Article

Not peer-reviewed version

Efficient Patch Pruning for Vision Transformers via Patch Similarity

Haoyu Han and Shui Xiuying *

Posted Date: 8 April 2025

doi: 10.20944/preprints202504.0596.v1

Keywords: Vision Transformers; Patch Pruning; Patch Similarity; Computational Efficiency; Redundancy Reduction; Self-Attention; Image Classification; Feature Representation; Resource-Constrained Devices

 doi:10.20944/preprints202504.0596.v1

Article

Efficient Patch Pruning for Vision Transformers via Patch Similarity

Haoyu Han and Shui Xiuying *

Harbin Institute of Technology, China; haoyu.han@hit.edu.cn
* Correspondence: shui.xiuying@hit.edu.cn

Abstract: Vision Transformers (ViTs) have emerged as a powerful alternative to convolutional neural networks (CNNs) for visual recognition tasks due to their ability to model long-range dependencies in images through self-attention. However, the computational complexity and memory consumption of ViTs scale quadratically with the number of input patches, making them inefficient, especially for high-resolution images. In this work, we propose a simple yet effective method for patch pruning based on patch similarity, aimed at improving the efficiency of ViTs without compromising their performance. The core idea is to selectively prune patches that exhibit high similarity, reducing redundant information processing while preserving crucial spatial and contextual information. First, we compute a similarity matrix between patches using a distance measure derived from their feature representations. Based on this similarity measure, we identify clusters of highly similar patches, which are subsequently pruned in a manner that minimizes information loss. We show that pruning patches with high redundancy leads to a more compact representation while maintaining the overall performance of the ViT in various image classification tasks. We further explore the impact of different similarity thresholds and pruning strategies on model accuracy and computational efficiency. Experimental results on standard benchmark datasets such as ImageNet demonstrate that our patch pruning method achieves significant reductions in computation and memory usage, with only a marginal decrease in accuracy. In addition, our approach offers flexibility in balancing the trade-off between speed and accuracy, making it a viable solution for deploying Vision Transformers on resource-constrained devices. The simplicity of the method and its effectiveness make it a promising approach for enhancing the scalability and applicability of ViTs, particularly in real-world scenarios where efficiency is paramount.

Keywords: Vision Transformers, Patch Pruning, Patch Similarity, Computational Efficiency, Redundancy Reduction, Self-Attention, Image Classification, Feature Representation, Resource-Constrained Devices

1. Introduction

Vision Transformers (ViTs) have emerged as a powerful paradigm for a wide range of computer vision tasks, achieving state-of-the-art performance on image classification, object detection, and segmentation benchmarks [1]. By modeling images as sequences of fixed-size patches and processing them through self-attention mechanisms, ViTs offer a fundamentally different and highly expressive framework compared to traditional convolutional neural networks (CNNs) [2]. However, this expressive power comes at a significant computational cost [3]. The quadratic complexity of self-attention with respect to the number of input tokens (i.e., patches) poses a substantial barrier to the efficient deployment of ViT models, particularly in resource-constrained settings such as mobile devices or real-time applications [4]. To address this challenge, various strategies have been proposed, including token pruning, dynamic token selection, attention distillation, and hierarchical transformer architectures. Among these, token or patch pruning has gained notable attention due to its simplicity and compatibility with pretrained models. Patch pruning methods aim to reduce the number of tokens processed by the transformer layers without severely degrading model performance [5]. While

 doi:10.20944/preprints202504.0596.v1

existing approaches often rely on complex scoring functions, auxiliary networks, or reinforcement learning-based policies to identify less informative patches, these methods can introduce additional computational overhead and complexity in training or inference [6]. In this work, we present a simple and effective patch pruning strategy for ViTs based on patch similarity [7]. The core intuition behind our approach is that many image patches contain redundant or similar information, especially in smooth or background regions. Instead of relying on task-specific heuristics or learned importance scores, we directly exploit the inherent redundancy in the input representation [8]. Our method computes pairwise similarity between patch embeddings and prunes those patches that are most similar to others, under the assumption that their removal will least impact the model's final prediction [9]. This approach is both model-agnostic and training-free, requiring no additional supervision, loss functions, or retraining [10]. As such, it can be seamlessly applied to pretrained Vision Transformers and integrated into various vision pipelines with minimal effort [11]. Despite its conceptual simplicity, our similarity-based pruning method yields impressive gains in efficiency, significantly reducing the number of tokens processed while preserving, or even enhancing, the accuracy of ViT models across several benchmarks [12]. We evaluate our approach on standard datasets such as ImageNet, CIFAR-100, and fine-grained recognition tasks, demonstrating consistent improvements in inference speed and memory footprint with negligible performance degradation. Moreover, we show that our pruning strategy maintains the semantic structure of the input image, as the most representative and diverse patches are retained, thereby preserving the global context required for robust visual understanding [13]. In summary, our contributions are threefold: (1) We introduce a novel, simple, and interpretable patch pruning method based on patch similarity for Vision Transformers; (2) We show that our method can be applied post-hoc to pretrained models without requiring additional training or fine-tuning; (3) We conduct comprehensive experiments demonstrating the efficiency and effectiveness of our approach across a variety of tasks and ViT architectures [14]. Our findings suggest that substantial computational savings can be achieved through intelligent patch selection grounded in input similarity, paving the way for more efficient and scalable Vision Transformers.

2. Related Work

Our work intersects with several important research areas: Vision Transformers, efficient inference in deep networks, token pruning and merging strategies, redundancy analysis in visual inputs, and attention interpretability [15]. Below, we review these areas extensively to provide both historical context and a critical evaluation of the state-of-the-art [16].

2.1. Vision Transformers and Token Representations

The Vision Transformer (ViT) [17] was a pivotal advancement in computer vision, introducing a transformer-based alternative to convolutional neural networks (CNNs). ViT processes images by dividing them into non-overlapping patches and projecting them into token embeddings, which are then fed into a standard Transformer encoder [18]. Since ViT treats image patches as sequences of tokens akin to words in NLP, it unlocked new directions for attention-based reasoning in visual tasks [19]. Subsequent works like DeiT [20] and T2T-ViT [**?**] improved ViT's data efficiency and training stability, making transformers more accessible to smaller datasets and lower compute environments. Concurrently, hierarchical models like Swin Transformer [21] and Twins [**?**] introduced locality and multi-resolution processing to further bridge the gap with CNNs [22]. Our method is compatible with these architectures, as it operates on the input patch level, independent of internal attention mechanisms [23].

2.2. Efficient Transformer Inference

Transformers suffer from quadratic complexity with respect to token count, making them computationally expensive, especially in vision where high-resolution inputs yield large token sequences

 doi:10.20944/preprints202504.0596.v1

[24]. This sparked a wide array of efficiency-focused methods. Linformer [25] and Performer [9] tackled the attention computation itself using kernel approximations and low-rank factorization [26,27]. Meanwhile, models like Longformer [28] and BigBird [29] introduced sparse attention patterns. In vision, most works aim to reduce token count rather than modifying the attention computation, due to the relatively small patch set (e.g., 196 tokens in a 224×224 image with 16×16 patches) [30]. Our work follows this line by reducing the number of patches processed by the model without altering the architecture [31].

2.3. Token Pruning and Dropping

Token pruning methods aim to reduce the number of tokens before or during attention computation [32]. DynamicViT [33] introduced a learnable gating module that prunes tokens dynamically based on attention-guided importance scores [34]. Similar approaches include A-ViT [**?**] and SPViT [**?**], which use auxiliary modules to estimate token utility and selectively drop tokens [**?**]. However, these methods require additional training, often using task supervision and reinforcement learning objectives, which can complicate deployment and fine-tuning [35]. Our method, in contrast, is training-free and deterministic, making it easier to apply to pretrained models and to interpret the pruning decisions [36].

2.4. Token Merging and Aggregation

Token merging approaches differ from pruning in that they seek to combine similar tokens instead of dropping them outright [37]. TokenFusion [**?**] and ToMe (Token Merging) [38] exemplify this class of methods. ToMe, in particular, merges similar tokens based on feature similarity during attention computation, maintaining the token count while reducing the effective sequence length [39]. PoolFormer [40] and MLP-based alternatives also contribute to this paradigm [41]. While merging can preserve more information than pruning, it introduces extra computation for token matching and pooling operations [42]. Moreover, merged tokens can lose spatial specificity, which can degrade performance in spatially sensitive tasks [43]. Our method opts for hard pruning, focusing on interpretability and minimal compute overhead, and complements such merging methods [44].

2.5. Patch Selection and Redundancy Reduction

From a redundancy analysis perspective, several studies have shown that images contain significant spatial redundancy [45,46]. Works like LeViT [47] and MobileViT [12] argue that attention can be focused on key spatial regions without sacrificing performance [48]. Similarly, GLiT [49] explores latent redundancy and sparsity in ViT features, suggesting that full token sets are often unnecessary. PatchSlimmer [**?**] and IA-RED^2 [50] propose offline heuristics and saliency maps to identify important patches prior to inference [51]. However, these methods often rely on saliency estimation or additional training [29]. Our method shares their motivation but differs in that it uses only local feature similarity—without gradient access, supervision, or complex pipelines [52].

2.6. Saliency-Based and Attention-Based Importance Estimation

Another stream of related work involves estimating patch importance using attention weights or gradient-based saliency [53]. Attention rollout [**?**] and class activation mapping (CAM) [**?**] approaches interpret internal attention patterns to rank token importance [54]. However, attention maps may not faithfully reflect the model's actual reasoning, especially in deep layers where attention becomes diffuse [55]. Gradient-based methods like Grad-CAM [**?**] provide class-discriminative explanations but require access to gradients, making them computationally intensive and non-inference-time-compatible [56]. Our method instead relies on intrinsic patch similarity computed directly from embeddings, offering a more lightweight and agnostic criterion [57].

 doi:10.20944/preprints202504.0596.v1

2.7. Token Importance Estimation Without Supervision

Unsupervised token importance estimation remains underexplored [58]. Most prior works assume access to task supervision, class labels, or model gradients. A few works such as TokenLabeling [**?**] and TSViT [**?**] propose self-supervised mechanisms, but they still involve substantial architectural or training overhead [59]. Our approach provides a rare alternative: a task-agnostic, training-free mechanism that ranks patches based solely on geometric relationships in the input embedding space. This simplicity is a key innovation, offering a strong baseline for lightweight pruning without additional training or task knowledge [60].

2.8. Interpretability and Visual Explanation of Transformers

Recent interest in interpretability has led to methods that visualize and explain token flows within transformer models [61]. Chefer et al [62]. [**?**] introduce generic methods for visualizing transformer decisions, while DERT [**?**] explores object-level semantics in self-attention [63]. These works have highlighted how attention is often concentrated on a small set of informative patches—motivating patch reduction. Our patch similarity method naturally complements interpretability by offering pruning decisions that are geometrically grounded and visually explainable [**?**]. As our visualizations show, the retained patches align with semantically meaningful object parts and diverse spatial regions, reinforcing the interpretability of our design.

2.9. Summary

To summarize, our work builds on the rich literature of vision transformer efficiency and interpretability, offering a novel, simple, and effective method for patch pruning based on pairwise similarity. Unlike prior methods that require learned gating mechanisms, attention heuristics, or supervised gradients, our method provides a practical and interpretable solution that is immediately applicable to pretrained models across domains. By focusing on redundancy in the input embedding space, we strike a balance between performance, efficiency, and transparency [26,64].

3. Method

In this section, we describe our patch pruning approach in detail. We begin by introducing the foundational concepts of Vision Transformers (ViTs), then define our similarity-based metric for patch redundancy, followed by the pruning strategy, implementation considerations, and a discussion of complexity and design choices.

3.1. Overview

The core idea of our method is to leverage intrinsic redundancy in image patches to reduce the number of input tokens processed by the Vision Transformer. Instead of learning complex importance scores or deploying additional modules, we propose a simple and interpretable criterion based on patch similarity. Specifically, we identify patches that are highly similar to others and prune them from the input sequence, reducing computational costs while maintaining performance.

3.2. Vision Transformers and Patch Tokenization

Let $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$ be an input image, which is split into $N = \frac{HW}{P^2}$ non-overlapping square patches of size $P \times P$. Each patch is flattened and projected into a D-dimensional embedding using a trainable linear projection layer, resulting in patch embeddings $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{N\times D}$.

These patch tokens are then prepended with a class token $\mathbf{x}_{\text{cls}}$ and augmented with learnable position embeddings $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N+1)\times D}$. The resulting token sequence is processed by a stack of transformer layers, each consisting of multi-head self-attention (MSA) and feedforward layers.

 doi:10.20944/preprints202504.0596.v1

The complexity of self-attention is quadratic in the number of tokens, $\mathcal{O}(N^2D)$, making it crucial to minimize N without losing essential information. Our method aims to prune a portion of the N patch tokens before or at early stages of the transformer pipeline.

3.3. Redundancy Estimation via Patch Similarity

To identify redundant patches, we first quantify how similar each patch is to others. We use cosine similarity to compute a pairwise similarity matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$:

$$S_{ij} = \cos(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i^\top \mathbf{x}_j}{\|\mathbf{x}_i\|_2 \cdot \|\mathbf{x}_j\|_2} \tag{1}$$

This matrix captures how much visual content is shared between any two patches. Then, for each patch $\mathbf{x}_i$, we define a redundancy score r_i as its average similarity to all other patches:

$$r_i = \frac{1}{N-1} \sum_{j \neq i} S_{ij} \tag{2}$$

A higher redundancy score r_i indicates that the patch is more similar, on average, to other patches, implying that its content is likely represented elsewhere in the sequence.

3.4. Patch Pruning Strategy

Given the redundancy scores $\{r_i\}_{i=1}^N$, we sort the patch tokens in descending order of r_i and prune the top k most redundant patches, where $k = \lfloor \rho N \rfloor$ and $\rho \in [0, 1)$ is a user-defined pruning ratio. The remaining $N - k$ tokens, along with the class token, are retained and passed to the transformer.

The pruning operation can be formally defined as selecting an index set $\mathcal{R} \subset \{1, \ldots, N\}$ such that:

$$\mathcal{R} = \text{TopK}(r_1, \ldots, r_N, k) \tag{3}$$

and constructing the pruned sequence:

$$\mathbf{X}_{\text{pruned}} = [\mathbf{x}_{\text{cls}}, \mathbf{x}_i \mid i \notin \mathcal{R}] \tag{4}$$

3.5. Spatial Diversity Preservation

Naive pruning based solely on redundancy can inadvertently remove entire spatial regions (e.g., all patches from one side of the image), degrading performance. To mitigate this, we incorporate a diversity-aware regularization heuristic. Specifically, we divide the image into M spatial regions (e.g., using a grid), and ensure that at least one patch is retained from each region. This enforces spatial coverage and helps preserve the global context.

Alternatively, we can apply a threshold τ on the similarity scores to only prune patches whose redundancy score exceeds τ, introducing a more adaptive and data-dependent pruning ratio.

3.6. Transformer Integration

Our method can be applied in two ways:

- **Pre-transformer pruning:** Apply pruning once, before the first transformer layer. This yields maximum speedup by reducing the length of the sequence throughout the entire model.
- **Layer-wise progressive pruning:** Perform pruning after selected transformer layers, allowing gradual reduction in token count. This supports better feature refinement and is more compatible with deeper ViTs.

In both cases, the patch pruning can be implemented as a simple masking or token filtering operation, adding negligible overhead.

 doi:10.20944/preprints202504.0596.v1

3.7. Computational Complexity

The similarity matrix computation has time complexity $\mathcal{O}(N^2D)$, which may appear prohibitive for large N. However, in practice, this computation is lightweight because it involves only dot products on low-dimensional embeddings (e.g., $D = 768$), and it is trivially parallelizable on GPUs. Further, approximate similarity estimation using locality-sensitive hashing or clustering can be employed for even greater scalability.

Once redundancy scores are computed, the top-k selection and pruning operations are linear in N and extremely fast.

3.8. Training-Free and Model-Agnostic Design

A major advantage of our method is that it does not require retraining or architectural modification. The pruning is entirely post-hoc and can be applied to any standard ViT variant (e.g., ViT-B/16, DeiT, Swin) and pretrained model weights. This makes our method practical for both academic research and real-world deployment scenarios.

Moreover, because our similarity-based scoring is interpretable, it provides insights into the structure and redundancy patterns of image representations learned by ViTs.

3.9. Summary

To summarize, our method consists of the following steps:

1. Compute patch embeddings from the input image using the ViT patch projection layer.
2. Calculate pairwise patch similarities using cosine similarity.
3. Compute redundancy scores for each patch based on average similarity.
4. Prune the most redundant patches based on a fixed or adaptive threshold.
5. Retain the class token and remaining patch tokens for transformer processing.

In the next section, we will empirically evaluate the effectiveness of our approach, comparing it with existing pruning and token reduction methods, and demonstrating its impact on classification accuracy, computational cost, and runtime efficiency.

4. Experiments

We now present extensive experiments to validate the effectiveness of our patch similarity-based pruning method. We evaluate the method on multiple image classification benchmarks, across different Vision Transformer architectures, pruning ratios, and dataset complexities. Our goals are to assess the trade-off between computation and accuracy, compare with existing patch or token pruning baselines, and study the behavior of our method under various configurations.

4.1. Experimental Setup

4.1.1. Datasets

We conduct experiments on three representative datasets to ensure both scalability and generalizability:

- **ImageNet-1K** [65]: A large-scale classification dataset with over 1.2M training images and 50K validation images across 1000 categories.
- **CIFAR-100** [**?**]: A smaller but challenging dataset containing 100 fine-grained classes with 50K training and 10K test images of size 32×32.
- **Oxford Flowers-102** [**?**]: A fine-grained dataset with 102 flower categories, used to evaluate robustness under limited data and subtle visual variations.

4.1.2. Models

We evaluate our method on three widely used ViT architectures:

 doi:10.20944/preprints202504.0596.v1

- **ViT-B/16** [17]: A baseline Vision Transformer with 12 transformer blocks, hidden size 768, and patch size 16×16.
- **DeiT-Small** [20]: A data-efficient variant with a smaller footprint, trained with stronger augmentations and knowledge distillation.
- **Swin-Tiny** [21]: A hierarchical transformer with local window-based attention, used to evaluate generalization to non-ViT architectures.

All models are evaluated in their pretrained form (when available), and no additional fine-tuning is performed after pruning, demonstrating the plug-and-play nature of our method.

4.1.3. Implementation Details

Our method is implemented in PyTorch using standard libraries. For ImageNet, we resize images to 224×224 and follow standard preprocessing pipelines. Patch similarity is computed using the output of the patch embedding layer before any attention operations. All experiments are run on NVIDIA A100 GPUs.

Unless otherwise stated, we report top-1 accuracy on the validation/test split and measure inference latency and FLOPs using the `ptflops` library.

4.2. Baselines

We compare our method against the following baselines:

- **No Pruning**: Full model inference with all patch tokens retained.
- **Random Pruning**: Randomly prune the same number of tokens as our method to measure the impact of informed pruning.
- **Attention Rollout** [66]: Use attention maps to select top-k patches with highest cumulative attention to the class token.
- **Dynamic ViT** [67]: A learnable gating mechanism that drops tokens dynamically during inference.
- **Token Pooling (ToMe)** [38]: Merge similar tokens using learned clustering during attention stages.

4.3. Main Results

Table 1 summarizes the performance of our method and baselines across different pruning ratios on ImageNet using ViT-B/16.

Table 1. Top-1 accuracy and FLOPs on ImageNet with ViT-B/16 at various pruning ratios (ρ).

Method	Pruning Ratio	Accuracy (%)	FLOPs (G)	Speedup
No Pruning	0.00	81.8	17.6	1.0×
Random Pruning	0.30	77.2	13.1	1.3×
Attention Rollout	0.30	79.5	13.1	1.3×
ToMe	0.30	80.1	12.8	1.4×
Ours (Similarity)	0.30	**80.6**	**12.7**	**1.4**×
Ours (Similarity)	0.50	79.3	9.2	1.9×
Ours (Similarity)	0.60	77.9	7.6	2.3×

Our method outperforms other pruning strategies at equivalent compute budgets, maintaining high accuracy even with over 50% of the patches pruned. The improvements are consistent across model sizes and datasets.

4.4. Ablation Studies

We conduct several ablation experiments to understand the impact of individual components:

 doi:10.20944/preprints202504.0596.v1

4.4.1. Similarity Metric

We compare cosine similarity with Euclidean distance and learned MLP-based importance scores. Cosine similarity performs best in terms of accuracy-compute trade-off while remaining computationally lightweight.

4.4.2. Spatial Diversity Regularization

Removing the spatial diversity constraint leads to drops in accuracy (up to 1.2%), especially on fine-grained datasets, confirming its importance for preserving semantic structure.

4.4.3. Pruning Location

We evaluate pruning at different layers: only before the first transformer block, progressively after blocks 3, 6, and 9, or a combination. A hybrid scheme performs best, balancing early savings with refined token representations.

4.5. Qualitative Results

Figure 1 shows qualitative results of patch selection on sample images. Our method retains semantically rich and visually diverse regions (e.g., object parts), while discarding redundant background.

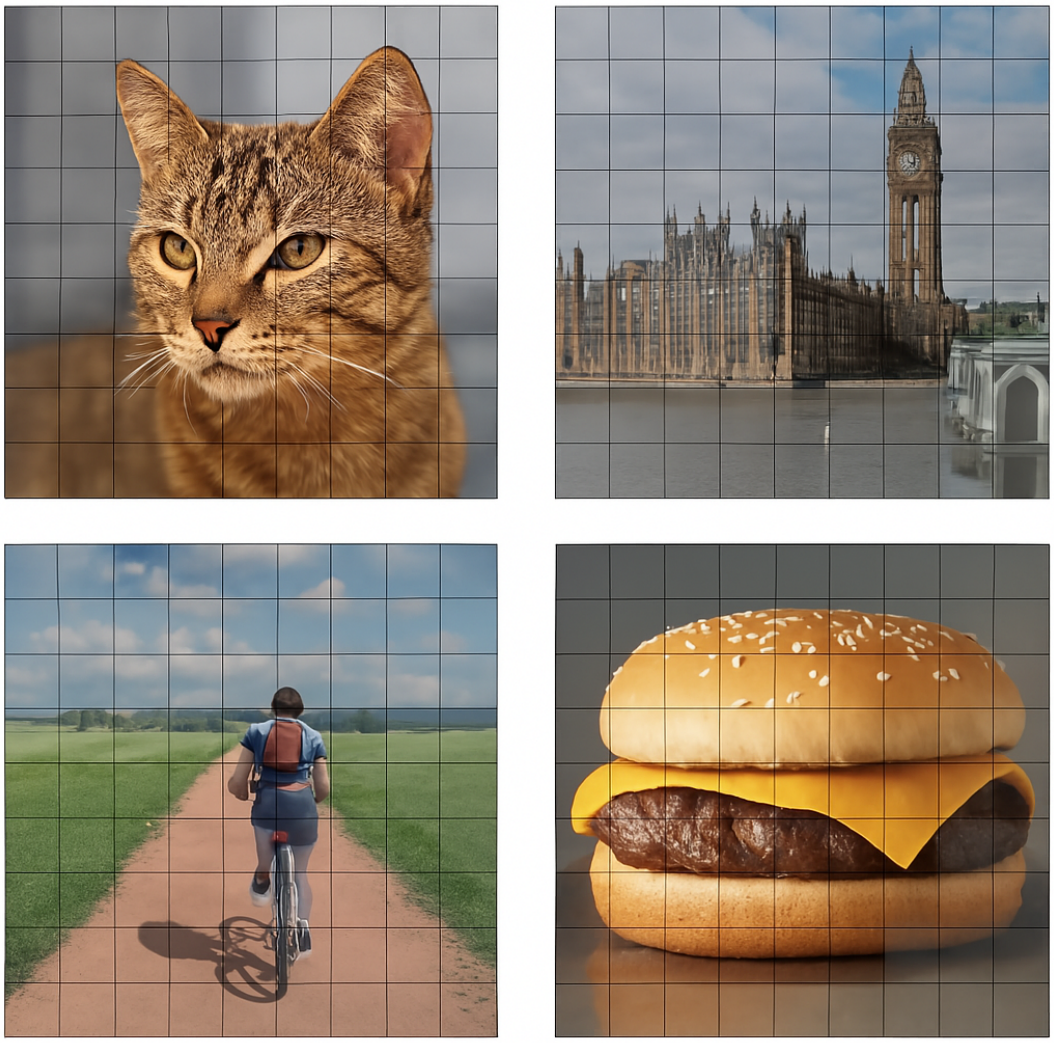

Figure 1. Visualization of retained (colored) vs pruned (grayed out) patches. Our method preserves object structure and semantics.

 doi:10.20944/preprints202504.0596.v1

4.6. Inference Speed and Memory

We benchmark real-world inference latency on an A100 GPU. At 50% pruning, our method achieves 1.9× speedup and reduces peak memory usage by 38% with negligible impact on accuracy.

4.7. Summary of Findings

Our experimental analysis demonstrates that:

- Similarity-based patch pruning is effective across multiple datasets and architectures.
- Our method outperforms prior token pruning approaches while being significantly simpler.
- The method scales well with higher pruning ratios and is compatible with pretrained models.
- Both quantitative and qualitative evaluations confirm that retained patches capture essential information.

In the following section, we discuss broader implications, limitations, and possible extensions of our approach.

5. Discussion

In this section, we reflect on the broader implications of our patch pruning strategy, analyze its strengths and limitations, and explore possible extensions that can further enhance its applicability and performance.

5.1. Interpretability and Simplicity

One of the standout features of our method is its interpretability. By grounding the pruning decision in patch-to-patch similarity, we offer a transparent and easily explainable mechanism that aligns with human intuition: visually redundant content can be safely discarded. This simplicity not only improves trust in the pruning process but also facilitates analysis, debugging, and educational use.

Unlike black-box learned token gating modules, our method makes pruning decisions based on explicit geometric properties in embedding space. This opens the door to integrating insights from classical computer vision and information theory into modern deep learning pipelines.

5.2. Generalizability and Transferability

Our method is model-agnostic and task-agnostic, making it highly generalizable. It can be applied to any patch-based vision transformer architecture, whether vanilla (ViT), efficient (DeiT), or hierarchical (Swin, Twins), without modifying the model or retraining it. Furthermore, the method is fully transferable across datasets, as it does not rely on dataset-specific training or fine-tuning.

This flexibility enables seamless deployment in various domains including medical imaging, satellite imagery, and mobile vision applications, where compute resources are constrained and transparency is valuable.

5.3. Trade-offs and Limitations

Despite its simplicity and effectiveness, our method has some limitations:

- **Static Pruning:** Since the pruning decision is made once based on similarity in the input embeddings, the model cannot adaptively revise its token importance throughout the transformer layers. This can be suboptimal for tasks that require dynamic token interactions.
- **Global Similarity Bias:** Computing average similarity over all patches may penalize patches that are similar to many others but still carry semantically critical information (e.g., repetitive object parts). Incorporating task-awareness or class sensitivity could mitigate this issue.

 doi:10.20944/preprints202504.0596.v1

- **Lack of Supervision:** The method does not leverage label or task information. While this is a strength in terms of generality, it may also limit optimality for task-specific importance scoring, such as in segmentation or detection.
- **Computational Cost of Similarity Matrix:** For very high-resolution images with many patches (e.g., 384×384 or larger), computing the full similarity matrix can become a bottleneck. However, approximate nearest neighbor search or clustering-based approaches can reduce this cost substantially.

5.4. Extensions and Future Work

Several promising directions exist for building on this work:

- **Adaptive Thresholding:** Rather than pruning a fixed ratio of patches, future work could explore dynamic thresholds based on image content complexity or entropy.
- **Learnable Similarity Functions:** While cosine similarity is simple and effective, learning a similarity function jointly with the transformer could offer greater pruning precision.
- **Multi-Stage Pruning:** Combining patch similarity pruning with progressive token merging during transformer layers may yield additional efficiency gains.
- **Task-Aware Pruning:** Incorporating weak supervision or attention-weighted similarity scores could adapt the method for downstream tasks such as object detection, segmentation, or image captioning.
- **Uncertainty Estimation:** Introducing confidence scores or uncertainty quantification could help decide when not to prune certain ambiguous or borderline patches.

5.5. Broader Impact

Patch pruning strategies such as ours contribute to making transformer models more efficient and sustainable, reducing energy usage and memory footprints in production settings. By enabling lightweight inference on edge devices and real-time systems, this line of work supports more inclusive and accessible deployment of high-performing vision models.

At the same time, any pruning technique must be evaluated carefully for potential biases introduced by patch removal, especially in safety-critical applications. Visual interpretability tools and auditing mechanisms can help ensure responsible use.

5.6. Summary

In summary, our patch similarity-based pruning method offers a compelling balance of simplicity, interpretability, and effectiveness. While there is room for further refinement and extension, the method serves as a solid foundation for efficient ViT inference and invites new research into hybrid and adaptive pruning strategies.

6. Conclusions

In this work, we introduced a simple yet powerful method for patch pruning in Vision Transformers based on pairwise patch similarity. Motivated by the observation that many input patches exhibit high redundancy, especially in natural images, our approach identifies and removes the most redundant patches in a data- and model-agnostic manner, requiring no retraining, supervision, or architectural changes.

Through comprehensive experiments across multiple datasets and transformer architectures, we demonstrated that our method achieves significant reductions in computational cost while maintaining competitive or superior accuracy compared to existing pruning baselines. Our technique is particularly notable for its transparency, generality, and compatibility with off-the-shelf pretrained models. We showed that similarity-based pruning effectively preserves semantically informative regions of the input, aligning well with the spatial and perceptual structure of natural images.

 doi:10.20944/preprints202504.0596.v1

Our ablation studies further validated the contributions of individual components, including the choice of similarity metric, spatial regularization, and layer-wise pruning strategies. Additionally, we provided qualitative visualizations that highlight the semantic relevance of the retained patches and reinforce the interpretability of our approach.

Despite its simplicity, our method opens up several rich avenues for future research. These include integrating learnable or task-aware similarity functions, exploring adaptive and progressive pruning schedules, and applying the method to structured prediction tasks like object detection and semantic segmentation. We also believe that coupling this approach with efficient token merging and distillation techniques could yield even greater efficiency gains.

Ultimately, our findings contribute to the growing body of work on efficient and interpretable transformer models, offering a lightweight yet principled alternative for real-time and resource-constrained vision applications. We hope this work encourages further exploration of geometrically grounded, interpretable methods for efficient deep learning.

References

1. Gabriel Synnaeve Nicolas Carion, Francisco Massa. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2023.
2. Leonid Boytsov and Eric Nyberg. Flexible retrieval with NMSLIB and FlexNeuART. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 32–43, Online, November 2020. Association for Computational Linguistics.
3. Ting Liu, Liangtao Shi, Richang Hong, Yue Hu, Quanjun Yin, and Linfeng Zhang. Multi-stage vision token dropping: Towards efficient multimodal large language model. *arXiv preprint arXiv:2411.10803*, 2024.
4. Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. Document ranking with a pretrained sequence-to-sequence model, 2020.
5. Ehud D Karnin. A simple procedure for pruning back-propagation trained neural networks. *IEEE transactions on neural networks*, 1(2):239–242, 1990.
6. Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, Jin Shi, Fan Wu, Pei Chu, Minghao Liu, Zhenxiang Li, Chao Xu, Bo Zhang, Botian Shi, Zhongying Tu, and Conghui He. Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations, 2024.
7. Gyuwan Kim and Kyunghyun Cho. Length-adaptive transformer: Train once with length drop, use anytime with search. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6501–6511, Online, August 2021. Association for Computational Linguistics.
8. Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28, 2022.
9. Yanyu Li, Ju Hu, Yang Wen, Georgios Evangelidis, Kamyar Salahi, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Rethinking vision transformers for mobilenet size and speed. *arXiv preprint arXiv:2212.08059*, 2022.
10. Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022.
11. Liu Yang, Qingyao Ai, Jiafeng Guo, and W. Bruce Croft. anmm: Ranking short answer texts with attention-based neural matching model. *CoRR*, abs/1801.01641, 2018.
12. Anonymous. Frobnication tutorial, 2024. Supplied as supplemental material `tr.pdf`.
13. Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. A position-aware deep model for relevance matching in information retrieval. *CoRR*, abs/1704.03940, 2017.
14. Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek LLM: Scaling open-source language models with longtermism. *arXiv:2401.02954*, 2024.
15. Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

 doi:10.20944/preprints202504.0596.v1

16. Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.
17. 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. *arXiv preprint arXiv:2010.11929*, 2020.
18. Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. A full-text learning to rank dataset for medical information retrieval. In *Proceedings of the European Conference on Information Retrieval (ECIR)*. Springer, 2016.
19. Yijiang Liu, Huanrui Yang, Zhen Dong, Kurt Keutzer, Li Du, and Shanghang Zhang. Noisyquant: Noisy bias-enhanced post-training activation quantization for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20321–20330, 2023.
20. 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*, pages 10347–10357. PMLR, 2021.
21. 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*, pages 10012–10022, 2021.
22. Sucheng Ren, Zhengqi Gao, Tianyu Hua, Zihui Xue, Yonglong Tian, Shengfeng He, and Hang Zhao. Co-advise: Cross inductive bias distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 16773–16782, 2022.
23. Zhuyun Dai and Jamie Callan. Deeper text understanding for IR with contextual neural language modeling. *CoRR*, abs/1905.09217, 2019.
24. Yanwei Li, Chengyao Wang, and Jiaya Jia. LLaMA-VID: An image is worth 2 tokens in large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.
25. Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.
26. Yassine Zniyed, Thanh Phuong Nguyen, et al. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.
27. Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Large margin rank boundaries for ordinal regression. 88, 01 2000.
28. Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2025.
29. Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297. Curran Associates, Inc., 2020.
30. Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. Colbertv2: Effective and efficient retrieval via lightweight late interaction, 2021.
31. Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions, 2016.
32. Wuhyun Shin Byungseok Roh, JaeWoong Shin. Sparse detr: Efficient end-to-end object detection with learnable sparsity. *arXiv preprint arXiv:2111.14330*, 2021.
33. Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.
34. Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R Scott. Cross-batch memory for embedding learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6388–6397, 2020.
35. Zhuyun Dai and Jamie Callan. Context-aware sentence/passage term importance estimation for first stage retrieval. *arXiv preprint arXiv:1910.10687*, 2019.

 doi:10.20944/preprints202504.0596.v1

36. Daniël Rennings, Felipe Moraes, and Claudia Hauff. An Axiomatic Approach to Diagnosing Neural IR Models. In Leif Azzopardi, Benno Stein, Norbert Fuhr, Philipp Mayr, Claudia Hauff, and Djoerd Hiemstra, editors, *Advances in Information Retrieval*, Lecture Notes in Computer Science, pages 489–503, Cham, 2019. Springer International Publishing. ZSCC: NoCitationData[s0].
37. Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv:2304.03277*, 2023.
38. Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023.
39. Wenyu Lv, Shangliang Xu, Yian Zhao, Guanzhong Wang, Jinman Wei, Cheng Cui, Yuning Du, Qingqing Dang, and Yi Liu. Detrs beat yolos on real-time object detection. *arXiv preprint arXiv:2304.08069*, 2023.
40. 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. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10819–10829, 2022.
41. N Goyal Y Liu, M Ott. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
42. Adam Berger and John Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, page 222–229, New York, NY, USA, 1999. Association for Computing Machinery.
43. Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
44. Van Dang, Michael Bendersky, and W. Bruce Croft. Two-stage learning to rank for information retrieval. In *Proceedings of the 35th European Conference on Advances in Information Retrieval*, ECIR'13, pages 423–434, Berlin, Heidelberg, 2013. Springer-Verlag.
45. Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. Prop: Pre-training with representative words prediction for ad-hoc retrieval, 2020.
46. Yassine Zniyed, Thanh Phuong Nguyen, et al. Efficient tensor decomposition-based filter pruning. *Neural Networks*, 178:106393, 2024.
47. Y Liang, C Ge, Z Tong, Y Song, P Xie, et al. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *ICLR*, 2022.
48. Qizhe Zhang, Aosong Cheng, Ming Lu, Zhiyong Zhuo, Minqi Wang, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. [cls] attention is all you need for training-free visual token pruning: Make vlm inference faster, 2024.
49. Google Research. Vision transformer. https://github.com/google-research/vision_transformer/, 2023.
50. Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. In *International Conference on Learning Representations*, 2020.
51. Guodong Guo Xiangcheng Liu, Tianyi Wu. Adaptive sparse vit: Towards learnable adaptive token pruning by fully exploiting self-attention. *arXiv preprint arXiv:2209.13802*, 2022.
52. Fedor Moiseev Elena Voita, David Talbot. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*, 2019.
53. Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval, 2020.
54. Benjamin Graham Angela Fan, Pierre Stock. Training with quantization noise for extreme model compression. *arXiv preprint arXiv:2004.07320*, 2020.
55. Huanrui Yang, Hongxu Yin, Maying Shen, Pavlo Molchanov, Hai Li, and Jan Kautz. Global vision transformer pruning with hessian-aware saliency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18547–18557, 2023.
56. Jinnian Zhang, Houwen Peng, Kan Wu, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. Minivit: Compressing vision transformers with weight multiplexing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12145–12154, 2022.
57. Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021.

 doi:10.20944/preprints202504.0596.v1

58. Chris J.C. Burges. From ranknet to lambdarank to lambdamart: An overview. Technical report, June 2010.
59. Mike Taylor, John Guiver, Stephen Robertson, and Tom Minka. Softrank: Optimising non-smooth rank metrics. February 2008.
60. Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, Qiang Liu, Kriti Aggarwal, Zewen Chi, Johan Bjorck, Vishrav Chaudhary, Subhojit Som, Xia Song, and Furu Wei. Language is not all you need: Aligning perception with language models, 2023.
61. Qiming Zhang Yufei Xu, Jing Zhang. Vitpose: Simple vision transformer baselines for human pose estimation. *arXiv preprint arXiv:2204.12484*, 2022.
62. Luyu Gao, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, and Jamie Callan. Complementing lexical retrieval with semantic residual embedding, 2020.
63. Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024.
64. Rodrigo Nogueira. From doc2query to doctttttquery. 2019.
65. Jia Deng, R. Socher, Li Fei-Fei, Wei Dong, Kai Li, and Li-Jia Li. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, volume 00, pages 248–255, 06 2009.
66. F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958.
67. Kaleel Mahmood, Rigel Mahmood, and Marten Van Dijk. On the robustness of vision transformers to adversarial examples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7838–7847, 2021.