TEXTURE AS SIGNAL: SPECTRAL ADAPTIVE TOKENIZATION FOR VISION TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision Transformers (ViTs) struggle with quadratic computational costs, a bottleneck often reduced by adaptive tokenization. Current methods predominantly rely on spatial sparsity (using edge detection or region homogeneity) to reduce token counts. However, this spatial tokenization fails in texture-dominated regimes (such as high-resolution aerial and microscopic imagery) where high-frequency gradients are dense. In these environments, edge-driven models either over-fragment the image or aggressively merge critical details, destroying both compression and performance. To resolve this, we introduce texture-aware adaptive tokenization. Rather than relying on spatial discontinuities, we formulate token saliency through spectral statistics to capture local frequency content and redundancy. We propose a general framework that dynamically partitions tokens based on texture complexity, requiring no modifications to the underlying ViT architecture. Supported by a theoretical analysis of spectral density failure modes, this approach is extensively evaluated across SAM, Swin, and ViT architectures on diverse, texture-rich datasets. The proposed texture-driven framework achieves up to a 41.5$\times$ speedup of training wall-clock time for SAM on wild blueberry aerial imagery, while simultaneously improving convergence dice scores (from 63.1% to 71.2%). Furthermore, under severe image perturbations, the texture partitioning mechanism maintains greater than 91% Structural Intersection over Union (SIoU) under heavy Gaussian noise and greater than 84% under severe Gaussian blur, preventing the catastrophic structural collapse observed in conventional edge-based tokenization (as low as 50.90%). Ultimately, these findings establish that spectral properties (rather than spatial boundaries) are the governors of token saliency in texture-dominated regimes, enabling the efficient deployment of heavy foundation models in complex visual domains.

1 INTRODUCTION

High-resolution imagery is essential across remote sensing, precision agriculture, biomedical microscopy, and materials analysis, where small structures and subtle spatial variations must be retained for downstream analysis (Wang et al., 2025; Bilodeau et al., 2023; Lucchi et al., 2013; Wu et al., 2018). For Vision Transformers (ViTs), however, preserving this resolution creates long token sequences whose computational and memory requirements can make large-scale or time-sensitive analysis impractical (Zhang et al., 2024). Reducing image resolution or aggressively limiting the number of tokens can improve computational feasibility, but may remove information required for accurate predictions. This creates a persistent trade-off between preserving image information and maintaining computational efficiency, restricting the practical use of ViTs in applications that require both.

1.1 THE LONG-SEQUENCE PROBLEM IN VISION TRANSFORMERS

ViTs process images by partitioning them into a sequence of tokens, typically obtained via a uniform grid of patches. Given an input image of resolution $Z \times Z$ and patch size $P \times P$, the resulting sequence length is $N = (Z/P)^2$. The self-attention mechanism computes pairwise interactions between tokens, leading to $\mathcal{O}(N^2)$ computational and memory complexity. As image resolution increases or patch size decreases, both desirable for dense prediction tasks such as segmentation, the sequence length grows rapidly, rendering standard self-attention prohibitively expensive. A large body of work has therefore focused on reducing the effective sequence length in vision models via adaptive tokenization (Ryoo et al., 2021; Rao et al., 2021; Liang et al., 2022; Bolya et al., 2022; Lee et al., 2025; Yan et al., 2025; Liu et al., 2021; Fan et al., 2021; Wang et al., 2021; 2022; Wu et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

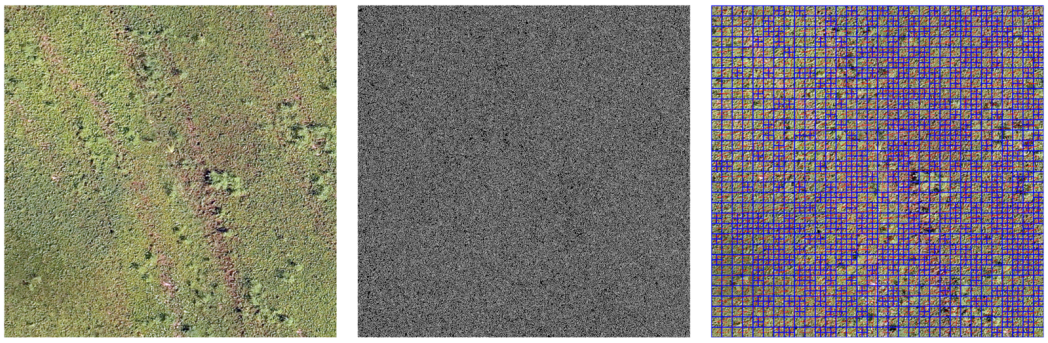


Figure 1: Failure of edge-based partitioning on textured (left). The Canny edge mask (thresholds 100-200) becomes densely populated by high-frequency gradients rather than sparse boundaries (middle), forcing the QuadTree algorithm to exhaust its token budget via recursive subdivision (right).

2019; Wang et al., 2016; Li et al., 2025; Zhang et al., 2024; 2025a; Bachmann et al., 2025; Duggal et al., 2025; Zhang et al., 2025b). These methods aim to preserve informative regions of the input while reducing redundancy, thereby improving the efficiency of attention without sacrificing accuracy. Broadly, such approaches attempt to exploit spatial structure in images to selectively allocate tokens where they are most needed.

1.2 The Edge/Region Assumption

Most existing adaptive tokenization methods rely, either explicitly or implicitly, on the assumption that salient visual information is spatially sparse. In particular, they prioritize tokens based on cues such as edge strength, region boundaries, or local feature importance. Edge-driven approaches use gradient magnitude or boundary detectors to allocate finer tokens near object contours, while region-based methods assume piecewise homogeneity and merge tokens in low-variance areas. Similarly, in-model token pruning and merging techniques often rely on learned importance scores that correlate with spatial discontinuities.

Underlying these approaches is a common assumption: that informative content is concentrated along sparse structures, and that large portions of the image can be coarsely represented without loss of fidelity. While this assumption holds for many natural images with well-defined objects and boundaries, it fails in settings where visual information is distributed more uniformly across space. This mismatch induces two failure modes: edge-based tokenizers either mark most regions as salient, causing over-fragmentation and little sequence reduction, or they satisfy a fixed token budget by merging regions that are not texturally redundant, erasing fine-scale structure. In texture-dominated images, the problem is therefore not noisy edge detection, but the use of spatial discontinuity as a proxy for information content.

1.3 The Texture-Dominated Regime

In texture-dominated images common to aerial imaging, microscopy, and astronomy (Wang et al., 2025; Ray et al., 2026; Dieleman et al., 2015; Inauen et al., 2025; Wu et al., 2018), visual content consists of dense, high-frequency patterns with ambiguous boundaries. In these regimes, non-sparse gradient magnitudes cause region homogeneity assumptions to fail. Consequently, edge- or region-based tokenization either over-fragments the image, yielding minimal sequence reduction, or aggressively merges tokens, destroying fine-grained structure (Figure 1). From a signal processing perspective, texture-dominated images exhibit broad spectral support, with energy distributed across a wide range of frequencies. Consequently, spatial heuristics based on discontinuities fail to capture the underlying redundancy structure of the signal. This leads to poor compression–accuracy trade-offs, as the criteria used to guide token allocation are misaligned with the true information content of the image. In this work, we revisit adaptive tokenization from a spectral perspective and introduce a texture-aware formulation of token saliency. Our contributions are as follows:

- **Failure Analysis of Edge-Based Tokenization:** We provide a formal and empirical analysis of the limitations of edge- and region-driven methods in texture-dominated regimes, showing that their underlying assumptions break down under high spectral density.

- **Texture-Aware Token Theory:** We introduce a principled framework for characterizing token informativeness based on local spectral statistics, including power spectral density and autocorrelation structure, and relate these to redundancy in token representations.
- **Adaptive Tokenization via Texture Analysis:** We propose a general method that allocates tokens dynamically based on texture complexity rather than spatial discontinuities, without modifying the underlying transformer architecture.
- **Comprehensive Evaluation Across Domains:** We conduct extensive experiments across diverse datasets, including aerial, microscopic, telescopic, and synthetic textures, and multiple model architectures, demonstrating consistent improvements in accuracy–efficiency trade-offs.

2 Background and Related Work

2.1 Adaptive Tokenization Methods

To mitigate the quadratic complexity of self-attention in high-resolution ViTs, sequence reduction strategies generally utilize three approaches: in-model reduction, hierarchical architectures, and adaptive input patching. In-model methods like DynamicViT (Rao et al., 2021), EViT (Liang et al., 2022), ToMe (Bolya et al., 2022), and TokenLearner (Ryoo et al., 2021) prune, drop, or merge redundant tokens based on attention scores or learned semantic biases. Hierarchical designs, notably Swin Transformer (Liu et al., 2021), bypass bottlenecks by limiting attention to local windows, though this imposes a rigid spatial bias. Input-level methods such as AP (Zhang et al., 2024) and SHF (Zhang et al., 2025a) employ QuadTree partitioning to downsample background regions while preserving structural details via gradient-driven biases. Additionally, FlexTok (Bachmann et al., 2025) and KARL (Duggal et al., 2025) predict variable sequence lengths or budgets prior to embedding. Appendix A.3 Table 4 summarizes existing adaptive tokenization methods.

2.2 Spectral Analysis in Vision

Frequency-domain and morphological representations have long served as foundational tools in computer vision (Bracewell & Kahn, 1966; Haralick et al., 1987; Jain & Farrokhnia, 1991), offering strong alternatives to spatial gradients by describing visual structure through frequency distributions and regional density rather than localized intensity discontinuities (Oppenheim et al., 1997; Petrou & Kamata, 2021). Traditional Fourier analysis captures broad periodicities (Bracewell & Kahn, 1966), while multiscale methods, such as wavelet decompositions and Gabor filtering, provide spatial-frequency localization (Mallat, 1989; Jain & Farrokhnia, 1991). Historically, these techniques drove early breakthroughs in texture classification (Haralick et al., 1973; Ojala et al., 2002; Petrou & Kamata, 2021), image compression (Antonini et al., 1992; Mallat, 1989), and image processing specifically containing dense micro-structures (Petrou & Kamata, 2021). Similarly, mathematical morphology and operations like the Top-Hat transform have been widely utilized in biomedical and satellite imaging to isolate fine-grained cellular features or structural anomalies from unevenly illuminated backgrounds (Haralick et al., 1987; Lucchi et al., 2013).

2.3 Texture Modeling

Texture was characterized by repetitive spatial arrangements and statistical properties rather than isolated intensity gradients (Petrou & Kamata, 2021; Haralick et al., 1973). Classical modeling relied on statistical feature extraction (Haralick et al., 1973; Ojala et al., 2002), utilizing local variance and the Gray Level Cooccurrence Matrix for second order dependencies (Haralick et al., 1973; Petrou & Kamata, 2021), alongside Local Binary Patterns to capture illumination invariant ordinal microstructures (Ojala et al., 2002; Petrou & Kamata, 2021). However, these rigid statistical descriptors lacked multiscale flexibility for dynamic, high resolution environments (Petrou & Kamata, 2021). To address this multiscale complexity, modern modeling employed spectral filtering and mathematical morphology (Jain & Farrokhnia, 1991; Haralick et al., 1987; Petrou & Kamata, 2021). Frequency domain representations, such as Power Spectral Density, captured dominant periodic structures (Bracewell & Kahn, 1966; Oppenheim et al., 1997). For joint spatial frequency localization, wavelet metrics aggregated multiscale coefficients (Mallat, 1989), while Gabor filters isolated oriented structural elements (Jain & Farrokhnia, 1991; Petrou & Kamata, 2021). Similarly, mathematical morphology probed topological properties (Haralick et al., 1987), where operations like the Top Hat transform evaluated regional density and shape to separate high frequency foreground textures from complex backgrounds (Haralick et al., 1987; Petrou & Kamata, 2021).

3 When Edge-based Tokenization Become Non-Selective

From an image to a quadtree token sequence. We study input tokenizers that partition an image into rectangular regions using a quadtree. The tree begins with a single node, called the root, representing the entire image. Splitting a region at its horizontal and vertical midpoints creates four child regions. A region that has not been split further is called a leaf; the current leaves form a nonoverlapping partition of the image. Repeated splits produce smaller regions where finer sampling is allocated, while unsplit regions remain larger. At the end of partitioning, each leaf is cropped from the image, resized to a common patch size, and embedded as one image token. Thus smaller leaves allocate more encoded samples per unit of original image area. The allocation rule determines which region to split next, while the token budget limits the number of regions produced. We first examine an edge-guided rule within this scheme, then introduce alternative texture responses for the same partitioning mechanism.

Let x be an image on a pixel domain Ω of size $H \times W$, and let $a_x : \Omega \to \{0, 1\}$ be a binary edge map, where $a_x(u) = 1$ indicates a detected edge at pixel u. An edge-count allocator assigns each current leaf region R the score

$$s_x(R) = \sum_{u \in R} a_x(u)$$

and splits a leaf with the largest score. This is a statement about edge-count-based allocation, not about all adaptive tokenizers.

Spatially widespread responses. Suppose, as an idealized model, that $\mathbb{P}(a_x(u) = 1) = \rho$ at every pixel. Then

$$\mathbb{E}[s_x(R)] = \rho|R|.$$

This identity does not require independence between pixels. Consequently, equal-area regions have equal expected scores even if their relevance to the prediction task differs. If their realized active fractions also concentrate around ρ, the allocation criterion becomes approximately area-based. In the limiting case $a_x(u) = 1$ everywhere, it selects a largest-area leaf at every split. On a square dyadic image this produces level-by-level refinement, up to the tie-breaking order and termination within a level.

Budgeted versus threshold-terminated refinement. A threshold-terminated tokenizer can lose compression when most regions trigger refinement. A budgeted tokenizer instead enforces a target sequence length M. Its potential failure is to spend that budget on unhelpful regions. For such a tokenizer, dense gradients alone do not imply that the compression ratio tends to one. We use

$$N_{\text{grid}} = HW/p^2, \qquad b = M/N_{\text{grid}}, \qquad \mathcal{C} = N_{\text{grid}}/M,$$

assuming H and W are divisible by the embedding patch size p. Here b is a token-allocation fraction, not the fraction of original image area retained.

Role of spectral information. High-frequency content can help explain why edge responses are widespread, but spectral support alone does not determine the number of threshold crossings. An isolated impulse has broad Fourier support and localized finite differences, whereas a periodic sinusoid has sparse Fourier support and spatially widespread variation. We therefore treat edge density as a measured, detector-dependent image property, rather than deriving it from broad spectral support.

4 Texture-Guided Budgeted Tokenization

Our method retains the quadtree construction described above and changes the image signal used to select regions for splitting. The pipeline has three stages: compute a scalar texture response at each pixel, threshold that response to obtain a binary allocation map, and use counts of active pixels to guide quadtree splitting. The resulting leaf regions are then resized and embedded as image tokens. The response map determines where the available resolution is spent; it is not itself the image representation passed to the transformer.

4.1 Response Maps and Their Roles

Let $r_x : \Omega \to \mathbb{R}_{\geq 0}$ be a scalar response map computed from the grayscale image. Our evaluated response families measure different aspects of local appearance:

- Black top-hat (BTH), $r_x = (x \bullet B) - x$, measures dark structures relative to morphological closing by a structuring element B.
- Local variance measures intensity dispersion within a fixed neighborhood.
- Local Fourier energy aggregates selected squared Fourier magnitudes within sliding windows.
- Wavelet detail energy aggregates squared detail coefficients at the implemented decomposition scale.

BTH and local variance are spatial-domain responses; they are not assumed to approximate a full power spectral density. Fourier and wavelet energies summarize selected frequency information and do not retain the full signal. These responses are alternative allocation cues, rather than interchangeable definitions of semantic redundancy.

For a nonconstant response map, the implementation applies per-image min–max normalization and integer quantization,

$$u_x(v) = \left\lfloor 255 \frac{r_x(v) - \min r_x}{\max r_x - \min r_x} \right\rfloor .$$

With a validity indicator $v_x(v)$, threshold τ, and polarity $\sigma \in \{+, -\}$, the active map is

$$a_x(v) = v_x(v) \begin{cases} \mathbf{1}\{u_x(v) > \tau\}, & \sigma = +, \\ \mathbf{1}\{u_x(v) \le \tau\}, & \sigma = -. \end{cases}$$

For the texture-response implementations, pixels with exactly zero grayscale intensity are excluded by v_x. This convention must not be interpreted as a general test of whether a pixel is background. Direct thresholding prioritizes high responses; inverse thresholding prioritizes low responses. Both the threshold and its polarity are therefore part of the tokenizer configuration.

4.2 Greedy Refinement and Serialization

Initially the root region covers the entire image and is the tree's only leaf, so $\mathcal{L}_0 = [\Omega]$. At iteration t, let $\mathcal{L}_t$ denote the ordered list of current leaf regions. Using the binary allocation map a_x defined above, we select the next region to split as

$$R_t \in \arg\max_{R \in \mathcal{L}_t} s_x(R), \qquad s_x(R) = \sum_{v \in R} a_x(v),$$

breaking ties by the first occurrence in the list. If the selected score is positive and both side lengths are at least two pixels, we replace the selected leaf by its four midpoint children, ordered top-left, top-right, bottom-left, bottom-right. Otherwise the current implementation stops. Refinement also stops when the requested leaf-count target is reached or exceeded.

Every successful split increases the number of leaves by three. Thus a complete tree has $1 + 3k$ leaves after k splits. Budget rounding, early termination, padding, and any truncation are reported separately from the requested allocation fraction. In particular, discarding excess leaves would remove image coverage rather than constitute a valid coarsening operation.

Each retained leaf R is cropped from the RGB image and resized to $p \times p$ using bicubic interpolation. For the sequence-based ViT and SAM adapters, its initial token is

$$z_R = W_{\text{emb}} \operatorname{vec}(\mathcal{N}(\text{Resize}_{p \times p}(x|_R))) + b_{\text{emb}} + g_\theta(c_R, \ell_R),$$

where $\mathcal{N}$ denotes channel normalization, c_R is the normalized region center, and ℓ_R is its normalized width. The positional map g_θ is a learned multilayer perceptron. The leaf-list order is the quadtree depth-first order induced by the child ordering above. The image responses guide refinement; their values are not supplied as a descriptor vector to the transformer.

The tokenizer does not compute pairwise texture distances, enforce a within-region spectral-variance tolerance, or merge nonadjacent patches. Its threshold τ controls the active response map, whereas its target length M controls sequence allocation. These are distinct parameters.

4.3 Computational Cost

Let $C_{\text{resp}}(H, W)$ be the response extraction cost, M the processed sequence length, and d the embedding dimension. A summed-area table costs $O(HW)$ to construct and gives $O(1)$ region sum queries. In the supplied implementation, each split scans and updates a Python leaf list, giving $O(M^2)$

worst-case partitioning cost over $O(M)$ splits. Fixed-stencil patch resampling costs $O(Mp^2)$ for a fixed number of channels. A global-attention block has cost $O(M^2d + Md^2)$, including attention and channel projections. The response extraction cost depends on the operator, support size, and implementation; it is not uniformly the cost of a local pixel transform.

Reducing M decreases the quadratic attention term, but does not guarantee an end-to-end speedup. Let

$$T_{\text{adaptive}} = T_{\text{response}} + T_{\text{partition}} + T_{\text{embedding}} + T_{\text{backbone}} + T_{\text{prediction}}.$$

The method is faster than a specified baseline only when $T_{\text{adaptive}} < T_{\text{baseline}}$ under the same timing protocol. Windowed or dense-image wrappers require their own cost analysis; their backbone cost cannot be inferred from the requested quadtree leaf count.

5 Allocation Quality and Supporting Analysis

Allocation quality is task-dependent. At the same processed token count, two partitions can preserve different spatial details because they assign different original regions to each resized patch. A useful allocation spends resolution where coarsening would damage prediction-relevant structure. Neither edge density nor a texture-response magnitude establishes that relevance by itself. The central empirical question is therefore whether changing the response improves prediction under a common predictor interface, and whether the gain can be connected to the structures preserved by its allocation.

A falsifiable allocation hypothesis. Texture guidance should help when the selected response allocates resolution to task-sensitive structures that edge-count refinement under-resolves at the same budget. Its advantage should diminish or reverse when that response emphasizes irrelevant texture, when different classes require incompatible cues, or when preprocessing cost exceeds the benefit of a shorter sequence. These are hypotheses to test through matched-budget comparisons and measured costs, not consequences of assuming that texture statistics encode semantics.

Supporting stability result. Appendix A.1 gives a sufficient condition for the implemented split sequence to remain unchanged. In brief, bounded response perturbations can change activity only near the threshold; if the selected region's score remains positive and its lead over competitors exceeds the possible score changes, the same regions are split. The guarantee depends on both threshold margins and split-score margins and is conditional on the stated stopping rule. For BTH, the unnormalized response additionally satisfies

$$\|r_x - r_{x'}\|_\infty \le 2\|x - x'\|_\infty. \tag{1}$$

This bound assumes a fixed structuring element and common boundary convention on the grayscale inputs. The appendix accounts for normalization, quantization, and validity masking. Stable geometry does not ensure useful allocation: even a content-independent grid can be perfectly stable. Prediction quality and geometric robustness must therefore be evaluated separately.

A circular shift preserves Fourier magnitudes but can change a linear embedding. Region resizing and positional encoding add further differences. Thus texture agreement is not a certificate of semantic preservation. Our main claim is empirical allocation quality; no general bound on downstream prediction error is inferred from response similarity.

6 Experimental Evaluation

6.1 Datasets

Experiments evaluated generalization across diverse texture domains and modalities (Appendix A.4 Table 6). Datasets included aerial RGB and multispectral imagery of wild blueberry fields containing hair fescue (*Festuca filiformis*) weeds (Bilodeau et al., 2023), alongside a tree plantation mapped using circlegrammetry (Bilodeau et al., 2025). The wild blueberry dataset was particularly challenging due to the masks being annotated at pixel-level granularity, following individual hair fescue structures rather than simplified or smoothed polygon boundaries. Consequently, small spatial offsets between the predicted and reference masks produce substantial pixel disagreement and disproportionately reduce Dice scores. Furthermore, 3D electron microscopy stacks were utilized for mitochondria segmentation (Lucchi et al., 2013), extracting specific $1024 \times 768 \times 165$ voxel subvolumes from larger 2048×1536 pixel slices to reduce annotation costs. All datasets were partitioned using an 80/10/10 training, validation, and testing split. Edge complexity of each dataset can be seen in Appendix A.4 Table 6.

6.2 Models and Baselines

Five foundation architectures were evaluated (Appendix A.3 Table 5): ViT-Base (Dosovitskiy et al., 2021) and SAM Base (Kirillov et al., 2023), alongside the Swin Transformer (Liu et al., 2021), DINOv3 (Siméoni et al., 2025), and RF-DETR (Robinson et al., 2026). Proposed variants (Texture-SHF, Texture-AP) were benchmarked against input-level spatial tokenizers (AP (Zhang et al., 2024), SHF (Zhang et al., 2025a)), in-model reducers (DynamicViT (Rao et al., 2021), TokenLearner (Ryoo et al., 2021)), and dynamic resource allocators (FlexTok (Bachmann et al., 2025), AdaFocusV2 (Wang et al., 2022)). Under matched token budgets (where applicable), all configurations were trained on a single NVIDIA H100 GPU (80 GB). Training configuration and specifics can be found in Appendix A.10.

6.3 Texture Spectrum Sweep

A qualitative grid-search analysis was conducted for each texture metric described in Section 4.1. The threshold τ was varied across both direct and inverted mappings (see Appendix A.5, Figure 2). This sweep served as an initial screening stage in which each metric was assessed according to its ability to concentrate fine-scale partitions on annotated structures, suppress fragmented responses within background regions, and maintain practical preprocessing costs. The screening retained four candidates representing two texture metric families: Local Variance and BTH as spatial-domain descriptors, and local PSD and Wavelet Energy as spectral-domain descriptors. For Texture-AP, an optional pretraining calibration was performed using the IoU between the generated texture mask $\mathcal{M}$ and the corresponding ground-truth target mask. This calibration identified the optimal threshold and direct or inverted mapping for each dataset. Although applied in this study, it is not required to use Texture-AP; rather, it represents a unique advantage of the process by allowing token allocation to be evaluated and optimized independently before model training. The four retained metrics were subsequently evaluated under a fixed 15% token budget, enabling controlled within-family comparisons of segmentation performance and computational efficiency while preserving representation from both spatial- and spectral-domain approaches.

6.4 Segmentation Results

The results in Table 1 validated that texture-aware partitioning optimizes the efficiency-fidelity tradeoff. General-purpose compressors (TokenLearner (Ryoo et al., 2021), FlexTok (Bachmann et al., 2025), DynamicViT (Rao et al., 2021)) discarded critical high-frequency details, yielding poor convergence Dice scores of 12.05–45.17% on the wild blueberry dataset (Bilodeau et al., 2023). The proposed method consistently mitigated foundation model bottlenecks on complex textures. On wild blueberry imagery, *Texture-SHF-SAM* achieved a 4.41× training speedup ($> 40\times$ relative to baseline SAM's (Kirillov et al., 2023) 0.11×) and improved convergence Dice from 63.17 to 71.20%. *Texture-SHF-Swin* achieved a 7.30× speedup during training(1.89× relative) and increased accuracy from 65.34 to 66.93%. On the Tree Plantation dataset (Bilodeau et al., 2025), *Texture-SHF-Swin* reached peak overall accuracy (97.33% Dice) with a 64.91× baseline training speedup. On the Mitochondria dataset (Lucchi et al., 2013), ViT-Base failed to reach the target Dice (67.10% max Dice), but *Texture-SHF-ViT* achieved the target and converged at 84.70% Dice. *Texture-SHF-Swin* reached 94.82% Dice, outperforming the baseline Swin speedup (6.08× vs 3.51×).

Furthermore, the texture-driven approach outperformed the edge-based SHF framework. Standard Canny-driven partitioning reduced compute but caused notable accuracy drops, decreasing Swin's Dice from 65.34 to 54.36% (wild blueberry) and SAM's from 95.55 to 89.67% (Tree Plantation). Conversely, *Texture-SHF* variants recovered peak accuracy and achieved greater computational speedups during training than edge-based counterparts without sacrificing resolution. Ultimately, allocating tokens based on texture density rather than arbitrary spatial grids or simple edge gradients freed the transformer backbone from processing large amounts of redundant background noise, allowing heavy vision models to be deployed efficiently in texture-rich environments. These results were further supported by the classification results which can be seen in Appendix A.7.

6.5 Compression vs Accuracy Tradeoff

Efficiency trade-offs were characterized by the cumulative time (seconds) required to reach target validation Dice scores as token allocations decreased from 75 to 15% of the 4096 sequence length (Table 2). Texture-guided tokenization consistently improved scaling at extreme compression. At 15% allocation, *Texture SHF* outperformed baseline SHF across all datasets, requiring only 477.93s vs. 4706.53s (Tree plantation), 943.03s vs. 1037.39s (Wild blueberry field), and 375.83s vs. 1519.36s (Mitochondria). Similarly, at 30% allocation, *Texture AP Spectral* outpaced baseline AP, reaching the target in 139.86s vs. 313.71s (Tree plantation) and 732.99s vs. 1091.68s (Mitochondria). These

Table 1: Performance comparison across adaptive tokenization methods using ViT-Base and SAM as speedup reference (patience=20).

Dataset	Method	↑ Conv. Dice (%)	↑ Conv. speedup	↑ speedup @ Dice (80)
*Wild Blueberry (Bilodeau et al., 2023)	ViT-B (Dosovitskiy et al., 2021)	67.64 ± 8.07	1.00	1.00
	Swin (Liu et al., 2021)	65.34 ± 13.48	5.60	1.13
	SAM (Kirillov et al., 2023)	63.17 ± 10.43	1.03	0.11−
	DINOv3 (Siméoni et al., 2025)	62.89 ± 9.31	1.12	0.65
	RF-DETR (Robinson et al., 2026)	$\mathbf{78.75 \pm 3.07}$	0.80	1.81
	AP-SAM (Zhang et al., 2024)	69.91 ± 2.80	3.16	1.01
	SHF-ViT (Zhang et al., 2025a)	68.78 ± 4.77	3.64	1.48
	SHF-Swin	54.36 ± 23.46	12.98	5.02
	SHF-SAM (Zhang et al., 2025a)	71.85 ± 4.59	3.11	3.38
	SHF-DINOv3	60.25 ± 7.87	1.62	0.94
	SHF-RF-DETR	73.16 ± 4.23	1.15	1.22
	TokenLearner (Ryoo et al., 2021)	39.37 ± 6.00	2.95	0
	DynamicViT (Rao et al., 2021)	45.17 ± 4.98	0.56	0
	AdaFocusV2 (Wang et al., 2022)	56.76 ± 5.51	2.22	1.96−
	FlexTok (Bachmann et al., 2025)	12.05 ± 2.92	1.01	0
	Texture-AP-SAM-Spectral (Ours)	70.25 ± 3.81	3.72	0.59
	Texture-AP-SAM-Spatial (Ours)	67.81 ± 6.60	4.24	3.23
	Texture-SHF-ViT (Ours)	67.00 ± 4.82	4.05	1.24
	Texture-SHF-Swin (Ours)	66.93 ± 5.66	10.60	**7.30**
	Texture-SHF-SAM (Ours)	71.20 ± 4.57	3.23	4.41
	Texture-SHF-DINOv3 (Ours)	60.06 ± 7.74	1.41	1.23
	Texture-SHF-RF-DETR (Ours)	74.71 ± 5.80	1.17	2.55
Tree plantation (Bilodeau et al., 2025)	ViT-B (Dosovitskiy et al., 2021)	88.87 ± 1.41	1.00	1.00
	Swin (Liu et al., 2021)	95.07 ± 1.09	2.21	4.88
	SAM (Kirillov et al., 2023)	95.55 ± 0.61	0.59	1.10
	DINOv3 (Siméoni et al., 2025)	91.54 ± 2.21	1.69	3.97
	RF-DETR (Robinson et al., 2026)	96.00 ± 1.22	0.90	22.33
	AP-SAM (Zhang et al., 2024)	92.16 ± 1.46	1.61	24.67
	SHF-ViT (Zhang et al., 2025a)	80.30 ± 3.90	3.91	5.35
	SHF-Swin	97.29 ± 0.29	3.43	61.92
	SHF-SAM (Zhang et al., 2025a)	89.67 ± 4.57	2.79	4.07
	SHF-DINOv3	87.47 ± 3.80	1.77	3.91
	SHF-RF-DETR	93.61 ± 1.71	1.36	15.34
	TokenLearner (Ryoo et al., 2021)	24.97 ± 13.81	2.98	0
	DynamicViT (Rao et al., 2021)	89.28 ± 5.28	0.34	2.52
	AdaFocusV2 (Wang et al., 2022)	86.12 ± 3.88	1.32	1.19
	FlexTok (Bachmann et al., 2025)	21.83 ± 4.01	1.30	0
	Texture-AP-SAM-Spectral (Ours)	92.33 ± 2.67	1.66	55.32
	Texture-AP-SAM-Spatial (Ours)	92.63 ± 2.05	2.37	23.53
	Texture-SHF-ViT (Ours)	81.50 ± 1.17	3.10	5.13
	Texture-SHF-Swin (Ours)	$\mathbf{97.33 \pm 0.06}$	4.42	**64.91**
	Texture-SHF-SAM (Ours)	95.10 ± 1.22	1.89	8.86
	Texture-SHF-DINOv3 (Ours)	87.73 ± 3.80	1.94	5.07
	Texture-SHF-RF-DETR (Ours)	94.37 ± 1.43	1.84	17.04
Electron Microscopic Mitochondria (Lucchi et al., 2013)	ViT-B (Dosovitskiy et al., 2021)	67.10 ± 2.14	0.48	0
	Swin (Liu et al., 2021)	83.81 ± 19.83	3.44	3.51
	SAM (Kirillov et al., 2023)	$\mathbf{95.17 \pm 0.64}$	1.00	1.00
	DINOv3 (Siméoni et al., 2025)	94.57 ± 0.38	0.31	0.13
	RF-DETR (Robinson et al., 2026)	94.22 ± 0.39	0.72	0.73
	AP-SAM (Zhang et al., 2024)	85.45 ± 0.88	2.09	0.49
	SHF-ViT (Zhang et al., 2025a)	75.02 ± 2.28	1.18	0.09
	SHF-Swin	94.61 ± 1.08	2.88	5.70
	SHF-SAM (Zhang et al., 2025a)	73.27 ± 7.48	1.85	0.31
	SHF-DINOv3	83.24 ± 5.83	0.89	0.09
	SHF-RF-DETR	87.32 ± 0.87	1.34	0.46
	TokenLearner (Ryoo et al., 2021)	69.25 ± 2.14	2.43	0.11
	DynamicViT (Rao et al., 2021)	91.67 ± 2.82	0.31	0.10
	AdaFocusV2 (Wang et al., 2022)	93.55 ± 0.87	1.27	2.86
	FlexTok (Bachmann et al., 2025)	13.10 ± 1.85	0.90	0
	Texture-AP-SAM-Spectral (Ours)	88.28 ± 0.09	1.79	0.73
	Texture-AP-SAM-Spatial (Ours)	86.39 ± 2.22	1.78	0.46
	Texture-SHF-ViT (Ours)	84.70 ± 5.36	0.82	0.23
	Texture-SHF-Swin (Ours)	94.82 ± 0.35	2.80	**6.08**
	Texture-SHF-SAM (Ours)	87.18 ± 0.38	1.10	1.22
	Texture-SHF-DINOv3 (Ours)	94.88 ± 0.20	0.41	0.21
	Texture-SHF-RF-DETR (Ours)	94.30 ± 0.49	0.77	0.85

∗ Target Dice threshold was changed to 70 due to dataset difficulty; − At least one run failed; 0 All runs failed.

trajectories confirm that morphological and spectral partitioning retain structural data more effectively than edge-driven algorithms under aggressive compression.

7 Ablation Studies

Robustness to perturbation severity ablation study can be seen in Appendix A.9.

7.1 Different Texture Metrics

An ablation study evaluated the impact of different texture metrics (Local Variance, Local PSD, BTH) on model efficacy under a fixed 15% token budget, with performance quantified via validation Dice scores and computation time. The morphological *BTH-SHF-SAM* reached convergence most efficiently, requiring only 426.07s (Tree plantation), 4724.82s (Wild blueberry), and 1120.31s (Mitochondria). While *Local Variance* and *Wavelet* metrics showed moderate efficiency on tree plantation data (928.40s and 787.72s), their demands escalated on complex wild blueberry imagery (5176.34s and 10641.18s), failing to match morphological performance. Conversely, *Local PSD* proved severely

Table 2: Performance comparison across adaptive patching (AP) and Symmetrical Hierarchical Forest (SHF) methods with respect to percent of token allocation (↓Time (seconds) to reach target Dice (80)).

Dataset	Method	75%	60%	45%	30%	15%
*Wild blueberry field	AP (Zhang et al., 2024)	2,498	**1,018**	**857**	4,324	4,134
	SHF (Zhang et al., 2025a)	5,372	3,472	2,515	1,293	1,037
	Texture-AP-Spec. (Ours)	21,224	13,091	1,304	7,343	5,955
	Texture-AP-Spat. (Ours)	**1,507**	2,319	1,037	1,352	3,828
	Texture-SHF (Ours)	4,329	4,156	1,727	**990**	**943**
Tree plantation	AP (Zhang et al., 2024)	**375**	**405**	261	313	353
	SHF (Zhang et al., 2025a)	2,910	3,260	2,802	1,903	4,706
	Texture-AP-Spec. (Ours)	268	286	**164**	**139**	219
	Texture-AP-Spat. (Ours)	619	**405**	325	328	**153**
	Texture-SHF (Ours)	2,387	2,130	1,658	872	477
Mitochondria cross-sections	AP (Zhang et al., 2024)	36,397	8,525	2,749	1,091	506
	SHF (Zhang et al., 2025a)	61,029	22,036	9,752	2,547	1,519
	Texture-AP-Spec. (Ours)	26,150	30,489	2,975	732	521
	Texture-AP-Spat. (Ours)	36,739	22,653	8,569	1,177	471
	Texture-SHF (Ours)	**1,197**	**599**	**605**	**439**	**375**

∗ Target Dice threshold was changed to 70 due to dataset difficulty

inefficient, demanding 6641.37s (Tree), 22563.08s (Blueberry), and 6943.97s (Mitochondria). These results establish that morphological compression isolates structural boundaries more effectively than frequency magnitudes under high compression.

Table 3: Cumulative time required (in seconds) to achieve a target validation Dice score across varying datasets and tokenization strategies.

Method	Tree Plantation	Wild Blueberry Field	Mitochondria Cross-sections
	↓ Time (s) to Dice = 0.80	↓ Time (s) to Dice = 0.70	↓ Time (s) to Dice = 0.80
PSD-SHF-SAM	6,641 a	22,563 a	6,943 a
Wavelet-SHF-SAM	787 b	10,641 b	1,297 b
LV-SHF-SAM	928 c	5,176 b	1,128 c
BTH-SHF-SAM	**426** d	**4,724** b	**1,120** c

Means that do not share a letter are significantly different.
Comparisons are within the column only.

8 Conclusion

A texture-driven adaptive tokenization framework resolved ViT bottlenecks in texture-dominated environments by partitioning tokens via spectral density and morphology rather than spatial boundaries. Evaluations confirmed consistent acceleration with matched or improved accuracy. In segmentation, the framework achieved a 41.5× speedup during training for SAM on agricultural imagery, increasing convergence Dice from 63.17 to 71.20%. Classification for a 90% accuracy target reached a 2.62× training speedup. At an aggressive 15% token allocation, Texture-SHF-SAM reached target convergence in 375.83s on cellular data. Ablations identified the morphological BTH transform as the most efficient extractor, requiring 426.07s on tree data vs. 6641.37s for PSD metrics. Morphological extraction maintained $> 91\%$ structural similarity under noise and $> 85\%$ under severe blur, while edge-driven methods collapsed below 60% SIoU. Spectral and morphological properties thus provide an efficient pathway for deploying foundation models in complex visual domains.

AI USE STATEMENT

Generative AI tools were used solely to assist with language editing and improve the readability of the manuscript. All suggested edits were reviewed by the authors, who take responsibility for the final content.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, Section 6.1 and Appendix A.4 describe the datasets and data splits, while Appendix A.10 specifies the optimization settings, hyperparameters, and random seeds used in our experiments. Model configurations and computational requirements are summarized in Table 5, with wall-clock breakdowns provided in Appendices A.8.1. Code will be made publicly available upon acceptance, together with a detailed README providing instructions for its use.

REFERENCES

Marc Antonini, Michel Barlaud, Pierre Mathieu, Ingrid Daubechies, et al. Image coding using wavelet transform. *IEEE Trans. Image Processing*, 1:20–5, 1992.

Roman Bachmann, Jesse Allardice, David Mizrahi, Enrico Fini, Oğuzhan Fatih Kar, Elmira Amirloo, Alaaeldin El-Nouby, Amir Zamir, and Afshin Dehghan. Flextok: Resampling images into 1d token sequences of flexible length. In *Forty-second International Conference on Machine Learning*, 2025.

Mathieu F Bilodeau, Travis J Esau, Craig B MacEachern, Aitazaz A Farooque, Scott N White, and Qamar U Zaman. Identifying hair fescue in wild blueberry fields using drone images for precise application of granular herbicide. *Smart Agricultural Technology*, 3:100127, 2023.

Mathieu F Bilodeau, Travis J Esau, Mason T MacDonald, and Aitazaz A Farooque. Circlegrammetry for drone imaging: Evaluating a novel technique for mission planning and 3d mapping. *ISPRS Open Journal of Photogrammetry and Remote Sensing*, pp. 100111, 2025.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

Ron Bracewell and Peter B Kahn. The fourier transform and its applications. *American Journal of Physics*, 34(8):712–712, 1966.

Sander Dieleman, Kyle W Willett, and Joni Dambre. Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly notices of the royal astronomical society*, 450(2): 1441–1459, 2015.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Shivam Duggal, Sanghyun Byun, William T. Freeman, Antonio Torralba, and Phillip Isola. Single-pass adaptive image tokenization for minimum program search. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=HkTOnCUQ1z. Poster.

Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. URL https://arxiv.org/abs/2104.11227.

Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6):610–621, 1973.

Robert M Haralick, Stanley R Sternberg, and Xinhua Zhuang. Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):532–550, 1987.

Cornelia M Inauen, Ingmar Nitze, Moritz Langer, Anne Morgenstern, Irena Hajnsek, and Guido Grosse. Using texture-based image segmentation and machine learning with high-resolution satellite imagery to assess permafrost degradation landforms in the russian high arctic. *Journal of Geophysical Research: Machine Learning and Computation*, 2(3):e2024JH000550, 2025.

Anil K Jain and Farshid Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern recognition*, 24(12):1167–1186, 1991.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

Dong-Jae Lee, Jiwan Hur, Jaehyun Choi, Jaemyung Yu, and Junmo Kim. Frequency-aware token reduction for efficient vision transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=Dr06Wjh45k. Poster.

Xue Li, Xiaonan Song, and Henry Hu. Efficient vision-language reasoning via adaptive token pruning. In *NeurIPS 2025 Workshop on Vision-Language Models for Real-World Deployment (VLM4RWD)*, 2025. URL https://openreview.net/forum?id=Vbqemx4YCC. Extended abstract.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2202.07800.

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*, pp. 10012–10022, 2021.

Aurélien Lucchi, Yunpeng Li, and Pascal Fua. Learning for structured prediction using approximate subgradient descent with working sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1987–1994, 2013.

Stephane G Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989.

Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.

Alan V Oppenheim, Alan S Willsky, and Syed Hamid Nawab. *Signals & systems*. Pearson Educación, 1997.

Maria MP Petrou and Sei-ichiro Kamata. *Image processing: dealing with texture*. John Wiley & Sons, 2021.

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.

Rahul D Ray, Utkarsh Srivastava, Rishi Mohapatra, and Naveen K. Shrivastava. The benchmarking void: A roadmap for domain-adapted computer vision in fuel cell defect detection. In *AI for Accelerated Materials Design - ICLR 2026*, 2026. URL https://openreview.net/forum?id=qo0yH1ewuf.

Isaac Robinson, Peter Robicheaux, Matvei Popov, Deva Ramanan, and Neehar Peri. Rf-detr: Neural architecture search for real-time detection transformers. In *International Conference on Learning Representations*, volume 2026, pp. 56087–56100, 2026.

Michael Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: Adaptive space-time tokenization for videos. *Advances in neural information processing systems*, 34:12786–12797, 2021.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Dongbo Wang, Zedong Yan, and Peng Liu. Fine-grained interpretation of remote sensing image: A review. *Remote Sensing*, 17(23):3887, 2025.

Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. URL https://arxiv.org/abs/1608.00859.

Yulin Wang, Zhaoxi Chen, Haojun Jiang, Shiji Song, Yizeng Han, and Gao Huang. Adaptive focus for efficient video recognition. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16229–16238. IEEE, 2021.

Yulin Wang, Yang Yue, Yuanze Lin, Haojun Jiang, Zihang Lai, Victor Kulikov, Nikita Orlov, Humphrey Shi, and Gao Huang. Adafocus v2: End-to-end training of spatial dynamic networks for video recognition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20030–20040. IEEE, 2022.

Guangming Wu, Xiaowei Shao, Zhiling Guo, Qi Chen, Wei Yuan, Xiaodan Shi, Yongwei Xu, and Ryosuke Shibasaki. Automatic building segmentation of aerial imagery using multi-constraint fully convolutional networks. *Remote Sensing*, 10(3):407, 2018.

Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S. Davis. Adaframe: Adaptive frame selection for fast video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. URL https://arxiv.org/abs/1811.11153.

Yaoyao Yan, Hui Yu, and Weizhi Xu. Attribution-driven adaptive token pruning for transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=MzN2Xi7EMa. Poster.

Enzhi Zhang, Isaac Lyngaas, Peng Chen, Xiao Wang, Jun Igarashi, Yuankai Huo, Masaharu Munetomo, and Mohamed Wahib. Adaptive patching for high-resolution image segmentation with transformers. In *SC24: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16. IEEE, 2024.

Enzhi Zhang, Peng Chen, Rui Zhong, Du Wu, Jun Igarashi, Isaac Lyngaas, Xiao Wang, Masaharu Munetomo, and Mohamed Wahib. Shf: Symmetrical hierarchical forest with pretrained vision transformer encoder for high-resolution medical segmentation. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 139696–139723, 2025a. doi: 10.52202/085713-4667.

Zhengqiang Zhang, Rongyuan Wu, Lingchen Sun, and Lei Zhang. Gpstoken: Gaussian parameterized spatially-adaptive tokenization for image representation and generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b. URL https://openreview.net/forum?id=BxoEDR2yQM. Poster.

A Technical Appendices and Supplementary Material

A.1 Proofs and Scope of the Allocation Analysis

Definitions and score stability. Consider two integer response maps u, u' on the same domain, with the same validity mask, threshold τ, polarity, and target length. Suppose $\|u - u'\|_\infty \le \eta$. Define

$$U_\eta = \{v : v \text{ is valid and } |u(v) - \tau| \le \eta\}, \qquad b_\eta(R) = |R \cap U_\eta|.$$

Only pixels in U_η can change their binary activity, so

$$|s_u(R) - s_{u'}(R)| \le b_\eta(R). \tag{2}$$

Proposition 1 (stability of the split sequence). Run the allocator on u, and suppose it reaches its budget through K successful splits. At every step $t < K$, assume that its selected leaf R_t satisfies

$$s_u(R_t) > b_\eta(R_t), \tag{3}$$

$$s_u(R_t) - s_u(R) > b_\eta(R_t) + b_\eta(R) \quad \text{for every } R \in \mathcal{L}_t \setminus \{R_t\}. \tag{4}$$

Then running the same allocator on u' produces the same ordered leaves and the same region geometry. These conditions are sufficient, not necessary; integer score ties may leave the certificate inconclusive even when the realized partition is stable.

Proof of the score perturbation bound. With fixed threshold and polarity, a pixel can change activity only if the interval between $u(v)$ and $u'(v)$ crosses the threshold. If $|u(v) - \tau| > \eta$ and $|u(v) - u'(v)| \le \eta$, this cannot occur. Since the validity mask is unchanged, each score difference is a sum of binary activity differences supported on $R \cap U_\eta$. The triangle inequality gives

$$|s_u(R) - s_{u'}(R)| \le \sum_{v \in R} |a_u(v) - a_{u'}(v)| \le b_\eta(R).$$

Proof of Proposition 1. Initially both runs contain the same root. Suppose their ordered leaf lists agree at step t. For every competitor $R \ne R_t$,

$$\begin{aligned} s_{u'}(R_t) - s_{u'}(R) &\ge s_u(R_t) - s_u(R) - b_\eta(R_t) - b_\eta(R) \\ &> 0. \end{aligned}$$

Moreover, $s_{u'}(R_t) \ge s_u(R_t) - b_\eta(R_t) > 0$. Because the clean run successfully splits R_t, its dimensions satisfy the geometric split condition in both runs. The selected rectangle, its midpoint children, and their insertion order therefore agree. Induction proves equality after all K splits. The same leaf counts imply the same budget termination. This statement assumes budget termination; it does not silently cover every early-stop case.

Identical-mask special case. If no valid pixel lies within η of the threshold, then $a_u = a_{u'}$. Every score, including ties, is identical. With the same deterministic tie-breaking rule, the complete execution is identical, including geometric or zero-score early termination. More generally, changes in threshold can be accommodated by replacing η with $\eta + |\tau - \tau'|$ in the uncertain-pixel set. If the validity masks differ, their symmetric difference must also be included in that set.

BTH before normalization. For vectors indexed by a common finite set, $|\max_i a_i - \max_i b_i| \le \max_i |a_i - b_i|$; the analogous inequality holds for minima. Flat grayscale dilation and erosion are thus nonexpansive in $\|\cdot\|_\infty$, using the same structuring element and boundary extension in both images. Their composition C_B is nonexpansive, giving

$$\|(C_B x - x) - (C_B x' - x')\|_\infty \le \|C_B x - C_B x'\|_\infty + \|x - x'\|_\infty \le 2\|x - x'\|_\infty.$$

Effect of min–max normalization. Let r, r' be nonconstant response maps with $\|r - r'\|_\infty \le \varepsilon_r$, and write $h = \max r - \min r > 2\varepsilon_r$. Then $h' = \max r' - \min r' > 0$, $|h - h'| \le 2\varepsilon_r$, and $|\min r - \min r'| \le \varepsilon_r$. Writing $A = r - \min r$ and $A' = r' - \min r'$, pointwise we have

$$\begin{aligned} \left|\frac{A}{h} - \frac{A'}{h'}\right| &\le \frac{|A - A'|}{h} + \frac{A'}{h'}\frac{|h' - h|}{h} \\ &\le \frac{4\varepsilon_r}{h}. \end{aligned}$$

If $u = \lfloor 255A/h \rfloor$ and $u' = \lfloor 255A'/h' \rfloor$, it follows that

$$\|u - u'\|_\infty \le \min\left\{255, \frac{1020\varepsilon_r}{h} + 1\right\}.$$

For BTH, setting $\delta = \|x - x'\|_\infty$ gives $\varepsilon_r \le 2\delta$, and therefore, when $h > 4\delta$,

$$\|u_x - u_{x'}\|_\infty \le \min\left\{255, \frac{2040\delta}{h} + 1\right\}.$$

These are conservative real-arithmetic bounds with the stated integer quantization; they do not model additional floating-point roundoff. Both maps must use the nonconstant normalization branch. The bound explicitly deteriorates for small response ranges.

Scope. The input perturbation bound is on realized grayscale arrays, not directly on a Gaussian noise standard deviation or blur-kernel size. To obtain a probabilistic guarantee, one would additionally need a probabilistic bound on the realized maximum-norm perturbation. Pixels excluded because their intensity equals zero may change validity after corruption; such changes must be counted separately. Stable partitions preserve region geometry, but their resized RGB contents and model predictions may still change.

A.2 Conditions for Approximating One Attention Output

Consider fine tokens partitioned into nonempty groups $G_1, \dots, G_m$, with $n_j = |G_j|$. For a query q, keys k_i, and values v_i, define

$$y(q) = \sum_i \alpha_i(q) v_i, \qquad \alpha_i(q) = \frac{\exp(q^\top k_i/\sqrt{d_k})}{\sum_l \exp(q^\top k_l/\sqrt{d_k})}.$$

Let $\bar{k}_j, \bar{v}_j$ be representative keys and values, and assume

$$\|q\|_2 \le Q, \quad \|k_i - \bar{k}_j\|_2 \le \delta_k, \quad \|v_i - \bar{v}_j\|_2 \le \delta_v \quad (i \in G_j), \quad \|\bar{v}_j\|_2 \le V.$$

Define multiplicity-corrected weights and their output by

$$\beta_j(q) = \frac{n_j \exp(q^\top \bar{k}_j/\sqrt{d_k})}{\sum_l n_l \exp(q^\top \bar{k}_l/\sqrt{d_k})}, \qquad \bar{y}(q) = \sum_j \beta_j(q)\bar{v}_j.$$

This weighting is equivalent to adding $\log n_j$ to the representative-token logits, as in proportional attention (Bolya et al., 2022).

Proposition 2 (conditional single-query bound). Under the stated assumptions,

$$\|y(q) - \bar{y}(q)\|_2 \le \delta_v + V \min\left\{2, \frac{Q\delta_k}{\sqrt{d_k}}\right\}.$$

Proof. Define fine-token logits $a_i = q^\top k_i/\sqrt{d_k}$ and $b_i = q^\top \bar{k}_{g(i)}/\sqrt{d_k}$, where $g(i)$ denotes the group containing i. Then $\|a - b\|_\infty \le Q\delta_k/\sqrt{d_k}$. For a probability vector $p = \mathrm{softmax}(z)$, the softmax Jacobian satisfies

$$J_p h = p \odot (h - \mathbb{E}_p h), \qquad \|J_p h\|_1 = \mathbb{E}_p|h - \mathbb{E}_p h| \le \sqrt{\mathrm{Var}_p(h)} \le \|h\|_\infty.$$

Integration along the line segment between a and b gives

$$\|\mathrm{softmax}(a) - \mathrm{softmax}(b)\|_1 \le \min\{2, \|a - b\|_\infty\}.$$

Let $\widetilde{\alpha} = \mathrm{softmax}(b)$. Summing its n_j equal entries within a group yields β_j. Thus

$$\begin{aligned}\|y(q) - \bar{y}(q)\|_2 &\le \left\|\sum_i \alpha_i (v_i - \bar{v}_{g(i)})\right\|_2 + \left\|\sum_i (\alpha_i - \widetilde{\alpha}_i)\bar{v}_{g(i)}\right\|_2 \\ &\le \delta_v + V\|\alpha - \widetilde{\alpha}\|_1,\end{aligned}$$

which proves the result.

Ordinary uncorrected attention. Suppose the compressed model instead uses query $\widetilde{q}$ and ordinary weights

$$\gamma_j(\widetilde{q}) = \frac{\exp(\widetilde{q}^\top \bar{k}_j/\sqrt{d_k})}{\sum_l \exp(\widetilde{q}^\top \bar{k}_l/\sqrt{d_k})}, \qquad y_{\mathrm{ord}}(\widetilde{q}) = \sum_j \gamma_j(\widetilde{q})\bar{v}_j.$$

Assume additionally $\|q - \widetilde{q}\|_2 \le \delta_q$ and $\|\bar{k}_j\|_2 \le K$. Then

$$\|y(q) - y_{\mathrm{ord}}(\widetilde{q})\|_2 \le \delta_v + V \min\left\{2, \frac{Q\delta_k}{\sqrt{d_k}}\right\} + V \min\left\{2, \frac{K\delta_q}{\sqrt{d_k}}\right\} + V\|\beta(\widetilde{q}) - \gamma(\widetilde{q})\|_1.$$

The final term measures the effect of discarding group multiplicity. It can be nonzero even with identical keys and values within each group. For example, take two groups of sizes three and one, all logits zero, and representative scalar values zero and one. The fine-token output is $1/4$, whereas ordinary attention over the two representatives gives $1/2$.

Relation to our implementation. Our tokenizer resizes image regions before embedding; it does not average existing transformer keys and values. The bounds above can be applied to a chosen fine/coarse correspondence only after the key, value, and query deviations have been established. These deviations must include resampling and positional effects. The supplied sequence adapters do not explicitly add $\log n_j$ to attention logits; their learned size embedding is not an algebraic substitute for that correction. Furthermore, arbitrary quadtree leaves need not be unions of the original $p \times p$ grid cells, so a fine/coarse grouping must be defined before measuring these quantities. This is a conditional one-layer statement, not a bound on the complete segmentation network or on Dice loss.

Why PSD agreement is insufficient. Let P' be a nontrivial circular shift of a real patch P. Their discrete Fourier transforms differ by a unit-modulus phase factor, so $|\widehat{P}|^2 = |\widehat{P'}|^2$. Nevertheless, if $P \neq P'$, the linear functional $w^\top \mathrm{vec}(P)$ with $w = \mathrm{vec}(P - P')$ separates them by

$$w^\top \mathrm{vec}(P - P') = \|P - P'\|_F^2 > 0.$$

Thus even exact equality of power spectra does not imply equality of unrestricted linear patch embeddings. Similarity of a coarser texture summary cannot provide that implication without additional assumptions.

A.3 Summary Tables of Relevant and Used Models

Table 4: Adaptive tokenization methods for vision transformer

Method	Adaptivity	Temp.	Applied	Arch. Change	Supervision	Remarks
In-model token selection / pruning / merging						
TokenLearner (Ryoo et al., 2021)	Pooling	✓	In-model	Yes	End-to-end	Learns token importance jointly
DynamicViT (Rao et al., 2021)	Pruning	✗	In-model	Yes	End-to-end	Layer-wise gating sparsity
EViT (Liang et al., 2022)	Dropping	✗	In-model	Yes	End-to-end	Early discard of inattentive tokens
ToMe (Bolya et al., 2022)	Merging	✓	In-model	Yes	Unsupervised	Online merge; faster inference
FreqAwareTR (Lee et al., 2025)	Freq-aware	✗	In-model	Yes	End-to-end	Preserves high-frequency tokens
AD-TP (Yan et al., 2025)	Attribution	✗	In-model	Yes	Teacher+Self-supervised	Instance-specific token budgets
Architectural hierarchical bias						
Swin (Liu et al., 2021)	Windows	✗	Arch.	Yes	–	Static hierarchical bias
MViT (Fan et al., 2021)	Multiscale	✓	Arch.	Yes	–	Learns temporal hierarchy
Adaptive frame / region selection at input						
AdaFocus (Wang et al., 2021; 2022)	Cropping	✓	Input	No	End-to-end	Crops salient frame regions
AdaFrame (Wu et al., 2019)	Frames	✓	Input	No	Reinforcement	Learns adaptive frame skipping
TSN (Wang et al., 2016)	Segments	✓	Input	No	Supervised	Non-adaptive temporal coverage
Multimodal adaptive token pruning						
ATP (VLM) (Li et al., 2025)	Cross-modal	✗	In-model	No	Train-free	Prunes tokens via cross-modal attention
Adaptive tokenization at input (pre-embedding)						
AP (Zhang et al., 2024)	Quadtree patching	✗	Input (pre)	None	Unsupervised (edges)	AMR-inspired adaptive patching; no model modification
SHF (Zhang et al., 2025a)	Hierarchical forest	✗	Input (pre)	None	Unsupervised (edges)	Adaptive patching + symmetrical depatching; removes decoder
FlexTok (Bachmann et al., 2025)	Var-length tokens	✗	Input (pre)	No	Self-supervised	Flexible 1D token sequences
KARL (Duggal et al., 2025)	Token budgeting	✗	Input (pre)	No	End-to-end	Predicts per-sample token count
GPSToken (Zhang et al., 2025b)	Spatial tokens	✗	Input (pre)	Yes	End-to-end	Gaussian-parameterized adaptive regions
Texture-guided adaptive tokenization (this work)						
Ours	Texture Quadtree	✓	Input (pre)	None	Unsupervised (texture)	Spectral tokenization for texture-dominated regimes

Table 5: Detailed comparison of evaluated model configurations and adaptive tokenization strategies.

Method	Adaptivity	Remarks
Architectural baselines		
ViT-Base (Dosovitskiy et al., 2021)	None	Uniform patch grid
Swin (Liu et al., 2021)	Windows	Static hierarchical windows
SAM-Base (Kirillov et al., 2023)	None	Uniform encoder tokens
DINOv3 (Siméoni et al., 2025)	None	Self-supervised ViT with uniform patch tokens
RF-DETR (Robinson et al., 2026)	Static windows	DINOv2 backbone with a DETR detection head
In-model token selection and pruning		
TokenLearner (Ryoo et al., 2021)	Pooling	Learns token importance jointly
DynamicViT (Rao et al., 2021)	Pruning	Layer-wise gating sparsity
Adaptive region selection at input		
AdaFocusV2 (Wang et al., 2022)	Cropping	Crops salient frame regions
Adaptive tokenization at input [pre-embedding]		
AP (Zhang et al., 2024)	Quadtree patching	Adaptive patching without model modification
SHF (Zhang et al., 2025a)	Hierarchical forest	Adaptive patching and symmetrical depatching
FlexTok (Bachmann et al., 2025)	Variable length	Flexible token sequences
Texture-aware adaptive tokenization [this work]		
Texture AP [Ours]	Texture quadtree	Texture-guided adaptive patching
Texture SHF [Ours]	Texture forest	Texture-guided hierarchical partitioning

A.4 Dataset Summarization

Table 6: Datasets used for evaluation and their edge-density statistics.

Dataset	Modality	Number of Images	Edge Density
Wild Blueberry Fields (Bilodeau et al., 2023)	Multispectral and RGB aerial imagery	1,572	35.85%
Tree Plantation (Bilodeau et al., 2025)	RGB aerial imagery	1,440	31.01%
Mitochondria cross-sections (Lucchi et al., 2013)	Electron Microscope	330	28.79%

A.5 Additional Visualization of Texture Metrics

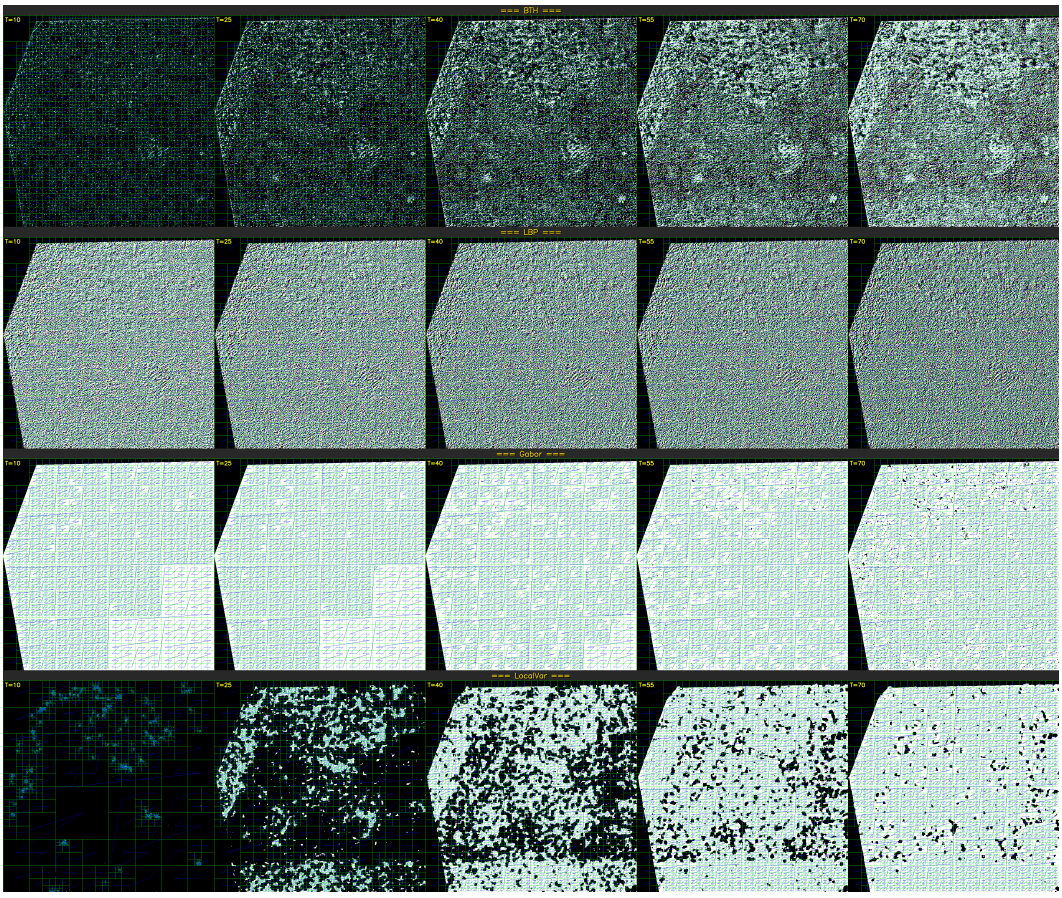


Figure 2: Qualitative grid-search of token allocation across sampled texture metrics and increasing thresholds (τ). The green overlays illustrate how different extractors partition the image under a fixed token budget.

A.6 Label-Free Threshold Selection for Texture-AP

The Texture-AP experiments in the primary analysis used ground-truth-calibrated texture thresholds, which are advantageous when spatial annotations are available because the texture mask can be optimized before training. However, this requirement limits applicability to unlabeled datasets. We therefore evaluate a label-free thresholding method that derives τ directly from the distribution of texture responses in the training images. This label-free approach combines a percentile rule with an elbow refinement. For each training image, a BTH or Wavelet response map is computed. These values are pooled across the dataset to form a dataset-level distribution, from which a single threshold is computed per dataset-method pair and applied consistently across the dataset.

For the fixed 30% token allocation, the initial threshold was set to the 70th percentile of the pooled response distribution. Consequently, approximately the highest 30% of BTH or Wavelet responses were designated as active. This active-response fraction guided the spatial distribution of the adaptive partitions, while fixed-length AP serialization enforced the final 30% token allocation.

The percentile threshold was refined using an elbow identified from the cumulative active-response curve. For each candidate threshold between 0 and 255, the active-response fraction was calculated as the proportion of pooled values exceeding that threshold. After normalizing both axes, the elbow was defined as the point with the greatest perpendicular distance from the line connecting the endpoints

of the curve. This point represented the strongest transition between the broadly distributed weaker responses and the smaller subset of stronger texture responses. The elbow was treated as a constrained refinement rather than an independent replacement for the percentile threshold. When the elbow and percentile thresholds differed by 20 intensity units or less, the elbow threshold was used. When the difference exceeded 20 intensity units, the percentile threshold was retained to prevent the active-response distribution from deviating substantially from the intended allocation. The selected value was subsequently restricted to the implemented threshold range of $[0, 100]$. This allowed the method to use a natural separation in the response distribution when it remained compatible with the target allocation, while the percentile threshold provided a fallback when the elbow was inconsistent with that budget.

A.6.1 Comparison of Selected and Label-Free Thresholds

The same label-free threshold-selection rule was applied independently to the BTH and Wavelet response distributions for the wild blueberry, tree plantation, and mitochondria datasets. Although the percentile-elbow procedure does not contain texture-specific selection criteria, the resulting threshold depends on the response distribution generated by the texture extractor. A separate label-free threshold was therefore obtained for each dataset-method pair (Table 7).

At the fixed 30% token allocation, each label-free threshold was compared with the corresponding ground-truth-calibrated threshold from the same texture method where the signed difference was defined as $\Delta\tau = \tau_{\mathrm{LF}} - \tau_{\mathrm{optimized}}$. Because each configuration produced one dataset-level threshold, these differences were evaluated descriptively. The effect of threshold selection on training efficiency was evaluated separately using the time-to-target Dice analysis (Section A.6.2).

Table 7 reports the calibrated threshold, label-free threshold, and signed difference for each dataset-method pair. Comparisons are made within each texture method; differences between the absolute BTH and Wavelet thresholds are not interpreted because their response distributions are generated and normalized independently.

Table 7: Within-method comparison of ground-truth-calibrated and label-free Texture-AP thresholds at a 30% token allocation.

Dataset	Texture method	$\tau_{\mathrm{Optimized}}$	τ_{LF}	$\Delta\tau$
Wild blueberry	BTH	33	50	17
Wild blueberry	Wavelet	7	8	1
Tree plantation	BTH	78	81	3
Tree plantation	Wavelet	38	29	9
Mitochondria	BTH	39	76	37
Mitochondria	Wavelet	27	55	28

As shown in Table 7, the absolute within-method differences between the calibrated and label-free BTH thresholds ranged from 3 to 37 intensity units across the three datasets. The corresponding Wavelet differences ranged from 1 to 28 intensity units. The greatest absolute difference occurred for BTH on the mitochondria dataset, whereas the closest agreement occurred for Wavelet on the wild blueberry dataset. These comparisons quantify how replacing ground-truth calibration with the label-free rule changed the operating point within each texture extractor.

A.6.2 Time-to-Target Dice Comparison

The practical effect of label-free threshold selection was evaluated using the cumulative training time required to reach the dataset-specific target Dice score. Ground-truth-calibrated and label-free configurations were trained independently at a fixed 30% token allocation, with all other preprocessing, model, optimization, and training settings held constant. Table 8 reports the mean and standard deviation across runs.

The label-free threshold had a lower mean time to target in three of the six dataset-method comparisons. For BTH, it reduced mean training time by 1.6% on tree plantation and 8.7% on mitochondria, but increased it by 25.3% on wild blueberry. For Wavelet, it increased mean time by 62.8% on wild

Table 8: Two-sample t-test comparisons of time-to-target Dice between ground-truth-calibrated and label-free Texture-AP thresholds at a 30% token allocation. Times are reported as mean ± SD in seconds.

Dataset	Texture method	$\tau_{\mathrm{Optimized}}$ time (s)	τ_{LF} time (s)
Wild blueberry	BTH	1051.13 ± 577.82	1317.29 ± 778.99
Wild blueberry	Wavelet	1565.13 ± 601.23	2548.02 ± 1533.42
Tree plantation	BTH	255.56 ± 92.94	251.51 ± 16.31
Tree plantation	Wavelet	108.70 ± 35.88	123.93 ± 43.77
Mitochondria	BTH	306.15 ± 5.59	279.58 ± 10.92
Mitochondria	Wavelet	349.52 ± 24.74	295.55 ± 7.23

blueberry and 15.4% on mitochondria, but increased it by 14.0% on tree plantation. Taken together, the results show that label-free threshold selection can achieve time-to-target Dice comparable to ground-truth-calibrated selection in several settings, although its effect varies by dataset and texture method.

A.7 Classification Results

The computational speedup achieved during training while reaching a top 1 classification accuracy of 90% for hair fescue weeds, utilizing imagery derived from the wild blueberry segmentation dataset, was evaluated at a fixed 30% token allocation (Table 9). The standard ViT-Base architecture served as the baseline performance metric, where the edge driven SHF approach accelerated computational performance to a 1.43× speedup for training wall-clock time. The Texture SHF configuration further optimized processing efficiency, achieving the 90% accuracy target with a 2.62× speedup on training wall-clock time. These benchmarks confirmed that texture guided hierarchical partitioning effectively minimized redundant background processing, yielding maximum classification efficiency in complex agricultural environments and furthering the findings from Section 6.4.

Table 9: Speedup (based off total time) to achieve a top 1 accuracy of 90% for hair fescue weed classification in aerial wild blueberry imagery.

Method	↑ Speedup @ Accuracy = 0.90
ViT [Base] (Dosovitskiy et al., 2021)	1.00 c
SHF (Zhang et al., 2025a)	1.43 b
Texture SHF [Ours]	**2.62** a
Means that do not share a letter are significantly different.	Comparisons are within the column only.

A.8 End-to-End Wall-Clock Breakdown

A.8.1 End-to-End Inference Breakdown

End-to-end inference wall-clock time was evaluated to determine whether reductions in backbone processing time offset the additional extraction and partitioning costs introduced by adaptive tokenization. Timing began when an input image became available in memory and ended after the final prediction was produced. Extraction time included generation of the Canny, BTH, or Wavelet response; partition time included construction of the QuadTree or forest; embedding time included patch extraction, resizing, ordering, and token embedding; and prediction time included the prediction head and output reconstruction. Disk I/O was excluded. Measurements were performed using a single image at a time approach. Each configuration was evaluated using 100 images sampled from the evaluation datasets following 5 warm-up iterations. Table 10 reports the mean wall-clock time in ms/image.

Table 10: End-to-end inference wall-clock breakdown across model and preprocessing configurations at different token allocations. Timing values are reported as mean ± SD in ms/image.

Model	Method	Token allocation	Wall-clock time (ms/image)						Speedup
			Extraction	Partition	Embedding	Backbone	Prediction	Total	
ViT-B	Uniform grid	100%	–	66.57 ± 14.74	47.77 ± 12.16	120.01 ± 5.68	2.19 ± 0.31	236.55 ± 15.92	1.00×
	SHF (Canny)	50%	1.61 ± 4.75	60.32 ± 28.39	24.53 ± 13.41	44.17 ± 4.20	3.02 ± 3.73	133.65 ± 32.51	1.77×
		30%	1.14 ± 4.06	30.63 ± 14.35	12.03 ± 7.91	24.74 ± 7.56	0.79 ± 1.42	69.33 ± 16.80	3.41×
		10%	1.61 ± 4.75	10.32 ± 7.69	4.59 ± 7.12	16.62 ± 24.25	0.23 ± 0.40	33.38 ± 26.31	7.09×
	Texture-SHF (BTH)	50%	14.91 ± 5.60	65.10 ± 18.04	19.32 ± 9.64	44.22 ± 7.22	8.43 ± 7.40	151.99 ± 26.90	1.56×
		30%	15.10 ± 5.48	33.10 ± 11.03	9.85 ± 7.64	26.23 ± 5.35	3.17 ± 5.81	87.44 ± 14.72	2.71×
		10%	16.34 ± 5.98	11.13 ± 8.67	4.12 ± 6.88	12.09 ± 2.22	0.22 ± 0.17	43.89 ± 11.01	5.39×
	Texture-SHF (Wavelet)	50%	22.35 ± 8.20	66.52 ± 19.10	18.09 ± 6.65	47.03 ± 10.46	8.52 ± 7.72	162.51 ± 25.77	1.46×
		30%	24.43 ± 8.39	32.72 ± 9.66	10.70 ± 7.26	29.87 ± 6.23	2.14 ± 3.23	99.86 ± 13.72	2.37×
		10%	27.41 ± 8.48	10.27 ± 7.51	5.02 ± 7.30	13.89 ± 1.51	0.26 ± 0.24	56.86 ± 10.43	4.16×
Swin	Uniform grid	100%	–	–	4.78 ± 7.20	23.72 ± 0.99	2.12 ± 0.17	30.63 ± 7.24	1.00×
	SHF (Canny)	50%	1.37 ± 4.42	71.40 ± 26.80	7.72 ± 7.91	95.83 ± 26.83	3.35 ± 0.52	179.68 ± 36.76	0.17×
		30%	1.70 ± 4.87	29.36 ± 12.08	6.16 ± 7.63	52.78 ± 11.18	2.56 ± 0.52	92.56 ± 17.88	0.33×
		10%	1.42 ± 4.49	8.43 ± 7.79	5.26 ± 7.77	31.71 ± 4.02	2.38 ± 0.58	49.19 ± 8.44	0.62×
	Texture-SHF (BTH)	50%	19.74 ± 8.34	82.39 ± 16.71	10.65 ± 9.79	35.89 ± 25.17	13.46 ± 22.51	162.14 ± 39.99	0.19×
		30%	19.84 ± 7.64	34.09 ± 8.90	7.39 ± 7.80	39.32 ± 19.73	3.19 ± 0.28	103.83 ± 21.08	0.29×
		10%	20.60 ± 7.38	7.48 ± 7.81	5.82 ± 11.12	26.14 ± 5.43	2.43 ± 0.69	62.47 ± 13.01	0.49×
	Texture-SHF (Wavelet)	50%	24.86 ± 11.07	82.05 ± 16.50	11.74 ± 9.64	32.49 ± 18.60	21.28 ± 29.37	172.43 ± 41.36	0.18×
		30%	27.65 ± 9.83	34.90 ± 11.38	7.77 ± 7.81	31.38 ± 12.82	3.17 ± 0.35	104.86 ± 18.30	0.29×
		10%	28.93 ± 6.65	7.39 ± 7.80	5.78 ± 7.54	27.23 ± 5.85	2.58 ± 0.76	71.90 ± 9.20	0.43×
SAM	Uniform grid	100%	–	43.80 ± 15.82	68.09 ± 13.76	117.86 ± 3.41	2.15 ± 0.29	231.89 ± 11.54	1.00×
	SHF (Canny)	50%	1.80 ± 4.99	65.20 ± 28.65	24.53 ± 13.18	45.59 ± 8.34	2.94 ± 4.14	140.06 ± 32.04	1.66×
		30%	1.28 ± 4.28	30.45 ± 15.53	12.50 ± 8.11	25.40 ± 7.41	0.70 ± 1.15	70.32 ± 18.27	3.30×
		10%	1.47 ± 4.56	9.66 ± 9.02	4.02 ± 6.83	14.66 ± 4.78	0.21 ± 0.29	30.02 ± 10.69	7.72×
	Texture-SHF (BTH)	50%	15.67 ± 5.51	66.15 ± 18.19	18.66 ± 9.25	44.16 ± 6.45	8.37 ± 7.24	153.00 ± 24.46	1.52×
		30%	15.77 ± 5.51	34.42 ± 11.46	9.61 ± 7.60	26.21 ± 5.61	3.47 ± 6.16	89.48 ± 14.49	2.59×
		10%	15.10 ± 5.21	10.61 ± 8.77	4.73 ± 7.18	12.28 ± 1.73	0.17 ± 0.21	42.90 ± 11.34	5.41×
	Texture-SHF (Wavelet)	50%	24.10 ± 8.06	64.77 ± 17.10	18.28 ± 6.58	47.00 ± 9.81	8.39 ± 7.47	162.54 ± 23.55	1.43×
		30%	24.38 ± 8.04	33.76 ± 10.76	10.46 ± 7.45	30.20 ± 7.70	2.16 ± 2.90	100.96 ± 13.72	2.30×
		10%	24.38 ± 8.91	10.80 ± 8.71	4.64 ± 7.14	13.72 ± 2.17	0.21 ± 0.19	53.75 ± 10.63	4.31×
DINOv3	Uniform grid	100%	–	63.35 ± 16.11	50.52 ± 13.30	119.47 ± 4.97	2.19 ± 0.51	235.53 ± 15.33	1.00×
	SHF (Canny)	50%	1.47 ± 4.56	60.61 ± 29.86	25.43 ± 13.25	43.44 ± 6.52	4.05 ± 4.78	135.00 ± 32.64	1.74×
		30%	1.56 ± 4.69	29.50 ± 15.88	12.93 ± 8.12	24.39 ± 8.21	0.85 ± 1.54	69.22 ± 18.74	3.40×
		10%	1.14 ± 4.06	9.94 ± 7.71	3.88 ± 6.75	13.62 ± 2.13	0.20 ± 0.19	28.78 ± 9.74	8.18×
	Texture-SHF (BTH)	50%	15.39 ± 5.77	64.63 ± 18.14	18.18 ± 8.65	43.96 ± 7.43	8.35 ± 7.28	150.51 ± 26.83	1.56×
		30%	15.77 ± 6.02	33.14 ± 10.57	10.61 ± 7.30	25.10 ± 4.59	3.05 ± 3.53	87.67 ± 13.93	2.69×
		10%	15.53 ± 5.01	9.61 ± 7.70	4.59 ± 7.12	11.33 ± 1.62	0.25 ± 0.31	41.32 ± 9.78	5.70×
	Texture-SHF (Wavelet)	50%	23.63 ± 8.18	64.63 ± 19.29	19.37 ± 7.30	43.18 ± 8.83	9.02 ± 9.13	159.83 ± 25.83	1.47×
		30%	23.82 ± 8.26	33.85 ± 9.88	10.61 ± 7.59	29.33 ± 6.73	2.57 ± 3.46	100.17 ± 13.53	2.35×
		10%	24.76 ± 8.17	11.22 ± 7.24	4.69 ± 7.16	12.97 ± 1.91	0.27 ± 0.24	53.92 ± 10.10	4.37×
RF-DETR	Uniform grid	100%	–	54.55 ± 17.38	178.82 ± 94.29	242.38 ± 91.77	68.71 ± 30.74	544.45 ± 134.25	1.00×
	SHF (Canny)	50%	1.64 ± 0.74	81.00 ± 33.60	94.92 ± 54.50	105.49 ± 28.52	91.20 ± 33.30	374.25 ± 82.80	1.45×
		30%	1.71 ± 0.73	41.06 ± 15.33	81.96 ± 45.75	51.72 ± 13.91	85.45 ± 35.15	261.89 ± 76.49	2.08×
		10%	1.71 ± 0.76	12.89 ± 2.86	78.63 ± 40.22	18.14 ± 2.48	80.22 ± 36.03	191.59 ± 71.10	2.84×
	Texture-SHF (BTH)	50%	19.43 ± 2.83	102.45 ± 95.44	84.19 ± 46.19	85.31 ± 33.07	109.48 ± 31.54	400.86 ± 127.71	1.36×
		30%	21.98 ± 3.57	51.79 ± 11.48	76.21 ± 87.56	44.97 ± 20.11	81.18 ± 34.64	276.13 ± 106.85	1.97×
		10%	20.34 ± 1.23	12.52 ± 0.73	62.69 ± 42.37	15.62 ± 4.60	77.82 ± 36.14	188.98 ± 75.80	2.88×
	Texture-SHF (Wavelet)	50%	24.52 ± 3.21	88.58 ± 13.38	88.24 ± 41.25	88.70 ± 32.32	109.79 ± 31.24	399.84 ± 61.83	1.36×
		30%	27.94 ± 4.06	47.89 ± 9.25	69.53 ± 42.85	48.83 ± 19.30	84.08 ± 34.79	278.27 ± 76.26	1.96×
		10%	31.15 ± 28.75	12.64 ± 1.66	48.92 ± 50.41	14.57 ± 4.57	69.32 ± 35.81	176.60 ± 92.19	3.08×

"–" indicates that the processing stage was not applicable.
Speedup was calculated relative to uniform-grid inference within each model.

As shown in Table 10, the adaptive SHF configurations generally achieved greater end-to-end speedups as token allocation decreased. Swin followed the same directional trend but did not achieve a net speedup at any evaluated allocation. Unlike the others, Swin already restricts attention to local windows and exhibited substantially lower uniform-grid inference latency. Consequently, the extraction and partitioning overhead, combined with the absence of a corresponding reduction in backbone latency, outweighed the computational savings associated with the reduced token count. Canny achieved the highest end-to-end speedups across the evaluated methods because its extraction time remained below 2 ms. BTH introduced approximately 15-20 ms of extraction overhead; nevertheless, at a 50% token allocation, it reduced backbone latency on ViT-B, SAM, DINOv3, and RF-DETR from 117.86-242.38 ms to 43.96-85.31 ms, representing a minimum backbone time saving of 73.70 ms. After accounting for extraction, partitioning, embedding, and prediction, BTH retained a minimum end-to-end speedup of 1.36× among the measured non-Swin models. Wavelet extraction incurred the greatest preprocessing overhead and therefore produced lower end-to-end speedups on average at equivalent token allocations.

A.9 Robustness to Noise

To evaluate reliability under typical sensor noise and optical blurring, the structural stability of the chosen BTH metric was compared against standard Canny edge extraction. Additive Gaussian noise ($\sigma \in \{1, 2, 4, 8\}$) and Gaussian blur ($k \in \{1, 2, 4, 8\}$) were injected into the input imagery. Fixed

QuadTrees were subsequently constructed across multiple token allocations (25, 50, and 75%) and evaluated against clean baselines. To quantify partitioning stability independently of downstream accuracy, QuadTree divergence was measured using two metrics:

1. **Centroid Chamfer Similarity ($S_{chamfer}$):** The normalized Chamfer distance between the center coordinates of clean and degraded tokens was computed to yield a similarity score $S_{chamfer} \in [0, 1]$, measuring spatial anchor retention.
2. **Structural Intersection over Union (SIoU):** The boundary IoU between QuadTree binary maps was calculated to assess hierarchical depth consistency.

The SIoU and ($S_{chamfer}$) metrics were applied across each dataset described in subsection 6.1 for both Canny and BTH methods of QuadTree construction. The averages for both impact from noise and blur were reported in respect to percent of token allocation (Table 11). From Table 11, it is apparent that Gaussian noise had little effect on BTH-guided QuadTree construction (greater than 91% for both $S_{chamfer}$ and SIoU) whereas Canny-guided construction was approximately 4% lower in the more degraded QuadTree constructions (σ level=3, at 50% token allocation) (greater than 85% for both $S_{chamfer}$ and SIoU). However, the dominant difference between the two QuadTree construction methods comes from the Gaussian blurring effect, where the BTH-guided QuadTrees were approximately 85% similar to no blurring at a minimum, but the Canny-guided QuadTrees degraded significantly, resulting in SIoU scores below 60% (Table 11).

Table 11: Robustness comparison across edge-based and morphological extractors under varying perturbation severities and token coverage budgets. Higher values denote better stability for both Centroid Chamfer Similarity ($S_{chamfer}$) and Structural Intersection over Union (SIoU).

Perturbation	Coverage	Method	Metric	Level 1	Level 2	Level 3	Level 4
Gaussian Noise (σ)	25%	BTH	$S_{chamfer} \uparrow$	99.51	99.01	98.94	98.34
			SIoU $\uparrow$	98.38	97.25	96.65	94.41
		Canny	$S_{chamfer} \uparrow$	99.21	98.96	98.62	97.88
			SIoU $\uparrow$	96.71	95.80	92.77	92.21
	50%	BTH	$S_{chamfer} \uparrow$	99.47	99.17	98.64	98.23
			SIoU $\uparrow$	97.52	96.28	92.44	91.17
		Canny	$S_{chamfer} \uparrow$	99.04	98.75	98.21	97.88
			SIoU $\uparrow$	94.48	93.11	87.02	92.21
	75%	BTH	$S_{chamfer} \uparrow$	99.53	99.14	98.79	98.34
			SIoU $\uparrow$	97.67	96.47	93.18	91.63
		Canny	$S_{chamfer} \uparrow$	99.24	98.99	98.71	97.91
			SIoU $\uparrow$	95.25	93.99	88.45	89.10
Gaussian Blur (k)	25%	BTH	$S_{chamfer} \uparrow$	100.00	99.18	98.83	98.13
			SIoU $\uparrow$	100.00	96.05	94.44	91.43
		Canny	$S_{chamfer} \uparrow$	100.00	97.07	95.02	87.11
			SIoU $\uparrow$	100.00	88.47	81.13	57.57
	50%	BTH	$S_{chamfer} \uparrow$	100.00	98.81	98.36	97.48
			SIoU $\uparrow$	100.00	92.25	89.66	84.99
		Canny	$S_{chamfer} \uparrow$	100.00	97.00	95.29	88.08
			SIoU $\uparrow$	100.00	84.79	77.71	53.42
	75%	BTH	$S_{chamfer} \uparrow$	100.00	99.03	98.64	97.87
			SIoU $\uparrow$	100.00	92.86	90.38	85.93
		Canny	$S_{chamfer} \uparrow$	100.00	97.32	95.64	88.43
			SIoU $\uparrow$	100.00	85.07	77.17	50.95

A.10 Training Configuration and Hyperparameters

All models were optimized using the AdamW optimizer with a constant learning rate of 10^{-4}. While the maximum training duration was set to 1000 epochs, the system utilized early-stopping criteria based on validation performance, both for target dice and patience based or a predefined compute budget to minimize unnecessary energy consumption. For the FlexTok variant, the tokenizer remained frozen while only the embedding table and convolutional decoder were trained. Similarly, ImageNet-pretrained weights were utilized for the Swin-Transformer, ViT, and SAM encoders where available. All configurations were trained using a fixed batch size of 16 across three randomly selected seeds: 2, 10, and 40.

A.10.1 Loss Function and Normalization

To optimize for both pixel-level accuracy and structural overlap, and since each dataset only had one target class, a composite DiceBCE Loss was utilized. This function combined standard binary cross-entropy (BCE) with a Soft-Dice coefficient loss to handle potential class imbalances in dense textures:

$$\mathcal{L}_{total} = \mathcal{L}_{BCE} + (1 - \text{Dice})$$

For SHF and AP variants, the loss was calculated across the reconstructed patch sequence.

A.11 Statistical Analysis

To evaluate the significance of performance variations across model architectures and texture extraction metrics, statistical significance was assessed using Analysis of Variance (ANOVA). Following the identification of significant main effects, Tukey's Honestly Significant Difference (HSD) post-hoc test was utilized to perform pairwise comparisons between the proposed texture-driven frameworks and spatial baselines. All tests were conducted at a significance level of $\alpha = 0.05$ using Minitab 21 (Minitab, LLC, Minitab 21, State College, PA, USA), with performance metrics aggregated across multiple experimental seeds to ensure the stability and reliability of the reported results.