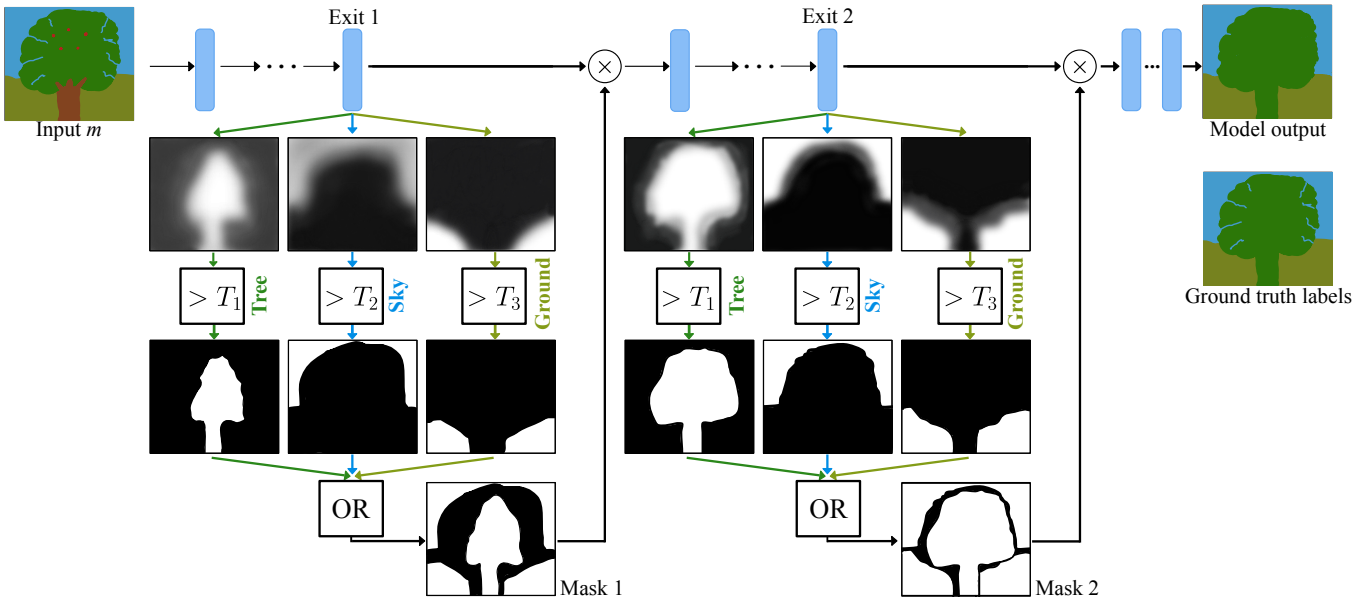

Fig. 2. Overview of our Class Based Thresholding (CBT) scheme at the inference time for an example network with $N = 2$ exit layers and $K = 3$ classes: Tree, ground, and sky. In contrast to [29] and [30], CBT utilizes different thresholds for different classes, considering their varying levels of inherent difficulty. The thresholds are determined as a function of only two non-trainable hyperparameters, independent of the number of classes or exits, thanks to our neural collapse inspired design. At each exit, the layer output is split into $K = 3$ channels, where each channel corresponds to one of the tree, ground, or sky classes. The channels are then transformed into masks using their corresponding distinct thresholds, and the resulting masks are merged. The methods presented in [29] and [30] thus become a special case of CBT where the thresholds for every class is the same. Mask 1 illustrates the confident (white) pixels after the merger at Exit 1, which is subsequently integrated into the following layers through multiplication. This integration ensures that the model avoids unnecessary computations for these confident pixels in subsequent layers. Exit 2 follows the same mechanism for inference. Mask 2 exhibits a greater number of confident pixels due to the input image passing through layers between Exit 1 and Exit 2. The exit predictions become progressively better.

the mean of the prediction probabilities of pixels in the training set, for each class. Then, the thresholds for each class are calculated via a simple transformation of the class means. These thresholds are then employed to allow the early termination of the computation for confidently predicted pixels at inference time. We show the effectiveness of CBT on the Cityscapes [24], ADE20K [25] and COCO-Stuff-10K [26] datasets using the HRNetV2-W18, HRNetV2-W48 and vision transformer models [27], [28]. By efficiently utilizing the neural collapse phenomenon, CBT can reduce the computational cost by up to 23% compared to the previous state-of-the-art method while preserving the model performance as shown in Fig. 1.

II. Class Based Thresholding

We build on the state-of-the-art early exit semantic segmentation method, "Anytime Dense Prediction with Confidence Adaptivity (ADP-C)" [29]. ADP-C adds early exit layers to the base semantic segmentation model and introduces a masking mechanism based on a single user-specified threshold value t to reduce the computational cost. If a pixel is predicted confidently at an exit layer, i.e., the maximum prediction probability over all classes is greater than the threshold t, that pixel is masked for all subsequent layers. Any masked pixel will not be processed again at later layers. The computational cost is reduced due to the induced feature sparsity. While it is possible to let every pixel exit at the same time [31], this approach performs worse at the boundaries of objects, and therefore we focus on ADP-C and pixel-wise early exiting.

A big room for improvement for ADP-C stems from the observation that the same user-specified threshold value t is used for every class. However, it is more plausible that different threshold values should be used for different classes, and the threshold values should reflect the dataset and class properties, rather than just being a user-specified number. The observation from [32] supports our hypothesis: "The distribution of max logits of each predicted class is significantly different from each other." This is because pixels belonging to different classes have different difficulty levels of being predicted correctly. For example, using $t = 0.998$ for *bicycle* class as in ADP-C makes sense because we may want to be really certain about pixels belonging to bicycles. However, pixels belonging to the *sky* class will be often easier to predict than pixels belonging to the *bicycle* class, which means the model will be confident about them much sooner. Therefore, a lower threshold value can be used for the *sky* class without significant penalty in prediction accuracy. Otherwise, more computation will have to be performed for the *sky* pixels.

Given a model trained on a semantic segmentation task with K classes, we propose using different masking threshold values per class, based on the dataset and class properties. Let $T = [T_1 \cdots T_K] \in [0, 1]^K$ be the threshold vector that we wish to determine, where the k^{th} element T_k corresponds to class k, and $k \in \{1, 2, \ldots, K\}$. Consider M training inputs, each of which have a height of H pixels, and a width of W pixels. Suppose that we utilize N exit layers in the model. The class prediction probabilities provided by the model at exit n for each (m, h, w) triplet can be represented by the function $\phi_n : \mathbb{R}^{M \times H \times W} \rightarrow [0, 1]^K$. Hence, given a pixel at height h, width w of input m, the prediction probabilities for the K classes at exit n are expressed as $\phi_n(m, h, w)$.

Let S_k denote the set of all pixels, or (m, h, w) triplets, whose ground truth is class k. At each exit layer n, for each class k in the training set, we calculate the mean of layer n's prediction probabilities using all training set pixels in S_k. This averaging helps obtaining a broad sense of information about the difficulty of pixels. This yields

$$p_{n,k} \triangleq \frac{1}{|S_k|} \sum_{(m,h,w)\in S_k} \phi_n(m,h,w) \in [0,1]^K. \quad (1)$$

The motivation of averaging in (1) comes from the neural collapse phenomenon, which states feature vectors converge to their average class means as one goes deeper in a network [13]. Indeed, the averages $p_{n,k}$ should empirically be a good estimate for the class probabilities, especially for a deep layer index n. Specifically, the i^{th} element of $p_{n,k}$ denotes the average probability of a pixel belonging to class i when the ground truth for that pixel is class k. Next, we compute

$$P_k = \frac{1}{N} \sum_{n=1}^{N} p_{n,k} \in [0,1]^K, \quad (2)$$

which is the average of $p_{n,k}$ over all layers. Hence, information across layers is shared to obtain a global estimate P_k on the difficulty of classes. The logic for the information sharing across layers is to leverage insights from both shallow and deep layers, and to make the thresholding less complex due to having only one set of thresholds for every exit. Information sharing in CBT can be seen as a naive version of feature reuse in other multi-exit network settings such as [33]–[37].

We then translate the estimates to classification thresholds as follows. We initialize the threshold T_k to be the difference between the largest and the second largest elements of P_k, because this initialization strategy has been shown to be a reliable confidence score, effectively capturing the importance of the most dominant element relative to the second-largest one [8], [9]. If the confidence score is high, then the masking threshold should be low so that the computation can terminate easily. After all components of T are initialized in this manner, we inversely scale T according to two non-trainable parameters α and β so that the maximum and minimum class confidence scores determined by T will be converted to masking threshold values α and β respectively, where $\alpha < \beta$. The rationale behind this inverse scaling is to guarantee that classes with high confidence scores will have low thresholds and vice versa. Specifically, the scaling is done via

$$T_k \leftarrow \left(1 - \frac{T_k - \min T}{\max T - \min T}\right)(\beta - \alpha) + \alpha. \quad (3)$$

The inference is performed as follows: Let $\pi \in [0,1]^K$ be the prediction probabilities for a pixel at an exit layer. Let $j = \arg\max \pi$. If $\pi_j > T_j$, this pixel will be marked as confidently predicted (predicted as class j) and will be incorporated to the mask M as in Fig. 2. By doing so, the outputs of subsequent layers at these locations will not be calculated. Instead, already computed values will be used. Note that once calculated, T is not updated in inference.

III. Experiments and Results

We compare CBT against ADP-C [29] and DToP [30]. ADP-C and DToP allow early prediction of pixels, but they use the same thresholds for all classes. We use Cityscapes, ADE20K, COCO-Stuff-10K datasets [24]–[26], and HRNetV2-W18, HRNetV2-W48, ViT models for evaluation [27], [28]. We use mean intersection over union (mIoU) as our performance metric and number of floating point operations (FLOPs) as our computational cost metric. We attach 3 early exit layers to HRNet models as in [29] and 2 to ViT models as in [30] with the same exit structures and positions. The training is done by using the weighted sum of the exit losses. We assign the same weight of 1 to exit losses.

We have evaluated CBT with numerous α-β pairs (denoted as CBT $[\alpha, \beta]$). We kept $\beta = 0.998$ for HRNet models, $\beta = 0.9$ for ViT-Base, and $\beta = 0.95$ for ViT-Large in our experiments for a fair comparison because ADP-C and DToP achieve the best performance with these values. For comparison purposes, we also included $\beta = 0.9$ for ADP-

TABLE I
Results on Cityscapes.

Method	Model	Exit 1		Exit 2		Exit 3		Exit 4	
		mIoU	GFLOPs	mIoU	GFLOPs	mIoU	GFLOPs	mIoU	GFLOPs
MDEQ [38]	S	17.3	521.6	38.7	717.9	65.6	914.2	72.4	1110.5
ADP-C	HRNetV2-W48	44.34	41.92	60.13	93.90	76.82	259.33	68.55	387.80
CBT [0.99, 0.998]		44.34	41.92	59.85	84.02	76.29	206.89	80.69	299.10
CBT-ns [0.99, 0.998]		44.34	41.92	59.82	84.00	76.28	206.88	80.74	299.17
CBT [0.95, 0.998]		44.34	41.92	57.97	71.57	72.86	155.77	76.60	222.65
CBT [0.9, 0.998]		44.34	41.92	56.05	65.91	68.92	132.49	72.29	186.31
ADP-C $\beta = 0.9$		44.34	41.92	54.86	53.27	67.10	118.25	69.31	157.48
ADP-C	HRNetV2-W18	40.83	23.68	48.19	33.27	68.26	45.40	77.02	58.90
CBT [0.99, 0.998]		40.83	23.68	48.07	31.74	67.98	41.40	76.57	51.26
CBT [0.95, 0.998]		40.83	23.68	46.97	29.51	64.88	36.25	71.18	43.35
CBT [0.9, 0.998]		40.83	23.68	45.79	28.39	61.32	33.72	67.45	39.42

TABLE II
Results on ADE20K.

Method	Model	Exit 1		Exit 2		Exit 3		Exit 4	
		mIoU	GFLOPs	mIoU	GFLOPs	mIoU	GFLOPs	mIoU	GFLOPs
ADP-C	HRNetV2-W48	4.12	6.20	5.16	15.42	12.15	52.47	42.82	100.28
CBT [0.9, 0.998]		4.12	6.20	5.15	15.07	12.09	50.48	41.85	94.31
CBT-ns [0.9, 0.998]		4.12	6.20	5.15	15.06	12.08	50.48	41.87	94.34
CBT [0.8, 0.998]		4.12	6.20	5.14	14.80	11.90	48.81	40.17	90.25
CBT [0.7, 0.998]		4.12	6.20	5.12	14.55	11.58	47.27	37.54	86.52
ADP-C	HRNetV2-W18	4.89	5.88	6.83	7.84	8.94	12.73	9.74	19.04
CBT [0.9, 0.998]		4.89	5.88	6.80	7.73	10.07	12.24	11.78	17.89
CBT [0.8, 0.998]		4.89	5.88	6.75	7.67	10.17	11.98	11.95	17.26
CBT [0.7, 0.998]		4.89	5.88	6.70	7.62	10.09	11.75	11.88	16.71

TABLE III
Comparison of CBT against Dynamic Token Pruning (DToP).

Method	Dataset	Model	Exit 1		Exit 2		Exit 3	
			mIoU	GFLOPs	mIoU	GFLOPs	mIoU	GFLOPs
DToP	ADE20K	ViT-Base	41.79	55.70	45.85	66.60	49.21	83.52
CBT [0.85, 0.9]			41.79	55.70	45.52	65.60	49.04	80.80
DToP		ViT-Large	37.86	208.96	47.97	352.32	52.18	452.3
CBT [0.9, 0.95]			37.86	208.96	47.82	336.01	51.69	421.93
DToP	COCOStuff10K		31.89	124.94	41.71	205.14	45.64	266.17
CBT [0.9, 0.95]			31.89	124.94	41.09	197.53	45.29	252.04

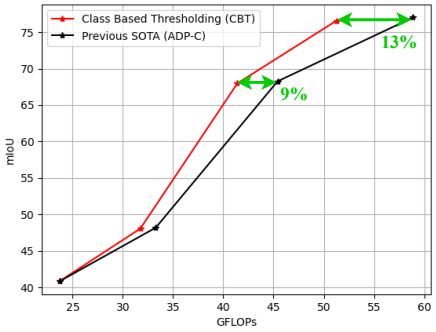

Fig. 3. Comparison of CBT with the previous state-of-the-art on the Cityscapes dataset for HRNetV2-W18 model.

C in Table I. Naturally, it has the lowest mIoU and GFLOPs because all classes use the same threshold of 0.9, the lowest among the experiment settings. As shown in Tables I, II and III, lower α facilitates pixels exiting early, and increasing it results in more confident predictions.

By Table I, CBT $[0.99, 0.998]$ decreases the computational cost by 23% while losing only 0.62 mIoU for HRNetV2-W48. For Exits 2 and 3, the computational cost is decreased by 10% and 20% respectively. By using smaller α, the computational cost can be decreased more, but mIoU starts degrading as well. Note that Exit 4 of CBT $[0.95, 0.998]$ can match the performance of Exit 3 of ADP-C while using 14% less computation. For HRNetV2-W18, the results follow the same trend: CBT $[0.99, 0.998]$ decreases the computational cost by 9% and 13% for exits 3 and 4 respectively as seen in Fig. 3.

According to Table II, we observe that CBT can also reduce the computational cost on the ADE20K dataset, which has significantly more classes as compared to the Cityscapes datasets. Specifically, CBT $[0.90, 0.998]$ decreases the computational cost by 6% while losing only 0.97 mIoU for HRNetV2-W48. The reason why the performances at the first three exit is low for both ADP-C and CBT is because the model cannot perform well enough due to large number of classes. It needs significantly more computation (e.g. 94.31 GFLOPs instead of 15.07, also seen in Table III with ViT) to have better performance. Also, this is why CBT cannot reduce the computational cost on ADE20K as much as it does on Cityscapes with HRNet models.

In Fig. 4, we illustrate CBT-calculated class thresholds for Cityscapes and ADE20K datasets. Due to the ADE20K dataset's large number of classes, only the 19 classes with the lowest thresholds are displayed for both datasets. For Cityscapes, with a total of 19 classes, we exhibit all class thresholds. Compared to ADE20K dataset, class thresholds are spread out more uniformly between $\alpha = 0.9$ and $\beta = 0.998$) for Cityscapes dataset ($\sigma = 0.033$). For ADE20K dataset on the other hand ($\sigma = 0.009$), the behavior is different: Most class thresholds lie between 0.997 and 0.998. This supports our observation that CBT can reduce the computational cost more when the number of classes is relatively low. We can also observe that for both datasets, simple classes such as "sky" have low thresholds, while more complex classes have typically have higher thresholds values, as expected.

Fig. 5 shows the relationship between different thresholds

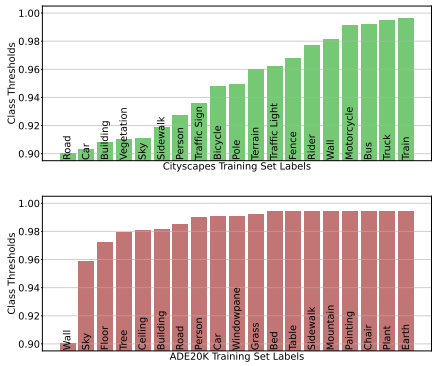

Fig. 4. CBT thresholds for Cityscapes and ADE20K ($\alpha = 0.9, \beta = 0.998$).

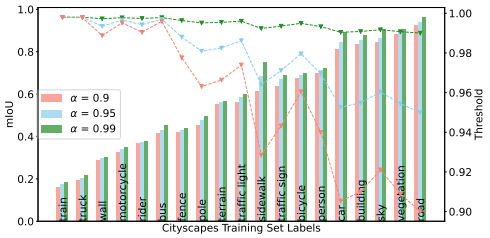

Fig. 5. Class-wise mIoU performances for HRNetV2-W48 at Exit 2 with various α values ($\beta = 0.998$). Dashed lines indicate the corresponding class thresholds.

and their corresponding class-wise mIoU performances. When a lower α is used, class-wise mIoU performances drop slightly, in line with the results in Tables I, II and III. The "sidewalk" and "car" classes are affected the most with the change of their thresholds. For easier, high-mIoU classes, the performance drop is not drastic, suggesting the effectiveness of CBT.

A. Ablation Study

CBT calculates the average of $p_{n,k}$ over all exits as in (2) to obtain one single vector P_k, which is later scaled to obtain the thresholds. This allows the information across the layers to be shared and reduces the number of total thresholds from $N \times K$ to K. Here, we disable the information sharing by not averaging and allowing each exit to have its own thresholds based on its $p_{n,k}$. This prevents information flow from deeper exits to shallower exits. Note that this is a more complex method due to having more thresholds. We include the results in Tables I and II only for the highest α values and denote by *CBT-ns*, due to lack of space. As seen from the numbers, there is no significant difference between CBT and CBT-ns, meaning the more complex CBT-ns is not superior to CBT.

IV. Conclusion

We proposed a novel algorithm that utilizes the naturally occurring neural collapse phenomenon to reduce the computational cost of early exit semantic segmentation models. Experiment results on different datasets and models suggest our method is effective in reducing the computational cost without significant penalty in model performance. We note that the ideas developed in this paper can be applied to multi-modal data, which inherently have different requirements for processing complexity [39]–[42]. In particular, the thresholds used for a neural network classifying text data should be different than the thresholds used for the image data. A detailed study is left as future work.

REFERENCES

[1] J. Sevilla, L. Heim, A. Ho, T. Besiroglu, M. Hobbhahn, and P. Villalobos, "Compute trends across three eras of machine learning," 2022. [Online]. Available: https://arxiv.org/abs/2202.05924

[2] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin *et al.*, "Opt: Open pre-trained transformer language models," *arXiv preprint arXiv:2205.01068*, 2022.

[3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[4] P. Li, H. Seferoglu, and E. Koyuncu, "Model distributed inference in multi-source edge networks," in *IEEE ICASSP Workshop on Timely and Private Machine Learning over Networks*, Jun. 2023.

[5] P. Li, E. Koyuncu, and H. Seferoglu, "Adaptive and resilient model-distributed inference in edge computing systems," *IEEE Open Journal of the Communications Society*, vol. 4, pp. 1263–1273, 2023.

[6] S. Teerapittayanon, B. McDanel, and H.-T. Kung, "Branchynet: Fast inference via early exiting from deep neural networks," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 2464–2469.

[7] P. Panda, A. Sengupta, and K. Roy, "Conditional deep learning for energy-efficient and enhanced pattern recognition," in *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2016, pp. 475–480.

[8] A. Görmez, V. R. Dasari, and E. Koyuncu, "E2cm: Early exit via class means for efficient supervised and unsupervised learning," in *2022 International Joint Conference on Neural Networks (IJCNN)*, 2022, pp. 1–8.

[9] Y. Kaya, S. Hong, and T. Dumitras, "Shallow-deep networks: Understanding and mitigating network overthinking," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3301–3310.

[10] E. Koyuncu, "Memorization capacity of neural networks with conditional computation," in *International Conference on Learning Representations*, 2023.

[11] A. Görmez and E. Koyuncu, "Pruning early exit networks," in *Workshop on Sparsity in Neural Networks*, Jul. 2022.

[12] A. Görmez and E. Koyuncu, "Dataset pruning using early exit networks," in *ICML Localized Learning Workshop*, Jul. 2023.

[13] V. Papyan, X. Han, and D. L. Donoho, "Prevalence of neural collapse during the terminal phase of deep learning training," *Proceedings of the National Academy of Sciences*, vol. 117, no. 40, pp. 24 652–24 663, 2020.

[14] L. Hui, M. Belkin, and P. Nakkiran, "Limitations of neural collapse for understanding generalization in deep learning," *arXiv preprint arXiv:2202.08384*, 2022.

[15] G. Li, L. Li, and J. Zhang, "Hierarchical semantic broadcasting network for real-time semantic segmentation," *IEEE Signal Processing Letters*, vol. 29, pp. 309–313, 2021.

[16] G. Zhang, J.-H. Xue, P. Xie, S. Yang, and G. Wang, "Non-local aggregation for rgb-d semantic segmentation," *IEEE Signal Processing Letters*, vol. 28, pp. 658–662, 2021.

[17] Y. Huang, Z. Tang, D. Chen, K. Su, and C. Chen, "Batching soft iou for training semantic segmentation networks," *IEEE Signal Processing Letters*, vol. 27, pp. 66–70, 2019.

[18] Y. Li, X. Li, C. Xiao, H. Li, and W. Zhang, "Eacnet: Enhanced asymmetric convolution for real-time semantic segmentation," *IEEE signal processing letters*, vol. 28, pp. 234–238, 2021.

[19] G. Li, L. Li, and J. Zhang, "Biattnet: bilateral attention for improving real-time semantic segmentation," *IEEE Signal Processing Letters*, vol. 29, pp. 46–50, 2021.

[20] Y. Yue, W. Zhou, J. Lei, and L. Yu, "Two-stage cascaded decoder for semantic segmentation of rgb-d images," *IEEE Signal Processing Letters*, vol. 28, pp. 1115–1119, 2021.

[21] E. Yang, W. Zhou, X. Qian, and L. Yu, "Mgcnet: Multilevel gated collaborative network for rgb-d semantic segmentation of indoor scene," *IEEE Signal Processing Letters*, vol. 29, pp. 2567–2571, 2022.

[22] I. Krešo, J. Krapac, and S. Šegvić, "Efficient ladder-style densenets for semantic segmentation of large images," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 4951–4961, 2020.

[23] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 1290–1299.

[24] 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 *IEEE CVPR*, June 2016.

[25] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *IEEE CVPR*, 2017.

[26] H. Caesar, J. Uijlings, and V. Ferrari, "Coco-stuff: Thing and stuff classes in context," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1209–1218.

[27] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3349–3364, 2020.

[28] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[29] Z. Liu, Z. Xu, H.-J. Wang, T. Darrell, and E. Shelhamer, "Anytime dense prediction with confidence adaptivity," *International Conference on Learning Representations (ICLR)*, 2022.

[30] Q. Tang, B. Zhang, J. Liu, F. Liu, and Y. Liu, "Dynamic token pruning in plain vision transformers for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 777–786.

[31] A. Kouris, S. I. Venieris, S. Laskaridis, and N. Lane, "Multi-exit semantic segmentation networks," in *European Conference on Computer Vision*. Springer, 2022, pp. 330–349.

[32] S. Jung, J. Lee, D. Gwak, S. Choi, and J. Choo, "Standardized max logits: A simple yet effective approach for identifying unexpected road obstacles in urban-scene segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 425–15 434.

[33] L. Yang, H. Jiang, R. Cai, Y. Wang, S. Song, G. Huang, and Q. Tian, "Condensenet v2: Sparse feature reactivation for deep networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3569–3578.

[34] G. Huang, D. Chen, T. Li, F. Wu, L. Van Der Maaten, and K. Q. Weinberger, "Multi-scale dense networks for resource efficient image classification," *arXiv preprint arXiv:1703.09844*, 2017.

[35] L. Yang, Y. Han, X. Chen, S. Song, J. Dai, and G. Huang, "Resolution adaptive networks for efficient inference," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2369–2378.

[36] M. Phuong and C. H. Lampert, "Distillation-based training for multi-exit architectures," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1355–1364.

[37] D. Zhang, H. Zhang, J. Tang, X.-S. Hua, and Q. Sun, "Self-regulation for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6953–6963.

[38] S. Bai, V. Koltun, and J. Z. Kolter, "Multiscale deep equilibrium models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5238–5250, 2020.

[39] K. Liu, Y. Li, N. Xu, and P. Natarajan, "Learn to combine modalities in multimodal deep learning," *arXiv preprint arXiv:1805.11730*, 2018.

[40] A. M. Shervedani, S. Li, N. Monaikul, B. Abbasi, B. Di Eugenio, and M. Zefran, "An end-to-end human simulator for task-oriented multimodal human-robot collaboration," *arXiv preprint arXiv:2304.00584*, 2023.

[41] X. Chen, X. Wang, L. Beyer, A. Kolesnikov, J. Wu, P. Voigtlaender, B. Mustafa, S. Goodman, I. Alabdulmohsin, P. Padlewski *et al.*, "Pali-3 vision language models: Smaller, faster, stronger," *arXiv preprint arXiv:2310.09199*, 2023.

[42] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 689–696.