CHASE: Channel-Wise and Spatial Attention for Early Exiting in Image Classification

Youva Addad, Alexis Lechervy, Frédéric Jurie
GREYC, Normandy University, UNICAEN, ENSICAEN, UMR CNRS 6072, France
first_name.last_name@unicaen.fr

***Abstract*—Dynamic early-exiting neural networks have been proposed for image classification to balance the trade-off between classification performance and inference cost. In this context, we propose a multi-exit neural network architecture that exploits the power of attention mechanisms, which improve performance but incur significant computational overhead. In CHASE, we introduce two attention-like mechanisms to go beyond existing multi-exit architectures. The first mechanism dynamically adjusts the importance of different feature channels and spatial locations, recalibrating channel-wise feature responses. The second mechanism, based on self-attention, aggregates features from different spatial locations at the end of the network. We evaluate the proposed architecture on the CIFAR and ImageNet datasets, comparing it with the original network and other state-of-the-art approaches. Our results show that the proposed architecture achieves competitive performance in terms of accuracy and computational efficiency.**

I. Introduction and related work

Deep learning has revolutionized computer vision through architectures like EfficientNet [1], [2], ResNet [3], and DenseNet [4], achieving state-of-the-art performance. However, increasing model complexity for higher accuracy raises computational demands. While attention mechanisms [5], [6] enhance feature focus, they increase computational costs.

Recent research has explored the integration of CNNs and attention mechanisms to create hybrid architectures that exploit the strengths of both approaches. CNNs excel at extracting local features and spatial hierarchies, while attention mechanisms are powerful at capturing long-range dependencies and focusing on relevant features. However, attention mechanisms can be computationally expensive due to their quadratic complexity with respect to the input sequence length. Several notable hybrid architectures have been proposed, such as LeViT [7], which presents a Conv-like design to speed up vision transformers, although Multi-Head Self-Attention (MHSA) remains computationally intensive on edge resources. EfficientFormer [8] introduces convolutional processes in the early stages and maintains attention in the final stages, while Mobile-Former [9] incorporates a parallel design of MobileNet and Transformer with a two-way bridge in between.

Dynamic early exit networks have emerged as a promising solution to balance performance and efficiency. These networks introduce auxiliary classifiers or ”exits” at different network depths, allowing adaptive inference by stopping the process earlier when the desired confidence level is reached. This approach reduces the overhead, especially for simpler samples. Key developments in dynamic early exit include BranchyNet [10], which introduced the concept of attaching classifier heads at different depths within the backbone, and MSDNet [11], which uses dense connections and a multiscale structure to mitigate interference between classifiers. RANet [12] performs early exiting first on low resolution features and then on high resolution features, while DVT [13] and CF-ViT [14] use full attention mechanisms and start early exiting with a small number of tokens and gradually increase to more tokens. Dyn-Perceiver [15] decouples the backbone and classifier to solve the interference problem, and L2W [16] proposes to assign a weight to each sample and train the whole network using meta-learning. Recent post-hoc methods such as Calibrated-DNN [17] and EENet [18] offer sophisticated ways to compute and calibrate final scores, ensuring proper calibration when choosing which classifier to exit.

While the literature on dynamic early exit networks has evolved with advances in architectures, sample importance assignment, and post-hoc methods, the integration of attention mechanisms has been limited and often restricted to classifier branches or a few exits due to computational cost. In this work, we propose a novel framework that synergistically combines CNNs and attention mechanisms to improve the performance and efficiency of dynamic early exit networks. By integrating these complementary components, we aim to balance local feature extraction with global context awareness. Our contributions include a novel architecture that fuses CNNs with attention mechanisms to enhance performance, a computationally efficient design that judiciously positions self-attention in the final stage while using squeeze and excitation for channel weighting in earlier stages, and extensive experimentation and validation on the CIFAR and ImageNet datasets to demonstrate the effectiveness and competitiveness of our proposed architecture with other dynamic neural networks.

II. CHASE’s Architecture

The proposed model, CHASE, builds upon the fundamental structure of MSDNet [11], a multi-scale, multi-output image classification model. It begins with a stem layer, whose role is to process images by transforming the initial RGB representation into a more compact yet informative representation.

This initial reduction in image size decreases the total number of FLOPs required for subsequent processing. We adopt the stem layer design from Addad [19], which demonstrated its efficiency and ability to enhance representativeness. The stem layer consists of four sequential convolutional layers, including

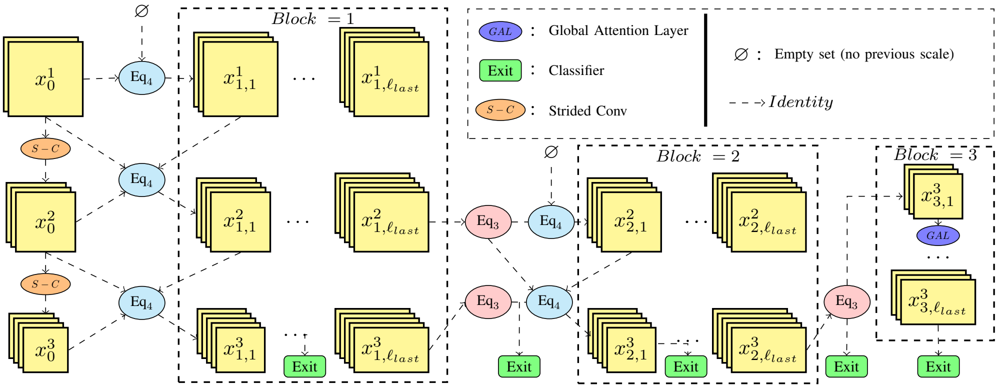

Fig. 1: Illustration of the proposed architecture, with $S = 3$ scales. It consists of 2 convolution blocks and an attention block, with a reduction in the number of scales within each successive convolution block. x_0^1 is the feature map from the stem.

$x_{b,\ell}^s$	$\ell = 1$	$\ell = 2$	$\ell = 3$	$\ell = 4$
$s = 1$	$h_{b,1}^1(x_{b,0}^1)$	$h_{b,2}^1(x_{b,0}^1 \oplus x_{b,1}^1)$	$h_{b,3}^1(x_{b,0}^1 \oplus x_{b,1}^1 \oplus x_{b,2}^1)$	$h_{b,4}^1(x_{b,0}^1 \oplus x_{b,1}^1 \oplus x_{b,2}^1 \oplus x_{b,3}^1)$
$s = 2$	$f_{b,1}^2(x_{b,0}^1 \oplus x_{b,1}^1)$ $\oplus$ $h_{b,1}^2(x_{b,0}^2)$	$f_{b,2}^2(x_{b,0}^1 \oplus x_{b,1}^1 \oplus x_{b,2}^1)$ $\oplus$ $h_{b,2}^2(x_{b,0}^2 \oplus x_{b,1}^2)$	$f_{b,3}^2(x_{b,0}^1 \oplus x_{b,1}^1 \oplus x_{b,2}^1 \oplus x_{b,3}^1)$ $\oplus$ $h_{b,3}^2(x_{b,0}^2 \oplus x_{b,1}^2 \oplus x_{b,2}^2)$	$f_{b,4}^2(x_{b,0}^1 \oplus x_{b,1}^1 \oplus x_{b,2}^1 \oplus x_{b,3}^1 \oplus x_{b,4}^1)$ $\oplus$ $h_{b,4}^2(x_{b,0}^2 \oplus x_{b,1}^2 \oplus x_{b,2}^2 \oplus x_{b,3}^2)$
$s = 3$	$f_{b,1}^3(x_{b,0}^2 \oplus x_{b,1}^2)$ $\oplus$ $h_{b,1}^3(x_{b,0}^3)$	$f_2^3(x_{b,0}^2 \oplus x_{b,1}^2 \oplus x_{b,2}^2)$ $\oplus$ $h_{b,2}^3(x_{b,0}^3 \oplus x_{b,1}^3)$	$f_{b,3}^3(x_{b,0}^2 \oplus x_{b,1}^2 \oplus x_{b,2}^2 \oplus x_{b,3}^2)$ $\oplus$ $h_{b,3}^3(x_{b,0}^3 \oplus x_{b,1}^3 \oplus x_{b,2}^3)$	$f_{b,4}^3(x_{b,0}^2 \oplus x_{b,1}^2 \oplus x_{b,2}^2 \oplus x_{b,3}^2 \oplus x_{b,4}^2)$ $\oplus$ $h_{b,4}^3(x_{b,0}^3 \oplus x_{b,1}^3 \oplus x_{b,2}^3 \oplus x_{b,3}^3)$

TABLE I: The different feature map computed in block b that make up our architecture. Scale 1 can be seen as a ”fusion” with scale 0, which consists of empty elements.

strided convolutions, depth-wise convolutions, and a residual connection. It generates lower-resolution representations that are then fed into the subsequent layers of the network.

Regarding the subsequent layers, CHASE follows principles common to most previous works, [11], [12], [19], consisting of a repetition of identical blocks as illustrated in Figure 1. After each block, the network can either exit by passing the computed representations through a classifier or transmit the representations to the next block, allowing for dynamic adjustment of the network's depth. Each block refines the representation for the classification task and adds a new exit. Each block operates on multi-scale feature map denoted as x_b^s, where s represents scale, and b stands for block number. x_0^1 refers to the feature map derived from the stem. These blocks contain intermediate layers indexed by ℓ, with $x_{b,\ell}^s$ representing the feature map of scale s in block b inputted to a layer at depth ℓ within that block. The layers at depth ℓ are structured into two configurations: $f_{b,\ell}^s(.)$ and $h_{b,\ell}^s(.)$, as depicted in Figure 2 (a). $f_{b,\ell}^s(.)$ divides the spatial resolution by 2, while $h_{b,\ell}^s(.)$ keeps it, allowing the scales to be aligned for fusion. The fusion stage in CHASE is the same as in [19]. This design facilitates the transfer of information from higher to lower resolutions, preserving semantic information as the network's depth grows. The fusion operation, depicted in Eq. 1, encapsulates this process.

$$x_{b,\ell}^s = f_{b,\ell}^s\left(\bigoplus_{i=0}^{\ell} x_{b,i}^{s-1}\right) \oplus h_{b,\ell}^s\left(\bigoplus_{i=0}^{\ell-1} x_{b,i}^s\right), \tag{1}$$

where $\oplus$ is the feature map concatenation operator along the channel axis. Employing Eq. 1 adheres to the conventions where $f(\varnothing) = h(\varnothing) = \varnothing$ and $x \oplus \varnothing = x$. Table I illustrates an example of how to calculate the feature map for a block b consisting of 3 scales and 4 layers.

Between the block b and $b+1$, the maximum scale of feature map is discarded. This is achieved as follow:

$$\forall b, s, \ell, \text{with } s < b, \quad x_{b,\ell}^s = \varnothing, \tag{2}$$

Eq. 1 can be used to define the feature map of other layers, even at scale $s < b$. When the maximum scale of the feature map is discarded, a transition layer T_b is introduced. T_b conducts a 1×1 convolution, followed by BatchNorm and $GELU$ activation. Its role is to reduce the size of the feature map by a specified reduction factor, which is set to 0.5 in this context. This prevents an increase in the size of the feature map. The input of the block $b+1$ is computed as follow:

$$x_{b+1,0}^s = T_b(x_{b,\ell_{\text{last}}}^s) \tag{3}$$

In the following sections, we detail the core of our approach, which involves incorporating self-attention at the lowest resolution and introducing an additional attention mechanism through channel weighting at intermediate resolutions.

A. Global Attention Layer (GAL)

Self-attention mechanisms have proven useful in recent architectures such as Visual Transformers [5], acting as a substitute for convolutional layers. However, self-attention token mixers result in notable computational overhead, especially at high resolutions. This complexity is expressed as $\mathcal{O}(HWC^2 + (HW)^2C)$ for a feature map $X \in \mathbb{R}^{H \times W \times C}$, where H, W, and C denote height, width, and channel count, respectively. The first term, HWC^2, encompasses the computational complexity of query, key, and value projections. This term dominates the overall complexity when C is large, which typically occurs in the later blocks of the network. The second term, $(HW)^2C$, pertains to the attention computation itself. This term becomes the dominant factor when H and W are large, which is the case in the initial blocks of the network. It should be noted that in this type of multi-exit architecture, the maximum size of the images decreases exponentially as they pass through the blocks.

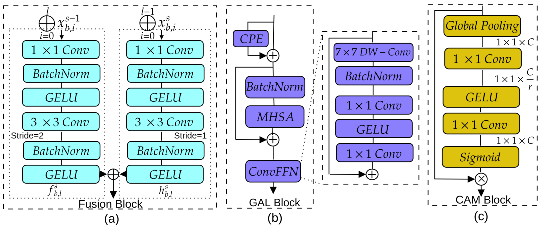

Fig. 2: Illustration of CHASE's components: The 'Fusion Block' module represents the h function or both, with $\bigoplus_{i=0}^{l} x_{b,l+1}^{s-1}$ as an empty set. 'GAL' is the Global Attention Layer, using Conditional Positional Encoding (CPE) for positional embedding. 'CAM' stands for Channel-level Attention Mechanism. Here, s is the current scale, b the current block, and l the current layer.

We address computational efficiency by incorporating attention exclusively in the final block (Fig. 2b), leveraging its benefits while controlling costs. This design routes the most challenging examples—fewer in number—to the network's final stage for enhanced processing. Empirical validation (see Experiments) confirms the approach's efficacy. The Global Attention Layer [20]–[22] dynamically focuses the classifier on critical input regions, using conditional positional encodings [23], [24] generated from local token neighborhoods.

B. Channel-level Attention Mechanism (CAM)

In the previous section, we noted that self-attention is computationally costly for high-resolution feature map. However, we believe that attention is important and aimed to introduce a cheaper mechanism by incorporating attention at the channel level of the feature map, which has a much lower computational cost. This channel weighting mechanism is integrated into the convolution layers for high-resolution feature map. We achieve this by incorporating a squeeze and excitation layer, as proposed by [25]. Figure 2 (c) illustrates the composition of Channel-level Attention Mechanism (CAM). It involves a sequence of transformations, including global pooling, followed by a 1×1 convolution (with GELU activation) without batch normalization, which reduces the number of channels by a factor of r (we find $r = 2$ to be effective). This is followed by another 1×1 convolution (without GELU activation or batch normalization). The CAM layer ends with a sigmoid activation. This process culminates in a channel-wise recalibration, achieved through element-wise multiplication, emphasizing the most informative channels while suppressing the less relevant ones. Consequetely, we replace the Eq. 1, immediately preceding the fusion in layers $f_{b,\ell}^{s}$ and $h_{b,\ell}^{s}$, by Eq. 4.

$$x_{b,\ell}^{s} = f_{b,\ell}^{s}\left(CAM\left(\bigoplus_{i=0}^{\ell} x_{b,i}^{s-1}\right)\right) \oplus h_{b,\ell}^{s}\left(CAM\left(\bigoplus_{i=0}^{\ell-1} x_{b,i}^{s}\right)\right), \quad (4)$$

The fusion layer amalgamates features from various scales, resulting in a more intricate representation that encompasses multiple abstraction levels. However, as elucidated in the introduction, not all channels or features contribute equally to the final representation in many instances. The primary objective of the channel compression layer is to dynamically reassign weights to and decorrelate each channel. This ensures that the most pertinent features are accentuated, while less significant ones are downplayed.

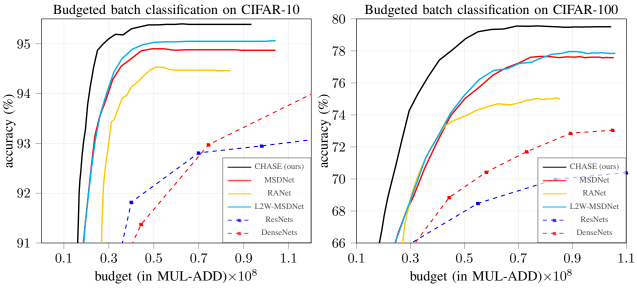

Fig. 3: Accuracy (top-1) of budgeted batch classification models as a function of average computational budget per image on CIFAR-10 (left) and CIFAR-100 (right).

C. Classifiers and loss function

Each exit k of the network has its own classifier g_k, comprising two convolutional layers followed by an average pooling layer and a fully connected layer. For the final output with an attention layer, the layer output is reshaped into a 2D feature map before being sent to the classifier.

The loss function $\mathcal{L}$, which is optimized during the training of the network, combines the loss functions of the individual exits, as follows:

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} \mathrm{CE}\left(g_k(x_{b_k,\ell_k}^{s_{\max}}), y_n\right) \quad (5)$$

Where CE represents the cross-entropy loss, N stands for the total number of training samples, K denotes the number of exits, g_k refers to the k^{th} classifier, and y_n represent the ground truth label of the n^{th} sample. The loss function gives equal importance to each g_k. The K exits are placed every κ layer of the B blocks, with the input for the k^{th} classifier denoted as $x_{b_k,\ell_k}^{s_{\max}}$ (with the dependency on the n^{th} example omitted for notation simplicity). $s_{\max}$ corresponds to the smallest resolution. Based on the decomposition of the total layer count into the base ℓ_{last}, we can derive that $b_k = \left\lceil \frac{(k-1)\kappa}{\ell_{last}} \right\rceil$ and $\ell_k = (k \times \kappa)$ modulo ℓ_{last}. The symbol $\lceil \cdot \rceil$ denotes the ceiling function, which rounds up to the nearest integer. If an exit occurs between two blocks, the input for the classifier comes after the transition layer. The last classifier uses the output of the last layer $x_{B,\ell_{last}}^{s_{\max}}$.

III. Experiments

A. Experiments with CIFAR

The results for Budgeted Batch Classification on CIFAR-10 and CIFAR-100 are presented in Fig. 3, highlighting performance in identifying optimal models across various computational budgets. Accuracy is computed on the test set, resulting in plotted curves for MSDNet [11], RANet [12], L2W-MSDNet [16], and our model. Our method shows better performance than L2W-MSDNet [16]. Nevertheless,

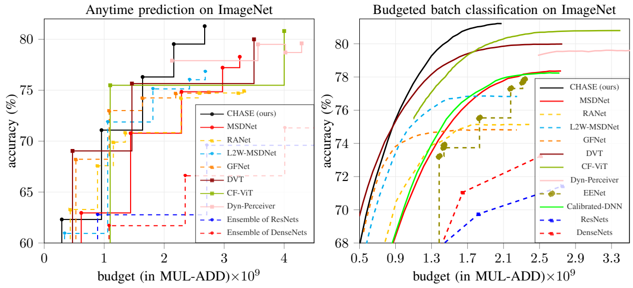

Fig. 4: ImageNet top-1 accuracy as a function of computational budget. Left: anytime prediction. Right: budgeted batch classification.

for budgets exceeding 3×10^7 FLOPs, our method clearly outperforms L2W-MSDNet [16] in CIFAR-100. Additionally, for budgets exceeding 2×10^7 FLOPs, our model outperforms L2W-MSDNet [16] in CIFAR-10. To achieve 95% accuracy on CIFAR-10, our CHASE requires only 2.7×10^7 FLOPs, and on CIFAR-100 it achieves 79% accuracy with only 5.5×10^7 FLOPs. In addition, CHASE achieves a superior level of accuracy comparable to MSDNet [11] while operating within a constrained budget of only 2.5×10^7 FLOPs for CIFAR-10 and 3.6×10^7 FLOPs for CIFAR-100. It also achieves the highest accuracy comparable to RANet [12] using only 2.8×10^7 FLOPs.

B. Experiments with ImageNet

Anytime prediction Experiments. Performance in the anytime prediction setting is given on the left side of Fig. 4. Our approach achieves 81.29% accuracy with 2.6×10^9 FLOPs, outperforming all methods, including the penultimate exit's 79.53% at 2.15×10^9 FLOPs. Compared to MSDNet [11], RANet [12], GFNet [26], and L2W-MSDNet [16], our model surpasses their final classifiers with over 34% fewer FLOPs than MSDNet and up to 50% fewer than RANet and GFNet. Against L2W-MSDNet, it achieves 76.30% with 40% fewer FLOPs. For transformer-based models, it performs comparably to DVT [27] with 38% fewer FLOPs and outperforms CF-ViT [14] using 33% fewer FLOPs. Similarly, it matches Dyn-Perceiver [15]'s 79.6% accuracy with 50% fewer FLOPs.

Budgeted Batch Classification Experiments. Results are shown on the right-hand side of Fig. 4. With a budget of 2.0×10^9 FLOPs, CHASE achieves 81.22% accuracy, outperforming competitors. At 1.0×10^9 and 0.6×10^9 FLOPs, it achieves 77.57% and 71%, respectively. Compared to convolutional models like MSDNet [11], RANet [12], GFNet [26], and L2W-MSDNet [16], CHASE achieves superior accuracy with fewer FLOPs: a 57% reduction compared to MSDNet, 62% to RANet, 56% to GFNet, and 45% to L2W-MSDNet. For transformer-based models, CHASE surpasses DVT [27] for budgets over 0.9×10^9 FLOPs, using 55% fewer FLOPs. Below 0.9×10^9, DVT shows slight superiority. CHASE outperforms CF-ViT [14] with 51% fewer FLOPs and exceeds Dyn-Perceiver's performance with nearly 60% less FLOPs.

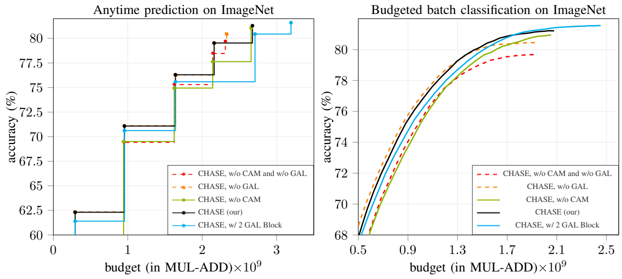

Fig. 5: Top-1 accuracy when GAL (Global Attention Layer) or CAM (Channel-level Attention Mechanism) are removed from our model.

Combining MSDNet [11] with EENet [18] or Calibrated-DNN [17] yields methods that perform equal to or worse than the original MSDNet, with no significant improvement observed.

C. Ablation Study on ImageNet

Fig. 5 showcases the performance of CHASE after the removal of the CAM layer and Global Attention layer, there is a decline of more than 1.5% in performance for an identical budget, resulting in an accuracy of 79.68% with 1.91×10^9 FLOPs. In contrast, when these layers are included, CHASE achieves an accuracy of 81.22% with a computational cost of just 2×10^9 FLOPs, and it attains 62.35% accuracy with a significantly reduced cost of 0.3×10^9 FLOPs. Exclusively considering the CAM layer results (the orange curve) in a performance decrease of over 0.75% for a budget of 1.94×10^9 FLOPs. The cyan curve in the Fig. 5 represents the results obtained by incorporating an Global Attention Layer into the last scale of the two final blocks. This brings the performance to 81.6% with 3.19×10^9 FLOPs in anytime prediction and 81.56% with 2.43×10^9 in budgeted classification. However, the cost experiences an increase of more than 15% for a relatively small gain of 0.3% on accuracy. Exclusively considering the GAL layer results on the worst performance for FLOPs lower than 1.3×10^9. In comparison to CHASE, accuracy lags behind by approximately 1% to 2% depending on budget.

IV. Conclusions

Our study explored dynamic early exit networks, aiming to optimize computational resources in computer vision tasks. We introduced CHASE, a novel architecture blending CNNs with attention mechanisms for improved efficiency and accuracy. Through rigorous evaluation on ImageNet dataset, CHASE demonstrated competitive performance compared to existing dynamic neural network architectures.

Acknowledgments. Research reported in this paper was supported by the Normandy Region, the ANR under award number ANR-19-CHIA-0017 and was performed using computing resources of CRIANN.

References

[1] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *ICML*, 2019.

[2] ——, "Efficientnetv2: Smaller models and faster training," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 10 096–10 106.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[4] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[6] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[7] B. Graham, A. El-Nouby, H. Touvron, P. Stock, A. Joulin, H. Jégou, and M. Douze, "Levit: a vision transformer in convnet's clothing for faster inference," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 2021, pp. 12 239–12 249.

[8] Y. Li, G. Yuan, Y. Wen, J. Hu, G. Evangelidis, S. Tulyakov, Y. Wang, and J. Ren, "Efficientformer: Vision transformers at mobilenet speed," *Advances in Neural Information Processing Systems*, vol. 35, pp. 12 934–12 949, 2022.

[9] Y. Chen, X. Dai, D. Chen, M. Liu, X. Dong, L. Yuan, and Z. Liu, "Mobile-former: Bridging mobilenet and transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 5270–5279.

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

[11] G. Huang, D. Chen, T. Li, F. Wu, L. van der Maaten, and K. Weinberger, "Multi-scale dense networks for resource efficient image classification," in *International Conference on Learning Representations*, 2018.

[12] L. Yang, Y. Han, X. Chen, S. Song, J. Dai, and G. Huang, "Resolution adaptive networks for efficient inference," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[13] M. Zhu, K. Han, E. Wu, Q. Zhang, Y. Nie, Z. Lan, and Y. Wang, "Dynamic Resolution Network," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 27 319–27 330.

[14] M. Chen, M. Lin, K. Li, Y. Shen, Y. Wu, F. Chao, and R. Ji, "Cf-vit: A general coarse-to-fine method for vision transformer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, 2022.

[15] Y. Han, D. Han, Z. Liu, Y. Wang, X. Pan, Y. Pu, C. Deng, J. Feng, S. Song, and G. Huang, "Dynamic perceiver for efficient visual recognition," Jun. 2023.

[16] Y. Han, Y. Pu, Z. Lai, C. Wang, S. Song, J. Cao, W. Huang, C. Deng, and G. Huang, "Learning to weight samples for dynamic early-exiting networks," in *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XI*. Springer-Verlag, 11 2022, pp. 362–378.

[17] L. Meronen, M. Trapp, A. Pilzer, L. Yang, and A. Solin, "Fixing Overconfidence in Dynamic Neural Networks," Apr. 2023, arXiv:2302.06359 [cs].

[18] F. Ilhan, K. Chow, S. Hu, T. Huang, S. F. Tekin, W. Wei, Y. Wu, M. Lee, R. Kompella, H. Latapie, G. Liu, and L. Liu, "Adaptive deep neural network inference optimization with eenet," in *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV*, 2024.

[19] Y. Addad, A. Lechervy, and F. Jurie, "Multi-exit resource-efficient neural architecture for image classification with optimized fusion block," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2023.

[20] Z. Liu, H. Mao, C. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 11 966–11 976.

[21] P. K. A. Vasu, J. Gabriel, J. Zhu, O. Tuzel, and A. Ranjan, "Fastvit: A fast hybrid vision transformer using structural reparameterization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 5785–5795.

[22] W. Yu, C. Si, P. Zhou, M. Luo, Y. Zhou, J. Feng, S. Yan, and X. Wang, "Metaformer baselines for vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 2, pp. 896–912, 2024.

[23] X. Chu, Z. Tian, Y. Wang, B. Zhang, H. Ren, X. Wei, H. Xia, and C. Shen, "Twins: Revisiting the design of spatial attention in vision transformers," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021.

[24] X. Chu, Z. Tian, B. Zhang, X. Wang, and C. Shen, "Conditional positional encodings for vision transformers," in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[25] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[26] Y. Wang, K. Lv, R. Huang, S. Song, L. Yang, and G. Huang, "Glance and focus: A dynamic approach to reducing spatial redundancy in image classification," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M.-F. Balcan, and H.-T. Lin, Eds., 2020.

[27] Y. Wang, R. Huang, S. Song, Z. Huang, and G. Huang, "Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.