PaPr: Training-Free One-Step Patch Pruning with Lightweight ConvNets for Faster Inference

Tanvir Mahmud[1] Burhaneddin Yaman[2] Chun-Hao Liu[3] Diana Marculescu[1]

[1]University of Texas at Austin [2]Bosch Research [3]Amazon Prime Video
[1]{tanvirmahmud, dianam}@utexas.edu [2]yaman013@umn.edu [3]liuch37@gmail.com

Abstract. As deep neural networks evolve from convolutional neural networks (ConvNets) to advanced vision transformers (ViTs), there is an increased need to eliminate redundant data for faster processing without compromising accuracy. Previous methods are often architecture-specific or necessitate re-training, restricting their applicability with frequent model updates. To solve this, we first introduce a novel property of lightweight ConvNets: their ability to identify key discriminative patch regions in images, irrespective of model's final accuracy or size. We demonstrate that fully-connected layers are the primary bottleneck for ConvNets performance, and their suppression with simple weight recalibration markedly enhances discriminative patch localization performance. Using this insight, we introduce PaPr, a method for substantially pruning redundant patches with minimal accuracy loss using lightweight ConvNets across a variety of deep learning architectures, including ViTs, ConvNets, and hybrid transformers, without any re-training. Moreover, the simple early-stage one-step patch pruning with PaPr enhances existing patch reduction methods. Through extensive testing on diverse architectures, PaPr achieves significantly higher accuracy over state-of-the-art patch reduction methods with similar FLOP count reduction. More specifically, PaPr reduces about 70% of redundant patches in videos with less than 0.8% drop in accuracy, and up to 3.7× FLOPs reduction, which is a 15% more reduction with 2.5% higher accuracy. Code is available at https://github.com/tanvir-utexas/PaPr.

1 Introduction

Deep neural networks have grown from simple convolutional neural networks (ConvNets) to complex transformer models, aiming for better accuracy with more computation [10,18]. Vision transformers (ViTs) excel by focusing on important parts of images with long-range attention and large-scale pre-training [17,39]. This success, however, comes at the cost of increased computational cost. Operating these large networks efficiently for downstream applications is crucial. Most visual tasks demand precisely pinpointing key image regions against complex backgrounds. Higher image resolution improves accuracy by capturing more details but also adds unnecessary background processing, burdening large models [44]. This highlights the need to cut down on redundant data in high-resolution images to maintain both speed and performance with advanced techniques [2,34].
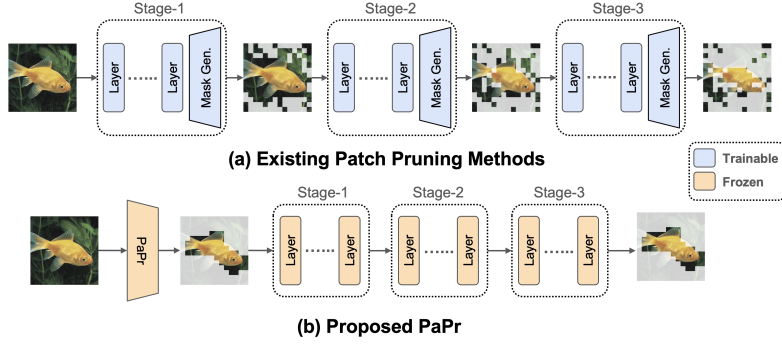
(a) Existing Patch Pruning Methods

(b) Proposed PaPr

Fig. 1: (a) Existing patch pruning methods gradually reduce patches over the model. This requires additional training of mask generators in intermediate layers. (b) Proposed PaPr directly prunes redundant patches early in the network by leveraging pretrained lightweight ConvNets and directly speeds-up *off-the-shelf* models without re-training.

Identifying key image regions demands a comprehensive understanding of the image and model operations. Incorrect key region estimation can impair pre-training performance by eliminating crucial areas. Traditional approaches [2,25,28, 34,56] for pruning redundant patch regions are hindered by three main limitations: (1) They often necessitate complex training of extra modules, which becomes increasingly difficult as baseline models evolve with more data, enhanced training methods, and deeper structures [25,41,56]. Re-training these modules for each model update is impractical. (2) Without a complete image understanding, these methods incrementally prune patches across the network, leading to unnecessary computations in early layers, which is particularly problematic for deeper models. (3) Many rely on specific architectural features for patch reduction, such as class tokens or attention maps, limiting their use to a narrow set of network designs [12,28]. Hence, there is a pressing need for a patch pruning solution that is adaptable to various modern architectures without additional training, can eliminate redundant patches in a single-step, thereby making it suitable for a broad spectrum of networks while streamlining each model update (See Fig. 1).

Recent work on patch reduction [2,23,55] focuses mostly on transformers rather than ConvNets due to their impressive performance on various tasks, their aforementioned limitations notwithstanding. While ConvNets achieve lower ImageNet-1k top-1 accuracy than large ViTs (68.7% in MobileOne-S0 [49] *vs.* 88.7% in ViT-Huge [44]), they exhibit a remarkable ability to efficiently process the key image regions with hierarchical inductive bias. Our empirical investigation reveals that, as we broaden the evaluation metric (increasing k in top-k evaluations, see Fig. 4), the benefits of deeper models diminish, especially with a large number of image classes (*e.g.*, 1000 in ImageNet). This suggests that shallower models excel at identifying discriminative areas as their bigger counterparts, rendering them ideal for patch pruning.

Leveraging this insight, we propose PaPr, a novel Patch Pruning method that employs pretrained lightweight ConvNets for efficient, one-step patch pruning in

a wide variety of deep learning models, maintaining accuracy while significantly cutting computational demands. Our findings show lightweight ConvNets have remarkable ability in identifying discriminative image regions but struggle in fine-grained prediction. To address this, PaPr relies on Patch Significance Maps (PSMs), which are generated using only the convolutional layers of ConvNets through uniform class weight recalibration in FC layers. Astonishingly, PSMs consistently highlight critical image regions across ConvNets of varying sizes and accuracies (Fig. 6), thereby amplifying the efficacy of ultra lightweight ConvNets.

ConvNets inherently preserve the positional property of patches for their inductive bias, while ViTs use cross-attention to blend patch features, leading to variability in patch relevance across models. Unlike previous methods that prune redundant patches gradually over multiple steps across intermediate layers of ViTs [2, 23, 41], our approach, PaPr, simplifies this process by eliminating non-essential patches at once, immediately after extraction (see Fig. 3), by leveraging lightweight ConvNets to assess patch significance. This direct, one-step pruning approach significantly cuts computational demands and is also compatible with other patch reduction techniques (see Fig. 5). PaPr's ability to separate crucial patch identification from fine-grained class prediction enhances a wide range of pre-trained models without further training, ensuring high accuracy with notable speed ups. By bypassing the complex training required for conventional patch selectors, PaPr capitalizes on the comprehensive capabilities of minimalist ConvNets for efficient patch pruning in larger models.

Our experiments demonstrate PaPr's effectiveness across various architectures and pre-training methods, achieving significant reduction in redundant patches for ViTs, large-scale ConvNets (*e.g.*, ConvNext [32]), and hybrid transformers (*e.g.*, Swin [31]), outperforming state-of-the-art (SOTA) patch reduction methods by a large margin. Notably, PaPr can be easily integrated with most existing patch reduction methods to reduce patches early in operation. More specifically, PaPr boosts ToMe [2] accuracy by 4.5% for a computational budget similar to ViT-B. Moreover, PaPr can accelerate training akin to token merging techniques, unlike most patch pruning methods that fail to boost training speed [34, 41, 56]. PaPr shows robust patch localization performance with ultra lightweight ConvNets ($<0.3\%$ accuracy loss) for $42\times$ reduction in proposal FLOPs, thereby enabling its use for larger *off-the-shelf* models. Remarkably, in video recognition, PaPr cuts down around 70% of redundant patches, resulting in up to $3.7\times$ FLOPs reduction with minimal impact on accuracy ($\approx 0.8\%$).

We summarize our main contributions as follows:

– We propose PaPr, a novel background patch pruning method that can seamlessly operate with ViTs, ConvNets, and hybrid transformers, without further training while leveraging batch processing.
– We propose a simple weight recalibration method in ConvNets to precisely and efficiently locate discriminative patches, irrespective of model size.
– We facilitate the use of ultra-lightweight ConvNets to speed-up large models, such as ViTs, with a seamless framework and negligible accuracy loss.

– We present extensive qualitative and quantitative results across numerous model architectures in both image and video applications.

2   Related Work

2.1   From ConvNets to Vision Transformers

ConvNets have been pivotal in computer vision, offering computational efficiency through kernel reuse and localization [8, 18–20, 29, 42, 46, 49, 59]. Yet, they struggle in capturing long-range dependencies, a gap bridged by ViT [10]. Inspired by the success of transformers from natural language processing (NLP), ViTs excel in long-range feature modeling through cross-attention, outperforming ConvNets albeit with higher computational demands and extensive training data requirements [7, 48, 51, 57]. Efforts to alleviate these issues include self-supervised [17, 47, 52, 60] and weakly-supervised [39, 44] pre-training, although challenges in computational complexity and optimization remain. Hybrid architectures, merging ConvNets' inductive biases with ViTs' cross-attention, offer a balanced solution by reducing computational load while maintaining performance [6, 9, 11, 15, 27, 30, 31, 33, 54]. Recent ConvNets advancements [32, 53], with improved training and large-scale data, now rival ViTs, questioning whether architectural innovations or enhanced training primarily drive performance gains.

Addressing the computational demands of these models, we focus on enhancing operational efficiency by pruning redundant patches without architectural modifications or re-training, maintaining performance while streamlining updates.

2.2   Class Activation Mapping for Explainable Deep Learning

Class Activation Mapping (CAM) techniques provide explainable visual reasoning for neural network predictions by highlighting activation regions[1] crucial for decisions [1, 21, 38, 43, 61]. While ConvNets utilize convolutional operations to preserve spatial information, making them apt for such visualizations, ViTs link attention weights to class tokens for prediction [4, 36]. However, CAM's reliance on accurate class predictions for effective feature localization is a significant drawback, as ConvNets' lower accuracy when compared to ViTs may lead to incorrect object localization. Despite ViTs' superior fine-grained classification capabilities, it is unclear if they prioritize the same discriminative regions as ConvNets. Additionally, CAM approaches often require gradient tracking [5, 43] or complex feature map decomposition [35], complicating batch processing and necessitating extra optimization steps.

Our work diverges from traditional CAM by aiming to consistently identify discriminative patches across different architectures, focusing computational resources on the most relevant areas without being constrained by class prediction accuracy. This approach enables more efficient processing by allowing larger models to concentrate on the most significant regions, identified by lighter networks, thus significantly reducing computational demands.

---

[1] "Patch" and "region" have been used interchangeably based on the context.

(a) Baseline Convolutional Model Feature Flow

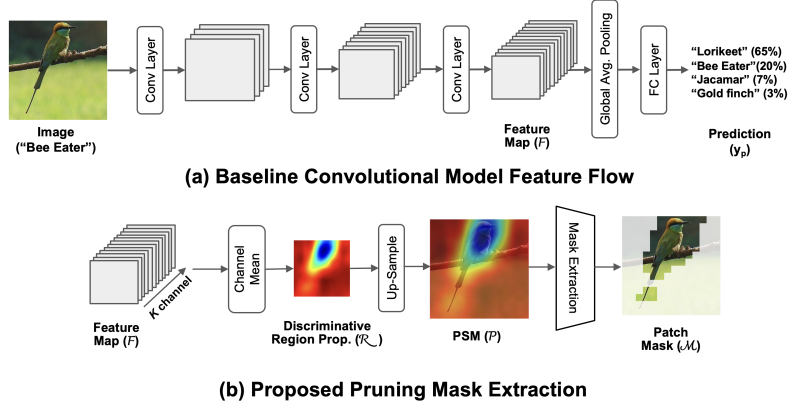

(b) Proposed Pruning Mask Extraction

Fig. 2: (a) Baseline ConvNet gradually reduces the feature map to produce $\mathcal{F} = \{f_k(x, y)\}_{k=1}^{K}$, followed by global average pooling and fully connected (FC) layers to predict y_p. (b) In PaPr, we operate on $\mathcal{F}$ by suppressing the FC layer. Initially, we extract pixel mean over K channels to produce discriminative region proposal $\mathcal{R}$. Later, simple upsampling operation generates the patch significance map (PSM) $\mathcal{P}$ of target dimension. Finally, patch mask $\mathcal{M}$ for top $z\%$ patches is obtained from $\mathcal{P}$.

2.3   Patch Reduction for Faster Inference

Several approaches have sought to enhance computational efficiency by reducing redundant patches in neural networks, with early strategies involving additional adapters or controllers to identify and prune less significant patches [24, 26, 41, 58]. These methods, however, necessitate separate adapter training for each network and are slow to adapt due to the need to learn from the dynamics of other layers. In the context of ViTs, efforts have leveraged architectural features such as class tokens and attention maps for patch relevance, yet these solutions often fail to generalize across various architectures like hybrid transformers [31] or larger ConvNets [32, 53], thus limiting their applicability [12, 25, 34, 56]. Furthermore, while some have explored patch merging in ViTs and a combination of merging and pruning [2, 3, 23], these approaches lack a holistic image understanding by not considering the entire image context in their optimizations, leading to incremental and sub-optimal patch reduction.

Our work diverges from these traditional patch reduction methods by proposing a single-step, early-network patch removal strategy, that seamlessly integrates with any architecture without re-training.

3   Methodology

Our methodology introduces a novel approach to discriminative patch pruning across various deep learning architectures, leveraging the innate capabilities of lightweight ConvNets. By generating a PSM, we efficiently identify and prune non-essential patches in a single step, enhancing computational efficiency without compromising accuracy. This process is universally applicable, seamlessly

integrating with ViTs, ConvNets, and hybrid models, thereby addressing the limitations of previous methods with a scalable, architecture-agnostic solution.

3.1   Extracting Discriminative Regions with ConvNets

Despite achieving relatively lower top-1 accuracy on the ImageNet benchmark, lightweight ConvNets exhibit a competitive edge in top-10 accuracy when compared to their larger ViT counterparts (Fig. 4). This phenomenon underscores ConvNets' capability to effectively localize regions of interest through their convolutional layers, despite potential limitations in fine-grained classification attributed to the fully-connected (FC) layer. Our methodology leverages this insight by proposing discriminative patch regions while minimizing the influence of the FC layer (see Fig. 2), thereby enhancing the large *off-the-shelf* model's focus on most salient image regions proposed by lightweight ConvNets (see Fig. 3).

Given an input image $X \in \mathbb{R}^{H \times W \times 3}$, with height H and width W, we denote by $f_k(x, y)$ the feature map generated by the kth kernel in the last convolutional layer of typical ConvNets, $\forall k \in \{1, 2, \ldots, K\}$. Here, each pixel (x, y) in $f_k(x, y)$ corresponds to a patch window of size $(H/d \times W/d)$ in the input image X, reflecting a spatial down-scaling factor of d through the convolutional layer stack. Typically, global average pooling (GAP) is applied to each feature map $f_k(x, y)$, yielding $F = \{F_k\}_{k=1}^{K} \in \mathbb{R}^K$, where $F_k = \sum_{x,y} f_k(x, y) \in \mathbb{R}$. Finally, a fully connected (FC) layer $W \in \mathbb{R}^{C \times K}$ with elements w_c^k processes F to generate class predictions $y_p \in \mathbb{R}^C$ across C classes as follows:

$$y_p = WF = \sum_c \sum_k w_c^k \sum_{x,y} f_k(x, y) = \sum_{x,y} R(x, y), \tag{1}$$

where $R \in \mathbb{R}^{h \times w}$ encapsulates the weighted mean class activation mapping of the convolutional feature map.

We note that the reliance on class activation weights w_c^k is influenced by the model's final accuracy, thereby posing a challenge for smaller models. Our objective transcends mere accuracy enhancement, aiming to precisely locate discriminative patches rich in information irrespective of the model size or its final classification performance. Recognizing that discriminative region localization is pivotal for detailed classification, and acknowledging the competitive top-10 accuracy of lighter ConvNets, we posit that an extremely lightweight ConvNet suffices for initial discriminative region proposal.

To counteract the influence of the weak linear FC layer W in convolutional region proposal, we propose an adjustment where $w_c^k = 1/KC$, $\forall k \in \{1, \ldots, K\}$, facilitating the generation of a discriminative region proposal $\mathcal{R} \in \mathbb{R}^{h \times w}$, where

$$\mathcal{R}(x, y) = \sum_c \sum_k \frac{1}{KC} f_k(x, y) = \frac{1}{K} \sum_k f_k(x, y). \tag{2}$$

This strategy allows us to leverage ConvNets for what they excel at: pinpointing critical image areas. By reducing reliance on class activation weights, we

efficiently generate discriminative region proposals directly from convolutional outputs. This approach not only enhances the interpretability and efficiency of the localization process but also enables the application of more complex models for subsequent detailed analysis, optimizing the use of computational resources.

3.2   Patch Significance Map

With the discriminative region proposal $\mathcal{R} \in \mathbb{R}^{h \times w}$, where each pixel (x, y) quantifies the significance of corresponding patches in the original image, we proceed to establish a precise mapping to our intended feature map (Fig. 2). This mapping ensures the preservation of spatial relationships within the feature map.

Consider a target feature map $\mathcal{F} \in \mathbb{R}^{h' \times w' \times K}$, with each pixel (x', y') encapsulating a feature vector from a specific image patch. To align $\mathcal{R}$ with $\mathcal{F}$, we employ an upsampling operation $U : \mathbb{R}^{h \times w} \to \mathbb{R}^{h' \times w'}$, transforming $\mathcal{R}$ into the Patch Significance Map (PSM) $\mathcal{P} \in \mathbb{R}^{h' \times w'}$. Consequently, each element of $\mathcal{P}$ directly corresponds to the patch significance within $\mathcal{F}$ for the given discriminative task. The next step involves utilizing $\mathcal{P}$ to discern and prune non-essential patch features from $\mathcal{F}$. By sorting the values within $\mathcal{P}$, we acquire a pruning mask $\mathcal{M}$:

$$\mathcal{M} = \text{reshape}(\text{argsort}(\text{flatten}(\mathcal{P}))), \mathcal{M} \in \mathbb{R}^{h' \times w'}, \tag{3}$$

enabling batch-wise patch pruning. Specifically, we identify the indices corresponding to the top-$z\%$ patches as per $\mathcal{M}$, facilitating the retention of only the most salient patches in $\mathcal{F}$, thereby enhancing computational efficiency in subsequent processing stages.

As part of our discussion on various architectures, we detail the application of this patch reduction technique. For enhanced visualization of the PSM, we apply *min-max* normalization to $\mathcal{P}$, adjusting for outliers and scaling the significance scores to visually depict the importance of different patches.

3.3   Integrating PSM with Vision Transformers

The ViT processes an input image $X \in \mathbb{R}^{H \times W \times 3}$ by extracting N-dimensional patch token features $X_p \in \mathbb{R}^{N \times d}$ from a $(k \times k)$ patch window using strided convolutions, where $N = HW/k^2$ (Fig. 3). Positional embeddings are then added to these patch tokens X_p to retain spatial information, followed by the application of successive cross-attention mechanisms. For classification tasks, the model either introduces a class token $t_{cls} \in \mathbb{R}^d$ or utilizes the mean of the output patch tokens.

The encoder employs multi-headed cross-attention (MHA) on the patch token embeddings X_p, with each MHA block consisting of multiple linear layers and a cross-attention layer. The computational complexity of these operations suggests that reducing the number of tokens N can significantly decrease computational demands. A key strength of the ViT architecture is its adaptability to varying numbers of tokens N, though at a higher computational cost. Prior work [2,41,56] has primarily aimed at reducing patch tokens within intermediate encoder layers, often requiring additional training due to the variability in token representations
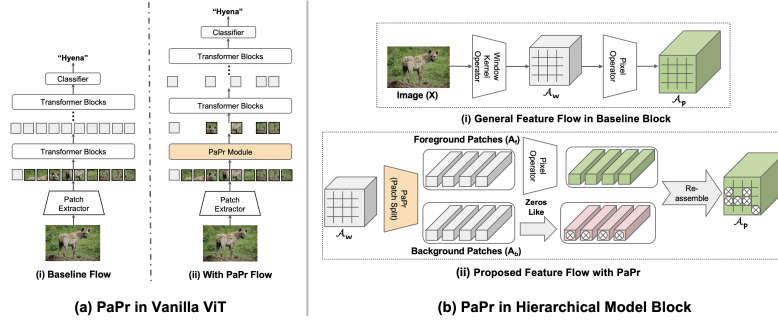
Fig. 3: (a) In vanilla ViT, PaPr operates right after the patch extractor module. Hence, all transformer blocks can operate only with the most discriminative patches. (b) Hierarchical model blocks comprise of window based kernel operator (*e.g.*, Conv$k \times k$/local attention), followed by pixel operator (*e.g.*, linear layer, Conv1x1). Pixel operator consumes more than 60% of total computation. PaPr is used to split the foreground patches to be operated with pixel operator. Background patches are zero-ed out, and finally, re-assembled with foreground output patches.

across different architectures. These methods typically achieve only gradual token reduction, limited by the distributed nature of information across tokens.

In contrast, our approach seeks to eliminate redundant tokens early in the processing pipeline, immediately following initial patch extraction. This strategy offers multiple advantages: (1) Leveraging the convolutional nature of the initial patch extraction allows for a direct mapping between our PSM $\mathcal{P}$ and the patch tokens X_p, obviating the need for additional mechanisms to determine token redundancy. (2) It facilitates generalizability across transformer models without the need for retraining, potentially accelerating training by concentrating on essential patches. (3) It permits seamless integration of existing intermediate layer token reduction techniques following our initial patch pruning process.

3.4   Integrating PSM with Hierarchical Models

ConvNets and hybrid transformers, such as Swin [31], primarily operate with window-based local operations in contrast to full-attention based operation in vanilla transformers. Such window-based local processing makes patch pruning particularly complicated, in contrast to vanilla transformers. However, these models maintain the location property of representative image patches all through the network, which leaves the door open to prune redundant patches. Nevertheless, the window-based operation is particularly difficult to prune.

In general, each block of hierarchical layers consists of window-based spatial kernel/attention operators, followed by pixel operators (usually, modeled with 1×1 convolutional layers or linear layers). In contrast to windowed convolutions or cross-attention, these pixel operators are particularly suitable for patch pruning. Interestingly, in the SOTA ConvNets and hybrid transformers, more than 60% of total computations are performed with such pixel operators (63.3% in Swin [31],
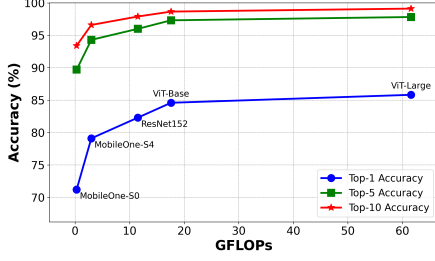
Fig. 4: ImageNet-1k evaluation for varying top-k accuracy targets. The accuracy gain with bigger model largely shrinks, as k increases. This suggests shallower ConvNets have understanding of object locations and visual property, despite their lower top-1 accuracy.
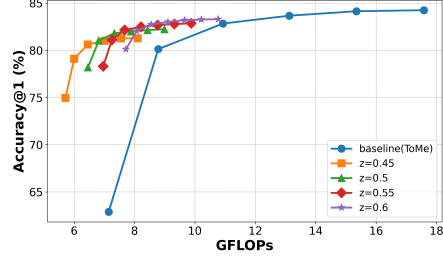
Fig. 5: Integrating PaPr with ToMe [2]. We use the Augreg pretrained ViT-B-16 architecture as the baseline. We sweep token merging ratio (r) for different pruning ratio (z). Integration of PaPr achieves Pareto-optimal performance, thus, PaPr can enhance existing patch reduction methods.

96.2% in ConvNext [32]). Based on the patch significance map from PaPr, we only use the pixel operator on most significant patches, and simply perform zero-padding on the remaining patch regions. The zero padding operations are performed to mostly recover the feature map shape to be used with subsequent spatial operators. Hence, the speed-up is mostly achieved by eliminating redundant computations in pixel operators (See Fig. 3).

Let's assume that the feature map before applying pixel operators is given by $\mathcal{A}_w$. Based on the PSM $\mathcal{P}$, we initially identify the foreground and background pixel features as A_f, and A_b, respectively. Later, the pixel operator layers are applied on foreground pixels A_f, and a zero-padded representation is used for the background pixels A_b. Finally, the output representation $\mathcal{A}_c$ is reassembled with modulated foreground pixels A_f and zero-padded background pixels A_z, given by Eq. (4):

$$A_f, A_b = \text{Split}(\mathcal{A}_w, \mathcal{P}),$$
$$A_z = \text{Zeros}(A_b), A_f = \text{Linear}(A_f), \tag{4}$$
$$\mathcal{A}_c = \text{Reassemble}(A_f, A_z).$$

4 Image Experiments

4.1 Experimental Setup

We experiment on image classification task on the ImageNet-1k benchmark dataset, following prior work [2, 12, 23, 34, 55]. We use the MobileOne-s0 [49] model as the proposal model for all architectures, unless otherwise specified. We report training-free results of PaPr, unless otherwise specified. For the performance metric, we report top-1 accuracy, GFLOPs, and the throughput (img/s). Throughput is measured on a single RTX-A5000 GPU with 24GB VRAM. We reproduced all baseline models under the same setup for a fair comparison.

Table 1: Performance comparison on Au-gReg models. PaPr achieves the best performance, while operating with token merging. z denotes patch keeping ratio in PaPr.

Table 2: Performance comparison on MAE models. Since MAE uses masked pretraining, PaPr is particularly suitable for MAE inference. PaPr achieves significantly higher performance than others. z denotes patch keeping ratio in PaPr.

Models	Methods	Acc1	GFLOPs	Img/s
	Baseline	81.39	4.61	975
	ToMe [2]	76.96	2.29	1978
	TokenFusion [23]	77.12	2.29	1982
ViT-S-16	GTP-ViT [55]	71.03	2.31	1970
	PaPr (z=0.45)	76.21	2.29	1988
	PaPr (z=0.55) (+ToMe)	**77.76**	2.19	2073
	PaPr (z=0.5) (+ToMe)	76.27	**1.97**	**2315**
	Baseline	84.59	17.59	307
	ToMe [2]	80.38	8.78	615
	TokenFusion [23]	80.7	8.78	618
ViT-B-16	GTP-ViT [55]	80.98	8.78	610
	PaPr (z=0.5)	82.11	8.98	605
	PaPr (z=0.55) (+ToMe)	**82.34**	8.21	660
	PaPr (z=0.5) (+ToMe)	80.88	**6.82**	**785**
	Baseline	85.82	61.61	91
	ToMe [2]	83.5	30.99	180
	TokenFusion [23]	83.91	30.99	182
ViT-L-16	GTP-ViT [55]	81.56	30.99	181
	PaPr (z=0.5)	83.87	30.83	183
	PaPr (z=0.55) (+ToMe)	**83.99**	26.33	210
	PaPr (z=0.55) (+ToMe)	83.5	**25.12**	**224**

Models	Methods	Acc1	GFLOPs	Img/s
	Baseline	83.74	17.59	307
	ToMe [2]	78.82	8.78	615
ViT-B-16	TokenFusion [23]	79.23	8.78	618
	GTP-ViT [55]	79.14	8.78	610
	PaPr (z=0.5)	**82.4**	8.98	605
	PaPr (z=0.4)	81.4	**7.72**	**700**
	Baseline	85.95	61.61	91
	ToMe [2]	84.24	30.99	180
ViT-L-16	TokenFusion [23]	84.33	30.99	182
	GTP-ViT [55]	84.15	30.99	181
	PaPr (z=0.5)	**85.06**	30.83	183
	PaPr (z=0.4)	84.76	**27.72**	**201**
	Baseline	86.89	167.4	36
	ToMe [2]	85.48	82.53	72
ViT-H-16	TokenFusion [23]	85.71	82.53	73
	GTP-ViT [55]	85.54	82.53	71
	PaPr (z=0.5)	**86.4**	83.04	71
	PaPr (z=0.4)	86.13	**74.59**	**81**

4.2   Performance on Various Vision Transformers

We study performance of PaPr in diverse ViT architectures, along with various pre-training methods. We also present training results, along with training-free results to compare with existing methods.

Training-free method comparison. We consider two different pre-training methods for comparative analysis on various ViT architectures. We use supervised pretrained Augreg models [45], and self-supervised pre-trained masked autoencoder (MAE) models [17], following prior work [2]. We compare with three recent training-free patch reduction methods, such as ToMe [2], TokenFusion [23], and GTP-ViT [55]. Additionally, we show the performance comparison on class-token free ViT models to highlight the architecture agnostic performance. In general, PaPr achieves better accuracy, with lower computational costs compared to state-of-the-art (SOTA) methods.

Augreg models: For AugReg pre-training with ViT-Base, PaPr achieves 2.14% higher accuracy over ToMe, with comparable FLOPs as shown in Tab. 1. In addition, by combining PaPr with ToMe, we achieve additional 22.3% FLOP reduction, while maintaining higher accuracy. We further study the compatibility of PaPr with existing patch reduction methods as ToMe. After the initial patch pruning with PaPr, we integrate ToMe only at the bottom-half layers due to its higher sensitivity in earlier layers. As shown in Fig. 5, integration of PaPr can directly boost ToMe performance thereby achieving Pareto optimality.

Table 3: Performance analysis on class-token free ViT models. PaPr performance gain is not limited to specific architectures.

Model	Method	Acc1	GFLOPs	Img/s
ViT-Medium GAP-16 256	Baseline	84.33	10.58	484
	GTP-ViT [55]	80.5	5.73	880
	ToMe [2]	80.21	5.73	890
	PaPr (z=0.5)	**81.8**	**5.5**	**932**
ViT-Medium GAP-16 384	Baseline	85.6	26.06	182
	GTP-ViT [55]	81.95	13.59	340
	ToMe [2]	82.5	13.59	344
	PaPr	**83.95**	**12.94**	**360**

Table 4: Training performance analysis on DeIT-s. PaPr achieves competitive performance with higher training speed.

Methods	Acc1	GFLOPs	Im/s	Batch Proc.	Train Speed
Baseline	79.8	4.6	960	✓	1x
DynamicViT [41]	79.3	2.9	1510	✗	1x
A-ViT [56]	78.6	2.9	-	✗	1x
ATS [12]	**79.5**	2.9	1512	-	1x
SP-ViT [25]	79.3	2.6	-	✗	1x
ToMe [2]	79.4	2.7	1575	✓	1.5x
TokenFusion [23]	**79.5**	2.7	1580	✓	1.5x
PaPr (z=0.55)	79.2	2.7	**1585**	✓	**1.6x**

Table 5: Performance comparison on ConvNext CNN models. PaPr can achieve competitive performance without training, and can seamlessly adapt to model updates.

Models	Methods	Train-Free	Acc1	GFLOPs	Img/s
ConvNeXt Base-1k	Baseline	N/A	83.84	15.38	265
	DynCNN [40]	✗	83.08	10.21	375
	PaPr (z=0.65)	✓	82.75	10.42	**395**
ConvNeXt Base-22k	Baseline	N/A	85.81	15.38	265
	PaPr (z=0.65)	✓	84.27	10.42	395
ConvNeXt Large-1k	Baseline	N/A	84.31	34.4	135
	PaPr (z=0.65)	✓	83.26	22.9	203
ConvNeXt Large-22k	Baseline	N/A	86.61	34.4	135
	PaPr (z=0.65)	✓	**85.67**	22.9	203

Table 6: Performance comparison on Swin hybrid transformer models. PaPr can adapt to much bigger models with higher operating resolutions without training.

Models	Methods	Train-Free	Acc1	GFLOPs	Img/s
Swin B-1k	Baseline	N/A	83.42	15.47	258
	DynSwin [40]	✗	83.18	12.1	**327**
	PaPr (z=0.65)	✓	81.7	12.25	325
Swin B-22k	Baseline	N/A	85.16	15.47	258
	PaPr (z=0.65)	✓	82.27	12.25	325
Swin L-22k	Baseline	N/A	86.25	34.53	135
	PaPr (z=0.65)	✓	84.53	26.96	175
Swin L-22k-384	Baseline	N/A	87.25	104.08	42
	PaPr (z=0.65)	✓	**86.47**	81.44	54

MAE models: MAE used self-supervised pre-training by reconstructing masked image patches to train large ViT models [17]. Later, the model is fine-tuned with full resolution images without masking, which cannot exploit its latent ability to learn from fewer patches. Interestingly, PaPr introduces masked inference, which is particularly suitable for MAE models. Hence, we observe significantly higher accuracy with PaPr compared to other training-free methods, *e.g.*, with ViT-B, PaPr achieves 4.5% higher accuracy over ToMe for similar FLOPs (see Tab. 2).

Class token-free models: Several existing patch reduction methods operate with the class token in ViTs to evaluate the patch relevance [12,28,34]. However, instead of class tokens, global pooling of patch tokens is used in many cases [16, 37]. Results presented in Tab. 3 demonstrate the superiority of PaPr for such cases.

Training-based methods comparison. We compare training performance of existing methods for the patch reduction on DeIT-s [48] model in Tab. 4. Prior pruning based methods cannot speed-up the training for learning the mask predictor [25, 41, 56]. However, PaPr achieves competitive performance as prior methods, while achieving large speed-up as other token merging methods [2,23,55].

Table 7: Sweeping ConvNet proposal model in PaPr with $z = 0.5$. PaPr can achieve similar performance irrespective of proposal model size. Thus, PaPr can use a much smaller and faster proposal model to speed-up larger models.

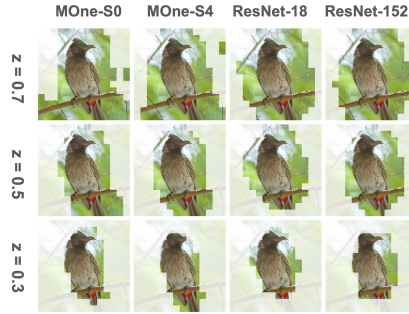| Proposal		Accuracy1(%)	
Model	GFLOPs	ViT-B-16	ViT-L-16
ResNet-18 [18]	1.81	81.1	83.84
ResNet-50 [18]	4.09	82.33	84.09
ResNet-152 [18]	11.51	**82.51**	84.08
MobileOne-S0 [49]	0.27	82.24	84.06
MobileOne-S2 [49]	1.35	82.28	84.16
MobileOne-S4 [49]	2.98	82.35	**84.32**

Fig. 6: Proposal models locate similar patches with PaPr for different pruning ratio (z) irrespective of model size.

Table 8: Training-free performance comparison on Kinetics-400 video evaluation. Each video has 16×224^2 input size. PaPr achieves significantly better performance for reducing spatio-temporal redundancy in videos.

Model	Method	Acc1	GFLOPs	Views
XViT	ATS [12]	80.0	259	1x3
TimeSformer-L	ATS [12]	80.5	3510	1x3
ViT-B MAE	Baseline	81.21	180	3x5
	PaPr(z=0.45)	**81.18**	76	3x5
	PaPr(z=0.35)	80.15	**59**	3x5
ViT-L MAE	Baseline	85.26	598	3x5
	ToMe [2]	84.5	281	1x10
	PaPr(z=0.5)	**85.12**	275	3x5
	ToMe [2]	82.5	184	1x10
	PaPr(z=0.3)	84.53	**160**	3x5

Table 9: Proposal ConvNet sweep in video recognition with ViT-B-MAE model as the baseline. PaPr achieves competitive performance with lighter proposal models for different z values.

Proposal		Keep Ratio(z)	Acc1 (%)	Views
Model	GFLOPs			
X3d-s [13]	1.25	0.5	81.11	3x5
X3d-m [13]	2.45	0.5	**81.19**	3x5
ResNet-50 [14]	41.9	0.5	79.96	3x5
ResNet-101 [14]	85.67	0.5	79.97	3x5
X3d-s [13]	1.25	0.4	80.86	3x5
X3d-m [13]	2.45	0.4	80.93	3x5
ResNet-50 [14]	41.9	0.4	78.85	3x5
ResNet-101 [14]	85.67	0.4	79.32	3x5

4.3 Performance on Various Hierarchical Models

We study the performance on two variants of hierarchical models, such as pure convolutional models, and hybrid transformer models. We compare with training based DynamicCNN [40], and DynamicSwin [40] methods (trained for 120 epochs), whereas PaPr operates without training.

Analysis on convolutional models. We use the SOTA ConvNext [32] architecture for the analysis as shown in Tab. 5. The training-free operation in PaPr makes it seamlessly usable with new model updates. We analyze ImageNet-1k and ImageNet-22k performance of same models. For the ConvNext-Base model, PaPr achieves 99.6% accuracy of DynamicCNN with 7.1% higher throughput. By simply using ImageNet-22k weights, PaPr can achieve additional 2% accuracy improvement without training, while having the same computation cost.
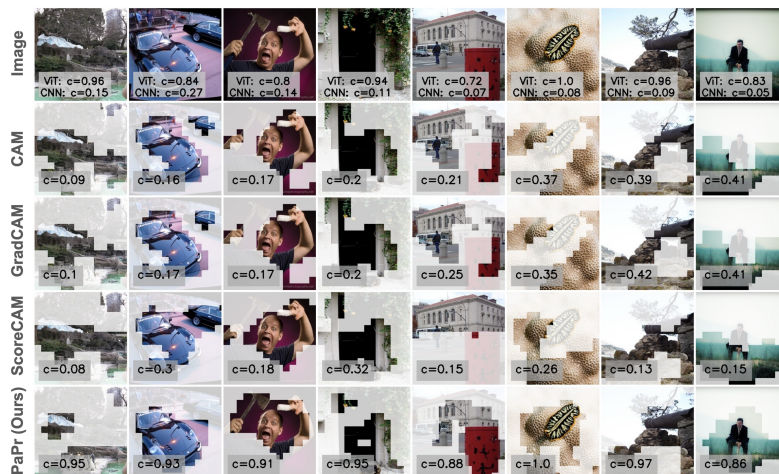
Fig. 7: Robustness of PaPr compared to CAM based methods. PaPr can perform even when the ConvNet proposal confidence (c) is very low. In contrast, existing CAM based methods fail in such cases, despite being significantly slower while not enabling batch processing and use of gradients in some cases.

Analysis on hybrid transformer models. We study the hierarchical Swin transformer models for patch reduction with PaPr, as given in Tab. 6. PaPr achieves 98% of DynamicSwin accuracy using similar FLOPs without re-training. Nevertheless, by leveraging bigger models, larger pre-training, and higher resolution, PaPr can seamlessly adapt to achieve higher accuracy.

4.4   Robustness of PaPr across various ConvNet proposals

PaPr can operate with ultra-lightweight ConvNets for generating robust proposals. We study different ConvNet architectures for proposal generation, as shown in Tab. 7. Interestingly, PaPr shows small reductions of final accuracy (0.3%), when using MobileOne-S0 based proposals compared to ResNet-152 (42$\times$ higher FLOPs) in ViT-B. When visualizing the PSM for different pruning ratio as in Fig. 6, we notice the similar PSMs irrespective of model size. Hence, PaPr can utilize ultra-lightweight ConvNets to mask patches without sacrificing accuracy.

4.5   Comparison with Class Activation Mappings (CAMs)

Existing CAM methods mostly focus on explainability to highlight image regions responsible for final prediction. However, such objectives rely on final prediction performance, which can be much lower for light ConvNets. Moreover, for localization, many of these methods use gradients, and optimization methods that limit batch processing. Nevertheless, we study the impact of such CAM methods ([43, 50, 61]) in PaPr framework, in challenging scenarios where the baseline ViT-B has higher prediction confidence (c) on target class, and the proposal

Fig. 8: Visualization of PaPr localization in videos. Video has inherent high sparsity. PaPr effectively localizes the discriminative regions for using holistic spatio-temporal understanding with small ConvNets. Thus, it significantly reduces the computational burden for larger models in downstream video recognition tasks.

MobileOne-S0 has lower confidence (see Fig. 7). In most cases, the baseline CAM method significantly lowers the final accuracy after patch pruning. In contrast, PaPr maintains robust confidence for its precise localization. Moreover, PaPr can enhance confidence of baseline models in several scenarios by suppressing the redundant patches, *e.g.*, in Sample 5, the c increased by 22% with PaPr.

5   Video Experiments

We study the training-free performance on Kinetics-400 [22] validation sets, as shown in Tab. 8. We use SOTA ATS [12], and ToMe [2] methods for comparison. We use PaPr on SOTA ViT-MAE [47] models with lightweight X3d-s [13] model for proposal generation. We follow the baseline [47] 3×5 view (3 spatial view and 5 temporal view) approach, and separate the views from model FLOP counts similar to other works as it can be arbitrarily chosen. Since ViT-MAE removes class tokens in fine-tuning, ATS [12] cannot be adapted to these models. For ViT-L, PaPr achieves $3.7 \times$ FLOPs reduction of the baseline model for a negligible 0.8% accuracy drop. We also study the impact of various proposal models on the final performance, as shown in Tab. 9. We use SlowOnly-ResNet [14] and X3d [13] models for comparison. Increasing model size for different pruning shows minimal impact on final performance. Finally, we visualize the patch masking of PaPr in videos, as shown in Fig. 8. By removing the redundant details, PaPr significantly boosts the performance of bigger models for fine-grained predictions.

6   Conclusion and Future Work

In this paper, we introduce a novel patch pruning method, namely PaPr, that can effectively speed-up *off-the-shelf* pre-trained models inference without re-training. We propose a simple modification of ConvNets, that allows extracting a precise discriminative PSM from ultra lightweight ConvNets with unparalleled speed. In PaPr, such PSMs are leveraged to directly reduce the redundant data regions in a single-step to guide larger model computations on most discriminative regions for achieving computational efficiency. Moreover, PaPr can be easily integrated with most existing patch reduction methods due to its simple structure.

Acknowledgement

References

1. Belharbi, S., Sarraf, A., Pedersoli, M., Ben Ayed, I., McCaffrey, L., Granger, E.: F-cam: Full resolution class activation maps via guided parametric upscaling. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3490–3499 (2022)
2. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster. In: The Twelfth International Conference on Learning Representations (2023)
3. Cao, Q., Paranjape, B., Hajishirzi, H.: Pumer: Pruning and merging tokens for efficient vision language models. arXiv preprint arXiv:2305.17530 (2023)
4. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9650–9660 (2021)
5. Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In: 2018 IEEE winter conference on applications of computer vision (WACV). pp. 839–847. IEEE (2018)
6. Chen, Y., Liu, H., Yin, H., Fan, B.: Building vision transformers with hierarchy aware feature aggregation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 5908–5918 (October 2023)
7. Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., Shen, C.: Twins: Revisiting the design of spatial attention in vision transformers. Advances in Neural Information Processing Systems **34**, 9355–9366 (2021)
8. Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., Sun, J.: Repvgg: Making vgg-style convnets great again. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13733–13742 (2021)
9. Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., Chen, D., Guo, B.: Cswin transformer: A general vision transformer backbone with cross-shaped windows. In: CVPR (2022)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2020)
11. Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., Feichtenhofer, C.: Multiscale vision transformers. In: ICCV (2021)
12. Fayyaz, M., Kouhpayegani, S.A., Jafari, F.R., Sommerlade, E., Joze, H.R.V., Pirsiavash, H., Gall, J.: Ats: Adaptive token sampling for efficient vision transformers. In: ECCV (2022)
13. Feichtenhofer, C.: X3d: Expanding architectures for efficient video recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 203–213 (2020)
14. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6202–6211 (2019)

15. Graham, B., El-Nouby, A., Touvron, H., Stock, P., Joulin, A., Jégou, H., Douze, M.: Levit: a vision transformer in convnet's clothing for faster inference. In: ICCV (2021)
16. Hatamizadeh, A., Yin, H., Heinrich, G., Kautz, J., Molchanov, P.: Global context vision transformers. In: International Conference on Machine Learning. pp. 12633–12646. PMLR (2023)
17. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: CVPR (2022)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
19. Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al.: Searching for mobilenetv3. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1314–1324 (2019)
20. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
21. Jung, H., Oh, Y.: Towards better explanations of class activation mapping. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1336–1344 (2021)
22. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv:1705.06950 [cs.CV] (2017)
23. Kim, M., Gao, S., Hsu, Y.C., Shen, Y., Jin, H.: Token fusion: Bridging the gap between token pruning and token merging. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1383–1392 (2024)
24. Kim, S., Shen, S., Thorsley, D., Gholami, A., Kwon, W., Hassoun, J., Keutzer, K.: Learned token pruning for transformers. arXiv:2107.00910 [cs.CL] (2021)
25. Kong, Z., Dong, P., Ma, X., Meng, X., Niu, W., Sun, M., Ren, B., Qin, M., Tang, H., Wang, Y.: Spvit: Enabling faster vision transformers via soft token pruning. In: ECCV (2022)
26. Lassance, C., Maachou, M., Park, J., Clinchant, S.: A study on token pruning for colbert. arXiv:2112.06540 [cs.CL] (2021)
27. Li, Y., Wu, C.Y., Fan, H., Mangalam, K., Xiong, B., Malik, J., Feichtenhofer, C.: Mvitv2: Improved multiscale vision transformers for classification and detection. In: CVPR (2022)
28. Liang, Y., Ge, C., Tong, Z., Song, Y., Wang, J., Xie, P.: Not all patches are what you need: Expediting vision transformers via token reorganizations. ICLR (2022)
29. Lin, J., Chen, W.M., Cai, H., Gan, C., Han, S.: Memory-efficient patch-based inference for tiny deep learning. Advances in Neural Information Processing Systems **34**, 2346–2358 (2021)
30. Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., et al.: Swin transformer v2: Scaling up capacity and resolution. In: CVPR (2022)
31. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV (2021)
32. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11976–11986 (2022)

33. Mehta, S., Rastegari, M.: Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. In: ICLR (2021)
34. Meng, L., Li, H., Chen, B.C., Lan, S., Wu, Z., Jiang, Y.G., Lim, S.N.: Adavit: Adaptive vision transformers for efficient image recognition. In: CVPR (2022)
35. Muhammad, M.B., Yeasin, M.: Eigen-cam: Class activation map using principal components. In: 2020 international joint conference on neural networks (IJCNN). pp. 1–7. IEEE (2020)
36. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
37. Pan, Z., Zhuang, B., Liu, J., He, H., Cai, J.: Scalable vision transformers with hierarchical pooling. In: Proceedings of the IEEE/cvf international conference on computer vision. pp. 377–386 (2021)
38. Patro, B.N., Lunayach, M., Patel, S., Namboodiri, V.P.: U-cam: Visual explanation using uncertainty based class activation maps. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7444–7453 (2019)
39. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
40. Rao, Y., Liu, Z., Zhao, W., Zhou, J., Lu, J.: Dynamic spatial sparsification for efficient vision transformers and convolutional neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)
41. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.J.: Dynamicvit: Efficient vision transformers with dynamic token sparsification. Advances in neural information processing systems **34**, 13937–13949 (2021)
42. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018)
43. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: ICCV (2017)
44. Singh, M., Gustafson, L., Adcock, A., de Freitas Reis, V., Gedik, B., Kosaraju, R.P., Mahajan, D., Girshick, R., Dollár, P., van der Maaten, L.: Revisiting weakly supervised pre-training of visual perception models. In: CVPR (2022)
45. Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., Beyer, L.: How to train your vit? data, augmentation, and regularization in vision transformers. TMLR (2022)
46. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: ICML (2019)
47. Tong, Z., Song, Y., Wang, J., Wang, L.: Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. arXiv:2203.12602 [cs.CV] (2022)
48. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: ICML (2021)
49. Vasu, P.K.A., Gabriel, J., Zhu, J., Tuzel, O., Ranjan, A.: Mobileone: An improved one millisecond mobile backbone. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7907–7917 (2023)
50. Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., Mardziel, P., Hu, X.: Score-cam: Score-weighted visual explanations for convolutional neural networks.

In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. pp. 24–25 (2020)

51. Wang, S., Li, B.Z., Khabsa, M., Fang, H., Ma, H.: Linformer: Self-attention with linear complexity. arXiv preprint arXiv:2006.04768 (2020)

52. Wei, C., Fan, H., Xie, S., Wu, C.Y., Yuille, A., Feichtenhofer, C.: Masked feature prediction for self-supervised visual pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14668–14678 (2022)

53. Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I.S., Xie, S.: Convnext v2: Co-designing and scaling convnets with masked autoencoders. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16133–16142 (2023)

54. Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., Zhang, L.: Cvt: Introducing convolutions to vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 22–31 (2021)

55. Xu, X., Wang, S., Chen, Y., Zheng, Y., Wei, Z., Liu, J.: Gtp-vit: Efficient vision transformers via graph-based token propagation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 86–95 (2024)

56. Yin, H., Vahdat, A., Alvarez, J.M., Mallya, A., Kautz, J., Molchanov, P.: A-vit: Adaptive tokens for efficient vision transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10809–10818 (2022)

57. You, H., Xiong, Y., Dai, X., Wu, B., Zhang, P., Fan, H., Vajda, P., Lin, Y.C.: Castling-vit: Compressing self-attention via switching towards linear-angular attention at vision transformer inference. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14431–14442 (2023)

58. Yu, H., Wu, J.: A unified pruning framework for vision transformers. arXiv:2111.15127 [cs.CV] (2021)

59. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6848–6856 (2018)

60. Zhou, A., Li, Y., Qin, Z., Liu, J., Pan, J., Zhang, R., Zhao, R., Gao, P., Li, H.: Sparsemae: Sparse training meets masked autoencoders. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16176–16186 (2023)

61. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2921–2929 (2016)