Gramian Attention Heads are Strong yet Efficient Vision Learners

Jongbin Ryu*
Ajou University
jongbinryu@ajou.ac.kr

Dongyoon Han*
NAVER AI Lab
dongyoon.han@navercorp.com

Jongwoo Lim†
Seoul National University
jongwoo.lim@gmail.com

Abstract

*We introduce a novel architecture design that enhances expressiveness by incorporating multiple head classifiers (*i.e.*, classification heads) instead of relying on channel expansion or additional building blocks. Our approach employs attention-based aggregation, utilizing pairwise feature similarity to enhance multiple lightweight heads with minimal resource overhead. We compute the Gramian matrices to reinforce class tokens in an attention layer for each head. This enables the heads to learn more discriminative representations, enhancing their aggregation capabilities. Furthermore, we propose a learning algorithm that encourages heads to complement each other by reducing correlation for aggregation. Our models eventually surpass state-of-the-art CNNs and ViTs regarding the accuracy-throughput trade-off on ImageNet-1K and deliver remarkable performance across various downstream tasks, such as COCO object instance segmentation, ADE20k semantic segmentation, and fine-grained visual classification datasets. The effectiveness of our framework is substantiated by practical experimental results and further underpinned by generalization error bound. We release the code publicly at: https://github.com/Lab-LVM/imagenet-models.*

1. Introduction

Supervised learning opened the door to the emergence of a plethora of milestone networks [52, 23, 59, 40, 11, 39] that achieved significant success on ImageNet [48]. Training a single network with the cross-entropy loss has been a simple standard for image classification; this also holds for training multiple networks or multiple features [43, 34, 74, 55, 51, 13]. The methods of extracting multiple features at different stages aim to aggregate diversified features from an architectural perspective. Previous works [43, 34, 55, 13] expand their architectures by incorporating many trainable layers to refine features, relying on the architectural perspective. Their success is likely attributed to extra heavy layers that promote feature diversification. However, it remains uncertain whether the architectures effectively promote learning favorable less-correlated representations [53, 7, 50, 29, 28]. Additionally, their intentional design for high network capacity with numerous trainable parameters increases computational demands.

In this paper, we present a new design concept of deep neural networks that learns multiple less-correlated features at the same time. Since motivated by feature aggregation methods [43, 34, 74], we aim to avoid excessively over-parameterized networks and realize performance improvement through the learning of multiple less-correlated features. Our architecture consists of multiple shallow head classifiers on top of the backbone instead of increasing depth or width and without employing complicated decoder-like architectures. Therefore, it is evident that our architecture offers a speed advantage, but the potentially limited expressiveness with lightweight heads is problematic. A question that naturally arises is *how can we improve the network capacity of shallow heads with limited trainable parameters?*

Our answer centers on the idea of introducing the Gramian matrix [17, 73] combined with the attention module [65]. The Gramian is identical to the bilinear pooling [36] that collects the feature correlations so that the attention can further leverage the information of the Gramian of features. Specifically, we compute the Gramian matrices of each output of heads before the final predictions and feed them into each attention as the query, which brings the pairwise similarity of features. This design principle is naturally scalable to any backbones, including Convolutional Neural Networks (CNNs) [23, 26, 59, 45, 40], Vision Transformers (ViTs) [11, 39], and hybrid architectures [9, 20, 75, 63].

We further introduce a learning algorithm that forces each head to learn different and less correlated representations. The algorithm is based on the proposed decorrelation loss, which performs like an inverse knowledge distillation loss [25]. Our proposed framework compels lightweight heads to learn distinguished and enhanced representations. It turns out that our trained models can replace complicated ones through empirical evaluations and effectively be gener-

*Equal contribution.
†This work was done when Jongwoo Lim was professor at Hanyang University.

alized to other downstream tasks.

We evaluate our models by training them on the CIFAR100 [32] and ImageNet-1K [48] datasets to showcase the effectiveness of our proposed method. We systematically compare the competing models with similar computational budgets, including the throughput of a model; ours beat the state-of-the-art CNNs, ViTs, and hybrid architectures in the accuracy and throughput trade-off. We also show the transferability of our pretrained models to downstream tasks, including instance segmentation and semantic segmentation on COCO [35] and ADE20k [83], respectively, and fine-grained visual classification datasets[1], including CUB-200 [67], Food-101 [3], Stanford Cars [31], FGVC Aircraft [42], and Oxford Flowers-102 [44].

Finally, we analyze the efficacy of our design elements, including hyper-parameters via Strength and Correlation theory [50, 4]. The theory manifests as Correlation is lowered while Strength increases, and classifiers learn a highly generalizable representation. While Correlation and Strength are usually proportional, our framework has elements lowering Correlation while increasing Strength, like evidence in Ryu *et al*. [50]. Therefore, this justifies that the ingredients are well-proposed to leverage low generalization error. We further support the theory through the analyses with the elements to showcase low validation errors in practice. We provide the following summary of our contributions:

(i) We introduce a new network design principle to intensify a backbone by incorporating *multiple lightweight heads* instead of using a complicated head or expanding model in width and depth directions.

(ii) We introduce a novel attention module that employs the Gramian of the penultimate features as a class token within an attention layer, thereby strengthening lightweight classifiers based on pairwise feature similarity. We call *Gramian attention*, which enhances expressiveness without compromising model speed.

(iii) We further propose a learning algorithm with a new loss that enforces multiple heads to yield less-correlated features to each other. Intriguingly, our learning method shows a faster convergence and yields strong precisions.

(iv) We provide an analysis tool for diagnosing design elements of a network and training methods to reveal the effectiveness of the proposed method based on Correlation and Strength with the generalization bound.

2. Method

This section first outlines the motivation for this work. We then present our network architecture and learning algorithm for less-correlated features.

[1]Experimental results of the fine-grained visual recognition tasks are found in Appendix.

2.1. Motivation

Class tokens for class prediction. Learning class tokens [11, 62, 72, 49] have gained popularity because of their effectiveness and simplicity in training ViTs. The class tokens are fed right after the patchification layer for long interactions with features following the original design choice [65]. Additionally, it has been revealed that having a shorter interaction on only later layers improves performance [62]. Longer interactions may harm the discriminability of the class token due to the low-level features, while short interactions can effectively capture high-level information in the later class tokens. We adopt a short-interaction-like design for our network.

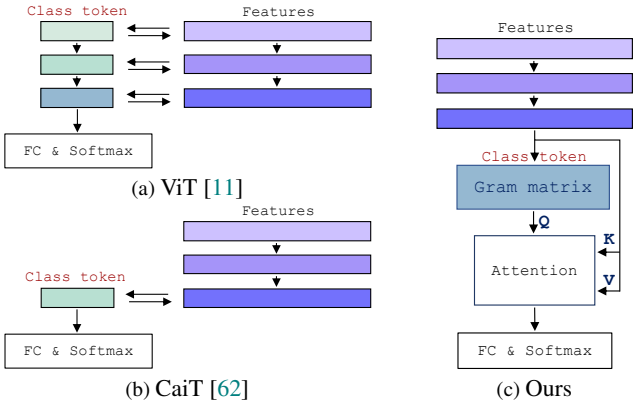
Second, using multiple class tokens [72, 49] has contributed to enhancing the interactions' discriminability. They passively let the class token be learned upon a random initialization rather than actively using the class token. We notice that no studies have been conducted to strengthen the class token itself. We argue that the way of utilizing class tokens in previous literature might not fully exploit the maximum capability of the learned model. We thus further imbue the class token by computing the Gramian from the feature to assign it as the class token.

Employing multiple heads. Previous works [56, 74, 55, 51, 13, 38, 60, 49] guide us that aggregating multiple features give significant benefits over single-path models such as ResNet [23]. Motivated by the success, we design our network learning multiple heads on the top of the backbone, barely spending a high computational budget; each head takes advantage of the aforementioned design manner. The design manner is also supported by the literature [4, 50], which tells us that weak learners (*i.e*., classifiers) should be strongly trained individually while diversifying the learned features for generalization.

Furthermore, to learn stronger heads, we focus on the underexplored correlation among learned features [34, 60, 13, 33, 51]. We propose a so-called less-correlated learning method to maximize feature diversity. In this light, we believe designing a network that branches lightweight head classifiers instead of a complicated network is an appropriate option for making a good combination of the proposed learning method and architecture.

2.2. Our Network Architecture

Gramian attention. We propose an attention-based module, dubbed Gram Attention, for aggregating visual tokens of a network more effectively. The primitive Transformer architecture with n-layers [65, 11] uses the C-dimensional class token $Z \in \mathbb{R}^C$ to formalize the network output Y as: $Y = f^n\left(\dots f^1([Z; X])\right)$, where $[Z; X]$ denotes the concatenation of the N visual tokens $X \in \mathbb{R}^{N\times C}$ and Z; f^1 and f^n stand for the patch extractor and final classifier. This formulation indicates that an early concate-



(a) ViT [11] (b) CaiT [62] (c) Ours

Figure 1: **Contrast illustration of different heads**. We conceptually compare our head classifier design with class token-based classifier heads [11, 62]. Ours employs Gramian with an attention layer to enhance the capability of the class token.

nation of class tokens with the input-side features subsequently updates class tokens through long interactions with visual tokens (see Figure 1a). Figure 1b shows a variation [62, 49] using the class token at later layers from an intermediate m-th layer, the output of which is formulated as $Y = f^n(\dots f^{m+1}\left([Z; f^m\left(\dots f^1\left(X\right)\right)]\right)$. Both of them are trained passively starting from the randomly initialized weights and may reach sub-optimal convergence points because optimizing both X and Z may not guarantee optimum at the same time. Unlikely, we alternatively assign the class token using features from a network.

Figure 1c illustrates our class token assigned by the Gramian computed with the penultimate features. This contrasts with the previous methods, where class tokens are initialized randomly and updated indirectly via feature interactions. Our aim is to directly influence on entire trainable parameters in a network, so we compute the Gramian of the last-computed feature $X^{n-1} \in \mathbb{R}^{N\times HW\times C}$ at the penultimate layer f^{n-1} as:

$$Y = f^n\left([X^{n-1}; \mathcal{G}_{X^{n-1}}]\right), \tag{1}$$

where $\mathcal{G}_X$ denotes the Gramian matrix of X (*i.e.*, an instance-wise computation $\mathcal{G}_X = X^T X \in \mathbb{R}^{N\times C\times C}$). We attribute the expressiveness of the Gramian to its computation of pairwise similarity. In practice, we compute $\mathcal{G}$ with the projected feature $V_X = XW_c$, where $W_c \in \mathbb{R}^{C\times\tilde{C}}$. This is because the Gramian computation here has the complexity of $\mathcal{O}(HWC^2)$, and it becomes more computationally demanding with a large C, so we reduce it by W_c to $\tilde{C} \ll C$ for efficiency. Reducing the inner dimension also improves efficiency, but it could harm the encoded localization information. We introduce a Gramian computation with the vectorized feature to compute pairwise similarity across all locations by the following formula:

$$\mathcal{G}_\mathcal{X} = \text{Vec}(V_X^T V_X)W_g, \tag{2}$$

where $\text{Vec}(\cdot)$ denotes the instance-wise vectorization, and $W_g \in \mathbb{R}^{\tilde{C}^2\times C}$ stands for another projection layer that restores the dimensionality to C, serving it as a class token for the subsequent attention layer.

Head classifier. Following Eq. (1), $\mathcal{G}$ in Eq. (2) is fed into f^n after concatenated with the input feature. We employ the attention [65] as the final layer f^n. We refer to this layer which computes the class embedding Y as the head classifier. Note that the computed Gramian becomes the query, which is similar to [62]. Despite the shallow architecture, it has a large capacity standalone by the pairwise similarity computed by the Gramian. This associating operation is identical to the bilinear pooling [36], which has been revealed as learning strong spatial representation [14, 54]. This operation is known to capture delicate spatial information across channel combinations, so it has been shown to improve the discriminative power of the object classification [6, 15, 16]. We leverage the expressiveness of the bilinear representation for the class tokens possessing a strong spatial representation.

Extending to multi-head architectures. Constructing multiple branches on top of the backbone is a simple way to build multi-head classifiers. We do not rely simply on the final feature but instead, take the aggregated features from a backbone for the head classifiers. This is to take advantage of using diverse multi-level features similar to feature aggregation networks [34, 74]. Since we re-encode the aggregated features using lightweight heads, the multiple heads barely involve extra computational budgets. Therefore, our multi-head architecture can be regarded as an efficient alternative to heavy head architectures [55, 34, 38, 60, 12] or the way building complicated architectures [82, 59, 78].

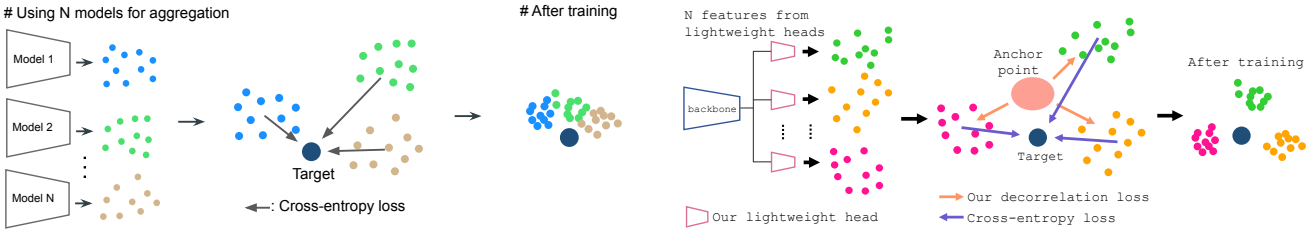
2.3. Training Multi-head Classifiers

On less-correlated multi-head classifiers. Here we introduce a novel less-correlated learning method to learn more expressive multi-head classifiers. Training multiple identical network architectures or branches without considering feature diversity may not yield advantages. Since the models are expected to converge to nearby local minima during training, the resulting models are likely to learn correlated representations [53, 7, 50, 29, 28] (see Figure 2a). We begin with the model averaging loss (*i.e.*, equally weighting the outputs) with the i-th output of h heads as:

$$\mathcal{L} = \sum_i^h \texttt{CE}_i = -\sum_i^h y^T \cdot \log f_i^n(x), \tag{3}$$

where $\texttt{CE}_i$ denotes the cross-entropy loss with the ground-truth label y, and $f_i(x)$ denotes the output of i-th head for the input x. For simplicity, we abbreviate f^n (*i.e.*, n-th layer's output) in previous notations to f.

Directly minimizing Eq. (3), the correlation among the predictions f_i is likely to be high, so we propose a new



(a) Traditional learning method

(b) Our learning method

Figure 2: **Schematic illustrations.** We compare our learning method with a traditional one. (a) Multiple features extracted from different models(or a complicated head) are trained under the task loss only, so they are likely to get close to the ground-truth labels as training progresses; (b) our method trains lightweight heads by ensuring the representations are not highly correlated with each other.

decorrelation loss to avoid it:

$$\mathcal{L} = -\sum_{i}^{h} y^T \cdot \log f_i(\hat{x}) + \lambda \mathcal{L}_{dec}, \tag{4}$$

$$\text{s.t. } \mathcal{L}_{dec} = \sum_{i}^{h} \sum_{j} \frac{f_j(\hat{x})^T}{n} \cdot \left(\log \sum_{k} \frac{f_k(\hat{x})}{n} - \log f_i(\hat{x}) \right),$$

where $\mathcal{L}_{dec}$ is coined by the decorrelation loss[2] (see Figure 2b), and λ is a tunable weighting parameter. Note that we use only negative λ in Eq. (4) to ensure the decorrelation loss functions in opposition to the cross-entropy loss.

Connection to knowledge distillation. One may speculate the proposed loss relates to Kullback Leibler Divergence used in the knowledge distillations [25, 47]. The canonical knowledge distillation methods use a positive value for λ; unlikely, our approach assigns a negative λ in Eq. (4), which lets the knowledge from the aggregated prediction be reversely transferred. Therefore, each prediction f_i would deviate and be less correlated (see Figure 2b). We argue that training with knowledge distillation (*i.e.*, using the KD loss [25]) may fail to let each head learn without high correlation. This result is obvious that a positive λ makes the distance between the aggregated prediction and each prediction get closer, so the predictions get similar, as shown in Figure 2a. Our claim is addressed in the later discussion section providing both qualitative and quantitive results (see the visualization in Figure 6e and compare it with Figure 6d.

3. Experiment

This section begins with the empirical analyses of the components of our method. We then demonstrate the superiority of our models through ImageNet classifications and transfer them to downstream tasks. We coin a network using our Gramian attention-included heads as GA-network.

[2]We use the term *decorrelation* here in the idiomatic context of reducing the relevance and correlation of output predictions.

$\mathcal{C}$	dim	Gram	Dec	FLOPs (G)	#Params (M)	Top-1 err (%)
1	32	-	-	0.26	2.5	21.8
1	32	✓	-	0.26	2.6	21.2
1	64	✓	-	0.35	4.0	19.8
2	64	✓	-	0.26	2.6	20.1
2	128	✓	-	0.35	4.1	18.4
8	128	✓	-	0.22	2.0	18.9
8	128	✓	✓	0.22	2.0	16.9

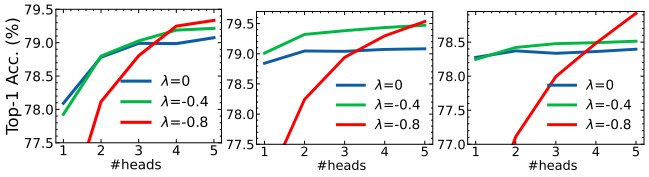
Table 1: **Factor analysis**. The cardinality ($\mathcal{C}$) and the reduced input channel (dim) of the head classifiers are studied. We mainly verify the impact of the proposed Gramian attention (Gram) and decorrelation loss (Dec). We experiment with ResNet110 on CIFAR100. A careful design significantly improves accuracy without added computational costs.

Net	Head	#heads	λ	#Params (M)	Top-1 acc (%)
ResNet50	GAP-FC	1 / 10	-	25.9 / 44.0	75.3 / 75.7
	CaiT	1 / 5	-	21.8 / 38.5	76.7 / 77.0
	Gram	1 / 5	0	22.4 / 41.3	**78.0** / 79.1
	Gram	1 / 5	-0.4	22.4 / 41.3	77.9 / 79.2
	Gram	1 / 5	-0.8	22.4 / 41.3	76.3 / **79.3**
ViT-S	GAP-FC	1 / 20	-	22.1 / 29.4	76.3 / 76.3
	ViT	1 / 20	-	22.1 / 29.4	75.3 / 75.4
	CaiT	1 / 5	-	22.8 / 27.3	75.2 / 75.3
	Gram	1 / 5	0	22.9 / 27.7	**78.3** / 78.4
	Gram	1 / 5	-0.4	22.9 / 27.7	**78.3** / 78.5
	Gram	1 / 5	-0.8	22.9 / 27.7	78.2 / **78.9**

Table 2: **Extended factor analysis.** We extend the analysis to ImageNet-1K, building upon learned insights from Tab. 1. We study the impact of head types (Head), the number of heads (#heads), and λ in the decorrelation loss. We include the global average pooling with a fully-connected layer (GAP-FC), ViT, CaiT, and ours (Gram) shown in Fig. 1. We report the accuracy of both single and multiple heads adjusted to have similar parameters (single/multiple heads).

3.1. Preliminary Factor Analyses

First, we study how each design element of the proposed method works on the CIFAR dataset. Table 1 shows that



(a) ResNet-50 (b) ConvNext-T (c) DeiT-S

Figure 3: **Empirical study on #heads and λ.** We examine ImageNet accuracy versus the number of head classifiers across different λ for the decorrelation loss. Single-head underperforms, whereas using more heads increases the performance across all backbones. Lower values of λ are more compatible with multiple heads; the best, with $\lambda = -0.8$, is achieved with five heads.

Network	FLOPs (G)	#Params (M)	Throughput (img/sec)	Top-1 acc (%)
RSB-ResNet50 [68]	4.1	25.6	3409	79.8
GA-ResNet50	5.2	41.3	2145	82.5
RSB-ResNet152 [68]	11.6	60.2	1463	81.8
ViT-S [11]	4.2	22.1	2556	79.8
GA-ViT-S	4.3	27.7	2289	80.9
GA-ViT-M	9.6	60.5	1322	82.6
ViT-B [11]	16.9	86.6	987	81.8
ConvNeXt-T [40]	4.5	28.6	2098	82.1
GA-ConvNeXt-T	6.3	48.7	1452	83.2
ConvNeXt-S [40]	8.7	50.2	1282	83.1
GA-ConvNeXt-S	10.5	70.4	967	83.9
ConvNeXt-B [40]	15.4	88.6	903	83.8
GA-ConvNeXt-B	19.0	124.3	668	84.3
ConvNeXt-L [40]	34.4	197.8	507	84.3

Table 3: **Our ImageNet-1K models.** We apply our method to the popular architectures, including ResNet [23], ConvNeXt [40], and ViT [11, 61]; we dub our models GA-ResNet, GA-ConvNeXt, and GA-ViT, respectively. All our models improve the baselines by large margins and enjoy faster speeds than each counterpart having similar accuracy.

using our proposed Gramian attention (Gram) and learning method with the decorrelation loss (Dec) boosts the accuracy significantly. The results also display a head classifier can be strengthened by increasing the aggregated dimension (dim) and cardinality ($\mathcal{C}$) under similar computational demands. Extending the analysis to the ImageNet-1K dataset, we investigate the effectiveness of our multiple head architectures and the proposed learning method in Table 2. All experiments are performed with identical network configurations to the baselines models (ResNet50, ViT-S), such as the stage configuration and channel dimension. We report accuracies training ResNet50s and ViTs for 50 and 100 epochs, respectively.

As shown in Table 2, we confirm the models with proposed multiple heads significantly outperform baseline networks trained with the naive global average pooling (GAP-

Network	FLOPs (G)	#Params (M)	Throughput (img/sec)	Top-1 acc (%)
RSB-ResNet50 [68]	4.1	25.6	3409	79.8
RSB-ResNet152 [68]	11.6	60.2	1463	81.8
ResNetY-8G [45]	8.0	39.2	827	82.1
ViT-S [11, 61]	4.2	22.1	2556	79.8
Swin-S [39]	8.5	49.6	1024	83.0
PoolFormer-M36 [75]	8.8	56.2	796	82.1
CoatNet-0 [9]	4.2	27.4	1781	81.6
CSwin-T [10]	4.3	23.0	1498	82.7
ConvNeXt-S [40]	8.7	50.2	1282	83.1
GA-ResNet50	5.2	41.3	2145	82.5
GA-ConvNeXt-T	6.3	48.7	1452	83.2
ResNetY-16G [45]	15.9	83.6	632	82.2
ViT-B [11, 61]	16.9	86.6	987	81.8
Swin-B [39]	15.1	87.8	731	83.5
PoolFormer-M48 [75]	11.6	73.5	601	82.5
CoatNet-1 [9]	7.6	41.7	985	83.3
InceptionNeXt-S [76]	8.4	49	-	83.5
CSwin-S [10]	6.9	35.0	933	83.6
MaxViT-T [63]	5.6	30.9	976	83.6
SLaK-S [37]	9.8	55	-	83.8
ConvNeXt-B [40]	15.4	88.6	903	83.8
GA-CSwin-T	6.1	42.0	1001	84.1
GA-ConvNeXt-S	10.5	70.4	967	83.9
ResNetY-32G [45]	32.3	145.1	378	82.4
InceptionNeXt-B [76]	14.9	87	-	84.0
SLaK-B [37]	17.1	95	-	84.0
CoatNet-2 [9]	14.5	73.9	629	84.1
CSwin-B [10]	15.0	78.0	549	84.2
ConvNeXt-L [40]	34.4	197.8	507	84.3
MaxViT-S [63]	11.7	68.9	636	84.5
CoatNet-3 [9]	32.5	165.2	360	84.5
GA-ConvNeXt-B	19.0	124.3	668	84.3
GA-ConvNeXt-B†	26.1	124.3	524	84.5
GA-CSwin-S	8.7	54.3	671	84.7

Table 4: **ImageNet-1K results.** Our models are compared with the state-of-the-art networks, including CNN, Transformer, and hybrid architectures on ImageNet-1K. We group the networks according to the computational budgets. All accuracies are borrowed from the original paper; RegNet accuracy is taken from [68]. We report the throughputs measured by ourselves, running on an RTX 3090 GPU. Our networks perform well over competitors with manageable resources and faster speed. We also provide the memory usage in the supplementary material. † uses 272×272 image size. GA extremely improves CSwin family; we presume the lower channel dimension of CSwin architectures is an underlying reason.

FC). Our Gramian attention remarkably outperforms existing ViT- and CaiT-like class token methods again. The proposed learning method with decorrelation loss (Dec) also contributes to performance, and this contribution is more significant with multiple heads and lowered λ across all ar-

chitectures. Figure 3 gives more information on the accuracy variation of the models with multiple heads concerning λ in the decorrelation loss. It verifies that the decorrelation loss can diversify learned features so that a higher λ ($\lambda = -0.8$) performs better than other lower λ cases (λ=0 and λ=-0.4).

3.2. ImageNet Classification

Implementation details. We employ ResNet [23], ConvNeXt [40], CSwin [10], and ViT [61] as our baseline networks, with each backbone branching out five heads. For ResNet50, we build our GA-network with some popular tweaks; we reduce the channel dimension of the last three residual blocks to 1024 and exploit SE [27], and design tweaks introduced in the previous work [24]. For ConvNeXt and ViT, we use the original architectures. For ViT, we encompass ViT-M having an intermediate model size between ViT-S and ViT-B, which has 576 channels with nine attention heads. For ConvNeXt and CSwin, due to the lower channel dimension compared to ResNet, we utilize a larger feature scale with minimal overhead. Note that we do not delve into investigating more compatible backbones for our method architecturally. Instead, our focus is to showcase the effectiveness of our method through performance improvements on popular and straightforward network architectures under minimal resources.

Comparison with state-of-the-arts. We compare the performance of GA-networks with the contemporary state-of-the-art network architectures regarding the accuracy and computational complexities. GA-networks competes with the recently proposed network architectures, including the CNN architectures of RSB-ResNet [68], RegNet [45], ConvNeXt [40], and SLaK [37]; the ViT [11]-related architectures, including ViT [61], Swin Transformer [39], and CSwin Transformer [10]; the hybrid architectures PoolFormer [75], CoatNet [9], MaxViT [63], and InceptionNeXt [76]. We systematically compare GA-networks, including scaled-up models shown in Table 3 with the competing models grouped by computational budgets, mainly focusing on throughput. Furthermore, we perform comprehensive comparisons with the popular contemporary models in Table 4, and it shows our models have clear advantages in throughput over their counterparts and outperform the competing networks, including the state-of-the-art CNN, ViT, and hybrid models.

3.3. Downstream tasks

We investigate the applicability of the proposed method to two downstream tasks, including instance segmentation and semantic segmentation. Compared with the previous state-of-the-art models, we train pretrained GA-networks on ImageNet-1k in Table 4. Following the setups in literature [39, 75, 49], we attach detection and segmentation networks to ours. As in the literature [34, 49], where the rear layers of the network are connected to the frontal layers, we attach dense prediction layers on our backbones. We train our model with the widely-used MMDetection and MMSegmentation libraries[3], and we report the performance of previous methods from the same training epochs or iterations.

Network	AP (box)	AP (mask)	#Params (M)
RegNetX-12G	42.2	38.0	64.1
Swin-T	42.7	39.3	47.8
Poolformer-S36	41.0	37.7	31.6
GA-R50	42.8	39.3	42.5
X101-64	48.3	41.7	140
Swin-S	51.9	45.0	107
ConvNeXt-S	51.9	45.0	108
GA-ConvNeXt-S	52.3	45.3	108

Table 5: **COCO instance segmentation results.** Our models ResNet50 (R50) and ConvNeXt-S outperform competing backbones using identical segmentation heads, respectively.

Object instance segmentation. We train the object instance segmentation model on COCO 2017 [35]. We exploit Mask R-CNN [22] for ResNet-50 and Cascade Mask R-CNN [5] for ConvNeXt-S as the baseline model. As shown in Table 5, ours outperform the models based on RegNet [45], Swin Transformer [39], and PoolFormer [75].

Semeantic segmentation. We train our models on the ADE20k semantic segmentation [83]. We employ two widely used heads: FPN [34] and UperNet [70] for the segmentation head in our model. As shown in Table 6, our networks exhibit competitive performance relative to models employing PoolFormer [75] and Swin Transformer [39] using each head.

3.4. Training Setups

ImageNet-1K. Recent state-of-the-art networks [80, 46, 68, 1, 21, 61] exploit training regimes with strong data augmentations, mostly based on timm library[4] [69]. We adopt a similar training regime, which employs Mixup [79], CutMix [77], and RandAugment [8] for data augmentation and use the cosine learning rate scheduling [41] with 300 epochs[5].

Downstream task. For fair comparisons, we follow the same training setup of the competing backbones. We exploit $1\times$ training schedule with 12 epochs in COCO. On ADE20k, we follow the same training setup of competitors again to train our segmentation model with iterations of 40k. We use 32 batch size for the 40k-iterations setup to compare ours with PoolFormer [75] and use the 120k-iterations setup in

[3] https://github.com/open-mmlab

[4] https://github.com/rwightman/pytorch-image-models/

[5] In our ablation study in Table 2, we primarily train networks with ResNet-based and ConvNeXt-based models for 50 epochs and exceptionally train ViT-based models for 100 epochs due to its late convergence.

Head	Network	Iter.	mIOU	#Params (M)
FPN	PoolFormer-S36	40k	41.6	34.6
	GA-R50	40k	**41.8**	26.6
UperNet	Swin-T	160k	44.4	59.9
	GA-R50	160k	**45.2**	67.3

Table 6: **ADE20k semantic segmentation results.** Our models outperform competing backbones with identical segmentation heads.

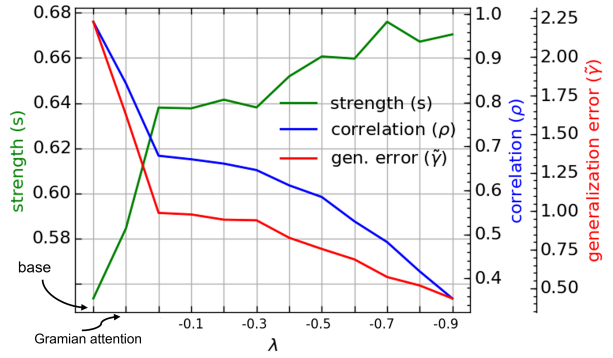


Figure 4: **Generalization error bound.** We visualize Correlation (ρ), Strength (s), and the upper bound of the generalization error ($\tilde{\gamma}$). We plot the metrics versus the architectural elements and different λ values in the decorrelation loss. The left tick of the Gramian attention on the x-axis shows that architecture elements contribute to lowering the generalization error bound, and λ in our less-correlated feature learning drops the bound on the right side.

Swin Transformer [39] with 16 batch size.

CIFAR100. We follow the standard 300-epochs training protocol with SGD [19, 77] with the initial learning rate of $1e{-}3$ decaying by 0.1 at 150 and 225 epochs. We use 64 batch size for training using two GPUs.

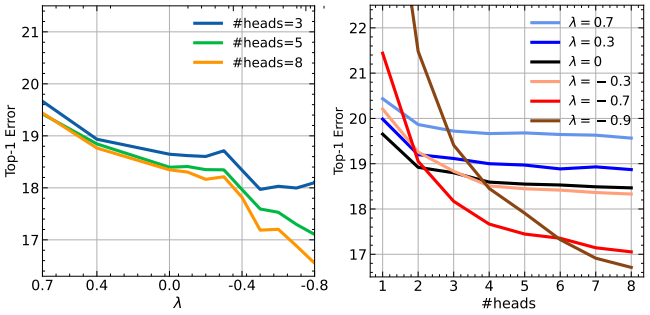
4. Discussions

In this section, we investigate our method through the generalization bound analysis and the visualization method.

4.1. Analyzing Our Method

Here we justify our proposed design principle and learning method based on the foundation theory [4, 50] that investigates the generalization capability of a model with multiple classifiers like ours. The theory is to compute the degree of Strength and Correlation for the generalization error bound [4, 50], and the magnitude of the metrics indicates how well the model generalizes [50].

Strength and Correlation. Strength s is firstly defined as the expectation of the margin between model prediction and the ground truth labels. The margin function is formulated as $f(Y_\phi, \hat{Y}) = P(Y_\phi = \hat{Y}) - \max_{j \neq \hat{Y}} P(Y_\phi = j)$, where Y_ϕ and $\hat{Y}$ denote the output labels of a head classifier ϕ and the ground-truth labels of the data points, respectively. The



(a) Error vs. λ (b) Error vs. #heads

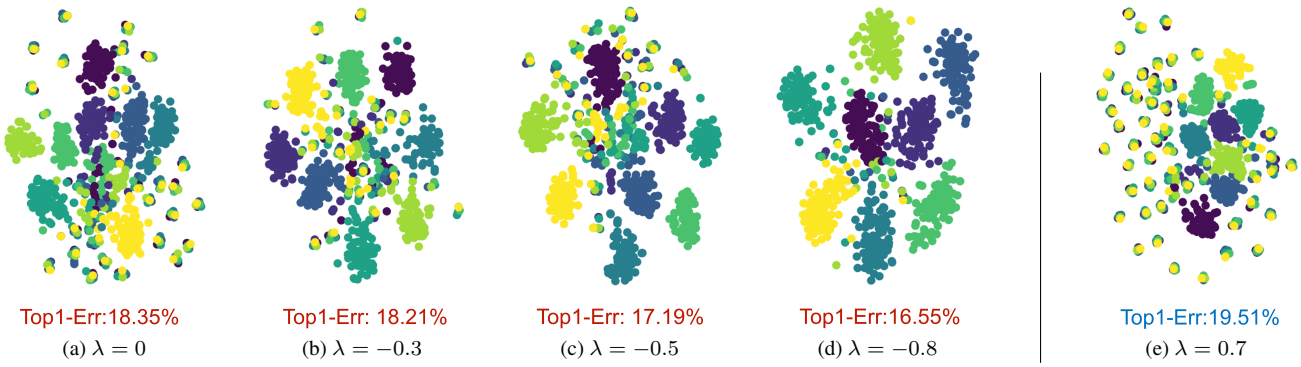
Figure 5: **Validation error trend w.r.t λ and #heads.** (a) top-1 error versus λ in the decorrelation loss; (b) top-1 error versus the number of heads. We observe that training with multiple heads with $\lambda < 0$ significantly reduce the top-1 error.

last term $\max_{j \neq \hat{Y}} P(Y_\phi = j)$ stands for a set of labels with the largest probability amongst wrong answers.

Correlation ρ is computed with the raw margin function ψ, which is defined as $\psi(Y_\phi, \hat{Y}) = \mathrm{I}(Y_\phi = \hat{Y}) - \mathrm{I}(Y_\phi = \max_{j \neq \hat{Y}} P(Y_\phi = j))$, where $\mathrm{I}(\cdot)$ is the indicator function. ρ is then computed by averaging the Pearson Correlation coefficient of ψ between all combinations of heads (ϕ_i, ϕ_j).

Generalization error bound. The upper bound of generalization error $\tilde{\gamma}$ is compute from Strength $\mathcal{S}$ and Correlation ρ, which is $\gamma \leq \rho(1 - s^2)/s^2$. This implies Correlation and Strength are opposite to each other to achieve a low generalization error; however, importantly, the previous literature [50] showed there could exist a method that trains a model to decrease correlation while increasing Strength. Based on the evidence, we conjecture that an appropriate design of the head may also achieve it again. We confirm this by measuring the metric – the generalization error bound – to be reduced for particular architectural or training-related elements. Figure 4 shows that the proposed architectural design elements and learning method significantly reduce the upper bound of the generalization error.

We further visualize Correlation and Strength metrics together and observe Correlation gets consistently lowered as appending the architectural elements and adjusting the degree of the correlation (adjusted by λ) in our learning method. This result indicates that our network architecture with multiple heads trained with our proposed learning method pushes the model to learn less-correlated and diversified features to contribute to the model's generalization capability. We further claim that the generalization bound is actually connected to performance in practice. We train models and visualize their validation errors in Figure 5a and Figure 5b. Along with Table 2 reporting the error decreases as architecture advances, the figures show a consistent trend with Figure 4.



(a) $\lambda = 0$ (b) $\lambda = -0.3$ (c) $\lambda = -0.5$ (d) $\lambda = -0.8$ (e) $\lambda = 0.7$

Figure 6: **t-SNE plots of the features extracted from learned head classifers**. We visualize how much the proposed learning method scatters the output features of each head. We extract the features from the images in the validation set and distinguish them from different head classifiers by color. We use features of a ResNet110 for (a) to (e). Specifically, (a) eight head classifiers without the decorrelation loss ($\lambda = 0$); (b), (c), (d), and (e) different weighting parameters λ, respectively; We observe that 1) accuracy is aligned with the feature correlation; 2) our proposed learning method (*i.e.* $\lambda < 0$) works to increase the feature diversity with lowered correlation; 3) learned features with ($\lambda > 0$) do not guarantee both low correlation and error (see (e).)

4.2. Visualizing Learned Features

We investigate the impact of the decorrelation loss in Eq. (4) with different λ by visualizing the output features with t-SNE [64]. Figure 6 shows the clear trend when using $\lambda < 0$; larger (to the negative direction) λ let the model learn less-correlated features; the performance follows the trend. All with negative λ outperforms the case of $\lambda = 0$ that does not use the decorrelation loss.

The performance with different λ gets clearer with Figure 5a, we achieve the best performance when the λ is near -0.8, and when $\lambda > 0$ the performance gets poorer than the model with $\lambda = 0$. Additionally, a comprehensive visualization both with the number of heads and different λ in Figure 5b reveals some interesting aspects. We observe that when λ reaches -0.7, the performance improves significantly as the number of heads increases. Performance gets saturated trained only with three heads when $\lambda \geq 0$, while negativeλ lets the model avoid saturation.

5. Related Work

Recent advance of the ImageNet networks. After the emergence of ResNet [23], EfficientNets [59] have dominated the field of ImageNet network architecture. Due to its low throughput compared to the low computational costs, ResNet [23] has been revisited by training it with more sophisticated training setups to maximize the performance and got new names called RS-ResNet [2] and ResNet-RSB [68]. After the emergence of Vision Transformers (ViT) [11], DeiT [61], which trained ViT more effectively, invaded CNNs and got dominated. After that, another milestone was Swin Transformer [39], which pioneered the hierarchical ViT. A hybrid architecture such as CoatNet [9] successively have showed another design principle using CNN and ViT effectively. ConvNeXt [40] was proposed to try to bring back the glory of CNN from ViT. Another hierarchical ViT, called CSwin [10], showed more improved performance over Swin Transformer. Our work does not lie in a dominant trend of architectural development but is being studied to complement all architectures like a plug-and-play module.

Network architectures with feature aggregation. Inception models [57, 30, 58, 56] showed aggregating multiple features could further bring performance improvements. Veit *et al*. [66] interpreted ResNet [23] as an ensemble of numerous shallow neural networks, resulting in learning various features intrinsically. Inspired by [43, 34], many previous works [55, 74, 51, 38, 13, 12, 49, 82] proposed to design advanced architectures by aggregating multiple features. They heavily rely on multi-path connections with extra trainable layers as head architecture. Albeit they showed outstanding task performance, the models are computationally heavy due to additional learnable parameters; the multiple paths may learn similar representations. Our work shares the similar concept of aggregating features, but the difference is that we leverage a lightweight design regime for head classifiers instead of a complicated head architecture for a strong prediction through aggregation. Furthermore, it turns out that our lighter model consisting of the operations above achieves better discriminative powers with less correlated features.

Training with lowering feature correlation. Despite the architectural advances, it has been reported that learned features are usually in high correlation [53, 7, 29, 81, 50, 28]. Algorithmic ways of training the features having a low correlation are also addressed in the literature [7, 71, 18, 84]. Our method has, in a similar line to [7, 84] which proposed distinctive losses that explicitly promote decorrelation at activation or filter, respectively. On the other hand, ours learn

less-correlated features for aggregation in an inter-feature (or inter-layer) manner, directly affecting the final classifier. Lan *et al*. [33] initially promoted ensemble branches by knowledge distillation, but the learned features were found to be highly correlated. Finally, it also turns out that our proposed architecture cooperates with the proposed learning technique towards improving the less-correlation property.

6. Conclusion

We have introduced a new learning framework with a network architecture leveraging lightweight heads. In contrast to traditional network architecture designs, we have proposed a novel approach using multiple lightweight head classifiers to create an expressive network. Our GA-network aggregates the features refined by lightweight head classifiers, where the computational budget is significantly low. Additionally, our proposed learning method with the proposed decorrelation loss made our network learn less-correlated features, and aggregating them boosts performance due to learned complementary features. Our network has demonstrated increased feature diversification when employing the proposed learning method. The experimental results have proven that only the lightweight architecture has sufficient capacity for learning. We have analyzed our proposed method's effectiveness based on the Correlation and Strength theory. We found that the generalization bound has been consistently reduced for each proposed element and learning method. Finally, our network architecture has significantly outperformed the recent state-of-the-art CNNs, ViTs, and hybrid architectures on the ImageNet evaluation. Furthermore, several downstream tasks, including the COCO instance segmentation and ADE20k semantic segmentation, showcased our models' superior transferability. We expect our network design principle and method can be applied to any network architecture to improve performance. We hope the overall proposed framework facilitates future research.

Limitations. Even though the proposed design of employing lightweight multiple heads has minimal computational budgets, it unavoidably incurs extra parameters due to the internal channel dimension. We did not train extremely large baseline models such as large vision transformers such as ViT-H/14 [11] or ViT-G/14 [78], we believe our method will be applicable to such large models.

Acknowledgements. This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea Government (MSIT) (Artificial Intelligence Innovation Hub) under Grant 2021-0-02068 and under the Artificial Intelligence Convergence Innovation Human Resources Development (IITP-2023-No.RS-2023-00255968) Grant.

References

[1] Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. In *NeurIPS*, 2021. 6

[2] Irwan Bello, William Fedus, Xianzhi Du, Ekin Dogus Cubuk, Aravind Srinivas, Tsung-Yi Lin, Jonathon Shlens, and Barret Zoph. Revisiting resnets: Improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 34:22614–22627, 2021. 8

[3] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014. 2

[4] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 2, 7

[5] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018. 6

[6] Joao Carreira, Rui Caseiro, Jorge Batista, and Cristian Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, pages 430–443. Springer, 2012. 3

[7] Michael Cogswell, Faruk Ahmed, Ross Girshick, Larry Zitnick, and Dhruv Batra. Reducing overfitting in deep networks by decorrelating representations. *arXiv*, 2015. 1, 3, 8

[8] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical data augmentation with no separate search. *arXiv*, 2019. 6

[9] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *NeurIPS*, 34:3965–3977, 2021. 1, 5, 6, 8

[10] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *CVPR*, pages 12124–12134, 2022. 5, 6, 8

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 1, 2, 3, 5, 6, 8, 9

[12] Ruoyi Du, Dongliang Chang, Ayan Kumar Bhunia, Jiyang Xie, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. Fine-grained visual classification via progressive multi-granularity training of jigsaw patches. In *ECCV*, pages 153–168. Springer, 2020. 3, 8

[13] Xianzhi Du, Tsung-Yi Lin, Pengchong Jin, Golnaz Ghiasi, Mingxing Tan, Yin Cui, Quoc V Le, and Xiaodan Song. Spinenet: Learning scale-permuted backbone for recognition and localization. In *CVPR*, 2020. 1, 2, 8

[14] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016. 3

[15] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact

bilinear pooling for visual question answering and visual grounding. *arXiv*, 2016. 3

[16] Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. Compact bilinear pooling. In *CVPR*, pages 317–326, 2016. 3

[17] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015. 1

[18] Shuqin Gu, Yuexian Hou, Lipeng Zhang, and Yazhou Zhang. Regularizing deep neural networks with an ensemble-based decorrelation method. In *IJCAI*, 2018. 8

[19] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *CVPR*, 2017. 7

[20] Dongyoon Han, YoungJoon Yoo, Beomyoung Kim, and Byeongho Heo. Learning features with parameter-free layers. *arXiv preprint arXiv:2202.02777*, 2022. 1

[21] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. In *NeurIPS*, 2021. 6

[22] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 6

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2, 5, 6, 8

[24] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 6

[25] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv*, 2015. 1, 4

[26] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *ICCV*, 2019. 1

[27] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 6

[28] Tianyu Hua, Wenxiao Wang, Zihui Xue, Sucheng Ren, Yue Wang, and Hang Zhao. On feature decorrelation in self-supervised learning. In *CVPR*, 2021. 1, 3, 8

[29] Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization. In *CVPR*, 2018. 1, 3, 8

[30] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 8

[31] Jonathan Krause, Jia Deng, Michael Stark, and Li Fei-Fei. Collecting a large-scale dataset of fine-grained cars. *In Second Workshop on Fine-Grained Visual Categorization*, 2013. 2

[32] A. Krizhevsky. Learning multiple layers of features from tiny images. In *Tech Report*, 2009. 2

[33] Xu Lan, Xiatian Zhu, and Shaogang Gong. Knowledge distillation by on-the-fly native ensemble. In *NeurIPS*, 2018. 2, 9

[34] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *ICCV*, 2017. 1, 2, 3, 6, 8

[35] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 6

[36] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *ICCV*, pages 1449–1457, 2015. 1, 3

[37] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Xuxi Chen, Qiao Xiao, Boqian Wu, Mykola Pechenizkiy, Decebal Mocanu, and Zhangyang Wang. More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. *arXiv preprint arXiv:2207.03620*, 2022. 5, 6

[38] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018. 2, 3, 8

[39] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 1, 5, 6, 7, 8

[40] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. 1, 5, 6, 8

[41] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 6

[42] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv*, 2013. 2

[43] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 1, 8

[44] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008. 2

[45] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *CVPR*, 2020. 1, 5, 6

[46] Tal Ridnik, Hussam Lawen, Asaf Noy, Emanuel Ben Baruch, Gilad Sharir, and Itamar Friedman. Tresnet: High performance gpu-dedicated architecture. In *WACV*, 2021. 6

[47] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015. 4

[48] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 1, 2

[49] Jongbin Ryu and Hankyul Kang. Mmcap: Learning to broadsight neural networks by class attention pooling, 2023. 2, 3, 6, 8

[50] Jongbin Ryu, Gitaek Kwon, Ming-Hsuan Yang, and Jongwoo Lim. Generalized convolutional forest networks for domain generalization and visual recognition. In *ICLR*, 2020. 1, 2, 3, 7, 8

[51] Jongbin Ryu, Ming-Hsuan Yang, and Jongwoo Lim. Dft-based transformation invariant pooling layer for visual classification. In *ECCV*, pages 84–99, 2018. 1, 2, 8

[52] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[53] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. 1, 3, 8

[54] Yumin Suh, Jingdong Wang, Siyu Tang, Tao Mei, and Kyoung Mu Lee. Part-aligned bilinear representations for person re-identification. In *ECCV*, pages 402–419, 2018. 3

[55] Shuyang Sun, Jiangmiao Pang, Jianping Shi, Shuai Yi, and Wanli Ouyang. Fishnet: A versatile backbone for image, region, and pixel level prediction. In *NeurIPS*, 2018. 1, 2, 3, 8

[56] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. In *ICLR Workshop*, 2016. 2, 8

[57] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 8

[58] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 8

[59] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv*, 2019. 1, 3, 8

[60] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *CVPR*, pages 10781–10790, 2020. 2, 3

[61] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, 2021. 5, 6, 8

[62] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 2, 3

[63] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *ECCV*, 2022. 1, 5, 6

[64] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008. 8

[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. 1, 2, 3

[66] Andreas Veit, Michael J Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. In *NeurIPS*, 2016. 8

[67] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2

[68] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *Arxiv*, 2021. 5, 6, 8

[69] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *Arxiv*, 2021. 6

[70] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, pages 418–434, 2018. 6

[71] Wei Xiong, Bo Du, Lefei Zhang, Ruimin Hu, and Dacheng Tao. Regularizing deep convolutional neural networks with a structured decorrelation constraint. In *ICDM*, 2016. 8

[72] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, and Dan Xu. Multi-class token transformer for weakly supervised semantic segmentation. In *CVPR*, pages 4310–4319, 2022. 2

[73] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *CVPR*, pages 4133–4141, 2017. 1

[74] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *CVPR*, 2018. 1, 2, 3, 8

[75] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. *Arxiv*, 2021. 1, 5, 6

[76] Weihao Yu, Pan Zhou, Shuicheng Yan, and Xinchao Wang. Inceptionnext: When inception meets convnext. *arXiv preprint arXiv:2303.16900*, 2023. 5, 6

[77] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 6, 7

[78] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022. 3, 9

[79] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 6

[80] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. *Arxiv*, 2020. 6

[81] Zijun Zhang, Yining Zhang, and Zongpeng Li. Removing the feature correlation effect of multiplicative noise. In *NeurIPS*, volume 31, 2018. 8

[82] Jingyu Zhao, Yanwen Fang, and Guodong Li. Recurrence along depth: Deep convolutional neural networks with recurrent layer aggregation. *Advances in Neural Information Processing Systems*, 34:10627–10640, 2021. 3, 8

[83] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 2, 6

[84] Xiaotian Zhu, Wengang Zhou, and Houqiang Li. Improving deep neural network sparsity through decorrelation regularization. In *IJCAI*, 2018. 8