Pixel-Level Predictions with Embedded Lookup Tables

Marius Aasan[1,2,*], Adín Ramírez Rivera[1,2]

[1]Department of Informatics, University of Oslo, Problemveien 11, 0313 Oslo, Norway
[2]SFI Visual Intelligence, P.O. Box 6050 Langnes, 9037 Tromsø, Norway

Abstract

Pixel-level prediction tasks inherently face constraints imposed by image resolution and the number of prediction classes. As image resolutions and dimensionalities increase, these constraints lead to significant memory bottlenecks when explicitly modeling high-dimensional embeddings for each individual pixel. Addressing these bottlenecks requires the development of alternative, more efficient representations to facilitate continued progress in the field. In this work, we discuss Embedded Lookup Tables (ELUTs) with indexed segmentation maps as an alternative data structure for more memory efficient representations in image processing. We show that ELUTs are inherently compatible with cost functionals and metrics for pixel-level prediction tasks with a significant reduction in memory overhead.

Keywords

Computer Vision, Segmentation, Representation Learning, Data Structures, Image Compression

1. Introduction

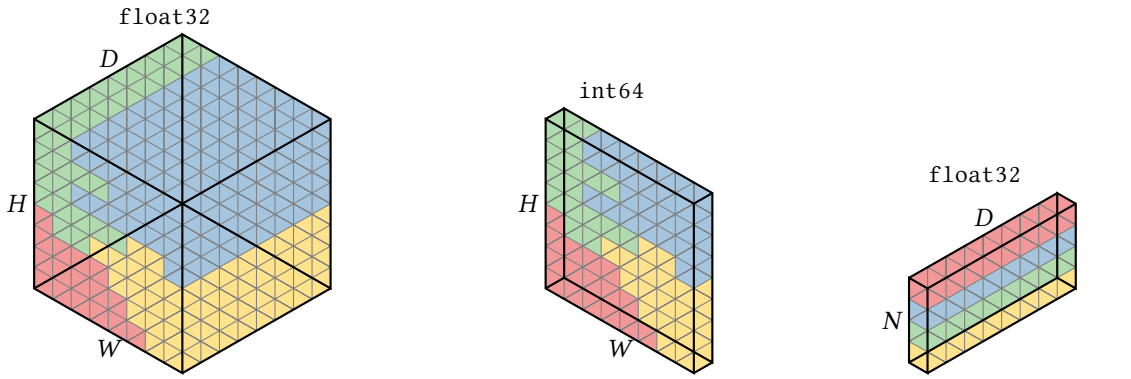
Neural models for dense prediction tasks are designed to extract descriptive features to perform accurate predictions for each individual pixel. Modern architectures typically leverage pyramidal structures for multilevel feature extraction [1, 2] to optimize pixel-level representations to delineate high-level object representations in images with a classification head. In standard formulations, dense image representations are modeled as $z \in \mathbb{R}^{D \times H \times W}$, where D is the feature dimension and H, W denote the spatial resolution. Established results from learning theory tells us that D should ideally be selected to be close to or higher than the number of prediction classes K for robust modeling in classification tasks [3, 4, 5]. Consequently, as number of classes K grows, so too must the feature dimension D.

Most imaging domains exhibit pronounced spatial redundancy [6, 7, 8]. A significant number of neighboring pixels will necessarily belong to the same semantic or instance level class, and naturally contain embeddings that are highly similar to adjacent pixel embeddings. In order to maintain pixel-level granularity in semantic and instance-level boundaries, intermediate representations and predictions need to match the original resolution in order to accurately delineate objects at the pixel-level.

While inherent redundancy is not an issue in and of itself, the problem is exacerbated by two clear trends in image processing. First of all, image resolutions increase over time as imaging advances and compute becomes more available [9]. Moreover, embedding dimension grow as categories or classes increase in line with modeling advances [10], and the shift towards dynamic open-vocabulary classification and self-supervised learning paradigms directly necessitates higher dimensional embeddings [11]. As a result, fully realized tensor representations $z \in \mathbb{R}^{D \times H \times W}$ become increasingly costly, and present a bottleneck for further scaling of pixel-level prediction models.

In this work, we introduce Embedded Lookup Tables (ELUTs) as a low-cost alternative to high-dimensional dense image representations. Grounded in principles from computer graphics [12, 13], ELUTs offer a compact and efficient formulation for pixel-level predictions in modern modeling pipelines. We derive theoretical bounds on memory and compute complexity and validate them empirically, showing order-of-magnitude reductions in peak memory.

NAIS 2025: Symposium of the Norwegian AI Society, June 17–18, 2025, Tromsø, Norway
*Corresponding author.
mariuaas@uio.no (M. Aasan); adinr@uio.no (A. R. Rivera)
0000-0003-2353-9984 (M. Aasan); 0000-0002-4321-9075 (A. R. Rivera)



(a) Dense representation. **(b)** Indexed segmentation. **(c)** Embedded lookup table.

Figure 1: Illustration of a fully realized $D \times H \times W$ dense representation **(a)**, compared to an $H \times W$ indexed segmented representation **(b)** with an $N \times D$ embedded lookup table **(c)**. Our proposition utilizes (c) and (b) to form a sparse representation of (a) with significantly lower memory overhead.

2. Related Work

Dense raster tensors are natural representations for convolutional networks (CNNs), whose design natively consumes and emits features in grids. As discussed in Sec. 1, the demands of higher resolution images in conjunction with higher embedding dimensions indicates that the $\mathcal{O}(DHW)$ overhead is a bottleneck. Current state-of-the-art foundational models [14] make use of over 131k pseudo-classes for instance- and patch-level predictions. Extending this to pixel-level predictions remains computationally intractable without alternative representations. While open-vocabulary approaches [15, 16] alleviate the issue by aligning with natural language embeddings instead of fixed classes, they necessitate high-dimensional embeddings and induce the same bottleneck.

Frame Buffers and Palettes. In the formative stages of computer graphics, Kajiya et al. [12] introduced the modern frame buffer for overcoming limitations of memory and compute. Early renderers were designed such that each pixel stores an index into a color lookup table (CLUT) with RGB color representations to enable compact frame buffers and real-time hardware color mapping. Later work expanded on quantization and dithering for optimizing color representations in imaging [13], paving the way for further developments with palette-based image quantization and lookups for hardware and software rendering. Color quantization and CLUTs is actively used for effective photometric transforms in modern GPU rendering [17] and compression in standard image formats [18, 19].

Superpixels and Objecthood. One remedy to the dense bottleneck is to decouple *where* a feature lives from *what* the feature is. If a colour-lookup table can recover an RGB triplet from an index, the same pattern generalises: store a D-dimensional feature vector in the table and let neighbouring pixels that share semantics reuse the index. Superpixels [20] were introduced for this exact purpose, and serve as a strong indicator for pre-segmentation by partitioning an image into $N \ll HW$ connected regions. By associating each region with an index and each index with a feature embedding, we can construct image representations that align with semantic content in an image.

Token-based Architectures. Research has seen widespread adoption of Vision Transformers [21] (ViTs), representing a move away from local receptive fields towards attention operators to model global interaction between discrete tokens. Tokenization has largely been driven by a simplified patchification procedure; square patches are extracted as tokens with positional embeddings to encode spatial information. ViTs for pixel-level predictions have focused on patch upsampling and concatenation of multiple layer activations [22, 23] to yield full resolution predictions. While studies have shown

that smaller patches are beneficial for predictive power [24, 25], the computational complexity of more granular tokens is quadratic [26, 27] due to the attention operator.

Recent work have looked to improve on the patch-based tokenization process in a move towards more adaptive tokenization methods [28, 29]. Adaptive superpixel tokenization have been successfully applied to allow ViTs to work with pixel level granularity [30, 31], and have been shown to improve state-of-the-art open-vocabulary segmentation models [32]. As every region carries an integer index, the image now is an index map; the per-region embeddings form a lookup table, giving exactly the palette-style pairs illustrated in Fig. 1.

Convolutions on Discrete Topologies. Although CNNs are used for feature extraction and superpixel tokenization [33], they are not inherently suited to discrete embedding spaces in the way transformers are. By design, a CNN natively consumes and emits features in grids necessitating a fully realized dense representation $z \in \mathbb{R}^{D\times H\times W}$. However, the fundamental neighborhood operations performed by a CNN can be extended to discrete topologies via graph neural networks [34, 35, 36] (GNNs). Superpixel-based GNNs have been successfully applied in medical imaging [37, 38], and the recent long range graph benchmark [39] (LRGB) include two superpixel graph datasets for segmentation, encouraging further research in vision based GNN approaches.

3. Embedded Lookup Tables with Indexed Segmentation Maps

Adaptive tokenization in ViTs with semantic edge detection provides an opportunity to rethink underlying data structures for pixel-level prediction. Because the multi-head self-attention operator is inherently permutation-invariant, spatial relationships must be injected through positional encodings rather than being implicit via grid topology. This observation suggests that we no longer need to commit to a fixed, dense lattice as the primitive representation. Instead, we encode each image as a pair

$$\big(I \in \{1,\dots,N\}^{H\times W},\ T \in \mathbb{R}^{N\times D}\big), \tag{1}$$

where $I(p)$ assigns each pixel p to one of N superpixel tokens, and $T_i \in \mathbb{R}^D$ is the D-dimensional feature for token i. In training, a batch of B images $\{(I_b, T_b)\}_{b=1}^{B}$ is combined by simple concatenation of tables

$$I_B(p) = I_b(p) + \sum_{j<b} N_j, \quad T_B = [T_1, T_2, \dots, T_B]. \tag{2}$$

No padding is required – each regions index range is shifted by the cumulative token counts of preceding samples. This is the same implementation used in GNNs; variable-size graphs are merged into a single supergraph by offsetting node indices and merging edge lists, preserving each subgraph's topology without forcing a common size [40].

Conceptually, ELUTs can be viewed an extension of indexed color representations in early frame buffers; I is a two-dimensional index map and T is a lookup table of learned embeddings. The memory savings follow immediately; instead of storing $D\times H\times W$ dense features, we store only $D\times N$ entries plus an $H\times W$ integer map. As long as the segmentation captures spatial redundancy this yields compression of pixel-level features, with support for batching of heterogeneous resolutions and token counts.

3.1. Implementation Details

An ELUT representation is straightforward to apply in ViTs, since token representations correspond directly to the embeddings in the lookup table. Any tokenization process that partitions the image into connected regions can be applied to extract segmentation maps $I_b(p)$. Different approaches can be applied for feature extraction; interpolation to a fixed feature size via bounding-boxes in addition to a joint histogram for positional embeddings provides commensurability with existing ViT approaches [30].

Alternatively, a vision encoder such as a lightweight CNN can be applied for feature extraction to construct a dense feature map $f(x) \in \mathbb{R}^{D \times H' \times W'}$ [31, 32], and features can be extracted by taking

$$T_{b,i} = \bigoplus_{p\,:\,I(p)=i} U\big(f(x)\big)_p \tag{3}$$

where U is an upsampling operator and $\oplus$ denotes an aggregation. Positional embeddings are aggregated by parametric or learnable positions for each region, with optional bounding box coordinates.

Both approaches to feature extraction provide strong results in downstream tasks, however the former approach is designed to be modularly commensurable with standard ViTs, while the latter can provide richer semantic features for predictive tasks. Variable length table sizes can be implemented for multi-head attention in ViTs either via a ragged tensor representation or via zero-padding all lookup tables to a fixed size.

Tokenization Strategies: Tokenization is central to construct an effective ELUT representation. Regions are constructed as partitions of an image for which each pixel is expected to have the same label in the downstream task. Superpixel segmentation with SLIC [41] is a common choice [33, 31] using centroid clustering, with the caveat that post-processing must be applied to get topologically consistent regions.

Recent approaches have explored alternative pre-segmentation strategies with stronger topological guarantees. EPOC [32] uses an transformer-based vision encoder to extract semantic edges with full receptive field, which is then segmented using watershed [42] in a morphological approach with semantic boundaries. SPiT [30] applies ground-up community detection mechanisms using a graph-based hierarchical region merging [43].

Parallel computations. Interestingly, watershed and graph region merging approaches give two different approaches to flood filling [44, 45]. Watershed applies filling based on a set of seeds given a gradient image, while graph region merging by processing all pixels uniformly in parallel while updating the graph representation. Specifically, both approaches can be viewed as optimal spanning forests [46, 47] and are equivalent to optimizing a random walk over the graph [48]. Efficient implementations in PyTorch [49] can be constructed without external dependencies and custom CUDA kernels [30].

3.2. Space Complexity Analysis

Let $P = H \times W$ denote the number of pixels in an image, D the feature dimension, and $N \leq P$ the number of regions. We compare the memory footprint of a standard dense feature map

$$M_{\text{dense}} = D \times P \tag{4}$$

against that of an ELUT representation, which is given by

$$M_{\text{ELUT}} = \underbrace{P}_{\text{index map } I \in \{1,\ldots,N\}^{H \times W}} + \underbrace{DN}_{\text{lookup table } T \in \mathbb{R}^{N \times D}}. \tag{5}$$

We can express the relative memory usage as

$$\frac{M_{\text{ELUT}}}{M_{\text{dense}}} = \frac{DN + P}{DP} = \frac{N}{P} + \frac{1}{D} = \alpha + \frac{1}{D}, \quad \alpha \equiv \frac{N}{P} \in (0, 1]. \tag{6}$$

Given this, ELUTs provide an overall memory *reduction factor*

$$\rho = \frac{M_{\text{dense}}}{M_{\text{ELUT}}} = \frac{1}{\alpha + 1/D}. \tag{7}$$

Sufficiently high-dimensional embeddings ($D \gg 1$) gives a "rule-of-thumb" estimate $\rho \approx \alpha^{-1}$. The savings can be illustrated by the following example; if N compress to $\alpha = 0.05$ of the pixel count, ELUTs use only $\sim 5\%$ of the dense footprint – a $\sim 20\times$ savings. For an extended empirical analysis, see Sec. 4.1.

Worst-case vs. best-case. For $N = P$ (no compression), $\alpha = 1$ and $M_{\text{ELUT}}/M_{\text{dense}} = 1 + 1/D$, hence ELUTs incur negligible extra overhead in high-dimensional regimes. In the ideal limit $N \ll P$, memory scales as $\mathcal{O}(P + DN) \ll \mathcal{O}(DP)$. This demonstrates how ELUTs improve on the $\mathcal{O}(DHW)$ bottleneck of dense tensor representations, trading off a linear-in-pixels index map for a much smaller, segmentation-driven embedding table.

Batch Processing. Eq. (2) shows that ELUT batching can be implemented by table concatenation without overhead. For a batch of B images with pixel counts P_b and token counts N_b,

$$M_{\text{dense}}^{(B)} = D \sum_b P_b, \quad M_{\text{ELUT}}^{(B)} = D \sum_b N_b + \sum_b P_b, \tag{8}$$

and the same ratio analysis applies, giving

$$\alpha_B = \frac{\sum_b N_b}{\sum_b P_b}. \tag{9}$$

3.3. Equivalence of Cost functionals and Metrics

ELUT representation (I, T) allows for standard computation of cost metrics – such as MSE, cross-entropy, or focal loss – as well as metrics such as Jaccard Index (IoU) or DICE coefficients. Given a set of predicted regions $I : \mathcal{P} \to \{1, \dots, N\}$ as well ground truth labels $G : \mathcal{P} \to \{1, \dots, K\}$, we define region sizes and the region–class contingency counts

$$n_i = \sum_{p \in \mathcal{P}} \mathbf{1}_{[I(p)=i]}, \qquad C_{ic} = \sum_{p \in \mathcal{P}} \mathbf{1}_{[I(p)=i]}\, \mathbf{1}_{[G(p)=c]}, \tag{10}$$

where $\mathbf{1}_{[\cdot]}$ denotes the Iverson bracket. Any loss or metric expressible as a sum over pixels $\mathcal{L} = \sum_{p \in P} \ell(z_p, G_p)$ can be rewritten exactly as a weighted sum over the contingency table

$$\mathcal{L} = \sum_{i=1}^{N} \sum_{c=1}^{K} C_{ic}\, \ell(T_i, c), \tag{11}$$

recalling that T_i is the embedding or logits emitted by token i and c is an target embedding or class index. Denote by $\mathcal{S} = \{(i, c) \,:\, C_{ic} > 0\}$ the set of *non-empty* region–class overlaps. We can then take (11) as an *identity*; it trades P pixel visits for the at most $S = |\mathcal{S}|$ non-empty overlaps.

Batch Processing. Computing cost functionals and metrics in a mini-batch regime requires only a slight modification for computing contingency tables, where each ground-truth segment needs to be unique within each sample. This can be achieved by concatenating $G_B = [G_b + (b-1)K]_{b=1}^{B}$, such that the original targets can be recovered using a simple modulo operation to determine the global frequency for the final contingency table.

Complexity reduction. Building the contingency table C_{ic} requires a single $\mathcal{O}(P)$ scan of the pixels; each visit increments the counter $(i, c) \mapsto C_{ic} + 1$. After that pass, any additive loss or metric of the form (11) touches only the S non-zero overlaps, plus the N region marginals. However, the sparsity of S is contingent on how well I matches G. Given that $N < S < NK$, the worst case yields

$$\begin{aligned} \mathcal{O}(P + S + NK) &= \mathcal{O}(P + 2NK) \\ &= \mathcal{O}(P + NK) \end{aligned} \tag{12}$$

Thus an ELUT incurs $\mathcal{O}(P + NK) = \mathcal{O}(P + \alpha PK)$ work per image, whereas a dense raster evaluates the same objective in $\mathcal{O}(PK)$. As with the space complexity, for $\alpha = 1$ (no compression) a dense representation outperforms ELUTs but since $\mathcal{O}(P + PK) = \mathcal{O}(P(K + 1)) = \mathcal{O}(PK)$, the representations technically have the same complexity. However, for the expected case $\alpha \ll 1$ we see significant reduction in compute for ELUT representations, tied to the reduction factor α. In a practical modeling setting, the sparsity of $\mathcal{S}$ also plays a role, conditional on how well I aligns with G.

Table 1
Empirical per-image memory footprint over ImageNet (224×224) using an adaptive tokenization framework [30] with ViT-B capacity ($D = 768$, `float32`). The dense tensor requires $M_{\text{dense}} = 4DP$ (147MB), while the ELUT cost is $M_{\text{ELUT}} = 8P + 4DN$ with `int64` indices. ρ denotes the reduction factor $M_{\text{dense}}/M_{\text{ELUT}}$.

Steps t	0	1	2	3	4
Num.reg. N	50 176	11 940	3 156	795	197
Mem.[MiB] M	147.4	35.2	9.4	2.5	0.8
Red.fac. ρ	0.99×	4.16×	15.27×	54.21×	158.13×

Table 2
Peak GPU memory (GB) and throughput (images/s) for a forward–back-prop pass of cross-entropy (224×224, `float32`). Measurements are on a single AMD MI250X using $t = 4$ tokenization steps.

	Dense		ELUT	
Batch Size	Mem. [GiB] ↓	img/ms ↑	Mem. [GiB] ↓	img/ms ↑
32	4.62	5.32	0.03	3.18
64	9.21	5.28	0.06	3.42
128	18.58	5.22	0.13	3.78
256	36.84	5.23	0.25	4.05

4. Experimental Results

Our experiments are designed to demonstrate feasibility for ELUT representations. In Sec. 4.1 we perform an empirical study on complexity, showing effective reduction in memory and computational costs, verifying derivations from Sections 3.2 and 3.3. We then demonstrate the effectiveness of hierarchical image representations in a simple lossless compression scheme.

4.1. Empirical Complexity Analysis

We compute empirical estimates of expected region counts over ImageNet [50] using an adaptive tokenization framework with $t \in \{1, 2, 3, 4\}$ merge iterations [30]. Table 1 show that even at conservative levels ($t = 1$), ELUTs quarter the memory footprint, while deeper merges further compress the feature map without altering pixel-wise losses or metrics – cf. Sec. 3.3.

Next, we compute memory overhead and wall-clock throughput for variable sized batches for images sampled from COCO [51]. We compare a dense feature representation extracted via upscaling [52] with $D = 768$ and $H = W = 224$ to an ELUT representation. Features are extracted via the same backbone over the same batch of images, so the only difference is their representation.

We compute equivalent formulations of cross-entropy for dense and ELUT representations, and report results in Tab. 2. While our results are not able to verify that ELUTs provide any significant empirical computational benefits in terms of throughput, the memory overhead is still significant. The loss is evaluated by gathering the N region logits from one contiguous lookup table and weighting them with the S sparse overlaps, so the only overhead comes from scatter/gather operations.

4.2. Information Content and Compression

For a data-structure to be effective, it should ideally result in low entropy representations. Empirically validating the informational content of general ELUTs is non-trivial, as the representation is highly contingent on the choice of tokenizer and features, which are typically task-dependent.

We construct a representation independent of feature representations by using a superpixel tokenizer and compress images using a hierarchical pyramid representation. Each pixel is associated with a parent region using SPiT tokenizer [30] to form a rooted tree, i.e., we aggregate regions until only a single node remains. We reuse the very same SPiT partitions that the model computes during training. We

Table 3
Compression results for individual test images and Kodak-24 [53] compared to lossless PNG compression. MiB is the size of the compressed image, CR denotes the compression ratio, and BPP denotes the bits-per-pixel.

	Pepper			Lena			Barbara			Baboon			Kodak-24† [53]		
Method	MiB ↓	CR ↑	BPP ↓	MiB ↓	CR ↑	BPP ↓	MiB ↓	CR ↑	BPP ↓	MiB ↓	CR ↑	BPP ↓	MiB ↓	CR ↑	BPP ↓
PNG	0.586	1.28	6.26	0.245	3.06	2.57	0.627	1.20	6.68	0.629	1.19	6.71	0.420	2.25	4.49
Ours	0.548	1.37	5.85	0.241	3.11	2.61	0.491	1.53	5.23	0.614	1.22	6.54	0.405	1.88	4.32

†: Average over full dataset.

then construct a graph Laplacian pyramid by representing all non-root RGB embeddings mapped to YCbCr with a difference transform over the depth of the tree. Since a region is typically similar to its parent this provides a sparse representation, particularly in leaf nodes representing individual pixels.

To encode the overall structure, we extract the Prüfer sequence of the graph [54] along with the ELUT. In conjunction with a Huffman encoding scheme, this yields a basic lossless compression format, useful for evaluating the informational capacity of ELUT representations. We measure the information density by evaluating the compression ratio (CR), bits-per-pixel (BPP) and contrast it with standard image formats, and report them in Tab. 3. Surprisingly, this simple strategy yields better compression than PNG, the industry standard. We note that while this is by no means a state-of-the-art approach, it shows that hierarchical ELUTs provides highly effective image representations.

5. Discussion and Conclusion

In this work, we introduce *Embedded Lookup Tables* as extensions of well-established representations in computer graphics to machine learning settings. We show that by factorising a $D \times H \times W$ map into an integer index image I and a compact table $T \in \mathbb{R}^{N \times D}$, ELUTs replace the $\mathcal{O}(DHW)$ memory wall with $\mathcal{O}(DN + HW)$. Our experiments provide empirical evidence that show how ELUTs provide an effective alternative in practical modeling tasks requiring pixel level granularity. Since objectives and metrics can be rewritten using contingency counts, arithmetic shrinks roughly in proportion to the factor $\alpha = N/P$.

While our results are not contingent on a single tokenization framework, future work would incorporate more extensive studies on the effect on training in dense prediction tasks and multiple tokenization framework. While our scope is limited to 2D imaging in this work, our theoretical results naturally extends to higher dimensional imaging. Extensions to volumetric data, light-field stacks, and video frames should benefit even further as the dense baseline grows with an extra dimension while the index map stays tractable.

Acknowledgments

Computations were performed on resources provided by Sigma2 (NRIS, Norway), Project NN8104K. We acknowledge Sigma2 for awarding access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CSC (Finland) and the LUMI consortium through Sigma2, Norway, Project no. 465001382. This work was funded in part by the Research Council of Norway, via the Visual Intelligence Centre for Research-based Innovation (grant no. 309439), and Consortium Partners.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-4o and Writefull for:[1] *Grammar and spelling checks*, and *Formatting assistance*. After using these services, the authors reviewed and edited the content as needed and takes full responsibility for the publication's content.

[1]Following the taxonomy from the CEUR-WS Policy

References

[1] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W. M. W. III, A. F. Frangi (Eds.), Inter. Conf. Med. Imag. Comput.-Assist. Interv. (MICCAI), volume 9351 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 234–241. doi:10.1007/978-3-319-24574-4_28.

[2] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, B. Xiao, Deep high-resolution representation learning for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 43 (2021) 3349–3364. doi:10.1109/TPAMI.2020.2983686.

[3] T. M. Cover, Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition, IEEE Transactions on Electronic Computers EC-14 (1965) 326–334.

[4] S. Shalev-Shwartz, S. Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014.

[5] W. Hardle, L. Simar, Applied Multivariate Statistical Analysis, Springer Berlin Heidelberg, 2007.

[6] D. J. Field, Relations between the statistics of natural images and the response properties of cortical cells, Journal of the Optical Society of America A 4 (1987) 2379–2394. doi:10.1364/JOSAA.4.002379.

[7] D. Kersten, Predictability and redundancy of natural images, Journal of the Optical Society of America A 4 (1987) 2395–2400. doi:10.1364/JOSAA.4.002395.

[8] E. P. Simoncelli, B. A. Olshausen, Natural image statistics and neural representation, Annual Review of Neuroscience 24 (2001) 1193–1216. doi:10.1146/annurev.neuro.24.1.1193.

[9] E. Seeram, Advances in imaging—the changing environment for the imaging technologist, Radiologic Technology 82 (2011) 417–438. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3076980/.

[10] F. Abramovich, M. Pensky, Classification with many classes: Challenges and pluses, J. Multivar. Anal. 174 (2019) 104536. URL: https://www.sciencedirect.com/science/article/pii/S0047259X19302763.

[11] Z. Liu, J. Zhang, W. Wu, Y. Wang, X. Zhang, Principled approach to the selection of the embedding dimension of networks, Nature Communications 12 (2021) 1–9. doi:10.1038/s41467-021-23795-5.

[12] J. T. Kajiya, I. E. Southerland, E. C. Cheadle, A random-access video frame buffer, Association for Computing Machinery, New York, NY, USA, 1998, p. 315–320. doi:10.1145/280811.281022.

[13] P. Heckbert, Color image quantization for frame buffer display, in: ACM Conf. Spec. Inter. Group Graphics Interact. Techn. (SIGGRAPH), Association for Computing Machinery, New York, NY, USA, 1982, p. 297–307. doi:10.1145/800064.801294.

[14] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. HAZIZA, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, P. Bojanowski, DINOv2: Learning robust visual features without supervision, Trans. Mach. Learn. Res. (2024). URL: https://openreview.net/forum?id=a68SUt6zFt.

[15] M. F. Naeem, Y. Xian, X. Zhai, L. Hoyer, L. Van Gool, F. Tombari, SILC: Improving vision language pretraining with self-distillation, in: European Conf. Comput. Vis. (ECCV), 2024. doi:10.1007/978-3-031-72664-4_3.

[16] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, R. Girshick, Segment anything, in: IEEE Inter. Conf. Comput. Vis. (ICCV), 2023. URL: https://openaccess.thecvf.com/content/ICCV2023/papers/Kirillov_Segment_Anything_ICCV_2023_paper.pdf.

[17] J. Selan, High-quality real-time rendering with multi-dimensional lookup tables, in: GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation, Addison-Wesley Professional, 2005.

[18] CompuServe Incorporated, Graphics interchange format (gif) specification, version 89a, Online, 1989. URL: https://www.w3.org/Graphics/GIF/spec-gif89a.txt.

[19] Adobe Systems Inc., Tiff revision 6.0 specification, Online, 1992. URL: http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf, originally published by Aldus Corporation as TIIF 6.0.

[20] X. Ren, J. Malik, Learning a classification model for segmentation, in: IEEE Inter. Conf. Comput. Vis. (ICCV), 2003, pp. 10–17 vol.1. doi:10.1109/ICCV.2003.1238308.

[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: Inter. Conf. Learn. Represent. (ICLR), 2021.

[22] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. S. Torr, L. Zhang, Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers, in: IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR), 2021.

[23] R. Ranftl, A. Bochkovskiy, V. Koltun, Vision transformers for dense prediction, in: IEEE Inter. Conf. Comput. Vis. (ICCV), 2021.

[24] F. Wang, Y. Yu, G. Wei, W. Shao, Y. Zhou, A. Yuille, C. Xie, Scaling laws in patchification: An image is worth 50,176 tokens and more, 2025. URL: https://arxiv.org/abs/2502.03738, arXiv preprint arXiv:2502.03738.

[25] R. Mojtahedi, M. Hamghalam, R. K. G. Do, A. L. Simpson, Towards optimal patch size in vision transformers for tumor segmentation, in: International Workshop on Multiscale Multimodal Medical Imaging (MMMI 2022), 2023. doi:10.1007/978-3-031-18814-5_11.

[26] K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlós, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, D. Belanger, L. Colwell, A. Weller, Rethinking attention with performers, in: Inter. Conf. Learn. Represent. (ICLR), 2021. ArXiv preprint arXiv:2009.14794.

[27] Y. Han, Z. Xu, X. Shang, L. Wang, Y. Yang, S. Li, X. Liu, B. Wu, J. Bai, et al., FLatten transformer: Vision transformer using focused linear attention, in: IEEE Inter. Conf. Comput. Vis. (ICCV), 2023.

[28] J. D. Havtorn, A. Royer, T. Blankevoort, B. E. Bejnordi, MSViT: Dynamic mixed-scale tokenization for vision transformers, in: IEEE Inter. Conf. Comput. Vis. Wksps. (ICCVW), 2023, pp. 838–848.

[29] T. Ronen, O. Levy, A. Golbert, Vision Transformers with Mixed-Resolution Tokenization , in: IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR), IEEE Computer Society, 2023. doi:10.1109/CVPRW59228.2023.00486.

[30] M. Aasan, O. Kolbjørnsen, A. Schistad Solberg, A. Ramírez Rivera, A spitting image: Modular superpixel tokenization in vision transformers, in: European Conf. Comput. Vis. Wksps. (ECCVW), 2024.

[31] J. Lew, S. Jang, J. Lee, S.-K. Yoo, E. Kim, S. Lee, J.-Y. C. J.-H. M. Y.-I. Mok, S. Kim, S. Yoon, Superpixel tokenization for vision transformers: Preserving semantic integrity in visual tokens, ArXiv abs/2412.04680 (2024). URL: https://api.semanticscholar.org/CorpusID:274581264.

[32] D. Chen, S. Cahyawijaya, J. Liu, B. Wang, P. Fung, Subobject-level image tokenization, ArXiv abs/2402.14327 (2024). URL: https://api.semanticscholar.org/CorpusID:267782983.

[33] V. Jampani, D. Sun, M.-Y. Liu, M.-H. Yang, J. Kautz, Superpixel samping networks, in: European Conf. Comput. Vis. (ECCV), 2018.

[34] T. N. Kipf, M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, in: Inter. Conf. Learn. Represent. (ICLR), 2017. URL: https://openreview.net/forum?id=SJU4ayYgl.

[35] K. Xu, W. Hu, J. Leskovec, S. Jegelka, How powerful are graph neural networks?, in: Inter. Conf. Learn. Represent. (ICLR), 2019. URL: https://openreview.net/forum?id=ryGs6iA5Km.

[36] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: Proceedings of the International Conference on Learning Representations, 2018. URL: https://openreview.net/forum?id=rJXMpikCZ.

[37] C. Playout, Z. Legault, R. Duval, M. C. Boucher, F. Cheriet, A Region-Based Approach to Diabetic Retinopathy Classification with Superpixel Tokenization , in: Inter. Conf. Med. Imag. Comput.-Assist. Interv. (MICCAI), volume LNCS 15005, Springer Nature Switzerland, 2024.

[38] Y. Lee, J. H. Park, S. Oh, K. Shin, J. Sun, M. Jung, C. Lee, H. Kim, J.-H. Chung, K. C. Moon, et al., Derivation of prognostic contextual histopathological features from whole-slide images of tumours via graph deep learning, Nature Biomedical Engineering (2022) 1–15.

[39] V. P. Dwivedi, L. Rampášek, M. Galkin, A. Parviz, G. Wolf, A. T. Luu, D. Beaini, Long range graph

benchmark, in: Adv. Neural Inf. Process. Sys. (NeurIPS), 2022. URL: https://openreview.net/forum?id=in7XC5RcjEn.

[40] M. Fey, J. E. Lenssen, Fast graph representation learning with pytorch geometric, in: Inter. Conf. Learn. Represent. Wksps. (ICLRW), New Orleans, USA, 2019. URL: https://arxiv.org/abs/1903.02428.

[41] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk, SLIC superpixels compared to state-of-the-art superpixel methods, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2012) 2274–2282. doi:10.1109/TPAMI.2012.120.

[42] L. Najman, M. Schmitt, Watershed of a Continuous Function, Signal Process. (1994). URL: https://hal.science/hal-00622129. doi:10.1016/0165-1684(94)90059-0.

[43] X. Wei, Q. Yang, Y. Gong, N. Ahuja, M. Yang, Superpixel hierarchy, IEEE Trans. Image Process. (2018). doi:10.1109/TIP.2018.2836300.

[44] H. Lieberman, How to color in a coloring book, ACM Conf. Spec. Inter. Group Graphics Interact. Techn. (SIGGRAPH) 12 (1978). doi:10.1145/965139.807380.

[45] U. Shani, Filling regions in binary raster images: A graph-theoretic approach, ACM Conf. Spec. Inter. Group Graphics Interact. Techn. (SIGGRAPH) 14 (1980).

[46] J. Cousty, G. Bertrand, L. Najman, M. Couprie, Watershed cuts: Minimum spanning forests and the drop of water principle, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2009) 1362–1374. doi:10.1109/TPAMI.2008.173.

[47] J. Stolfi, R. de Alencar Lotufo, A. X. Falc?, The Image Foresting Transform: Theory, Algorithms, and Applications , IEEE Trans. Pattern Anal. Mach. Intell. 26 (2004) 19–29. doi:10.1109/TPAMI.2004.10012.

[48] C. Couprie, L. Grady, L. Najman, H. Talbot, Power watershed: A unifying graph-based optimization framework, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2011) 1384–1399. doi:10.1109/TPAMI.2010.200.

[49] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: an imperative style, high-performance deep learning library, Adv. Neural Inf. Process. Sys. (NeurIPS) (2019).

[50] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR), IEEE, 2009, pp. 248–255.

[51] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: common objects in context, in: D. J. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), European Conf. Comput. Vis. (ECCV), 2014. doi:10.1007/978-3-319-10602-1_48.

[52] S. Fu, M. Hamilton, L. E. Brandt, A. Feldmann, Z. Zhang, W. T. Freeman, Featup: A model-agnostic framework for features at any resolution, in: Inter. Conf. Learn. Represent. (ICLR), 2024. URL: https://openreview.net/forum?id=GkJiNn2QDF.

[53] R. Franzen, Kodak lossless true color image suite, 2012. URL: https://r0k.us/graphics/kodak/.

[54] S. Caminiti, I. Finocchi, R. Petreschi, On coding labeled trees, Theoretical Computer Science 382 (2007) 97–108.