Region-Based Representations Revisited

Michal Shlapentokh-Rothman[1*] Ansel Blume[1*] Yao Xiao[1] Yuqun Wu[1]
Sethuraman T V[1] Heyi Tao[1] Jae Yong Lee[1] Wilfredo Torres[2] Yu-Xiong Wang[1]
Derek Hoiem[1,2]
[1] University of Illinois at Urbana-Champaign
[2] Reconstruct

Abstract

We investigate whether region-based representations are effective for recognition. Regions were once a mainstay in recognition approaches, but pixel and patch-based features are now used almost exclusively. We show that recent class-agnostic segmenters like SAM can be effectively combined with strong self-supervised representations, like those from DINOv2, and used for a wide variety of tasks, including semantic segmentation, object-based image retrieval, and multi-image analysis. Once the masks and features are extracted, these representations, even with linear decoders, enable competitive performance, making them well suited to applications that require custom queries. The representations' compactness also makes them well-suited to video analysis and other problems requiring inference across many images.

1. Introduction

Over the past ten years, recognition capabilities have improved dramatically. For broader application, developing scalable, flexible, and interpretable representations has become more important than ever. For example, we may want to search large image collections with custom queries, create an interactive learning system, or perform complex inferences over many images or video frames.

Region-based representations could serve an important role in these applications. Consider if an image could be fully represented with a few dozen embeddings that represent surfaces, objects, parts, and other meaningful portions of the scene. Compared to embeddings over 16x16 patches, we could then reduce computation and memory for downstream tasks by 10-20x, enabling aggregation of information across regions from many images and efficient object-based searches of image collections. We could simplify interaction by enabling people to operate on the level of regions that correspond to intuitive portions of the scene, rather than at the pixel or patch level. In the past, region-based representations were seen as a critical part of the recognition solution (e.g. [25, 29, 30, 42, 48, 52]), but have fallen to the wayside as deep network architectures excel at processing pixels and patches. Now, given advances in automatic segmentation and unsupervised feature learning, it is time to reexamine the capabilities, potential, and limitations of region-based representations.

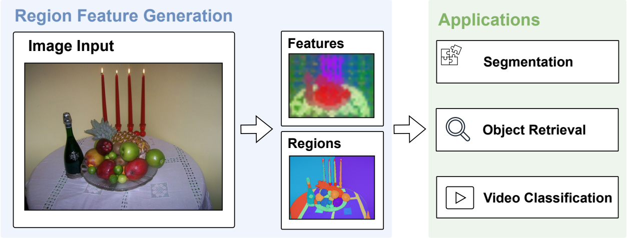

Figure 1. Our framework revisits the use of region features for downstream applications. We generate region features by first segmenting an image, extracting image features, then pooling the image features across the region masks.

In this paper, we explore design choices for region-based representations and investigate their effectiveness for a variety of applications. Some of the design decisions include:

- *How to generate regions?* We ideally want a small number of regions that provide good segmentations for all the surfaces, objects, and salient parts. We explore SAM and some of its recent variants [32, 59], and SLIC [34] as a complementary mechanism to improve completeness.
- *What features are effective in regions?* We compare features from CLIP [49], ImageNet [13], DINOv1 [6] and DINOv2 [44].
- *How to pool features?* We find upsampling the features and then averaging to work better than alternatives.

We explore applications of image semantic segmentation, object-based image retrieval, multi-view semantic segmentation, and activity classification. With semantic segmentation, we explore the design decisions of regions and features and region and image-level decoders. We evaluate one-shot object-based image retrieval, which is useful

[*] Equal Contribution. Correspondence:{michal5,blume5}@illinois.edu

for image search and a foundation for efficient image labeling in an interactive learning setting. For multi-view scene analysis, we explore using 3D positional embeddings and prediction based on multiple viewpoints. For multi-frame activity classification, we further explore using transformers to aggregate region information across frames. These take advantage of the compactness of region-based representations for multi-image inference.

Altogether, our investigations show that region-based representations are much more powerful than would have been possible just one or two years ago and also point to where more work is needed to increase their effectiveness.

In summary, our main contributions are:

1. Investigate key design decisions for region-based representations, including recent methods for mask generation and feature generation, and the efficacy of simple decoders.
2. Propose SAM+SLIC as a simple method to achieve good coverage with few regions.
3. Demonstrate competitive performance across several applications and discuss the current applicability, limitations and potential of region-based representations.

2. Related Work

Region-based representations have a long history in recognition. Recently, feature learning has progressed to create patch-based encodings that are effective for recognition and correspondence, even without full fine-tuning. Recent work has also made tremendous progress in generating a small number (dozens, not thousands) of regions that correspond well to surfaces, objects, and parts. We describe some of the most relevant.

2.1. Region-based Recognition

Segmentation has long been proposed as a pre-process to image analysis. Compared to pixels or patches, well-segmented regions provide better spatial support for features and more compact image representations, enabling faster inference or retrieval and reduced memory usage. Unsupervised segmentation methods (e.g. [23, 34, 53]) are unreliable, so many methods [25, 29, 30, 42, 48, 52] use hierarchies [4] or bags of regions from multiple segmentations [29, 48, 55] generated with different parameters, or formed during the image analysis [25, 54]. Object proposal methods [3, 7, 16] aim to produce a small number of regions that could represent the most depicted objects. Such proposal mechanisms, particularly Selective Search [55], were important components in early deep network object recognition methods like Fast-RCNN [24], but, for speed, architectural simplicity, and end-to-end training, the use of preprocessed regions has given way to generating boxes or labels based on feature grids [50] or tokens, aggregating information across the image using convolution, pooling, and/or attention mechanisms [15].

2.2. Feature learning for patch-level representations

Self-supervised pre-training on large amounts of data has been shown to be an effective visual representation learning approach [6, 27, 28, 44, 49]. Using self-supervised pretraining techniques, both DINO models (DINOv1 [6], DINOv2 [44] produce image encodings that perform well on a wide variety of correspondence, dense prediction, and image classification tasks, even when using simple decoders. DINOv2 incorporates data curation to build a larger pretraining dataset than in DINOv1. CLIP [49] is contrastively trained to match images with text, enabling open vocabulary image classification, and MaskCLIP [62] provides mechanisms to extract useful patch-level features from the CLIP image encoder for open vocabulary semantic segmentation.

We investigate the effectiveness of many of these features when mask-pooled to create region representations. While many of these representations are similarly effective when tuned for downstream tasks, we find large differences when used as region representations.

2.3. Segmentation

SAM [36] is a class-agnostic segmentation model composed of a prompt encoder, vision encoder, and mask decoder. Given a prompt in the form of a point or bounding box, SAM generates a set of pixel masks and selects those with the highest scores. The generation and scoring models are learned from a large training set. To automatically generate many masks for an image ("segment everything"), SAM can be provided a set of points, e.g. on a 32x32 grid, then generate many regions, selecting a subset based on stability scores, quality scores, and non-maximum suppression. SAM does not partition the image: one pixel may be in multiple regions while another is in none.

In short order, others have built on SAM. For example, MobileSAMv1 [59] distills a smaller encoder from the original SAM encoder for faster mask generation, while claiming performance "on par" with SAM. HQ-SAM [33] augments SAM's decoder to produce higher quality masks. Concurrent to SAM, SEEM (Segment Everything Everywhere All at Once) [63] generates high quality masks based on text and a variety of user annotations. Also concurrent, Qi et al. [41] propose a dataset and model that achieves high quality semantic segmentation on many labels.

Prior to the deep learning era, superpixel [37, 43, 51, 56], algorithms were widely used for many tasks including unsupervised segmentation. Starting with a grid of points, Simple Linear Iterative Clustering (SLIC) [34] performs local K-means clustering to efficiently generate superpixels, small regions that partition the image consistent with image boundaries. Later implementations, such as FastSLIC [35], improves the speed by an order of magnitude to 30 ms per image.

The SAM-based methods mainly evaluate based on generation of individual masks from points or bounding boxes based on detections or ground truth masks, leaving their relative efficacy for complete image segmentation largely

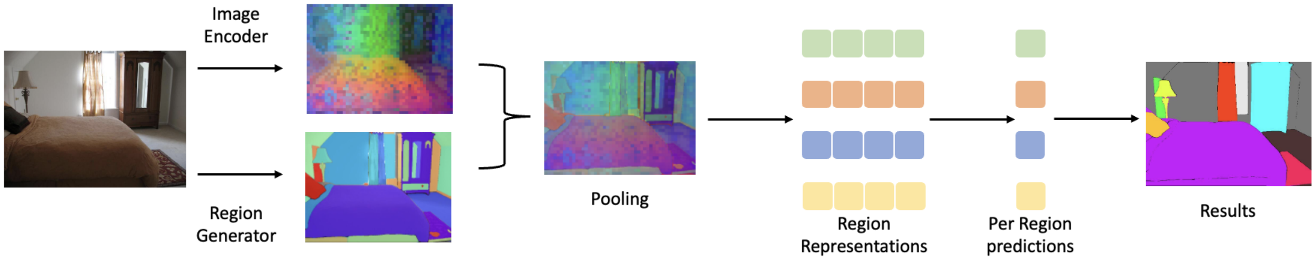

Figure 2. Method overview. We generate masks using class-agnostic segmenters, such as SAM, and patch-based features using strong representations, such as DINOv2. The features are average-pooled in the masks, creating region-based representations, which can then be decoded with linear classifiers or decoders for a variety of tasks.

untested. We evaluate and compare SAM, MobileSAM(v1), HQ-SAM, and FastSLIC for speed, compactness, coverage, and utility in semantic segmentation tasks. We also propose a simple method to improve coverage without adding many regions by combining SAM and SLIC.

2.4. SAM-based Recognition

Many works and software repositories have sought to combine SAM with recognition. Grounded SAM [26] applies SAM to segment boxes from Grounding DINO [40], which in turn builds on DINOv1 [6] and BERT [14]. Semantic Segment Anything [8] refines labels from semantic segmentation models with SAM. These mainly use SAM as a region refinement post-process. Segment Anything with CLIP [46] generates regions with SAM, crops the image around each region, and classifies each crop using CLIP. By contrast, we directly encode regions by pooling features in masks and use the region representations directly for downstream tasks, which is simpler, faster, and often more effective than encoding crops with CLIP.

3. Methods

We first describe how to build region representations by generating masks and image features, and then pooling the features within the masks. Despite our method's simplicity, our experiments show that the details matter. Next, we describe how to use these region representations for semantic segmentation of images, object-based image retrieval, multi-view semantic segmentation, and activity classification.

3.1. Generating and Representing Regions

See Figure 2 for an overview.

SAM. Masks produced by SAM [36] tend to correspond to intuitive portions of the scene, such as whole objects, parts of objects, surfaces, and shadows. The masks may overlap. For example, one mask may contain all pixels pertaining to a car, while others correspond to a tire or license plate. The quality and number of masks produced depends on the set of input point prompts and parameters such as the stability threshold. Denser grids of points tend to increase coverage but take more time to process. A higher stability score increases the quality of the masks but decreases the number (and coverage) of masks generated.

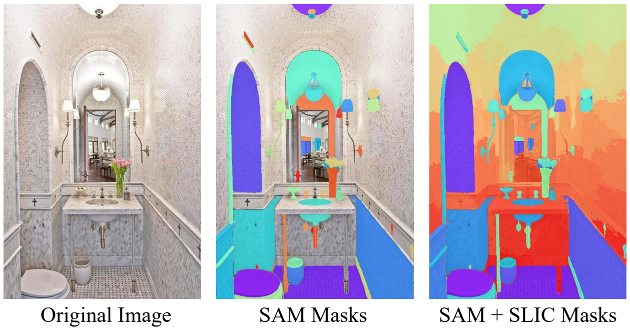

Figure 3. A comparison of region coverage when using SAM and SAM with SLIC. SLIC fills in many of the uncovered regions, leaving few holes.

Augmenting SAM with SLIC. As shown in Figure 3, SAM-generated masks may fail to cover significant portions of an image. Reducing the stability threshold alleviates the problem, but results in many poor-quality masks. Iterative use of SAM to try to cover unmasked regions would be prohibitively slow. Instead, we use the FastSLIC implementation of SLIC [34] to generate a moderate number of regions and intersect them with pixels that are not covered by any SAM mask. We generate superpixels with fifty components and a compactness of 8, keeping masks which intersect at least 300 pixels of unmasked image regions. We find the combination of SAM and SLIC to be an efficient and effective way to increase coverage without greatly increasing the number of regions. In Figure 3, we see the increase in region coverage when augmenting SAM with SLIC.

Features and Pooling. For a given SAM mask, we wish to aggregate image features within the mask. We focus on patch-based feature representations produced by vision transformers [15] due to their usage in state-of-the-art methods [28, 44]. In a vision transformer, an input image of shape (h, w) is divided into a flattened sequence of N patches of resolution (p, p) where $N = \frac{hw}{p^2}$ (assuming h, w are divisible by p; padding or cropping may be applied to achieve this). The output is a sequence of N patches which can be reshaped to have dimension $d \times h/p \times w/p$, where d is the embedding size and p is the patch size.

To create features for a SAM mask, we require the image feature maps and generated masks to be the same size so they can be superimposed. Two straightforward options include downsampling the masks to the feature map dimensions, or to upsample the features to the image dimensions. We found that upsampling the $d \times h/p \times w/p$ features to $d \times h \times w$ was most effective across datasets—, downsampling the SAM masks sometimes reduced small regions to a single point in the patch features, or shrank them to nonexistence. Upsampling the image features to the image size retains the regions' fine granularity.

With the region masks and feature maps the same size, the features are pooled within each mask to serve as each region's representation. We experiment with max and average pooling and choose the latter as it works best. After pooling, each region is represented by a d-dimensional vector, and an image can be represented by its collection of region vectors and their masks.

3.2. Application of Region Based Representations

We apply region-based representations to several applications. The same representation can be used for semantic segmentation, retrieval, or classification, and its compactness enables fast and flexible queries and information to be aggregated across many images.

Semantic Segmentation is the task of predicting a label for every pixel. Producing accurate label maps is challenging, since patch-based representations and convolutional layers typically have much lower resolution than the input image. A common approach is to fine-tune and augment patch-level representations with adapters [9] and specialized decoders [10, 11, 57] to improve the precision.

Once we encode regions, we can treat semantic segmentation as a *region classification* problem. Given a region represented by its region features, we predict the label probabilities for the entire region. We set the probability that a pixel is assigned a label to the average probability that its containing regions are assigned that label.

To derive region labels for training, we assign a region a label if it contains at least some threshold percent of pixels with that label, according to the pixel-level ground truth. Regions without any assigned label are excluded from training. We train with cross-entropy loss, weighting regions proportionally to the number of pixels they contain. We experiment with linear, MLP, and transformer [15] decoders.

Multi-view Semantic Segmentation. A 3D scene may be represented by multiple views, e.g. based on video or photos, and one may wish to recognize, count, or infer objects and properties within the scene. Such processes currently require 3D models and cumbersome inference.

We explore a region-based approach to multi-view semantic segmentation that is a simple extension of image-based semantic segmentation. We add a 3D positional encoding to each region based on underlying 3D points, and use a transformer decoder to jointly process regions across many images of the same scene. This is only possible as each image can be represented with a few dozen region encodings instead of ≈ 1000 patch encodings.

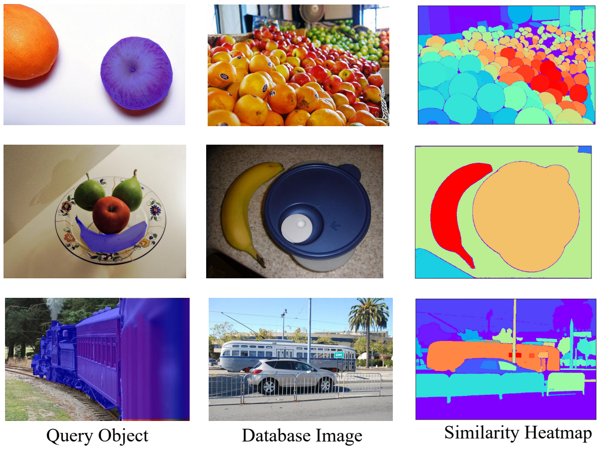

Figure 4. Examples of object retrieval with region representations. The query object is highlighted in the first column. The second column contains the database images, and the third column shows the similarity score between all of the regions in the database image and the query object. Our method matches objects in database images to the query object under different settings.

Object-based Image Retrieval is the task of retrieving images containing a query object. This task has many practical applications: for example, this might enable a field engineer to find all the step ladders on a job site, or an individual to recall where they put their keys. These problems can be solved by showing an example of the desired item and retrieving images containing similar objects. Object-based image retrieval is also highly useful in an interactive learning loop. Starting from an initial example, images are retrieved containing other examples of the object, a model is updated, and more examples are found. The main challenges lie in creating a query from a single example, then efficiently and effectively searching an image collection based on that query. Whole-image representations such as CLIP [49] may inadequately represent small objects, and powerful detectors [5, 22] are difficult to train from a few examples and slow to search through thousands of images.

Object-based image retrieval is a natural application for region representations. Given an encoding of a region or a linear classifier trained on encoded regions, we can efficiently search a database of regions using FAISS [31] or similar libraries. We experiment with one-shot retrieval, using a single query object based on a ground truth mask and pooled features. Dot-product similarity between the query and all region encodings in the image collection is used to sort images, based on the most similar region in each image. In Figure 4, we visualize the similarity scores between all regions in the database image and query image. Our method can detect multiple instances of query objects in an image even when the objects are small or are not the focus of the

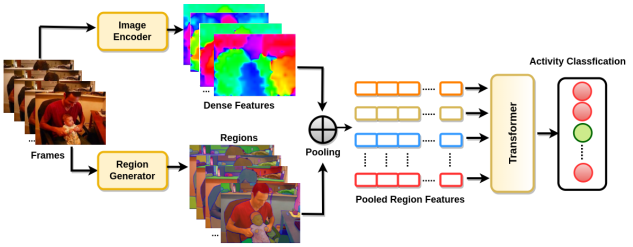

Figure 5. Video Activity Classification Method Overview. By pooling regions across video frames, we can categorize a video using a small fraction of the number of tokens that would be required for patch-based representations.

image. In the first row of Figure 4, our method is able to differentiate between the query object (apple) and objects of similar shape and size (oranges).

Activity Classification. Many works have successfully adapted image-based foundation models to the video domain by integrating adapters or through fine-tuning [39, 45, 58]. Despite the success of these models, it remains difficult to process multiple images or frames and to capture temporal dynamics. For example, ViT-L/14 creates 1,369 patches for one 518x518 image. Joint training on patches across many images is therefore not tractable with commonly used GPUs. Some approaches, e.g. [58], decouple self-attention to operate on one patch position across frames and many patch positions within each frame, but this approach may not fully aggregate information across moving objects.

A region-based representation is ideal for multi-frame inference. In our approach, frames have on average 20 to 30 SAM-generated masks. If we sample 8 frames per video, self-attention is computed at most 240 times for the entire video without separating the spatial and temporal components. Additionally, we can track regions across frames and use such temporal information as part of our representation.

We pick eight evenly spaced frames from each video and extract region features for each frame. These features are then combined as a collection of tokens to get the video representation. Since the number of masks can vary from one video to another, we pad the region features to 400 per video. We then train a classifier to identify video activities using three transformer blocks.

4. Experiments

In Sec. 4.1, we experiment with a variety of region generation methods, feature types, architectures, and pooling methods on semantic segmentation. This informs our design choices for applications. In Sec. 4.2, we test on per-image and multi-view semantic segmentation, object-based image retrieval, and activity classification.

Unless otherwise specified, regions are generated using SAM ViT-H [36] with features generated by the DINOv2 ViT-L/14 backbone [44]. Baseline methods use the same frozen DINOv2 features. Pascal VOC [21] and ADE20K [60, 61] results are evaluated on the validation sets unless otherwise indicated. Additional parameters are included in the supplemental material.

4.1. Region Representation

Region Generation. In Table 1, we compare region generation approaches, including three SAM [36] variants and SLIC [34] superpixels. We report the time to compute regions, the average number of regions produced, the actual segmentation accuracy, and the "oracle" segmentation accuracy. *Actual* uses predictions from a trained linear decoder, and *oracle* assigns regions with the most common label of its pixels. Of the SAM variants, HQ-SAM performs best with the fewest regions and highest actual and oracle performance. Mobile-SAMv1 is only slightly faster, as the time is dominated by the decoder, and generates worse results for this use case. Surprisingly, the unsupervised and super-fast SLIC outperforms MobileSAMv1 in generating regions for semantic segmentation. Inspection showed that MobileSAMv1 would frequently leave large amounts of the image unsegmented (and unsegmented regions receive a score of zero), whereas SLIC segments the entire image. We find SAM has a good trade-off between speed and quality, so we use it with the default hyperparameters as our primary region generation method. Combining SLIC with SAM adds only 15 regions on average and almost no time, but significantly boosts performance. A breakdown of total inference time for each step using SAM (ViT-H) and DINOv2 (ViT-L/14) is in Table 2.

Pooling. In Table 3, we compare average vs. max pooling and upsampling features vs. downsampling masks. Upsampling features and average pooling gives the best results.

Feature Type and Model Size. We experiment with several types of image features: DINOv1 [6], DINOv2 [44], CLIP [49], MaskCLIP [62], and a pre-trained ImageNet vision transformer, as well as different model sizes. For MaskCLIP, we use the vanilla version, as we found no benefit to the other augmentations used in MaskCLIP+ when pooled with SAM masks. In early tests, we also found that region-pooled vanilla MaskCLIP outperforms MaskCLIP+ on ADE20K zero-shot semantic segmentation. As shown in Table 4, DINOv2 outperforms all other models by a large margin. Based on the results for different DINOv2 variants (Table 5), we choose DINOv2 ViT-L/14 for our experiments.

4.2. Applications

Semantic Segmentation. In Table 6, we compare patch-based representations to region-based representations for semantic segmentation using linear classifiers. The patch-features are bilinearly interpolated to the image resolution to compute per-pixel prediction and loss. Region features soundly outperform patch features on both datasets across all feature types. Although DINOv2 performs best, the biggest patch-to-region performance jump is in MaskCLIP, likely because the smoothing that mask-pooling provides is more important for MaskCLIP features.

Table 1. **Comparison between region generation approaches on semantic segmentation.** In order: average time to process an image; average number of regions per image; actual and (oracle) mIoU on PASCAL VOC 2012 and ADE20K. Oracle performance assigns the label probability of a region as the fraction of the pixels in the region with that label. Average time and number of regions are measured on ADE20K.

	s/im	reg/im	VOC	ADE20K
SAM (ViT-H) [36]	4.61	90.3	83.6 (91.9)	50.2 (77.5)
Mobile-SAMv1 (ViT-T) [59]	3.22	38.7	52.4 (58.1)	29.9 (46.5)
HQ-SAM (ViT-H) [33]	7.36	74.8	85.1 (92.6)	50.7 (79.5)
SLIC [2]	0.027	47.6	74.1 (82.8)	40.7 (62.1)
SAM + SLIC	4.64	106	**87.2** (95.6)	**52.8** (77.9)

Table 2. **Time (s/img) for region representations on 1 A40.**

SAM	SLIC	DINOv2	Pooling	Classification
4.61	0.03	0.01	0.46	0.13

Table 3. **Comparison of pooling methods on semantic segmentation.** Upsampling the DINOv2 ViT-L feature grid to the mask size and average pooling works best.

Pooling Type	Pascal VOC	ADE20K
None (Patch based)	82.1	47.7
Upsample Features, Max	81.6	47.3
Downsample Masks, Average	76.3	44.5
Upsample Features, Average	**83.6**	**50.2**

Table 4. **Comparison of region features on semantic segmentation.**

Feature Type	Architecture	Pascal VOC	ADE20K
DINOv1	ViT-B/16	66.2	33.0
DINOv2	ViT-L/14	**83.6**	**50.2**
CLIP	ViT-B/32	65.7	28.6
MaskCLIP	ViT-L	76.7	41.2
ImageNet	ViT-L	54.6	24.2

Table 5. **Comparison of DINOv2 model sizes.**

Architecture	Pascal VOC	ADE20K
DINOv2 ViT-S	75.1	46.1
DINOv2 ViT-B	81.2	48.6
DINOv2 ViT-L	83.6	**50.2**
DINOv2 ViT-G	**84.2**	49.7

In Table 7, we find further gain using a per-region MLP (hidden layer of 1000 nodes) or transformer decoder (1 block), with not much difference between the two. These experiments also add the original DINOv2 positional embedding to the patch features before pooling, which provides a negligible gain of 0.001. Our SAM+SLIC result on VOC 2012 Test is the highest of all existing methods that do not use extra training data[1], outperforming dozens of recent approaches. On ADE20K (Table 8), our performance is lower than SotA but the linear performance of 52.9 mIOU is quite good considering we have only 154K tunable parameters, along with no data augmentation, test-time augmentation, long-tail modifications, or other tricks.

Table 6. **Comparison between region and patch representations for semantic segmentation.** Regions are generated by SAM with ViT-H. Region features outperform patch-based features across several different models.

Architecture	Pascal VOC		ADE20K	
	Patch	Region	Patch	Region
DINOv1 ViT-B/16	58.1	**66.2**	26.7	**33.0**
DINOv2 ViT-L/14	79.9	**83.6**	43.4	**50.2**
CLIP ViT-B/32	51.1	**65.7**	22.3	**28.6**
MaskCLIP ViT-L/14	59.2	**76.7**	30.5	**41.2**

Table 7. **Comparison of decoders on semantic segmentation.** MLP and transformer decoders with SAM+SLIC regions perform best. All decoders outperform patch based metrics. * denotes scores reported by DINOv2 [44]. The top section contains evaluations on validation splits, the bottom section on test splits (ADE20K does not have a test split).

Feature Type	Architecture	Pascal VOC	ADE20K
Patch [44]	Linear	81.2*	47.7*
SAM+SLIC	Linear	86.9	52.9
SAM+SLIC	MLP	**88.4**	53.3
SAM+SLIC	Transformer	88.1	**53.5**
Patch [44]	Linear	83.0 (Test)*	-
SAM+SLIC	MLP	88.4 (Test)	-

Table 8. **Comparison of Semantic Segmentation Methods**

Method	Decoder	Extra Training Data	ADE20K
InternImage [20]	Mask2Former+ ViT-Adapter	✓	62.5
DINOv2	Mask2Former+ ViT-Adapter	✓	60.2
DINOv2	Linear	✗	47.7
Region Representation	Linear	✗	52.9

Multi-view Semantic Segmentation. In Table 9, we evaluate multi-view semantic segmentation on the ScanNet [12] 2D semantic label benchmark. Standard approaches use provided 3D point clouds to aid prediction and fuse per-

[1] https://paperswithcode.com/sota/semantic-segmentation-on-pascal-voc-2012

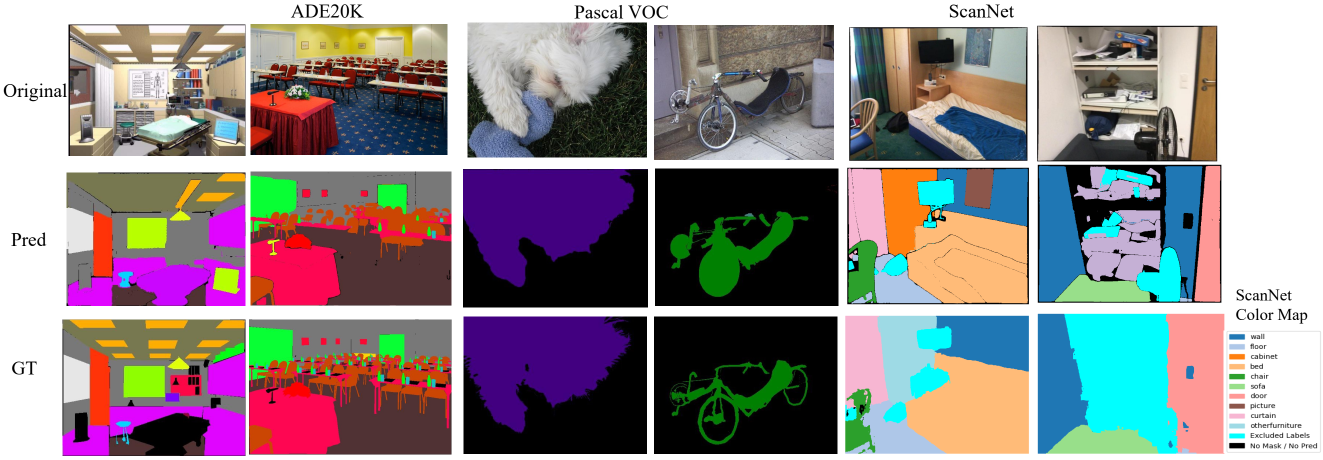

Figure 6. **Semantic segmentation examples**. In the first column, the ceiling lights may be missed because SAM did not segment out each light, or a region over the ceiling and its lights had more weight. The adherence to image boundaries and ability to segment fine objects is not perfect, but very good, e.g. chairs, bottles, cups as shown in column 2. The scores for the images in the last two columns show that our predictions are very precise, while the ground truth is often more noisy. However, the mIoU scores of 68.1 and 49.6 in the last two columns indicate that these numerical evaluations do not fully capture the actual performance.

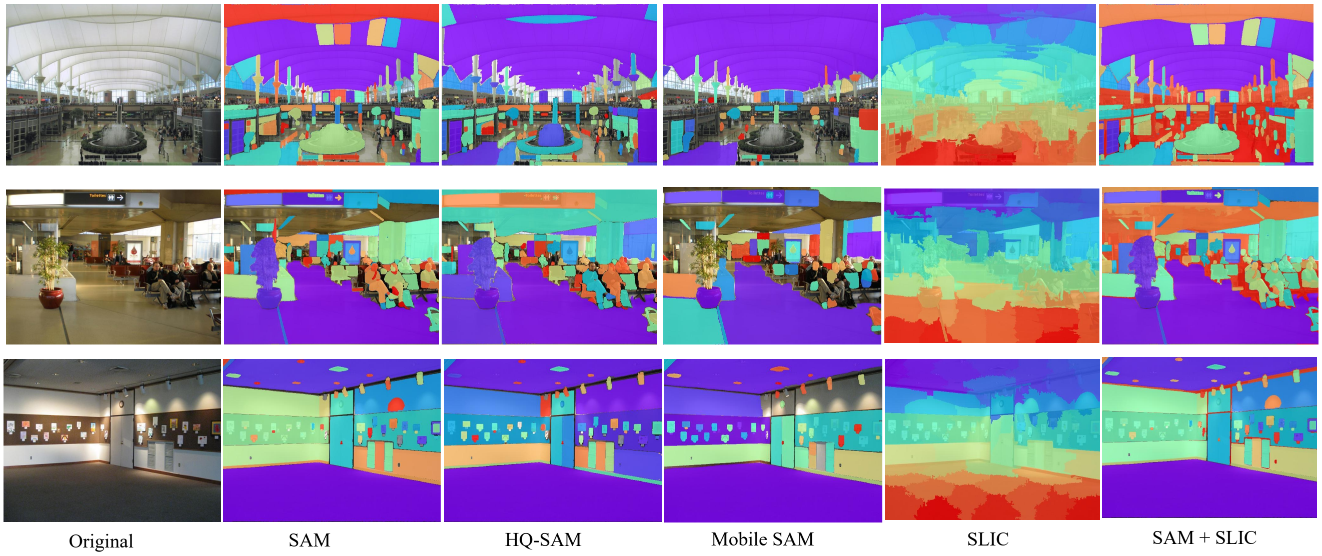

Figure 7. **Region generation examples**. Regions are indicated by different colors. SAM and HQ-SAM regions are high-quality but frequently do not cover the entire image. MobileSAMv1 regions have considerably less coverage. SLIC completely partitions the image but frequently does not respect object boundaries. SAM+SLIC guarantees excellent coverage while benefiting from high quality SAM regions.

image predictions. Our region representations enable a simpler approach, embedding each region with 2D and 3D positional embeddings and using a transformer to predict on all images (or a large subset) within each scene.

We evaluate different design decisions, comparing per-region linear probe against per-image transformers and multi-view transformers, and we compare the impact of 2D and 3D embeddings. We use the author-provided train, val, and test splits, and measure the mIOU per pixel for all the validation scenes. The embeddings are not very helpful for per-region prediction. The per-image transformer performs slightly better than the per-region linear probe, and the per-scene transformer improves further.

In Table 10 we compare our region-based approach with current state-of-the-art methods for multi-view segmentation. While our method does not have SotA performance, we found that the ground truth labels are not very accurate (Fig. 6), and the actual performance of our approach is often much better than the numbers indicate.

Object-based Image Retrieval. We use the COCO dataset [1] for one-shot object-based image retrieval. For each class or object type in COCO, we sample 50 ground truth masks

Table 9. **Multi-View Semantic Segmentation with regions on ScanNet [12].** " An "Image" input source implies the use of regions from a single image, whereas "Scene" indicates the use of regions from the whole scene. Each "Emb." represents addition of embedding features to the visual features. Evaluations are performed on the validation set.

Model	Input Source	Embeddings			mIOU ↑
		Image Regions	2D Pos.	3D Pos.	
Linear Probe	Image	✓			66.0
Linear Probe	Image	✓	✓		66.0
Linear Probe	Image	✓	✓	✓	66.1
Transformer	Image	✓	✓		66.4
Transformer	Image	✓	✓	✓	66.6
Transformer	Scene	✓	✓	✓	**67.5**

Table 10. **Comparison of Multi-View Segmentation Methods**

Method	ScanNet (Val)
Virtual Multi-view Fusion [18]	74.9
Region Representation	67.5
BPNet [17]	66.5

Table 11. **Object retrieval results.** Region representations significantly outperform single token-based representations

Method	COCO mAP	COCO@50
CLIP-Crop [49]	0.27	0.38
DINOv2 [44]	0.13	0.33
Region Representation (Ours)	**0.45**	**0.58**

of the object. Each mask comes from a different image. These masks become the query instances for that particular class. The COCO validation set acts as the image database. We compare our method as described in Section 3.2 to two baselines. Our method compares the query region features to all the database regions and sorts images by their maximum region similarity scores. "DINOv2" [44] computes the similarity between the average (DINOv2) feature in the query object (mask) and the CLS token for each of the images. "CLIP-cropped" [49] computes the CLIP CLS token of an image cropped around the region. The extracted CLS token of the cropped image is used to compute the similarity with the CLS token of database images. For each query image, we compute the mAP and precision@50, averaging over the 50 images in each class and across classes.

As shown in Table 11, using a region representation greatly outperforms the two baselines. Both baselines use a single token for the entire image, so objects from different parts of the image are unlikely to be well-encoded. Based on these results, region-based representations have the potential to be highly effective for retrieval and interactive learning applications.

Activity Classification. To compare the effectiveness of region features with patch features, we follow DINO's linear probe setting on video action recognition. We pick eight evenly-spaced frames in the video, extract region features for the selected frames, and train a three-layer transformer with the region features. Training a full cross-attention patch-based transformer would require 10,952 patch tokens, whereas our approach needs at most 400 region tokens. The results (Table 12) on the Kinetic 400 dataset indicate that using the region features yields a decent improvement over the patch-based method without using video-specific architecture like in ATM [19].

Table 12. **Comparison of Activity Classification Methods**

Method	Decoder	Kinetics-400
ATM [19]	Temporal Transformer	89.4
DINOv2	Linear	76.3
Region Representation	Transformer	79.5

5. Conclusion

One year ago, region-based representations would not have performed well. Now, simple mask-pooled feature representations, while not SotA, perform competitively even with linear classifiers. The main advantage of region-based representations is that, once region masks and features are computed, image collections can be efficiently queried and inference performed jointly on many related images. This is especially beneficial for multiview and multiframe inference and applications that require customizable queries.

The main disadvantage currently is that SAM is slow. If efficient prediction is needed for one well-defined task, it makes more sense to use patch-based decoders. However, continued advances in region and feature generation will likely make region-based representations increasingly useful. For example, the PyTorch team released an implementation of SAM that is 8x faster and of the same quality [47].

Beyond better mask and feature extractors, region-based representations have much untapped potential. For example, for activity classification, embeddings of human pose and optical flow could be added to the appearance-based region features. Multi-view scene analysis could potentially count objects in a scene and do other tasks that require many images, even without an underlying 3D model.

In conclusion, we provide insights on how to best construct region-based representations and demonstrate their efficacy on a range of tasks. These representations are already useful when customizability or interaction is important and will become increasingly useful as methods progress. True progress, one might argue, is not advancing the state-of-the-art but advancing the baseline, and region-based representations advance the baseline.

Acknowledgement This work is supported in part by the following awards: ONR N00014-23-1-2383, ONR N00014-21-1-2705, DARPA HR0011-23-9-0060, NSF IIS 23-12102. The views and conclusions expressed are those of the authors, and not necessarily representative of the US Government or its agencies.

References

[1] Papers with code: Object detection on coco test-dev. https://paperswithcode.com/sota/object-detection-on-coco. 7

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and Sabine Süsstrunk. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012. 6

[3] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34, 2012. 2

[4] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011. 2

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 4

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Herv'e J'egou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640, 2021. 1, 2, 3, 5

[7] João Carreira, Rui Caseiro, Jorge Batista, and Cristian Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, pages 430–443, 2012. 2

[8] Jiaqi Chen, Zeyu Yang, and Li Zhang. Semantic segment anything. https://github.com/fudan-zvg/Semantic-Segment-Anything, 2023. 3

[9] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. In *The Eleventh International Conference on Learning Representations*, 2023. 4

[10] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. 2021. 4

[11] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. 2022. 4

[12] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 6, 8, 1, 3

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 3

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, abs/2010.11929, 2021. 2, 3, 4

[16] Ian Endres and Derek Hoiem. Category independent object proposals. In *European Conference on Computer Vision*, 2010. 2

[17] Hu et al. Bidirectional projection network for cross dimensional scene understanding. In *CVPR*, 2021. 8

[18] Kundu et al. Virtual multi-view fusion for 3d semantic segmentation. In *ECCV*, 2020. 8

[19] Wu et al. What can simple arithmetic operations do for temporal modeling? In *ICCV*, 2023. 8

[20] Wang et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *CVPR*, 2023. 6

[21] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88:303–308, 2009. Printed version publication date: June 2010. 5

[22] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023. 4

[23] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005. 2

[24] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2

[25] Stephen Gould, Tianshi Gao, and Daphne Koller. Region-based segmentation and object detection. In *NIPS*, 2009. 1, 2

[26] Grounded-SAM Contributors. Grounded-Segment-Anything. https://github.com/IDEA-Research/Grounded-Segment-Anything, 2023. 3

[27] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, Los Alamitos, CA, USA, 2020. IEEE Computer Society. 2

[28] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Doll'ar, and Ross B. Girshick. Masked autoencoders are scalable vision learners. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988, 2021. 2, 3

[29] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Geometric context from a single image. In *ICCV*, 2005. 1, 2

[30] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172, 2007. 1, 2

[31] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. 4

[32] Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality, 2023. 1

[33] Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. *arXiv preprint arXiv:2306.01567*, 2023. 2, 6

[34] S. Khorasgani, Y. Chen, and F. Shkurti. Slic: Self-supervised learning with iterative clustering for human action videos. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16070–16080, Los Alamitos, CA, USA, 2022. IEEE Computer Society. 1, 2, 3, 5

[35] Alchan Kim. FastSLIC optimized SLIC superpixel. https://github.com/Algy/fast-slic, 2019. 2

[36] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 2, 3, 5, 6

[37] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi. Fast superpixels using geometric flows. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, to appear. 2

[38] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 1

[39] Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard de Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. In *European Conference on Computer Vision*, pages 388–404. Springer, 2022. 5

[40] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 3

[41] Qi Lu, Jason Kuen, Shen Tiancheng, Gu Jiuxiang, Guo Weidong, Jia Jiaya, Lin Zhe, and Yang Ming-Hsuan. High-quality entity segmentation. In *ICCV*, 2023. 2

[42] Tomasz Malisiewicz and Alexei A. Efros. Improving spatial support for objects via multiple segmentations. In *BMVC*, 2007. 1, 2

[43] Greg Mori. Guiding model search using segmentation. In *ICCV*, 2005. 2

[44] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 1, 2, 3, 5, 6, 8

[45] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 5

[46] Curt Park. Segment anything with CLIP. https://github.com/Curt-Park/segment-anything-with-clip, 2023. 3

[47] PyTorch.org. Accelerating generative ai with pytorch: Segment anything, fast, 2023. https://pytorch.org/blog/accelerating-generative-ai/ [Accessed: (11/17/2023)]. 8

[48] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *ICCV*, 2007. 1, 2

[49] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2, 4, 5, 8

[50] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015. 2

[51] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *ICCV*, 2003. 2

[52] Bryan C. Russell, Alexei A. Efros, Josef Sivic, William T. Freeman, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006. 1, 2

[53] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8), 2000. 2

[54] Zhuowen Tu, Xiangrong Chen, Alan L. Yuille, and Song Chun Zhu. Image parsing: Unifying segmentation, detection, and recognition. *International Journal of Computer Vision*, 63(2):113–140, 2005. 2

[55] Jasper R.R. Uijlings, Koen E.A. van de Sande, Theo Gevers, and Arnold W.M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 2013. 2

[56] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009. 2

[57] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Neural Information Processing Systems (NeurIPS)*, 2021. 4

[58] Taojiannan Yang, Yi Zhu, Yusheng Xie, Aston Zhang, Chen Chen, and Mu Li. Aim: Adapting image models for efficient video action recognition. *arXiv preprint arXiv:2302.03024*, 2023. 5

[59] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight sam for mobile applications, 2023. 1, 2, 6

[60] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 5, 1

[61] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. 5, 1

[62] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision (ECCV)*, 2022. 2, 5

[63] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2