Upsampling DINOv2 features for unsupervised vision tasks and weakly supervised materials segmentation

Ronan Docherty
Department of Materials
Department of Design Engineering
Imperial College London
ronan.docherty18@imperial.ac.uk

Antonis Vamvakeros
Department of Design Engineering
Imperial College London
a.vamvakeros@imperial.ac.uk

Samuel J. Cooper
Department of Design Engineering
Imperial College London
samuel.cooper@imperial.ac.uk

Abstract

The features of self-supervised vision transformers (ViTs) contain strong semantic and positional information relevant to downstream tasks like object localization and segmentation. Recent works combine these features with traditional methods like clustering, graph partitioning or region correlations to achieve impressive baselines without finetuning or training additional networks. We leverage upsampled features from ViT networks (*e.g* DINOv2) in two workflows: in a clustering based approach for object localization and segmentation, and paired with standard classifiers in weakly supervised materials segmentation. Both show strong performance on benchmarks, especially in weakly supervised segmentation where the ViT features capture complex relationships inaccessible to classical approaches. We expect the flexibility and generalizability of these features will both speed up and strengthen materials characterization, from segmentation to property-prediction.

1 Introduction

The rise of 'foundation models' - large networks (usually transformers[1]) trained on vast corpuses of data - has been a strong theme in machine learning for the past five years[2]. Initially these were language models, such as BERT[3], T5[4] or the GPT[5,6] series, but has also included some generative vision models like CLIP[7], DALL-E[8] or Stable Diffusion[9]. Foundation models targeting fundamental vision problems like segmentation and object detection have also been developed, notably Meta's Segment Anything Model (SAM)[10] and the YOLO series[11,12] that display strong zero- or few-shot performance. The analysis of micrographs is ubiquitous in scientific workflows, for example in the characterization of crystals in material science or cells in biology. Segmentation (assigning every pixel/voxel a class) is prerequisite before any such analysis can take place, be it phase quantification, defect detection or transport simulations. Finding ways to use these models to improve materials segmentation is therefore of great interest.

One class of vision foundation model that has seen progress recently is the 'feature foundation model', designed to learn "all-purpose visual features"[13]. These features can then be used in downstream tasks, usually by freezing the foundation model and training a small 'head' network to map from the image features to the specific objective. Examples include DINO[14], DINOv2[13] and I-JEPA[15]. Like other foundation models, they are predominantly trained via self-supervised learning. ViTs trained in

38th Conference on Neural Information Processing Systems - Ai4Mat Workshop (NeurIPS 2024).

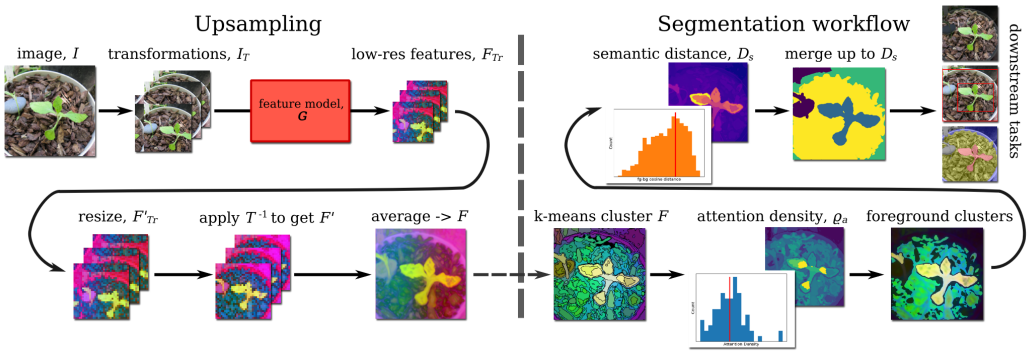

Figure 1: Diagram of our feature upsampling method (left) and its application in unsupervised downstream vision tasks like object detection, saliency detection and semantic segmentation.

this way capture rich semantic information in their features that convolutional neural networks do not[14,16].

DINO and DINOv2 produce spatialised features, albeit at a resolution limited by the model's patch size. These semantically meaningful patch-level features can then be combined with classical techniques (clustering, spectral methods *etc.*) in workflows that aim to leverage these features in an unsupervised manner, *i.e,* without further training[17]. Example tasks that can be achieved using these features unsupervised include (multi-)object detection[18–20], saliency detection[20,21] and (semantic) segmentation[16,21]. These workflows require no labelled training data (unlike, say, training a head network) and are therefore attractive in domains where labelled training data is scarce and expensive, like biological, medical or materials imaging[22].

In such domains, dense (pixel-level) features are often desirable for tasks like semantic segmentation or localised property prediction. For example, in materials science, zero-shot semantic segmentation is often achieved by training a classifier (usually a Random Forest) to map from classical image features like average local intensity, edge strength and textures to user-drawn labels[23–26]. The semantic information offered by ViTs could improve tasks like segmentation where classical image features fail, but are currently limited by being patch- rather than pixel-level. Work on improving the feature resolution of these models has been done, including reducing the stride of the convolution in the model's patch embedding layer[16], training a single forward-pass upsampling filter or fitting an implicit model for each image[27].

In this work we present three main contributions - the first is a novel single-pass method for feature upsampling that is model agnostic and works without any further training. It works by shifting the input image a certain number of pixels (less than the patch size) in each direction, computing the features with a foundation model, resizing the features back to the original image size, shifting in the opposite direction and averaging. This method is compatible with the strided approach and works for other invertible transformations like flipping, rotating *etc*.

The second is an unsupervised segmentation workflow based on clustering these high resolution feature maps, where the attention density can be used to create a robust foreground/background distinction, and from that a 'semantic distance' between classes in the image can be estimated. Clusters are agglomeratively merged up to this distance, producing a 'semantic segmentation' relative to the image (and model). This segmentation can then be processed for object detection, saliency detection or semantic segmentation. A diagram of the feature upsampling and its use in the unsupervised workflow is presented in Figure 1.

Finally, we integrate these high resolution features into a zero-shot weakly supervised semantic segmentation app and qualitatively demonstrate its ability to handle both pixel-level and semantic distinctions when applied to a case study of cell nuclei segmentation and a series of industrially-relevant materials - namely, battery cathodes, alloys, oxide layers and organic crystals.

2 Background

Vision transformers

Vision Transformers extend the token sequence-modelling of language transformers to images by breaking an image into (non-overlapping) embedded patches with a convolutional layer with given kernel size (commonly 8, 16 or 32 pixels) and stride equal to kernel size[28]. These patches are given a positional encoding relative to their position in the image (*i.e,* raster-order[28] or sinusoidal[29]) and then treated as tokens in the transformer as normal. It is common to append a non-spatial class token (called the `[CLS]` token) to the sequence to capture global information.

The number of patch tokens, n, increases quadratically as the edge length of the square image increases, or patch size decreases. Given the attention mechanism of transformers is $O(n^2)$[1] this represents a sharp quartic scaling to achieve higher resolution. It is therefore common to train these networks at a large patch size, say 16 pixels, which limits the resolution of extracted features - not usually a problem for natural images but has implications when handling micrographs of materials, which often have pixel-level features.

Self-supervised feature learning

Self-supervised representation learning with transformers on large datasets has proven both popular and successful in Natural Language Processing (NLP), learning by mapping between transformations of an input text sequence - like causal token masking for GPT[5] or single-token masking for BERT[3].

A direct analogue to this approach for ViTs is the Masked Autoencoder (MAE), which learns by reconstructing an image with around 70% of the input patches masked; it works in different modalities and as part of wider feature learning setups[15,30–33]. Other approaches view self-supervised learning as a self-distillation problem, employing a student-teacher framework to learn patch-level features from either varying local/global view of an image, or distillation token or in combination with masked token modelling. Examples of these approaches are DINO[14], DEiT[34] and iBot[35]. The most recent feature foundation model is DINOv2[13], which showed strong performance across various benchmarks.

It has been shown features of supervised models are more discriminative (and therefore less flexible) than those of self-supervised models, and that self-supervised ViTs learn semantic information that comparable CNNs do not[14,34,36]. Different self-supervised training schemes result in different features: the features of MAE-trained ViTs are better at separating instances of the same object whereas DINO-trained ViTs are better at semantic separation[37].

These feature models tend to use the same ViT architecture, varying the training data and objective. The ViT architecture tends to come in varying sizes: (S)mall, (B)ase, (L)arge, (H)uge and (G)iant, each with increasing parameter counts and hidden dimension output. They also have varying patch size, usually 16 but sometimes 14 or 8. We will use the notation `model-size-patch`, *i.e* DINO-S-16, in the following sections.

Feature upsampling

Standard interpolation approaches (bilinear, bicubic, nearest-neighbour) can be used to upsample patch-level features but risks blurring or missing high-resolution details. For ViTs, reducing the stride of the patch embedding layer (creating overlapping patches)[16] can increase the feature resolution, but this increases the number of input patches and therefore causes the time and memory scaling problem discussed in Section S5. It can also cause blurring and at very small strides (1 or 2) cause numerical errors.

FeatUp[27] presented two approaches for general feature upsampling: a learned Joint Bilateral Upsampling (JBU)[38] filter which operates in a single forward pass and an implict model that is fitted to each image. A feature upsampler and downsampler are trained simultaneously to predict the change in features after small transformations of the input image. The JBU filter is fast, though sometimes produces blurry feature maps and the implicit approach produces very sharp features but requires training for each image.

Unsupervised feature adaption: object localization and segmentation

Various methods have been proposed for applying unsupervised image features to different tasks - most leveraging DINO as their featuriser[17]. LOST[19] detects objects by finding patches correlated to a given 'seed' patch. MOST[18] performs entropy-based box analysis on the patch feature correlations to detect foreground objects.

Other approaches use clustering, usually to cluster features into classes (and later superclasses) for semantic segmentation[16,39]. Spectral methods have also been used, either decomposing images via eigenvectors of the Laplacian of the feature affinity matrix[21], or graph/spectral clustering of the features[37,40], or a graph-cut approach[20,41]. The memory cost of these methods scales prohibitively as the number of tokens (and therefore graph size) increases.

Each of these applications needs some heuristic for separating foreground and background classes, be it thresholding attention[14], number of pixels touching the edges of the image[37] or assuming the largest class to be the background[19]. They also must contend with the limited resolution of the patch features, improving them by changing the model stride[16], fusing additional colour information[21], using a Bilateral Solver (BS)[41] or Conditional Random Field (CRF)[16] or training a model on their workflow's pseudo-labels ('self-training').

Weakly supervised segmentation for materials

Weakly- or semi-supervised segmentation in computer science is a wide field with a variety of models, including CNNs, generative adversarial networks (GANs) and transformers[42–45]. Materials science is a field that spans a range of materials and length scales, and that often has limited access to computational hardware or large quantities of data. As such, cheap methods that generalise well on new data are preferable. Tools like Weka[23], ilastik[24] and napari-apoc[25] train a random forest to map from classical image features (Gaussian, Sobel, Hessian, Difference of Gaussians, Laplacian, *etc.*, filters) to user-drawn labels. These features are computed for every pixel over a range of scales, so are full resolution, though are limited in the complexity of the relationships they can express (relative to a neural network).

As the featurisation only happens once and it is quick to train a random forest, users can add new labels in an active learning style[46], correcting wrong or uncertain outputs to improve the segmentation[23,24]. Once the classifier is trained, it can be applied to new examples without the user needing to add labels (*i.e,* in an automated analysis workflow).

3 Method

3.1 Increasing feature resolution

Our method for upsampling features is simple. Starting with a set of invertible transformations, $T = \{t_1, t_2, ...\}$, an input image $I \in \mathbb{R}^{h \times w \times 3}$, we compute the set $I_T = \{t(I) \mid t \in T\}$. We feed this set (sequentially or as a batch) into our model $G(\boldsymbol{x}; \boldsymbol{\theta})$, which takes input $\boldsymbol{x}$ and has (frozen) weights $\boldsymbol{\theta}$. It has a patch size P and hidden dimension D so produces a set of features $F_T = \{G(i) \mid i \in I_T\}$.

Each $f \in F_T$ has dimensions $f \in \mathbb{R}^{(h/P) \times (w/P) \times D}$ and so we nearest-neighbour resize each f back to the original image dimensions to get F'_T with elements $f' \in \mathbb{R}^{h \times w \times D}$. Next we apply each inverse transformation in $T^{-1} = \{t_1^{-1}, t_2^{-1}, ...\}$ to F'_T to get F'. Finally we average over the t in F'_T to get $F \in \mathbb{R}^{h \times w \times D}$, our upsampled features for I.

As a concrete example, consider $T = \{t_1, t_2, ...\}$ to be the set of pixel-shift operators for each direction in a Moore neighbourhood for distances $\{d \mid 0 < d < P/2; d \in \mathbb{Z}\}$. When we compute the features of these shifted inputs, some of the spatial information from pixel p will spill into neighbouring patches, such that when the features are resized and the inverse transformation (*i.e,* the shift in the opposite direction) applied, the information is 'put back in the right place' at pixel p.

This method extends to other transformations, including rotations and flips (useful for averaging away positional information) and arbitrary combinations thereof; this is realised via partial functions. We also extract high-resolution attention maps from the model at the same time in a similar manner. Similar to FeatUp, this approach upsamples the features by gaining new information from querying

the model with transformations of the input - in FeatUp this information is encoded in a network (learned upsampling kernel or implict), in ours it is in a single forward pass.

There are several advantages: this method is model agnostic (assuming they use spatialised features), requires no additional training and works with the strided approach discussed in Section 2. The memory and time cost is linear with the number of transforms T (see Section S5) if batched; for sequential processing the memory cost is constant (relative to the memory and time costs of the strided method). Examples of the feature resolution can be seen in Sections 4.1 and S2. Limitations are discussed in Section S6. Discussion on the impact of these transformations, as well as the number needed for a given stride is available in Section S9.

3.2 Unsupervised segmentation workflow

These high-resolution features can then be used directly (*i.e,* without including additional colour information or self-training networks) in unsupervised downstream tasks. Our approach is to produce a 'Class-Agnostic Segmentation' (CAS)[47] - drawing parallels to class-agnostic object detection - using only the information available in the features, and from that perform tasks like object localization, saliency detection and semantic segmentation. A full diagram of the workflow is shown in Figure 1.

We begin by clustering the high-resolution features into $C = 80$ clusters using k-means clustering. To generate a foreground/background distinction we measure the 'attention density', ρ_A, which is the attention per unit area of the `[CLS]` token in a cluster. Clusters with $\rho_A > \bar{\rho}_A$ are deemed foreground clusters. The use of ρ_A is to create a well-separated distribution over the clusters, where background clusters are large and have low attention, so have a small resulting ρ_A. Not relying on the 'largest class/cluster = background'[19] or 'most border pixels = background'[37] heuristics allows more flexibility on what is counted as foreground. This is especially useful when moving away from natural images towards less centralised, more homogeneous micrographs.

We then measure the cosine distance between each foreground and background cluster taking the modal distance as the 'semantic distance', a measure of inter-class distance in feature space for the image I. That this semantic distance merging will produce a good segmentation assumes that the distance between a class in the foreground and a different one in the background is similar enough to the distance between two different classes in the foreground (*i.e,* a bike vs. the woods and the rider vs. the bike).

Agglomerative merging with complete linkage is then used to merge clusters that have a distance less than the semantic distance, producing a rough CAS map of the image. This can be refined with a CRF[48] to ameliorate the blurring problem. Multiplying the semantic distance by a scalar factor λ allows control of the granularity of the merging - varied between 0.95 and 1.1 during experiments. Combining the CAS map and attention information enables various downstream tasks: like object detection by drawing bounding boxes around unconnected regions in each class or saliency detection via attention density of the CAS map.

Like the feature upsampling, this approach works per-image, for any model, and without extra training. We note that whilst FeatUp could be used for upsampling the features, it does not upsample the model attention maps at the same time, which are needed for the foreground/background separation.

3.3 Weakly supervised segmentation

Following existing work in the materials imaging community which train a random forest classifier[23–26] to map from classical image features to user labels, we train a logistic regression classifier to map from our upsampled pixel-wise features from ViT models to user labels. These user labels are from 'paint-brush' style annotation, (see Figure 3 for an example) so may be sparse. Each D-dimensional vector that describes a (labelled) pixel is used alongside its associated label as a training example for the classifier.

Logisitic regression was chosen for its simplicity and regularizability relative to random forests. We use a random forest classifier for the classical features to ensure similarity to existing schemes[23–26] and because it showed the best performance.

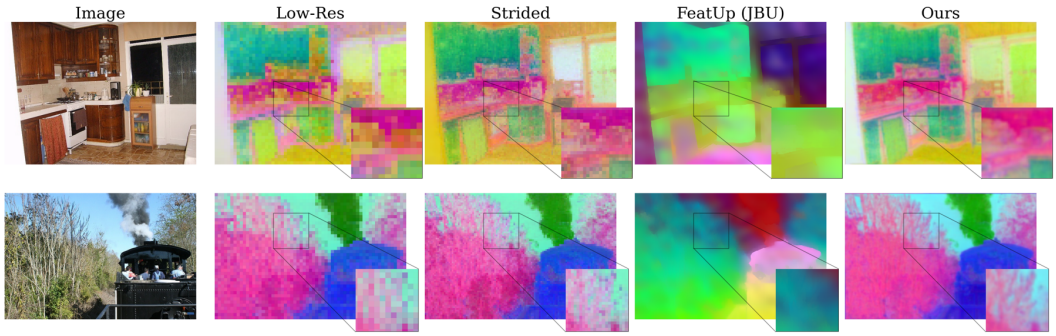

Figure 2: Feature resolution comparisons on two example images from VOC07 for different upsampling methods for DINO-S-8 features. FeatUp (JBU) is able to produce sharp edges for some objects but can blur others or miss fine details *i.e,* the cups in the top image. Our method can capture such details, though the blurring introduced by using strided approach causes softer boundaries. Note the FeatUp featurizer was DINOv2-S-14.

4 Results & discussion

4.1 Qualitative comparisons of features

We compare the 3-component PCA of our upsampled features for DINO-S-8 qualitatively to the original low-resolution features, the result of setting the model stride to 4 and FeatUp in Figure 2, noting an increase in resolution for fine-details and faithfulness to the original features. It should also be noted this approach works alongside the strided approach, and that the improvement from using our approach is relatively modest.

Further comparisons are available in Figure S1. Like FeatUp, our approach works across all models that produce spatialised features, this can be seen in Figure S2. We discuss limitations, including blurring and boundary effects in Figure S7. In each comparison we compare to FeatUp's fast, single-forward pass JBU approach as it is most similar to the aims of our approach - their implicit approach would produce higher-resolution features but would take far longer.

Following FeatUp we perform small-object retrieval[27] in Figure S4 - searching for the most similar point in a target image to a query point in another image where both images have been featurised with our upsampling approach. A good match indicates the features are high-resolution and semantically relevant (*i.e,* they are useful in downstream tasks). We add more keypoints to their image and track their matches, finding good agreement. When comparing to FeatUp (JBU), we find our feature similarities are better localised to the relevant objects (*i.e.*, the traffic cones). A comparison using DINO-S-8 and showing the full target image is available in Figure S5.

4.2 Quantitative feature comparisons via linear probes

To demonstrate the effectiveness of the learned features of DINOv2, a series of linear probes were trained to perform dense tasks like semantic segmentation and depth estimation given bilinear upsamplings of the patch features. Assuming our method produces faithful and useful upsamplings, these linear probes should still be directly applicable to them and produce similar (if not better) results. The results of these linear probes applied to our features for semantic segmentation on VOC2012[49] is reported in Table S1. There is a performance drop from applying the probe to the features produced by our method, this aligns with FeatUp[27]'s findings for the strided method of higher resolution but noisier segmentation maps.

4.3 Unsupervised segmentation experiments

Method	VOC07	VOC12
DINO[14,17]	0.458	0.462
LOST[19]	0.620	0.640
Melas-Kyriazi *et al.*[21]	0.627	0.664
MOST *et al.* (multi)[18]	**0.748**	**0.774**
Ours	0.554	0.572
Ours (multi)	0.718	0.725

Table 1: CorLoc on VOC07 & VOC12 for various unsupervised object detection schemes. 'Multi' refers to methods which produce multiple object bounding box predictions. Other values quoted from[17,18,21] - we chose not to include methods which used a further self-training step.

Method	CUB	DUTS
OneGAN[50]	0.555	-
Voynov *et al.*[51]	0.683	0.498
Melas-Kyriazi *et al.*[21]	0.769	0.514
MOST[18]	-	0.538
Deep Cut[40]	0.777	0.560
simSAM[52]	0.770	0.582
TokenCut[20]	**0.795**	0.624
Ours	0.785	**0.654**

Table 2: IoU of foreground object segmentation across the CUBS & DUTS datasets. Other values quoted from[17,18,21,52]

4.3.1 Object localization

For unsupervised object detection we simply take a bounding box around each connected component of each foreground class in the CAS map discussed in Section 3.2. A class is a foreground class if its attention density is higher than the mean attention density - similar to the definition of foreground clusters. Our approach will occasionally decompose an object into multiple parts (*i.e,* head/body, car/doors, *etc.*) - to ameliorate this we introduce a 'superbox' around the largest connected component of all foreground classes. If this superbox has more than an 80% intersection over union (IoU) with another box, we retain only the other box.

We then apply this approach to VOC07[53] and VOC12[49], standard unsupervised object detection benchmarks, reporting the results in Table 1. The success metric is 'CorLoc' - the percentage of images where at least one of the predicted boxes has a greater than 50% IoU with at least one of the ground truth boxes. We report two results: the CorLoc when only using the superbox (single object detection) and when using all predicted foreground boxes (multi-object detection). Our method shows comparable performance to state-of-the-art in both single- and multi- object detection.

4.3.2 Foreground object segmentation

For foreground object segmentation (also called 'saliency detection') we use the same approach as in Sections 3.2 and 4.3.1, producing a binary segmentation where all the foreground classes are counted as foreground, the rest as background. We apply our approach to two standard benchmark datasets, CUBS[54] and DUTS[55], reporting the results in Table 2. Our approach shows good performance on both datasets, surpassing state-of-the-art on DUTS. The success metric is '(mean) Intersection over Union' (mIoU) - the area of the overlap between the prediction for the foreground class and ground truth label, divided by the union of both areas. For multiple classes (*i.e,* in Section 4.4), this is averaged over the classes.

4.4 Featuriser for zero-shot weakly supervised segmentation

Following Section 3.3, our aim is to use the richer semantics of these ViT models to perform complex semantic segmentation tasks; as an example we chose a dataset of 135 Transmission Electron Microscopy (TEM) micrographs of human T-cells[56] with three classes: background, cell, nucleus. We show an example in Figure 3, where the model using ViT features outperforms the classical method.

Next we trained classifiers on partial labels across a set of six cells that cover the range of variation in the dataset, namely varying exposure, presence of background cells and multiple nuclei per cell. We then apply these trained classifiers to the rest of the (unlabelled) dataset and measure the mIoU for the three classes (background, nucleus, cell) relative to the ground truth annotations, presenting the

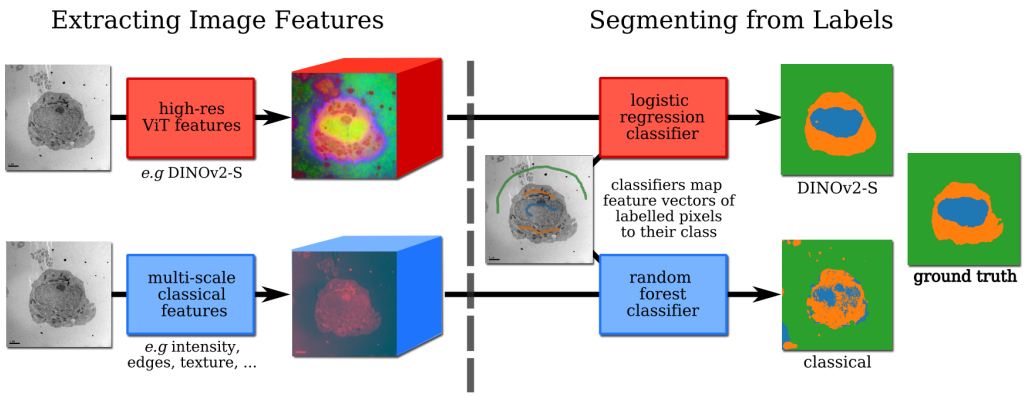

Figure 3: Explanation of the weakly supervised segmentation workflow, using an example of nucleic segmentation of a TEM image of a Jurkat human T-Cell[56]. Our method using high-resolution ViT features is able to capture the semantic information (interiority, foreground vs. background) needed for a good segmentation, compared to the classical image features + random forest approach. This disparity in feature richness can be seen in their respective feature spaces.

Feature-set	**mIoU**	**mIoU** (+CRF)
Classical	0.404	0.439
FeatUp (JBU)	0.795	0.816
Ours (DINO-S-8)	0.776	0.803
Ours (DINOv2-S-14)	0.797	0.827
Ours (Hybrid)	0.809	**0.842**

Table 3: mIoU of the three classes across the T-cell dataset for classifiers trained on the same set of labels with different pixel-features. Classifiers trained with upsampled ViT features (FeatUp, DINO-, hybrid) perform far better than when trained on classical features.

results in Table 3 and some example predictions on unseen data in Figure 4. The stride of the ViT model was set to 4. The labels and cells used to train the classifiers are available in Figure S12.

We note a CRF[48] was used to improve the segmentations for the case study, results without a CRF can be found in Section S8. Also included in Section S8 are more example segmentations on unlabelled cells, both with and without a CRF. Our method using the DINOv2 features shows markedly better performance across the dataset than the standard classical features + random forest approach. We attribute this to the richer feature space allowing concepts like interiority and foreground/background distinction to be expressed.

Despite the success of the CRF in correcting the (somewhat) blurry segmentations from the upsampled features for large phases like the cell or nucleus, we found it tended to remove small tertiary phases (*i.e,* the organelles of the cell). This is a problem in materials science, where small, high frequency features like hairline cracks in battery materials are of great interest.

To ameliorate this problem we experiment with concatenating the classical and DINOv2 features before training the classifier, in order to improve the accuracy on small, complex phases. We call this the "hybrid" approach, and present the results in Figure 5 for three micrographs, again with the same set of labels and noting that a CRF was **not** used in these examples.

The first micrograph is from a high resolution Scanning Electron Microscope (SEM) image of lithium nickel manganese cobalt oxide cathode[57,58] - the ViT features are able to distinguish between in-plane (light grey, flat) and out-of-plane (light grey, textured) active material (AM) present due to the 'pore-back' effect[57,58]. The ViT features also admit the distinction between voids inside an AM particle and the out-of-plane AM (which have similar textures).

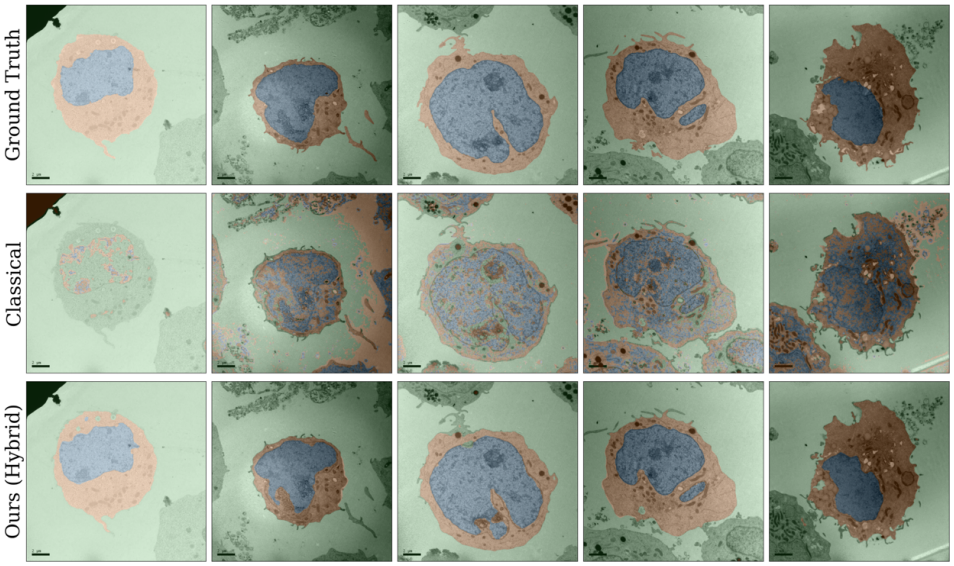

Figure 4: Predictions of the trained classifiers using classical or hybrid features on unlabelled examples from the cell dataset. The hybrid features produce good segmentations, demonstrating their ability to generalise well. Note a CRF was used for the hybrid cells and not for the classical.

The second micrograph is a silicon carbide environmental barrier coating[59], where the goal is to segment the thermally-grown oxide layer in the middle. Similar to the cells example, the classical features struggle to separate the phases, which have similar greyscale intensity. The last micrograph is a Reflected Light Microscope (RFM) image of "cast iron with magnesium induced spheroidised graphite"[60] - the hybrid features can distinguish between the interior graphite and occasionally similar-looking iron. Two further examples of the segmentation of a nickel superalloy micrograph[59] and an X-ray Computed Tomography (XCT) slice of α- and β-polymorphs of glutamic acid are available in Figure S11.

We note that these experiments were performed using the default Weka feature set[23]; choosing a set more suited to the problem may improve performance, though this represents significant trial-and-error. Discussion of the the limitations can be found in Section S6, focusing on the practical implications of using these features in a materials context.

5 Conclusion

To conclude, we have demonstrated a novel single-forward pass upsampling technique for features of vision transformers, like DINOv2. This was then applied to perform unsupervised object detection and segmentation, achieving strong results compared to baselines. One potential use for this unsupervised workflow is generating high-quality semantic region proposals for interactive segmentation datasets[10,61].

Finally, we demonstrated the use of these upsampled features in weakly supervised materials segmentation, where they capture complex relationships that the current classical approaches are unable to express and achieve highly accurate segmentations. We expect that the ability to perform well across different materials, instruments and imaging conditions mean that the use of deep ViT features will greatly improve and expedite automated materials characterization.

There are many avenues for future work: improving the resolution and speed of feature extraction, utilizing the features in novel setups (*i.e,* defect detection/classification) or using them for spatialised property prediction (regression) tasks.

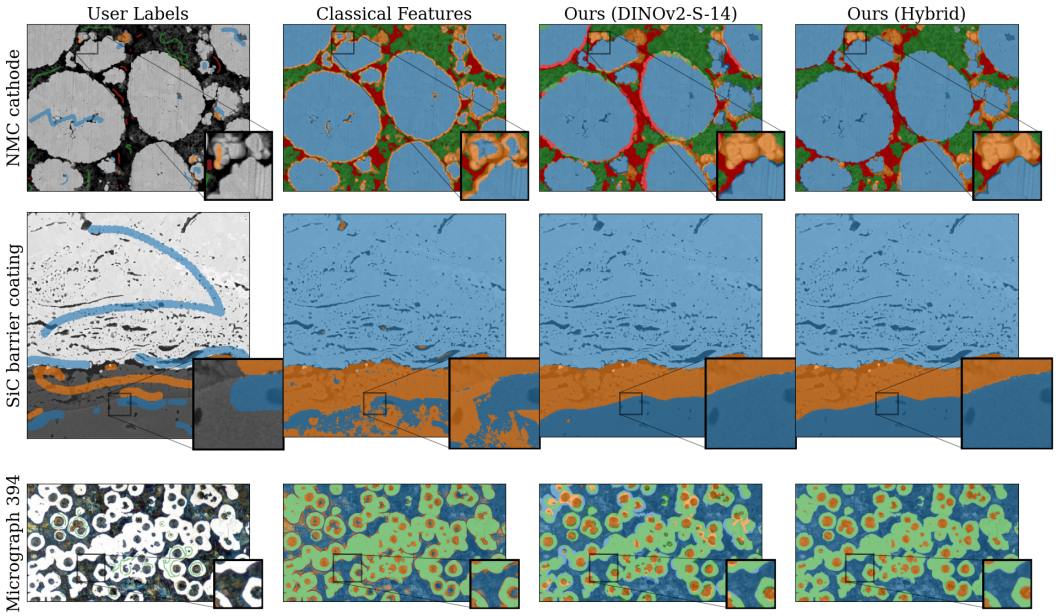

Figure 5: Application of classical, DINOv2 and classical + DINOv2 (hybrid) features for weakly supervised segmentation of materials such as NMC cathodes[57,58], SiC oxide layers[59] and cast iron alloys[60]. The hybrid scheme is able to combine the strong semantic features of DINOv2 with the high-resolution classical features for good segmentations of complex tertiary and quaternary phases.

Acknowledgements

This work was supported by funding from the the EPRSC and SFI Centre for Doctoral Training in Advanced Characterisation of Materials (EP/S023259/1 received by RD) and the Royal Society (IF\R2\222059 received by AV as a Royal Society Industry Fellow).

Code Availability

The code needed to reproduce the results of the paper is available at `https://github.com/tldr-group/HR-Dv2` with an MIT license agreement.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[2] C. Zhou, Q. Li, C. Li, J. Yu, Y. Liu, G. Wang, K. Zhang, C. Ji, Q. Yan, L. He, H. Peng, J. Li, J. Wu, Z. Liu, P. Xie, C. Xiong, J. Pei, P. S. Yu, and L. Sun, "A Comprehensive Survey on Pretrained Foundation Models: A History from BERT to ChatGPT," *arXiv preprint arXiv:2302.09419*, 2023.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2019.

[4] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv preprint arXiv:1910.10683*, 2023.

[5] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[6] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray,

B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language Models are Few-Shot Learners," *arXiv preprint arXiv:2005.14165*, 2020.

[7] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, and J. Clark, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[8] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-Shot Text-to-Image Generation," *arXiv preprint arXiv:2102.12092*, 2021.

[9] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," *arXiv preprint arXiv:2112.10752*, 2022.

[10] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment Anything," *arXiv preprint arXiv:2304.02643*, 2023.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *arXiv preprint arXiv:1506.02640*, 2016.

[12] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction*, vol. 5, pp. 1680–1716, Nov. 2023. Publisher: MDPI AG.

[13] M. Oquab, T. Darcet, T. Moutakanni, H. 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, and P. Bojanowski, "DINOv2: Learning Robust Visual Features without Supervision," *arXiv preprint arXiv: 2304.07193*, 2023.

[14] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging Properties in Self-Supervised Vision Transformers," 2021.

[15] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas, "Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture," *arXiv preprint arXiv: 2301.08243*, 2023.

[16] S. Amir, Y. Gandelsman, S. Bagon, and T. Dekel, "Deep ViT Features as Dense Visual Descriptors," 2022.

[17] O. Siméoni, E. Zablocki, S. Gidaris, G. Puy, and P. Pérez, "Unsupervised Object Localization in the Era of Self-Supervised ViTs: A Survey," *arXiv preprint arXiv: 2310.12904*, 2023.

[18] S. S. Rambhatla, I. Misra, R. Chellappa, and A. Shrivastava, "MOST: Multiple Object localization with Self-supervised Transformers for object discovery," *arXiv preprint arXiv: 2304.05387*, 2023.

[19] O. Siméoni, G. Puy, H. V. Vo, S. Roburin, S. Gidaris, A. Bursuc, P. Pérez, R. Marlet, and J. Ponce, "Localizing Objects with Self-Supervised Transformers and no Labels," *arXiv preprint arXiv: 2109.14279*, 2021.

[20] Y. Wang, X. Shen, S. Hu, Y. Yuan, J. Crowley, and D. Vaufreydaz, "Self-Supervised Transformers for Unsupervised Object Discovery using Normalized Cut," *arXiv preprint arXiv: 2202.11539*, 2022.

[21] L. Melas-Kyriazi, C. Rupprecht, I. Laina, and A. Vedaldi, "Deep Spectral Methods: A Surprisingly Strong Baseline for Unsupervised Semantic Segmentation and Localization," *arXiv preprint arXiv: 2205.07839*, 2022.

[22] A. Goetz, A. R. Durmaz, M. Müller, A. Thomas, D. Britz, P. Kerfriden, and C. Eberl, "Addressing materials' microstructure diversity using transfer learning," *npj Computational Materials*, vol. 8, p. 27, Feb. 2022.

[23] I. Arganda-Carreras, V. Kaynig, C. Rueden, K. W. Eliceiri, J. Schindelin, A. Cardona, and H. Sebastian Seung, "Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification," *Bioinformatics*, vol. 33, pp. 2424–2426, Aug. 2017.

[24] S. Berg, D. Kutra, T. Kroeger, C. N. Straehle, B. X. Kausler, C. Haubold, M. Schiegg, J. Ales, T. Beier, M. Rudy, K. Eren, J. I. Cervantes, B. Xu, F. Beuttenmueller, A. Wolny, C. Zhang, U. Koethe, F. A. Hamprecht, and A. Kreshuk, "ilastik: interactive machine learning for (bio)image analysis," *Nature Methods*, Sept. 2019.

[25] R. Haase, D. Lee, D. D. Pop, and L. Žigutytė, "haesleinhuepf/napari-accelerated-pixel-and-object-classification: 0.14.1," Nov. 2023.

[26] J. Luethi and M. Hess, "napari-feature-classifier: An interactive classifier plugin to use with label images and feature measurements."

[27] S. Fu, M. Hamilton, L. Brandt, A. Feldman, Z. Zhang, and W. T. Freeman, "FeatUp: A Model-Agnostic Framework for Features at Any Resolution," *arXiv preprint arXiv: 2403.10516*, 2024.
[28] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv preprint arXiv: 2010.11929*, 2021.
[29] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng, "Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains," *arXiv preprint arXiv: 2006.10739*, 2020.
[30] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked Autoencoders Are Scalable Vision Learners," *arXiv preprint arXiv: 2111.06377*, 2021.
[31] P.-Y. Huang, H. Xu, J. Li, A. Baevski, M. Auli, W. Galuba, F. Metze, and C. Feichtenhofer, "Masked autoencoders that listen," *arXiv preprint arXiv: 2207.06405*, 2023.
[32] Z. Tong, Y. Song, J. Wang, and L. Wang, "Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training," *arXiv preprint arXiv: 2203.12602*, 2022.
[33] A. Bardes, Q. Garrido, I. Misra, J. Ponce, X. Chen, M. Rabbat, Y. LeCun, M. Assran, and N. Ballas, "Revisiting feature prediction for learning visual representations from video," *preprint*, 2024.
[34] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," *arXiv preprint arXiv: 2012.12877*, 2021.
[35] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, and T. Kong, "iBOT: Image BERT Pre-Training with Online Tokenizer," *arXiv preprint arXiv: 2111.07832*, 2022.
[36] S. Paul and P.-Y. Chen, "Vision transformers are robust learners," *arXiv preprint arXiv:2105.07581*, 2021.
[37] P. Engstler, L. Melas-Kyriazi, C. Rupprecht, and I. Laina, "Understanding Self-Supervised Features for Learning Unsupervised Instance Segmentation," *arXiv preprint arXiv: 2311.14665*, 2023.
[38] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," *ACM Trans. Graph.*, vol. 26, p. 96–es, jul 2007.
[39] Y. S. J. Cheung, X. Chen, L. Yang, and H. Zhao, "A Lightweight Clustering Framework for Unsupervised Semantic Segmentation," *arXiv preprint arXiv: 2311.18628*, 2023.
[40] A. Aflalo, S. Bagon, T. Kashti, and Y. Eldar, "Deepcut: Unsupervised segmentation using graph neural networks clustering," *arXiv preprint arXiv: 2212.05853*, 2023.
[41] Y. Wang, X. Shen, Y. Yuan, Y. Du, M. Li, S. X. Hu, J. L. Crowley, and D. Vaufreydaz, "Tokencut: Segmenting objects in images and videos with self-supervised transformer and normalized cut," *arXiv preprint arXiv: 2209.00383*, 2023.
[42] X. Chen, Y. S. J. Cheung, S.-N. Lim, and H. Zhao, "Scribbleseg: Scribble-based interactive image segmentation," *arXiv preprint arXiv: 2303.11320*, 2023.
[43] N. Souly, C. Spampinato, and M. Shah, "Semi and weakly supervised semantic segmentation using generative adversarial network," *arXiv preprint arXiv: 1703.09695*, 2017.
[44] L. Ru, Y. Zhan, B. Yu, and B. Du, "Learning affinity from attention: End-to-end weakly-supervised semantic segmentation with transformers," *arXiv preprint arXiv: 2203.02664*, 2022.
[45] L. Chan, M. S. Hosseini, and K. N. Plataniotis, "A comprehensive analysis of weakly-supervised semantic segmentation in different image domains," *International Journal of Computer Vision*, vol. 129, p. 361–384, Sept. 2020.
[46] B. Settles, "Active Learning Literature Survey," 2010.
[47] C. Zhang, G. Lin, F. Liu, R. Yao, and C. Shen, "Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning," *arXiv preprint arXiv: 1903.02351*, 2019.
[48] P. Krähenbühl and V. Koltun, "Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials," *arXiv preprint arXiv: 1210.5644*, 2012.
[49] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results." http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.
[50] Y. Benny and L. Wolf, *OneGAN: Simultaneous Unsupervised Learning of Conditional Image Generation, Foreground Segmentation, and Fine-Grained Clustering*, p. 514–530. Springer International Publishing, 2020.

[51] A. Voynov, S. Morozov, and A. Babenko, "Object segmentation without labels with large-scale generative models," *arXiv preprint arXiv: 2006.04988*, 2021.
[52] C. G. Kamra, I. D. Mastan, N. Kumar, and D. Gupta, "Simsam: Simple siamese representations based semantic affinity matrix for unsupervised image segmentation," *arXiv preprint arXiv: 2406.07986*, 2024.
[53] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results." http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.
[54] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "Caltech-ucsd birds-200-2011 (cub-200-2011)," Tech. Rep. CNS-TR-2011-001, California Institute of Technology, 2011.
[55] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan, "Learning to detect salient objects with image-level supervision," in *CVPR*, 2017.
[56] V. Morath, M. Keuper, M. Rodriguez-Franco, S. Deswal, G. Fiala, B. Blumenthal, D. Kaschek, J. Timmer, G. Neuhaus, S. Ehl, O. Ronneberger, and W. W. A. Schamel, "Semi-automatic determination of cell surface areas used in systems biology.," *Frontiers in bioscience (Elite edition)*, vol. 5, pp. 533–545, Jan. 2013. Place: Singapore.
[57] S. J. Cooper, S. A. Roberts, Z. Liu, and B. Winiarski, "Methods—Kintsugi Imaging of Battery Electrodes: Distinguishing Pores from the Carbon Binder Domain using Pt Deposition," *Journal of The Electrochemical Society*, vol. 169, p. 070512, July 2022. Publisher: IOP Publishing.
[58] B. Winiarski and P. Barthelemy, "Kintsugi imaging of battery electrodes with plasma FIB-SEM."
[59] J. Stuckner, B. Harder, and T. M. Smith, "Microstructure segmentation with deep learning encoders pre-trained on a large microscopy dataset," *npj Computational Materials*, vol. 8, p. 200, Sept. 2022.
[60] R. F. Cochrane, "Micrograph 394," 2002. https://www.doitpoms.ac.uk/miclib/micrograph_record.php?id=394.
[61] K. Li, Y. Zhao, Z. Wang, Z. Cheng, P. Jin, X. Ji, L. Yuan, C. Liu, and J. Chen, "Multi-granularity interaction simulation for unsupervised interactive segmentation," *arXiv preprint arXiv: 2303.13399*, 2023.
[62] M. N. Rabe and C. Staats, "Self-attention does not need $o(n^2)$ memory," *arXiv preprint arXiv: 2112.05682*, 2022.
[63] T. Li, "Particle size and pore size selection on ordered mesoporous silica," *ChemRxiv*, 2023.
[64] T. D. Turner, P. Gajjar, I. S. Fragkopoulos, J. Carr, T. T. H. Nguyen, D. Hooper, F. Clarke, N. Dawson, P. J. Withers, and K. J. Roberts, "Measuring the Particle Packing of l-Glutamic Acid Crystals through X-ray Computed Tomography for Understanding Powder Flow and Consolidation Behavior," *Crystal Growth & Design*, vol. 20, no. 7, pp. 4252–4263, 2020.

Supplementary

S1 Linear probe evaluation

Method	Pascal VOC
Bilinear	0.806
FeatUp (JBU)	**0.825**
Ours	0.701

Table S1: Drop-in performance when applying a linear classifier trained on bilinear-upsampled DINOv2-S-14 features to FeatUp upsampled features and features upsampled using our method (strided + shift transforms).

S2 Further resolution comparisons

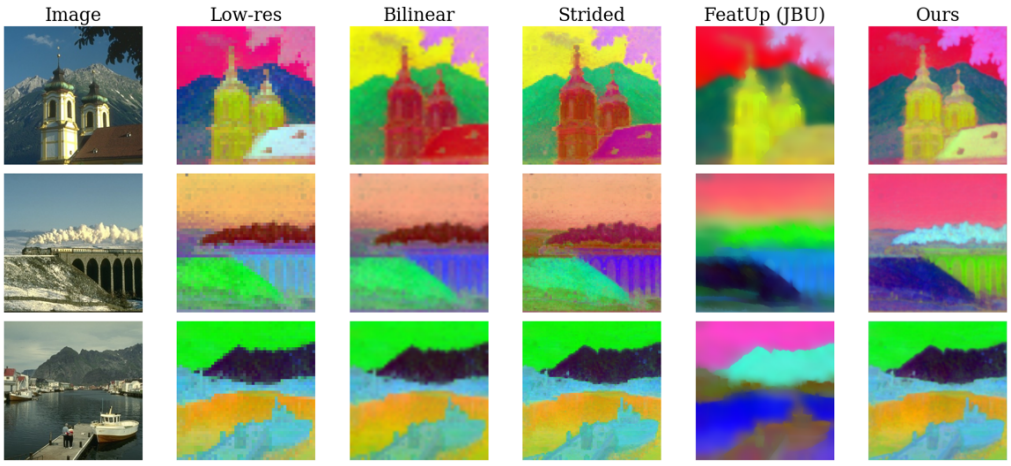

Figure S1: Cross-method comparison of feature upsampling methods on square crops of images from the BSD300 dataset. The model used for the all upsamplings except 'FeatUp (JBU)' was DINO-S-8 - 'FeatUp (JBU)' did not have a pretrained checkpoint for this resolution, so DINO-S-16 was used instead. The input image size was (384, 384).

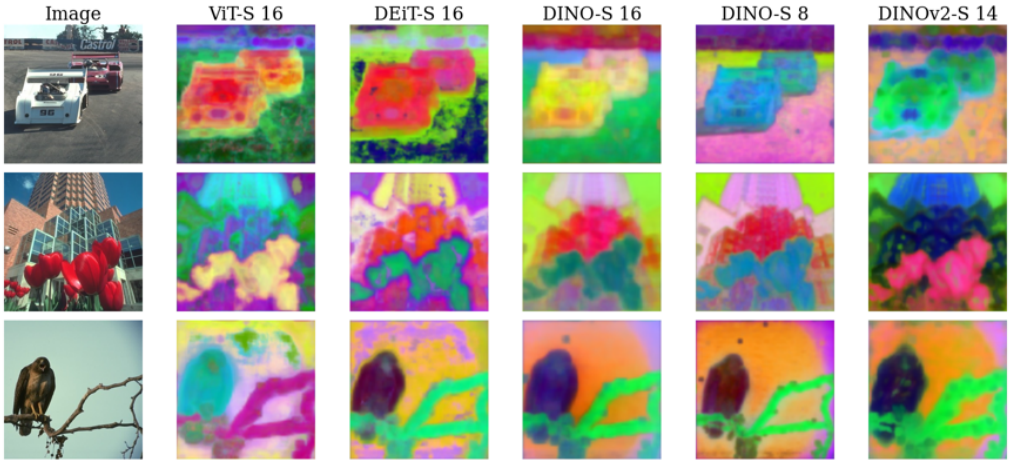

Figure S2: Cross-model comparison of our upsampling method on square crops of images from the BSD300 dataset with shift and flip transforms applied and a stride of 4. The input image size was (224, 224), the training resolution of ViT-S and DEiT-S. Note the DINO models can be applied to arbitrary resolutions, and improve as the image size increases. DINOv2 suffers the worst blurring, possibly because its patch size of 14 is not a multiple of the stride.

S3 Averaging positional effects

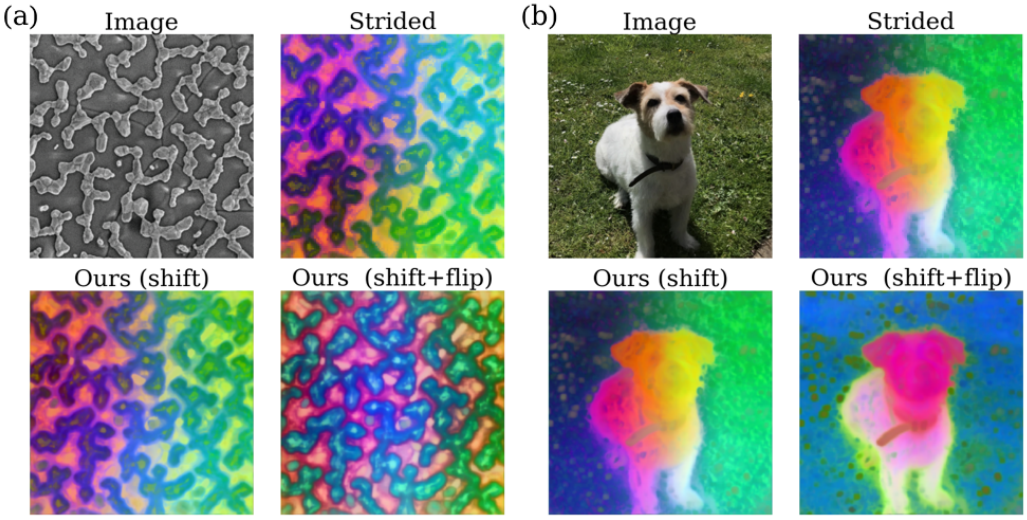

Figure S3: Demonstrating the positional effects present when striding DINOv2 features in **(a)** a Scanning Electron Microscope (SEM) image of a solid-oxide fuel cell and **(b)** a photo of a dog. These can be ameliorated by including flip transformations during the upsampling process, which can be used in conjunction with shift transforms at the cost of computing $N_{\text{shift}} \times N_{\text{flip}}$ feature maps, allowing the features to express more semantic information when displayed in a 3-component PCA. The input image size was (518, 518). Note there is still a center-image bias for the SOFC even after flip transforms have been applied.

S4 Small object retrieval and keypoint matching

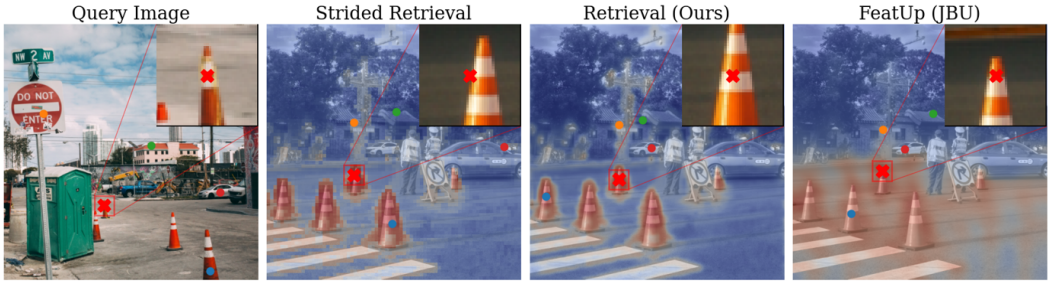

Figure S4: Small object retrieval (red cross) and keypoint matching (coloured dots) for various feature upsampling methods with DINOv2-S-14, stride 7. Following FeatUp, we compute high-resolution features for a query and target image, then find the most similar point in the target image to a given query point (and additional keypoints). The colourmap indicates similarity to the query point, red for similiarity, blue for dissimilarity. We note our similarities are more localised than FeatUp's (JBU).

Figure S5: Small object retrieval and other keypoint matching with DINO-S-8 with stride 4. The similarity maps for the the query point (cross) are sharper than with DINOv2-S-14 (Figure S4), but emphasise colour information more *i.e,* the cone is similar to the orange kerb and car and the stop signpost keypoint is matched to the red and white road sign rather than the background signpost. Using our approach gives a slightly more accurate retrieval than just using strided features.

S5 Practical considerations

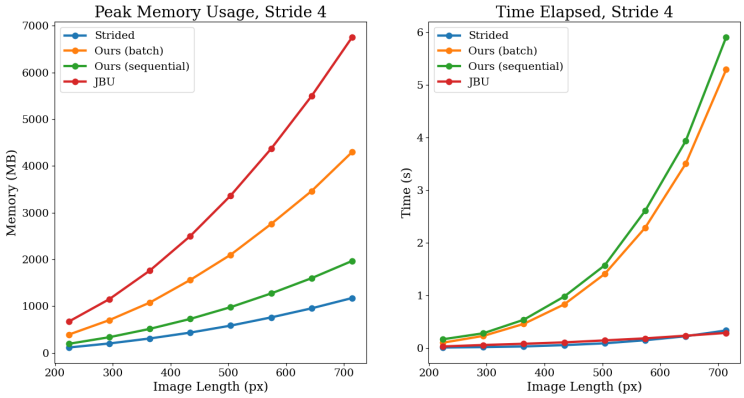

Figure S6: Memory and time usage of various upsampling methods as function of square image length for DINOv2-S-14. We fix the stride for both strided and our methods to be 4 and use shift transforms with a distance of 2 and in an 8-neighbourhood for a total of 17 transformations. 'Batch' refers to computing the features of the transformed image batch in one forward pass, 'Sequential' to computing them one at a time. We note JBU has a harsher memory scaling as image length increases but far better time scaling. We achieve this better memory scaling than the expected n^2 via memory efficient attention[62], which can be added post-hoc into any existing ViT network and admits $\sqrt{n}$ memory cost with n tokens (though still an n^2 time cost). If the stride were to be increased beyond 4, the memory cost would exceed JBU, though this is undesirable due to numerical errors encountered at low strides/high upsampling factors. Float16 precision was used for both measurements; note for image length l, $n \propto l^2$. Values measured on an NVIDIA RTX A6000.

S6 Limitations

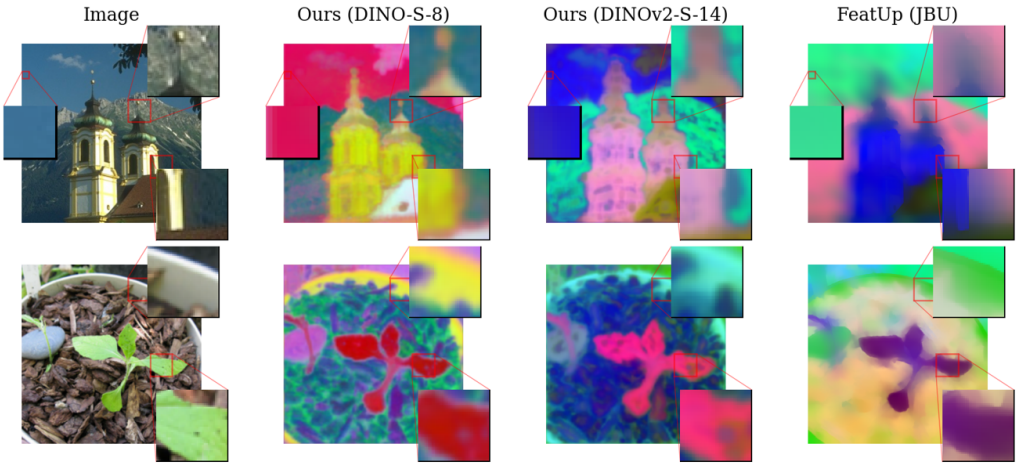

Figure S7: We highlight the limitations of our feature upsampling with two example images, the most major being the blurring caused by reducing the stride of the patch embedding layer - this is most evident around the plant pot's edges when compared to FeatUp's JBU. Comparing our approach applied to DINO-S-8 vs DINOv2-S-14 shows that the resolution is best improved by using models with a smaller patch size, which relies on the group training the feature model; FeatUp does not suffer this problem. One minor problem is edge effects caused by using pixel-shifts with wrap boundary conditions - this causes information from opposite sides of the image to mix, albeit for a small (< 4) number of pixels. Another problem is the increasing time-cost as a function of image resolution (see Figure S6), which can prove prohibitive for real-time applications in domains with high-resolution images, like materials science.

The primary limitation of our feature upsampling method is the blurring introduced by the striding (see Figure S7), which limits the resolution of the later downstream tasks. There are also practical problems with decreasing the model stride (and therefore increasing the number of tokens, n): although the memory cost does not scale n^2 thanks to memory-efficient attention[62], the time-cost still scales with n^2 (*i.e,* N^4_{pixels}). See Figure S6 for more discussion. Materials science frequently handles high resolution volumetric data - reducing the featurisation time is important to ensure these data can be processed.

The unsupervised segmentation workflow tends to decompose large foreground objects into parts (head/body, *etc.*) and treat those as separate classes. This could be improved by merging the features of individual pixels (rather than clusters), but this would be expensive computationally.

The resolution problems and time-cost of generating the high-resolution features limit the applicability of the method in user-facing weakly supervised materials segmentation. Furthermore, it is possible for the classifiers to overfit to positional information present in the deep features - this requires additional homogeneous labelling to overcome or some augmentations to reduce their importance, see Supplementary Section S3. Empirically, combining classical and deep features can make the trained classifiers 'brittle', *i.e,* less sensitive to semantic information and therefore worse at generalizing.

Figure S8: Examples of failure modes in the unsupervised segmentation workflow. First row: over-decomposition of foreground object into legs and body - this can be improved by increasing the merging threshold distance, but can also merge more unrelated classes. Second row: merging of unrelated classes *i.e,* the shadow of an object and the object. Third and fourth row: over-merging objects close in space *i.e,* the rider and the horse/bike.

S7 Unsupervised segmentation examples

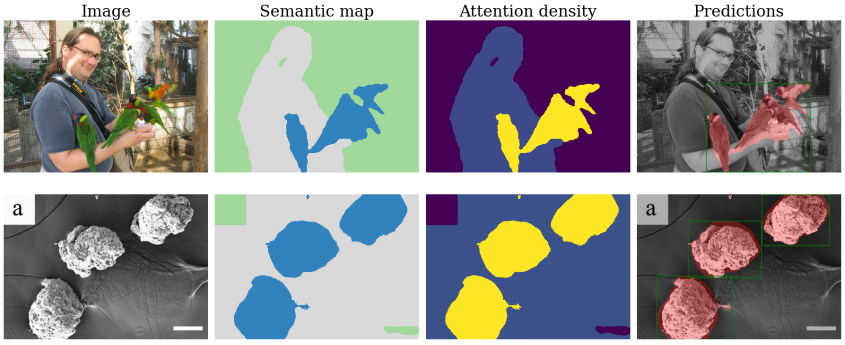

Figure S9: Examples of the unsupervised segmentation workflow applied to a natural scale image and an SEM of mesoporous KIT-6 silica[63], showing how the unsupervised semantic map and attention densities can be used for object detection and localization.

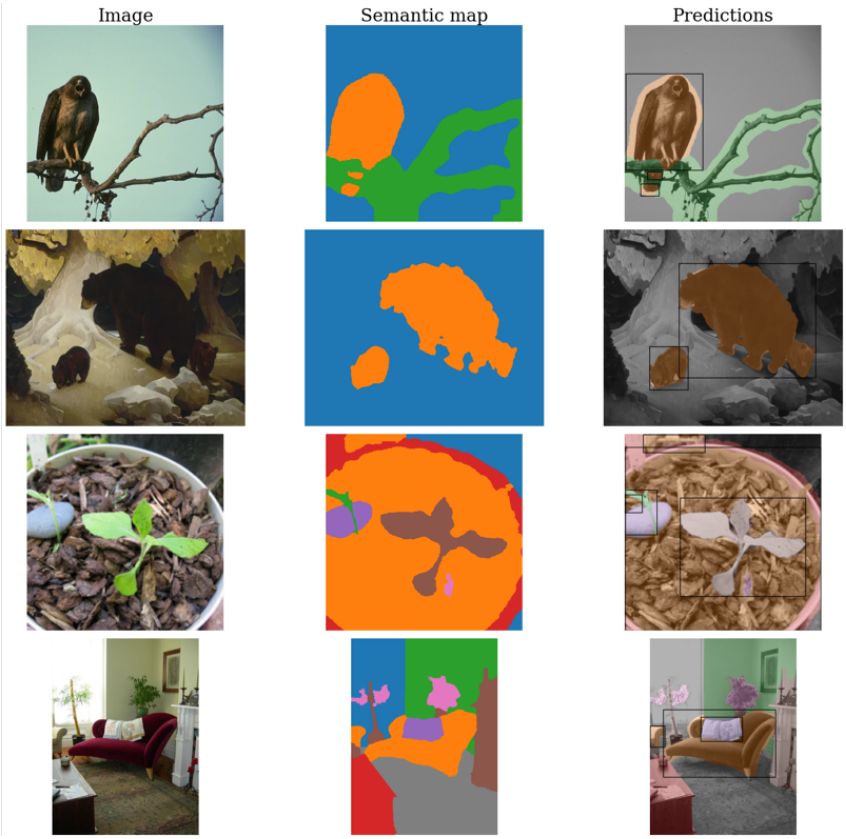

Figure S10: Example predictions of foreground objects and their segmentation using the unsupervised workflow. The background class (*i.e,* the one with the lowest attention density) is show in blue. Note the blurring in the first image - this is more frequent in uncluttered scenes and worse at lower resolutions.

S8 Weakly supervised segmentation

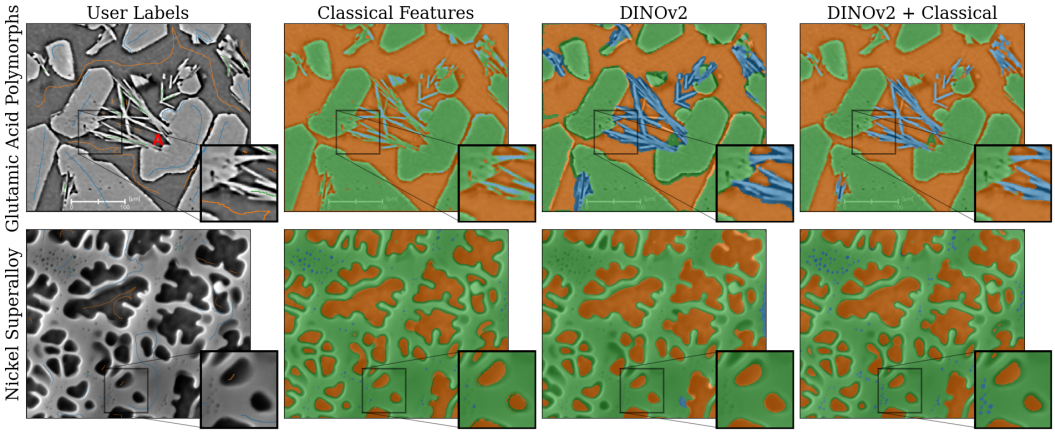

Figure S11: Further examples of weakly supervised segmentation of materials. The first is a slice from an X-ray Computed Tomography (XCT) of glutamic acid[64], displaying two crystal polymorphs: the larger, blocky α-polymorph and the smaller, needle-like β-polymorph[64]. The classical features are unable to distinguish between the two phases, struggling to delineate between edges of a large α block and the β needles, whilst the DINOv2 features succeed but are blurred. The combination of the two produces a good segmentation of both phases - this ability to classify using structural information (rather than focussing on pixel-value) is useful in distinguishing between polymorphs for characterization techniques that have elemental contrast (like XCT or SEM), as the polymorphs will have similar pixel-values[†]. The second micrograph is a Scanning Electron Microscopy (SEM) image of a nickel-based superalloy[59] with three phases: a large, connected matrix phase, secondary precipitates (large blobs) and tertiary precipitates (small blobs). Again, we find that the hybrid approach produces a better segmentation, especially of the third phase.

[†] Phase contrast for polymorphs will be similar in XCT assuming the polymorphs have similar densities and SEM assuming similar depth and that the shape is not so different that edge effects become important.

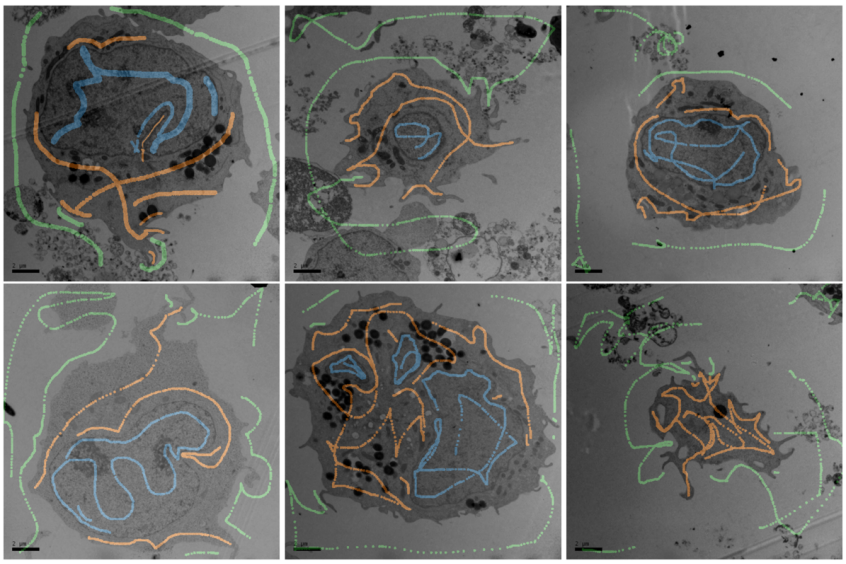

Figure S12: The labels and cells used to train each classifiers in the case study. Classfiers are trained to map from image features (be they classical, from a high-resolution ViT or a combination thereof) to the user-drawn labels, then applied to the unlabelled feature vectors, both of the cells here and the rest of the 129 unlabelled cells in the dataset. Note all cells (train + test) were resized from (1024, 1024) to (518, 518) during training and segmentation.

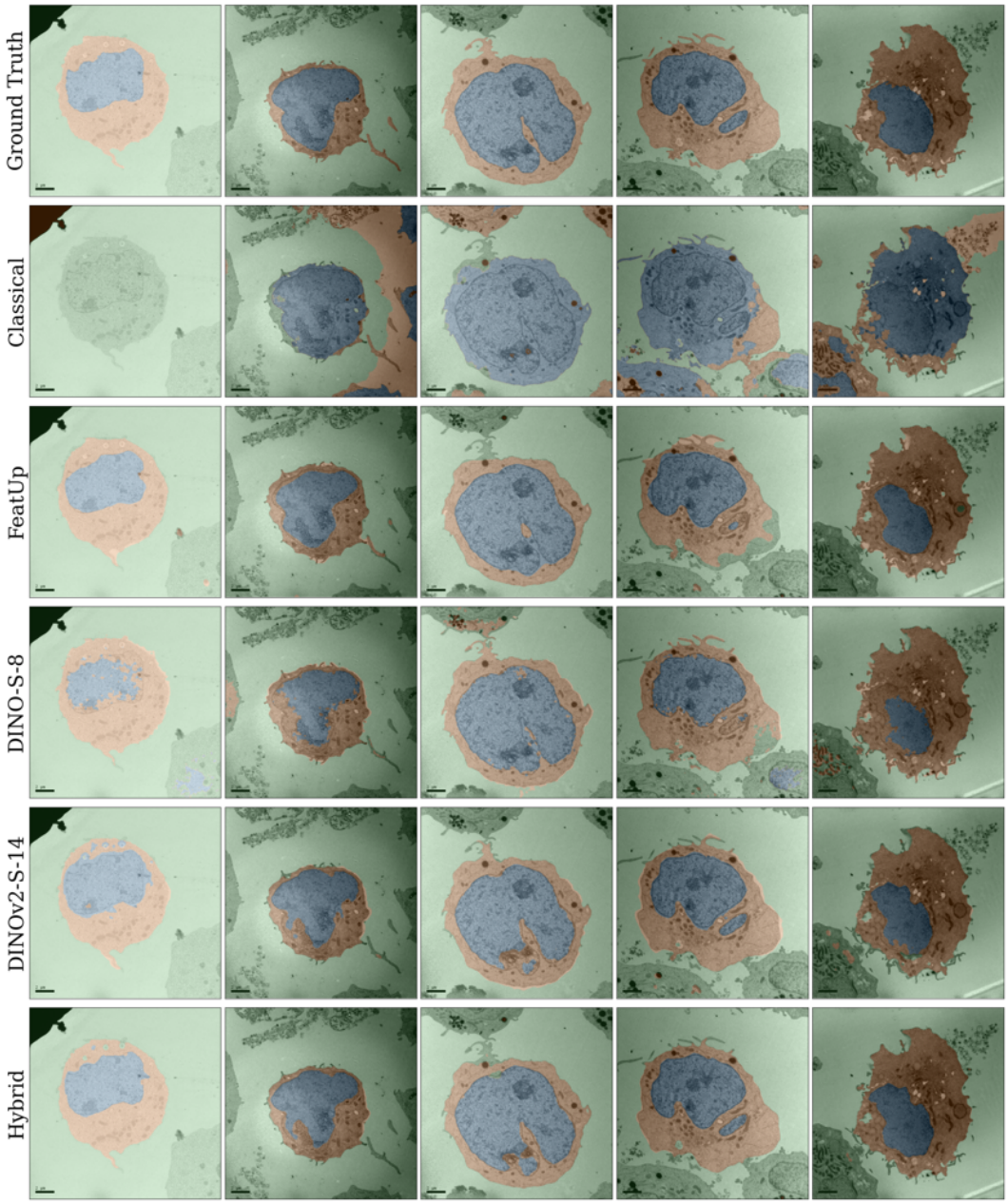

Figure S13: Example segmentations from applying classifiers trained on various feature-sets to unlabelled examples in the T-cell dataset. We note the upsampled ViT features produce segmentations that much more closely align with the ground truth, able to ignore other cluttered cells, handle the presence of multiple nuclei and varying exposure. These predictions have had a CRF applied.

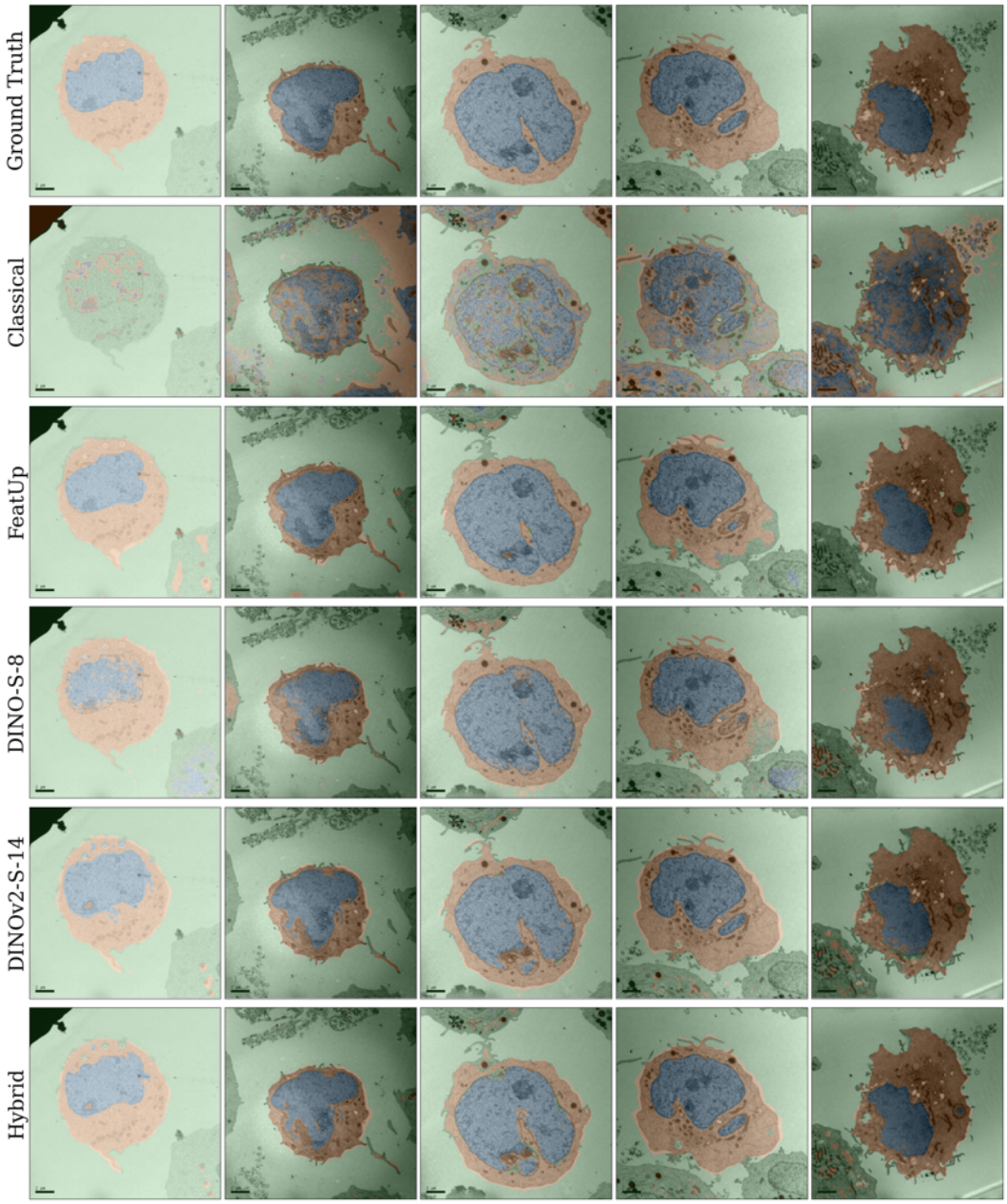

Figure S14: The same predictions as in Figure S13 without a CRF applied.

S9 Transformations

Based on experiments and shown in Figure S15, for a ViT model with stride S, we found the best range of distances for the shift transforms to be $[1, S/2]$. When combined with the flip transforms this gives 4 (flips) $\times$ 8 (directions) $\times$ $S/2 = 32S$ transformations. Usually $S = 4$ so the number of transforms was 64. This could be reduced by using a 4-neighbourhood rather than an 8-neighbourhood, at the cost of slightly reduced performance. Although 4 flip transforms were used (no flip, flip vertical, no flip, flip horizontal), three would also work (no flip, flip vertical, flip horizontal).

For the object detection and localization studies, a stride of 4 was used, as were the shifts [1, 2] as well as flip transforms. Flip transforms were useful to average out positional information to decrease the semantic distance between instances of the same class on different sides of the image.

For the cell segmentation case study, a stride of 4 was used as well as the shifts [1, 2]. No flip transforms were used, positional information was useful in this case to identify the central cell.

For the weakly supervised segmentation examples, a stride of 4 was used, as were the shifts [1, 2] as well as flip transforms (materials segmentation is a more homogenous problem).

Feature-set	**mIoU**
DINO-S-8 (no shift)	0.7446
DINOv2-S-14 (no shift)	0.7915
DINO-S-8 (shift 1, 2)	0.7764
DINOv2-S-14 (shift 1, 2)	0.7974

Table S2: Performance on the cell TEM dataset for ViT features (stride 4, no CRF) without and with shift transforms applied - the performance gain is modest for DINOv2 and more marked for DINO.

Feature-set	**mIoU**
DINOv2-S-14 (no transforms)	0.638
DINOv2-S-14 (flip, no shift)	0.6465
DINOv2-S-14 (no flip, shift)	0.6477
DINOv2-S-14 (flips, shift 1, 2)	0.654

Table S3: Performance on DUTs foreground object segmentation as a function of transforms applied

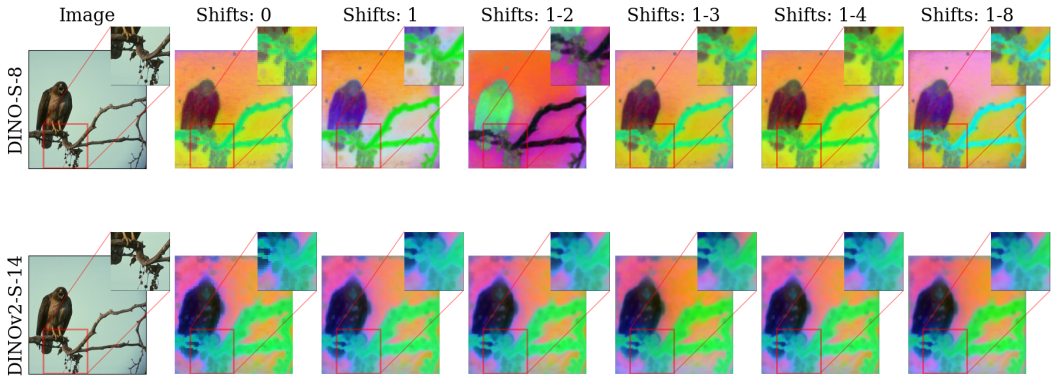

Figure S15: Visualisation of the impact of adding more shift transforms on the feature resolution for different ViTs with stride 4 - we note past shifts 1, 2 (*i.e,* stride / 2) we see little, if any, improvement.