

Multi-scale and Cross-scale Contrastive Learning for Semantic Segmentation

Theodoros Pissas^{1,2}, Claudio S. Ravasio^{1,2},
 Lyndon Da Cruz^{3,4} *, and Christos Bergeles^{2*}

¹ Wellcome/EPSRC Centre for Interventional and Surgical Sciences, University College London (UCL)

² School of Biomedical Engineering & Imaging Sciences, King's College London (KCL)

³ Moorfields Eye Hospital, London

⁴ Institute of Ophthalmology, University College London
 rmaptpi@ucl.ac.uk

Abstract. This work considers supervised contrastive learning for semantic segmentation. We apply contrastive learning to enhance the discriminative power of the multi-scale features extracted by semantic segmentation networks. Our key methodological insight is to leverage samples from the feature spaces emanating from multiple stages of a model’s encoder itself requiring neither data augmentation nor online memory banks to obtain a diverse set of samples. To allow for such an extension we introduce an efficient and effective sampling process, that enables applying contrastive losses over the encoder’s features at multiple scales. Furthermore, by first mapping the encoder’s multi-scale representations to a common feature space, we instantiate a novel form of supervised local-global constraint by introducing cross-scale contrastive learning linking high-resolution local features to low-resolution global features. Combined, our multi-scale and cross-scale contrastive losses boost performance of various models (DeepLabv3, HRNet, OCRNet, UPerNet) with both CNN and Transformer backbones, when evaluated on 4 diverse datasets from natural (Cityscapes, PascalContext, ADE20K) but also surgical (CADIS) domains. Our code is available at https://github.com/RViMLab/MS_CS_ContrSeg.

Keywords: contrastive learning, segmentation, mutli-scale, cross-scale

1 Introduction

Supervised deep learning has driven remarkable progress in semantic segmentation, catalyzed by advances in convolutional network architecture design and the availability of large-scale pixel-level annotated datasets. Regarding the former, the standard paradigm involves convolutional encoders [12,30,34] to extract non-linear embeddings from images followed by a decoder that maps them

* The last two authors contributed equally.

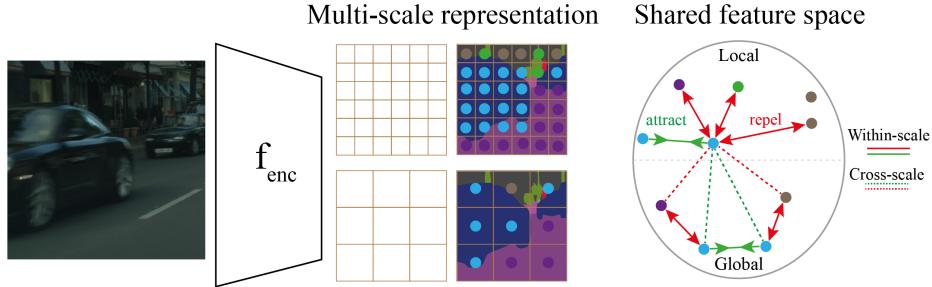


Fig. 1: **Key idea:** We leverage the multiscale representation provided by semantic segmentation encoders to propose a supervised-contrastive learning loss that is applied both at multiple scales and across them, within a shared feature space.

to a task-specific output space. For semantic segmentation, the seminal work of fully convolutional networks [28] demonstrated end-to-end learning of both these components by simply supervising the decoder with per-pixel classification losses. Further, considerable research has focused on designing inductive biases in convolutional architectures that enable complex, both local and global, visual relations across the input image to be encoded [43,5,6,44,34], leading to impressive results on challenging visual domains [9,50].

Recently, contrastive learning has been revisited as a tool for shaping network feature spaces under desired constraints over their samples, while avoiding representation collapse [35]. It has achieved strong results in unsupervised representation learning, and has established that contrastive pretraining over global [11,8,41,13] or dense [37] features can give rise to encoders that are on par with their supervised counterparts when fine-tuned on downstream tasks. This progress has motivated the incorporation of contrastive loss functions and training methodologies for the tasks of supervised classification and segmentation.

For the latter, recent works have shown that dense supervised contrastive learning applied over the final encoder layer can boost semantic segmentation performance when labelled examples are scarce [48], or in semi-supervised learning [1]. When using the full datasets, the method of [36] showed significant overall gains when enhanced with a class-wise memory bank maintaining a large diverse set of features during training. While recognizing and delineating objects at multiple scales is essential for segmentation, these methods directly regularize only the feature space of the final encoder layer thus operating at a single scale.

We instead apply multi-scale contrastive learning at multiple model layers. This direct feature-space supervision on early convolution/attention layers, usually learnt by back-propagating gradients all the way through many layers starting from the output space, can allow them to capture more complex relations between image regions, complementary to the usual function of early layers as local texture or geometry feature extractors. Intuitively, we treat the network as

a function that maps an image to a multi-scale representation, distributed across its different stages, and endow it with class-aware, feature-space constraints.

A second central element of contrastive approaches is the sampling of positive and negative samples to respectively attract and repel same-class embeddings. The usual sample generation mechanisms for unsupervised learning are augmentations, which as extensively discussed in [8,11,40], is a non-trivial and lengthy tuning step. Similarly, maintaining a class-wise memory bank, while providing a large set of examples, introduces hyper-parameters such as its size, update frequency and sample selection heuristic that can be dataset dependent [36].

We propose a simpler alternative to those options, by collecting samples from the feature spaces of multiple layers. Essentially, we leverage internal, intermediate information from the encoder that is available simply through the feed-forward step of the network without the need to resort to external steps such as data augmentation or online storing of samples via memory banks. Specifically, to find diverse *views* of the data, we sample them from within the model’s multi-scale feature spaces and link them both within and across scales (Fig 1). Conclusively, our contributions are the following:

- A batch-level hyper-parameter-free anchor sampling process that balances the contributions from frequent and rare classes while allowing efficient application of multiple contrastive loss terms.
- The introduction of supervised contrastive loss terms at multiple scales, and a novel cross-scale loss that enforces local-global consistency between high-resolution local and low-resolution global features emanating from different stages of the encoder.
- A model-agnostic overall method that can be successfully applied to a variety of strong baselines and state-of-the-art methods, comprising both transformer and CNN-based backbones, on 4 challenging datasets leading to improved performance.

Notably, on ADE20K, our method improves OCRNet [44] by **2.3%** and UPerNet with Swin-T and Swin-S, by **1.4%** and **1.3%**, respectively (for single scale evaluation). Further, on Cityscapes, our approach achieves an improvement of **1.1%** for both HRNet [34] and UPerNet with ResNet-101. Finally, on a challenging surgical scene understanding dataset, CaDIS, we outperform the state-of-art [25], especially improving rare classes by **1.2%**.

2 Related works

We outline connections between several research areas and our method and discuss its differences with existing methods.

Context aggregation for semantic segmentation: An intrinsic property of convolutional encoders is that information is aggregated from each pixel’s local neighborhood, the size of which can be expanded at the sacrifice of spatial feature resolution. The latter, however, is crucial for segmentation. The simple and effective aggregation approach of [28] underpinned extensive research into

more sophisticated context-aggregation mechanisms such as dilated-convolution-based approaches [5,43,7,6], attention [44,33], feature pyramid networks [39], or maintaining a high resolution representation throughout the network [34]. Recently, transformer and hybrid architectures have also been proposed. By design, these enable long-range context aggregation at the cost of increased computational requirements for training and inference [26,31]. Broadly, these approaches only support information aggregation from within a single input image. On the contrary, our method links features from different images during training, while being compatible with any architecture with hierarchical structure.

Contrastive learning is a feature learning paradigm under which, given a known assignment of samples into classes, the objective is to minimize the distance between samples of the same class while maximizing the distance between samples of different classes. From an information-theoretic perspective, minimizing an instantiation of this objective, termed the *InfoNce* loss [24], maximizes a lower bound of the mutual information (MI) between same class features. The same loss was used to learn a pretext task for unsupervised representation learning leading to impressive downstream task results [38,11,8,37,41], while in [19] it was shown to perform on par with standard cross-entropy for classification.

Supervised contrastive learning for segmentation: Concurrently with this work, improvements on strong baselines were demonstrated for segmentation by employing a **single** supervised contrastive loss term as an auxiliary loss applied after the inner-most layer of the encoder [36,15]. We instead explore **multiple** contrastive loss terms computed over different encoder layers, directly regularizing their feature spaces. Both [36,15] employ a memory bank used to maintain an extended set of pixel-embeddings or class-wise averaged region-embeddings. Instead, we opt for a simpler memory-less approach that avoids the need to tune memory size and memory update frequency, and instead collect samples from within and across different encoder layers. Further, while [36,15] focused on ResNet and HRNet, we demonstrate effective application of our method on a wider set of architectures and backbones, also exploring UPerNet and transformers. Finally, in [48], a similar contrastive loss term was used as a pretraining objective for DeepLabv3 leading to significant gains in performance when labelled examples are scarce. However, it provided small benefits when using the complete datasets, and required long ($300 - 600K$ steps) 2-phase training schedules, which is close to $3-6\times$ the training steps for our approach. Notably, in [48], contrasted feature maps had to spatially downsampled, for efficiency, which we circumvent by employing a balanced sampling approach.

Local-global representation learning: Contrastive learning with local and global representations of images has been studied for unsupervised pretraining. [14] proposes training an encoder by maximizing the MI between local and global representations to force the latter to compactly describe the shared information across the former. The method of [46] is similar; the maximization of MI between features encoding different local windows of a sentence, and global sentence embeddings, is used as a pretraining objective in natural language processing leading to improved downstream transfer. Moreover, in [2], the InfoNCE loss is

optimized over local and global features computed for two augmented views of an image, while in [4] it is applied on medical images separately over global features, and local features. In the latter, computation of the local loss term assumes the presence of common anatomical structures in medical scans to a relatively fixed position, an assumption invalid for datasets with diverse structures and scenes. There, strict prior knowledge on the location of an object/region is unrealistic. Finally, in [41] with the goal of pretraining for object detection, multiple InfoNCE objectives are optimized: globally, by extracting a feature vector for the whole image, locally by extracting multiple feature vectors each describing a single random patch of the same image, and across local-global levels by forcing whole-image and patch-level features of the same image to be similar.

Our approach is supervised, allowing to directly align the negative/positive assignment with the task of segmentation. We do not employ augmentation to obtain views of the data as in [2,4], as tuning it is laborious [8,40]. Instead, we leverage each dataset’s spatial class distribution as a source for diverse samples. Finally, our local-global loss term is better aligned to segmentation than [41]: instead of enforcing scale invariance of globally pooled encoder features, our loss is computed over dense features from multiple layers.

3 Method

We now provide an introduction to the InfoNCE loss function and the proposed multi-scale and cross-scale losses, as well as a sampling procedure to perform balanced and computationally feasible contrastive learning.

3.1 Preliminaries

Let $I \in \mathbb{R}^{H \times W \times 3}$ be an image, with H its height and W its width. Semantic segmentation frameworks usually comprise a backbone encoder and a segmentation head. The backbone encoder, $f_{enc} : \mathbb{R}^{H \times W \times 3} \mapsto \mathbb{R}^{h \times w \times c}$, maps the input image to a c -dimensional feature space. The segmentation head, $f_{seg} : \mathbb{R}^{h \times w \times c} \mapsto [0, 1]^{H \times W \times N_c}$, decodes the features $F = f_{enc}(I)$ to produce the output per-pixel segmentation $\hat{Y} = f_{seg}(F)$. These two modules are usually trained with a *cross-entropy* loss $\mathcal{L}_{ce}(\hat{Y}, Y)$, where Y denotes the ground truth per-pixel labels. Both f_{enc}, f_{seg} can be learned end-to-end with only supervision of f_{seg} . Under this training paradigm, the learning signal is restricted to unary classification errors.

An extension to this paradigm is to produce gradients that consider the distance of encoded image regions relative to other regions (not necessarily from the same image) in feature space. To directly (rather than implicitly via the decoder’s gradients) shape the encoder’s latent space, a supervised contrastive loss term [19] can be used. The loss forces features from image regions belonging to the same (different) class to be pushed close (apart). In this work, to identify the classes in each image, we use the labels downsampled to the spatial dimensions of the feature space, which we denote by \tilde{Y} . Given this class assignment, the InfoNCE [24] loss can be computed over a set of feature vectors, which is

usually termed *anchor* or *query* set; we denote this set by \mathcal{A} . For $\forall z_i \in \mathcal{A}$, there exist the sets of positives and negatives denoted by \mathcal{P}_i and \mathcal{N}_i respectively, which in the supervised setting that we examine, are identified according to the labels. The availability of dense rather than global features and labels allows a simple way to identify positive samples, referred also as “views”, without the need to craft a set of appearance and geometric perturbations as done in other unsupervised [8,11] and supervised approaches [48]. Thus, instead of requiring a dataset-specific data augmentation pipeline, we exploit the natural occurrences of same or different class-pixels across the scene and across different images.

$$\mathcal{L}_c(\mathcal{A}) = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \frac{1}{|\mathcal{P}(i)|} \sum_{j \in \mathcal{P}(i)} \mathcal{L}(z_i, z_j), \quad (1)$$

where

$$\mathcal{L}(z_i, z_j) = -\log \frac{\exp(z_i \cdot z_j / \tau)}{\exp(z_i \cdot z_j / \tau) + \sum_{n \in \mathcal{N}(i)} \exp(z_i \cdot z_n / \tau)} \quad (2)$$

This formulation is identical to the supervised contrastive loss for classification proposed in [19]. There, however, the choices of \mathcal{P}_i and \mathcal{N}_i are straightforward as each image in a batch has only a single global label. Crucially, as is standard practice in contrastive learning over convolutional feature maps [11,8], the loss is not directly computed over the encoder features F , but rather over a non-linear projection using a small FCN $f_{proj} : \mathbb{R}^{h \times w \times c} \mapsto \mathbb{R}^{h \times w \times d}$ such that $Z = f_{proj}(F)$. For the rest, z_i refers to a d -dimensional feature vector from the i -th spatial position of Z . We now motivate and describe the proposed anchor sampling process and the multi- and cross-scale loss terms.

3.2 Fully-dense contrastive learning

In the general case, \mathcal{A} can consist of all feature vectors from all spatial positions of F , the set of which is hereby denoted as Ω_F . For each element z_i of \mathcal{A} and knowing the downsampled labels \tilde{Y} , we select as $\mathcal{N}_i = \{z_j \in \mathcal{A} : j \neq i, \tilde{Y}(j) \neq \tilde{Y}(i)\}$ and $\mathcal{P}_i = \{z_j \in \mathcal{A} : j \neq i, \tilde{Y}(j) = \tilde{Y}(i)\}$. The computational complexity of this operation is quadratic in the spatial dimensions of the feature vector, i.e., $\mathcal{O}(h^2w^2)$. As most semantic segmentation methods, e.g. [7,34,44], require a small output stride for f_{enc} , it can become prohibitively expensive. Additionally, the quadratic complexity of this choice becomes even more prohibitive when introducing contrastive losses at multiple layers. Further, it is a well studied property of contrastive losses that the number [11,8] and the hardness [27,17,24] of the negatives affect the learned representations. Therefore, minimizing (1) over Ω_F can become trivial due to the consideration of many easy samples.

3.3 Anchor-set sampling

An alternative is to sample from Ω_F while maintaining a balanced number of feature vectors from each present class. We generate \mathcal{A} across a batch of images,

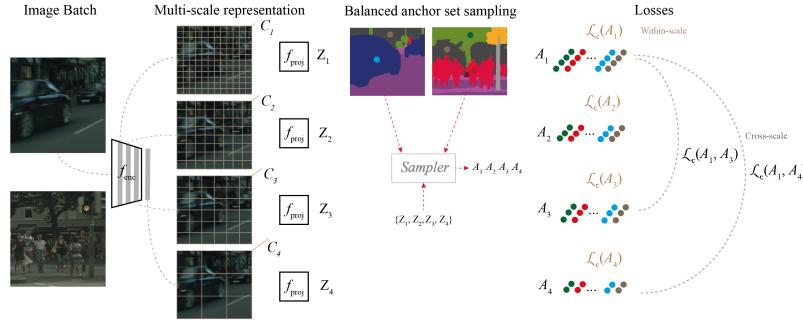


Fig. 2: Method overview: (1) An input batch is mapped to a multi-scale representation F_1, \dots, F_4 , each having varying spatial dimensions and channels C_1, \dots, C_4 , using the convolutional encoder. (2) Each dense feature is projected using a separate f_{proj} that preserves spatial dimensions but maps all features to a common d -dimensional space. (3) Each projected feature map Z_s undergoes a label-based balanced sampling process (Sec. 3.2) to produce an anchor set \mathcal{A}_s . (4) The feature vectors in \mathcal{A} and their class assignments are used to calculate the contrastive loss of (1) within each scale. Further, pairs of anchor sets from two different scales are utilized to compute a cross-scale loss (Sec. 3.5).

comparably to [36]. The intuition behind this is to further expand the diversity of samples considered by not restricting semantic relations between samples to span a single image. Rather, we allow samples to extend across different images.

Specifically, we sample K anchors per class present in the batch, equally divided among instances of that class. The sets of positives and negatives of each anchor are populated with elements of \mathcal{A} . This sampling process can be described as $\mathcal{A} \sim \{i \in \cup_{b=1}^B \Omega_F^{(b)}\}$, such that \mathcal{A} has at most A_{max} samples, with B being the batch size. Importantly and contrary to [36,48], K is selected on-the-fly rather than as a hyper-parameter, and is the number of samples from the class with the least occurrences in the batch. This is motivated by the observation that classes of small objects or regions will only occupy a small fraction of Ω_F , even more so due to the spatial stride of network encoders. This heuristic enables a balanced contribution of semantic classes in the loss and removes the requirement for tuning K . Additionally, it reduces the computational cost of each contrastive term enabling the multi-scale and cross-scale extensions described next.

3.4 Multi-scale contrastive loss

Having obtained an efficient way to compute the loss of (1), we extend it to multiple scales. This extension regularizes the feature space of different network layers, by pushing same-class features closer, and maximizes the MI of features and their semantic labels. Importantly, while applying a pixel-space classification

loss would also achieve the latter, the way we generate the anchor set allows us to attract features of the same class *from across different images*.

The hierarchical design of convolutional (ResNet [12], HRNet [34]) or transformer (Swin [20]) encoders provides a natural interface over which our loss can be applied. The stages followed are also outlined in Figure 2. We independently generate \mathcal{A}_s (Sec. 3.3) using the projection of features F_s , Z_s and labels \tilde{Y}_s at each scale s . The overall loss is computed as a weighted sum across scales:

$$\mathcal{L}_{cms} = \sum_{s=1}^S w_s \mathcal{L}_c(\mathcal{A}_s) \quad (3)$$

where the weights w_s control the contribution of each scale in the overall loss, and S is the number of different scales. As described in Section 3.3 generating \mathcal{A}_s involves randomization, thus it is ensured that the image regions involved in computing the above loss at each scale are independent.

3.5 Cross-scale contrastive loss

We further push same-class features from across different scales closer together. Specifically, we push high-resolution local features to be close to lower resolution global features. Given that global features encapsulate high level semantic concepts of the encoded image, guided by L_{ce} , we require that local features also encode those concepts by forcing them to lie close by in the projector’s feature space. Intuitively, this enables local features describing parts of objects/regions to be predicative of their global structure of the object and vice versa.

Importantly, we note that directly requiring that the features be similar across scales would be very hard to satisfy without causing collapse. The use of separate small non-linear convolutional projector at each scale (Figure 2) provides a compromise between a hard contrastive constraint on the encoder’s features and the lower dimensional common space spanned by f_{proj} wherein the cross-scale loss is calculated. Therefore, we compute it using the independently generated anchor sets \mathcal{A}_s and $\mathcal{A}_{s'}$, from scales s and s' :

$$\mathcal{L}_{ccs} = \sum_{(s,s')} w_{s,s'} \mathcal{L}_c(\mathcal{A}_s, \mathcal{A}_{s'}) \quad (4)$$

Negatives and positives for samples of one scale are collected from the anchor set of the other. Concretely, for each element $z_i \in \mathcal{A}_s$ we specify $\mathcal{N}_i = \{z_j \in \mathcal{A}_{s'} : j \neq i, \tilde{Y}_{s'}(j) \neq \tilde{Y}_s(i)\}$ and $\mathcal{P}_i = \{z_j \in \mathcal{A}_{s'} : j \neq i | j, \tilde{Y}_{s'}(j) = \tilde{Y}_s(i)\}$. Finally, the gradients derived from this loss are backpropagated to both involved anchor sets thus instantiating a form of bidirectional local-global consistency for learning the encoder. Combining all the above terms, our complete objective is:

$$\mathcal{L}_{total} = \mathcal{L}_{ce} + \lambda_{cms} * \mathcal{L}_{cms} + \lambda_{ccs} * \mathcal{L}_{ccs} \quad (5)$$

Table 1: (a),(c): Ablations for each component of our contrastive loss on **Cityscapes val** and **ADE20K**, respectively, reporting the mean and std deviation of each variant across 4 random seeds. (b), (d): Comparisons with auxiliary losses on **Cityscapes** (results taken from [36]) and **ADE20K**, respectively.

(a)						(b)		
Network	Loss			mIoU (ss)	Model	Loss	mIoU (ss)	
Model Backbone	\mathcal{L}_c	\mathcal{L}_{cms}	\mathcal{L}_{ccs}	Scale pairs	mean±std			
HRNet	HR48v2			-	79.1 ± 0.2	HRNet	CE	79.1
HRNet	HR48v2	✓		-	80.0 ± 0.1	HRNet	AFF [18]	78.7
HRNet	HR48v2		✓	-	80.7 ± 0.2	HRNet	RMI [47]	79.9
HRNet	HR48v2	✓	✓	1	81.2 ± 0.1	HRNet	Lovasz[3]	80.3
HRNet	HR48v2	✓	✓	2	81.4 ± 0.1	HRNet	ours	81.5

(c)						(d)		
Network	Loss			mIoU (ss)	Model	Loss	mIoU (ss)	
Model Backbone	\mathcal{L}_c	\mathcal{L}_{cms}	\mathcal{L}_{ccs}	Scale pairs	mean±std			
UPerNet	Swin-T			-	44.5 ± 0.4	OCRNet	CE	44.5
UPerNet	Swin-T	✓		-	45.1 ± 0.2	OCRNet	Lovasz	44.7
UPerNet	Swin-T	✓	✓	2	45.8 ± 0.1	OCRNet	ours	46.8

4 Experiments

4.1 Datasets

We benchmark our approach on the following challenging datasets from natural and surgical image domains using the mIoU as the main performance metric:

ADE20K [50] comprises 20210 and 2000 and train and val images, respectively, capturing 150 semantic classes from natural scenes.

Cityscapes [9] consists of 5000 images of urban scenes on which 19 classes are pixel-level labelled. We train, and evaluate, on the *train*, and *val* sets, respectively. We also report performance on the server-withheld *test* set.

Pascal-Context [23] comprises 4998 and 5105 and train and val images, respectively, capturing 59 classes from natural scenes.

CaDIS [10] comprises 25 cataract surgery videos and 4671 pixel-level annotated frames with labels for *anatomies*, *instruments* and *miscellaneous objects*. We experiment on tasks 2 and 3 defined in [10], comprising 17, 25 classes respectively. We follow [25] and also report the average of the per-class IoUs for anatomies, surgical tools and rare classes (present in less than 15% of the images).

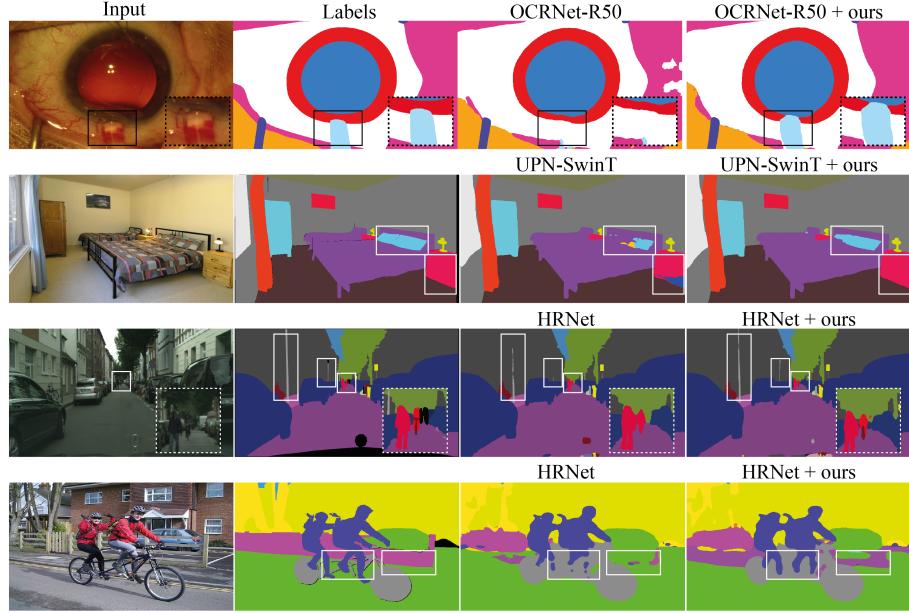


Fig. 3: **Qualitative results:** Qualitative comparisons across all 4 datasets. More results are provided in the supplementary material.

4.2 Ablations and comparisons to state-of-the-art

For ablations, we train for $40K$ steps with a batch size of 8 and for $160K$ steps with batch size of 16 on Cityscapes and ADE20K, respectively.

Ablation of loss components: First, we conduct an ablation study to demonstrate the importance of each component of our proposed loss on Cityscapes using HRNet, reported in Table 1(a). We repeat this ablation on ADE20K using UPerNet with a Swin-T backbone, shown in Table 1(c). We report the average mIoU across 4 runs with different random seeds to demonstrate the stability of the ranking of methods with respect to initialization and training stochasticity.

Comparisons to other auxiliary losses: Tables 1(b), 1(d) compare the proposed loss to other auxiliary losses on Cityscapes and ADE20K, respectively.

Comparisons to contrastive learning approaches: Table 2 compares our approach to concurrent contrastive learning methods for semantic segmentation.

Performance improvements: In Tables 3(a), 3(b), 4 and 5, we report performance improvements obtained by training a variety of models with our loss (Eq. 5) (referred to as "ours") on 4 datasets. We refer to cross-entropy as "CE" in tables. We experiment with DeepLabv3, HRNet, OCRNet and UPerNet with ResNet-101 and Swin (T,S,B,L) Transformer backbones. To train models using our proposed loss we leave all other settings the same as for the referenced or implemented baselines (i.e without our loss) to enable fair comparisons.

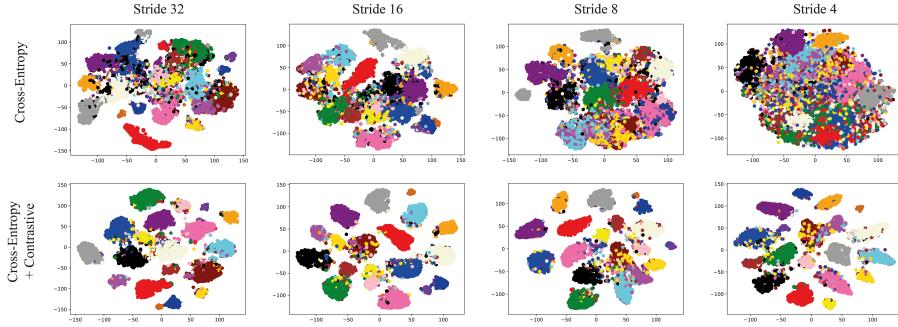


Fig. 4: TSNE [22] visualisation of the feature spaces of HRNet, on Cityscapes, trained without (top) and with (bottom) our proposed contrastive loss. Color indicates each sample’s class.

Table 2: Comparison with concurrent supervised contrastive learning methods for semantic segmentation on **Cityscapes val** with single scale evaluation. Our proposed loss matches the performance of those methods without using a memory bank and relying only on anchors sampled from multiscale features.

Network	Loss				mIoU	
	Model	Function	Memory	\mathcal{A} Selection	@40K	Best
HRNet (ours)	$\mathcal{L}_{cms}, \mathcal{L}_{ccs}$	-	Multi-scale Balanced		81.5	82.2
HRNet [36]	\mathcal{L}_c	Pixel	Hard-example	80.5	-	
HRNet [36]	\mathcal{L}_c	Region+Pixel	Hard-example	81.0	82.2	
HRNet [15]	\mathcal{L}_c+aux	Region	Averaging	-	81.9	

4.3 Implementation details

Contrastive Losses: All instantiations of the loss of Eq. (1) utilize a temperature $\tau = 0.1$. A distinct projector comprising 2 *Conv1x1-ReLU-BN* layers and a linear mapping with $d = 256$, is attached to features of each scale for multi-scale and cross-scale loss variants. We utilize the $C2$, $C3$, and $C5$ features, with output strides of 4, 8 and 8, respectively, for models with a dilated ResNet backbone. When using HRNet as the backbone, we utilize features from all 4 scales, with output strides of 4, 8, 16, 32. The cross-scale loss is applied for scale pairs (4, 32) and (4, 16). When using UPerNet we attach our loss on all 4 scales of the backbone on Cityscapes and on the feature pyramid net (FPN) outputs on ADE20K. The weights w_s in Eq. (3) are set to 1, 0.7, 0.4, 0.1 for feature maps of strides 4, 8, 16, 32 respectively in all experiments. The two latter choices are supported by ablations provided in the supplementary material.

Training settings: For experiments on all datasets/models, to enable fair comparisons we closely follow the training settings specified in [34, 44, 6, 20] and their respective official implementations. Unless otherwise stated, on Cityscapes for

CNN backbones we use SGD with a batch size of 12 for 120K steps while for transformer backbones we use AdamW [21] with a batch size of 8 for 160K steps. The crops size used is 512×1024 for all models. On ADE20K and Pascal-Context, the only differences are that all models are trained with a batch size of 16, a crop size of 512×512 , for 160K and 62K steps, respectively. On CaDIS and train for 50 epochs and use repeat factor sampling, following [25]. More detailed training settings are described in the supplementary material.

5 Results and Discussion

As shown in Tables 1(a) and 1(c), using both multi- and cross-scale contrastive terms provides the highest improvement relative to the baseline. Additionally, as shown in Tables 1(b) and 1(d), our loss outperforms other auxiliary losses.

Table 3: (a): Comparisons on **ADE20K** with single/multi-scale evaluation (ss/ms). If no reference is provided the value is obtained by our implementation and OHEM refers to the loss of [29]. (b): Comparisons on **CaDIS** tasks 2 and 3 that define 17 and 25 classes respectively. \dagger : Imagenet-22K pretraining.

(a) ADE20K

Model	Backbone	#Params (M)	Source	Loss		mIoU	
						ss/ms	Improvement
DeepLabv3	R101	63	[45]	CE		-/44.1	
DeepLabv3	R101	63	-	ours	44.2/45.6	(-/+ 1.5)	
DeepLabv3	R101	63	[15]	$\mathcal{L}_c + \text{Mem} + \text{OHEM}$		-/46.8	
OCRNet	HR48v2	71	[44]	OHEM	44.5/45.5		
OCRNet	HR48v2	71	-	ours	46.8/47.4	(+2.3/+ 1.9)	
UPerNet	R101	86	[39]	CE	42.0/42.7		
UPerNet	R101	86	-	ours	43.8/45.3	(+1.8/+ 2.6)	
UPerNet	Swin-T	60	-	CE	44.7/45.5		
UPerNet	Swin-T	60	[20]	CE	44.5/45.8		
UPerNet	Swin-T	60	-	ours	45.9/46.6	(+1.4/+ 0.8)	
UPerNet	Swin-S	81	-	CE	48.1/49.2		
UPerNet	Swin-S	81	[20]	CE	47.6/49.5		
UPerNet	Swin-S	81	-	ours	48.9/50.0	(+1.3/+ 0.5)	
UPerNet	Swin-B \dagger	121	[20]	CE	50.1/51.6		
UPerNet	Swin-B \dagger	121	-	ours	51.3/52.2	(+1.2/+ 0.6)	
UPerNet	Swin-L \dagger	234	[20]	CE	52.0/53.5		
UPerNet	Swin-L \dagger	234	-	ours	52.9/53.3	(+0.9/- 0.2)	

(b) CaDIS

Model	Backbone	Loss	mIoU		Anatomies		Tools		Rare	
			Task2	Task3	Task2	Task3	Task2	Task3	Task2	Task3
OCRNet	R50	CE	81.02	76.24	90.80	90.87	74.65	70.82	74.46	67.58
OCRNet	R50	+ours	81.47	77.67	90.66	90.91	75.58	72.79	77.63	73.09
Improvement			(+0.45)	(+1.43)	(-0.14)	(+0.04)	(+0.97)	(+1.97)	(+3.17)	(+5.51)
OCRNet	R50	Lovasz	82.36	77.77	90.63	90.59	76.89	72.96	77.52	71.44
OCRNet	R50	+ours	82.56	78.25	90.59	90.76	77.41	73.11	78.55	72.67
Improvement			(+0.20)	(+0.48)	(-0.04)	(+0.17)	(+0.52)	(+0.15)	(+1.03)	(+1.23)

Table 4: (a): Comparisons with strong baselines on **Cityscapes val** with single scale evaluation (ss). If no reference is provided the value is obtained by our implementation. The denoted improvements are relative to the baseline with the highest mIoU between our implementation and previously reported results. (b): Comparison on **Cityscapes test** with all models trained on **train+val**.

(a) Cityscapes val				(b) Cityscapes test							
Model	Network	Backbone	Source	Loss		mIoU (ss)		Improvement			
PSPNet	R101	[16]	Metric-learning	78.2							
DeepLabv3	R101	[6]	CE	77.8							
DeepLabv3	R101	-	CE	78.2							
DeepLabv3	R101	-	ours	79.0	(+0.8)						
HRNet	HR48v2	-	CE	81.0							
HRNet	HR48v2	[34]	CE	81.1							
HRNet	HR48v2	-	ours	82.2	(+1.1)						
HRNet	HR48v2	[36]	$\mathcal{L}_c + \text{Mem}$	82.2							
OCR	HR48v2	-	CE	81.2							
OCR	HR48v2	[44]	CE	81.6							
OCR	HR48v2	-	ours	81.9	(+0.3)						
UPerNet	R101	-	CE	78.0							
UPerNet	R101	-	ours	79.1	(+1.1)						
UPerNet	Swin-T	-	CE	79.2							
UPerNet	Swin-T	-	ours	79.9	(+0.7)						
UPerNet	Swin-S	-	CE	80.9							
UPerNet	Swin-S	-	ours	81.7	(+0.8)						
UPerNet	Swin-B	-	CE	82.0							
UPerNet	Swin-B	-	ours	82.6	(+0.6)						

Regarding **other supervised contrastive losses**, on **Cityscapes**, where we are able to provide a comparison between all methods, our loss is competitive, matching [36] and outperforming [15], despite not using a memory bank, hard-example mining or region averaging (Table 2). On **ADE20K**, [15] only report results with DeepLabv3 and achieve higher improvement over the baseline than ours (+2.7% vs +1.5%), albeit using both memory, OHEM and an intermediate auxiliary loss. Importantly, we showcase improvements for a wider range of models on this dataset. On **Pascal-Context**, we marginally outperform [36] when using HRNet and are outperformed when using OCRNet.

Notably, our loss improves performance for various CNN and Transformer-based models across datasets: On the challenging **ADE20K** dataset, (Table 3(a)), we obtain an improvement for OCRNet by +2.3% (ss) and +1.9% (ms), for UPerNet by +1.8% (ss) and +2.6% (ms) and for DeepLabv3 by +1.5% (ms). We also improve the state-of-the-art model of [20] using UPerNet with Swin Transformer for increasing backbone sizes (Swin-T,-S,-B, -L) by +1.4%, +1.3%, +1.2% and +0.9% (ss). Overall, with only a single-scale input, CNN models and Swin variants trained with our loss achieve performance close to that of the baseline when the latter employs the computationally expensive (several seconds per image) multiscale test-time augmentation.

On **Cityscapes-val** (Table 4), we improve both HRNet and UPerNet with ResNet101 by +1.1%, DeepLabv3 by +0.8% and UperNet with Swin-T and

Table 5: Comparisons on **Pascal-context val** with multi-scale evaluation (ms).

Model	Network	Backbone	Source	Loss		mIoU (ms)
HRNet	HR48v2	-		CE	53.5	
HRNet	HR48v2	[34]		CE	54.0	
HRNet	HR48v2	-	ours	55.3	(+1.3)	
HRNet	HR48v2	[36]	$\mathcal{L}_c + \text{Mem}$	55.1		
OCRNet	HR48v2	-		CE	55.8	
OCRNet	HR48v2	[44]		CE	56.2	
OCRNet	HR48v2	-	ours	56.5	(+0.3)	
OCRNet	HR48v2	[36]	$\mathcal{L}_c + \text{Mem}$	57.2		

Swin-S by **+0.7%** and **+0.8%** respectively. Our approach with HRNet and OCRNet, did not significantly improve mIoU on the **test** set but in both cases outperforms baselines reported in [34,44] in terms of **iIoU** and **iIoU-Cat** (Table 4(b)) by notable margins. This can be attributed to our sampling approach, that balances loss contributions from all present classes (Sec. 3.3) and the fact that our loss enhances the discriminative power of features at multiple scales, as exemplified in Figure 4.

On **Pascal-Context**, adding our loss to HRNet leads to **+1.3%** and a small improvement for OCRNet (Table 5) while doing so on the method of [25], on **CaDIS** results in state-of-the-art performance, especially favouring the rarest of classes, respectively for tasks 2 and 3, by **+1.0%** and **+1.2%**, when combined with the Lovasz loss, and by **+3.1%** and **+5.2%** when combined with CE. While the mean mIoU is a standard metric for assessing semantic segmentation, focusing it over the rarest classes is crucial to assess long-tailed class performance. This is especially important in the surgical domain where collecting data of rarely appearing tools, under real surgery conditions, can be particularly difficult.

6 Conclusion

We presented an effective method for supervised contrastive learning both at multiple feature scales and across them. Overall, we showcased significant gains for most of the strong CNN and Transformer-based models we experimented with on 4 datasets. Notably, our approach achieved maximal gains on the challenging ADE20K dataset, which contains a large number semantic concepts (150 classes), where the recognition component of segmentation greatly benefits from the class-aware clustered feature spaces of our method.

Acknowledgements: This work was supported by an ERC Starting Grant [714562].

Supplementary Material

A Selecting loss hyperparameters

The proposed loss (Eq. (5)) requires selecting certain hyperparameters, namely: the number of feature scales, the choice of cross-scale pairs, the per scale and overall weights of contrastive losses. Our results are obtained with minimal model or dataset-specific tuning of those parameters. Specifically, for **all models and datasets** we set both weights of Eq.(5) to 0.1 and use 2 scale-pairs (s4-s32, s4-s16) based on the results of the ablation in Tables 1(a) and 1(c) of the main paper. We further tested two different per-scale weight and cross-scale pair choices using a single model (HRNet) on Cityscapes (Table 1.b) and adopt per-scale weights as a decreasing function of the output stride. Finally, we tested two different alternatives regarding the position of the loss when using the UPerNet architecture, where the loss can be applied either on the FPN outputs or the directly over the backbones features. For Cityscapes the optimal choice is the latter while on ADE20K it is the former (Table 1(a)) and we adopt these choices for all other experiments on each dataset when using UPerNet. Thus, with minimal tuning our approach is effective while potentially further model- or dataset-specific tuning can boost performance even more.

Table 6: Ablation on (a) the position of application of the multi-scale and cross-scale losses for models using the UPerNet architecture and (b) on values of weights w_s of the multi-scale loss of Eq. (4).

(a)				(b)			
Network	Loss position	Dataset	mIoU (ss)	w_s	mIoU (ss)		
UPerNet R101	Backbone	CTS	79.1	1.0 1.0 1.0 1.0	81.8		
	FPN	CTS	78.4	1.0 0.7 0.4 0.1	82.2		
UPerNet Swin-S	Backbone	CTS	81.7				
	FPN	CTS	80.9				
UPerNet Swin-S	Backbone	ADE20K	47.9				
	FPN	ADE20K	49.0				

B Additional ablations

We report additional ablations regarding the effect of using longer training schedules and the importance of using the sampling process described in Section 3.3. As can be seen our method benefits by a longer training schedule and a bigger batch size while staying ahead of the baseline in all cases (Table 7(b)). Further, as shown in Table 7(a), our use of anchor sampling is necessary to allow an extension of contrastive losses to multiple scales as memory consumption exceeds our utilized hardware’s capacity ($4 \times 24\text{GB-GPUs}$). Further we find that even with a single contrastive loss term (\mathcal{L}_c)

our choice to perform anchor sampling results in better performance than using all available anchors in the batch which is equivalent to obtaining a number of anchors per class (denoted by \mathbf{K}) according to the class distribution p_{data} , which is imbalanced.

Table 7: (a) Comparison with alternative sampling options (40K steps, with a batch size of 8 and using 4). We denote the number of samples per class by \mathbf{K} . (b) Ablation of training schedules/batch sizes. All results are on **Cityscapes val** using single scale evaluation.

(a)							
Model	Scales	Scale Pairs	Loss	Sampling	\mathbf{K}	mIoU	Mem/GPU (GB)
HRNet	1	-	\mathcal{L}_c		$\sim p_{data}$	79.4	14.2
HRNet	1	-	\mathcal{L}_c	✓	Sec. 3.3	80.2	7.4
HRNet	4	2	$\mathcal{L}_{cms} + \mathcal{L}_{ccs}$		$\sim p_{data}$	-	OOM
HRNet	4	2	$\mathcal{L}_{cms} + \mathcal{L}_{ccs}$	✓	Sec. 3.3	81.5	9.7

(b)							
Network	Settings		mIoU				
	Model	Loss	Batch	40K	80K	120K	
HRNet	CE	8	8	79.1	79.7	80.5	
HRNet +ours		8		81.5	81.7	81.6	
HRNet	CE	12	-	-	-	81.0	
HRNet +ours		12		-	-	82.2	

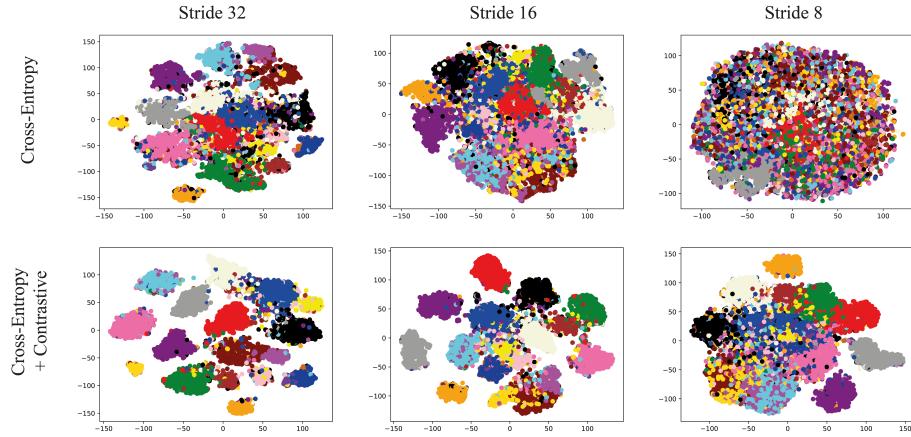


Fig. 5: TSNE [22] visualisation of the feature spaces of UPerNet with ResNet-101 backbone, on Cityscapes, trained without (top) and with (bottom) our proposed contrastive loss. Color indicates each sample’s class.

C Additional comparisons on ADE20K

We provide more comparisons between our results using UPerNet with Swin backbones and other state-of-the-art transformer models, on ADE20K. Notably, our result using Swin-B outperforms other competitive models despite having close to a third of the parameters in comparison to Segmener [32] and SETR [49].

Table 8: Additional results and comparisons with SOTA on **ADE20K val.**

Model	Backbone	#Params (M)	Source	Loss		mIoU
				ss/ms	Improvement	
UPerNet	Swin-B†	121	[20]	CE	50.1/51.6	
UPerNet	Swin-B†	121	-	ours	51.3/52.2 (+1.2/ + 0.6)	
UPerNet	Swin-L†	234	[20]	CE	52.0/53.5	
UPerNet	Swin-L†	234	-	ours	52.9/53.3 (+0.9/ - 0.2)	
SegFormer	MiT-B5	84	[42]	CE	51.1/51.8	
Segmenter	ViT-L/16†	307	[32]	CE	50.7/52.2	
SETR	T-Large†	310	[49]	CE	48.6/50.3	

D Qualitative results

We provide qualitative results of models trained with our proposed loss on ADE20K (Fig. 6), Cityscapes (Fig. 7) and CaDIS (Fig. 8). We also compare the feature spaces of UPerNet with ResNet-101 backbone, without and with our contrastive loss (Fig. 5).

E Training and testing settings

In Tables 10, 9, 11 and 12 we provide the settings used for our experiments. We closely follow each baseline’s implementation details found in its official code publication. Regarding testing, when multi-scale (flipping and scaling) inference is used, the scaling factors used are 0.5, 0.75, 1.25, 1.5, 1.75 on **ADE20K** and 0.5, 0.75, 1.25, 1.5, 1.75, 2.0 on **Cityscapes-test** and **Pascal-Context**.

Table 9: Training settings on **Cityscapes**.

Network		Settings					
Model	Backbone	crop	lr	decay	w_d	Batch/steps	optim
HRNet	HR48v2	512×1024	10^{-2}	poly	5×10^{-5}	12/120K	SGD
OCRNet	HR48v2	512×1024	10^{-2}	poly	5×10^{-5}	12/120K	SGD
DeepLabv3	R101	512×1024	10^{-2}	poly	5×10^{-5}	12/120K	SGD
UPerNet	R101	512×1024	10^{-2}	poly	5×10^{-5}	12/120K	SGD
UPerNet	Swin-T	512×1024	6×10^{-5}	linear	10^{-2}	8/120K	ADAMW
UPerNet	Swin-S	512×1024	6×10^{-5}	linear	10^{-2}	8/120K	ADAMW
UPerNet	Swin-B	512×1024	6×10^{-5}	linear	10^{-2}	8/120K	ADAMW

Table 10: Training settings on **ADE20K**.

Network		Settings					
Model	Backbone	crop	lr	decay	w_d	Batch/steps	optim
OCRNet	HR48v2	512×512	10^{-2}	poly	10^{-4}	16/160K	SGD
DeepLabv3	R101	512×512	10^{-2}	poly	10^{-4}	16/160K	SGD
UPerNet	R101	512×512	10^{-2}	poly	10^{-4}	16/160K	SGD
UPerNet	Swin-T	512×512	6×10^{-5}	linear	10^{-2}	16/160K	ADAMW
UPerNet	Swin-S	512×512	6×10^{-5}	linear	10^{-2}	16/160K	ADAMW
UPerNet	Swin-B	512×512	6×10^{-5}	linear	10^{-2}	16/160K	ADAMW
UPerNet	Swin-L	640×640	6×10^{-5}	linear	10^{-2}	16/160K	ADAMW

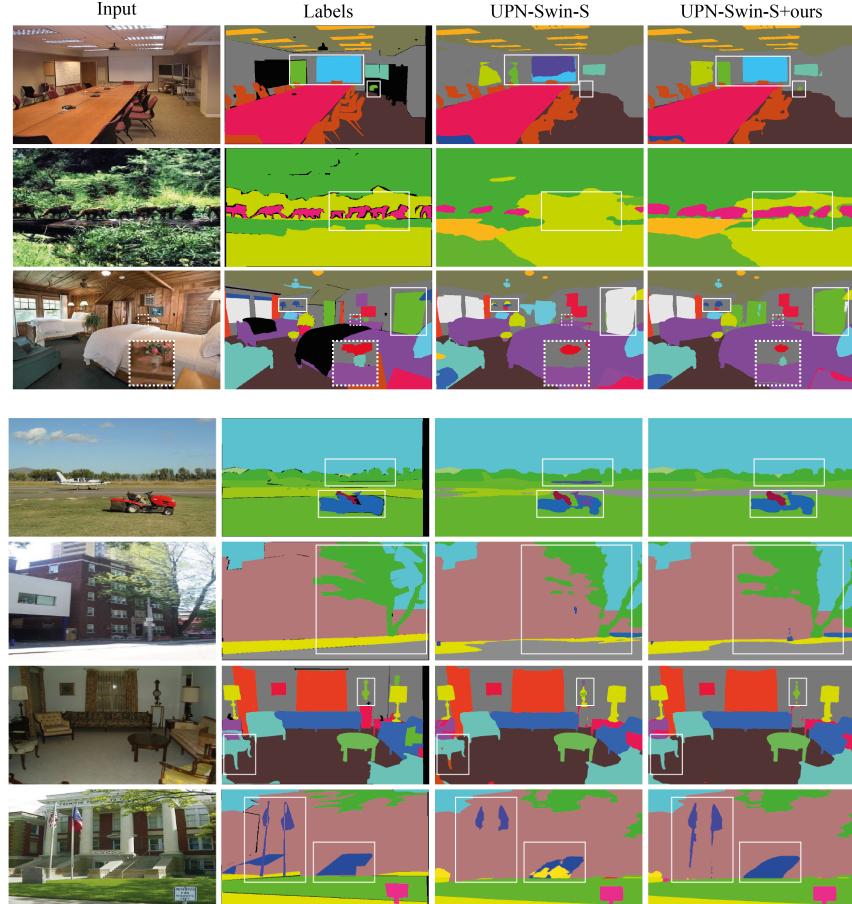


Fig. 6: Qualitative comparisons on ADE20K val: we compare UPerNet-Swin-S trained with only CE to the same model trained with also our multi- and cross-scale losses. White bounding boxes indicate some examples where our model performs better in cases where the foreground class is difficult to distinguish from the background (2nd, 5th row) or when it recognizes and segments smaller/thinner objects missed by the baseline (1st, 3rd, 6th, 7th)

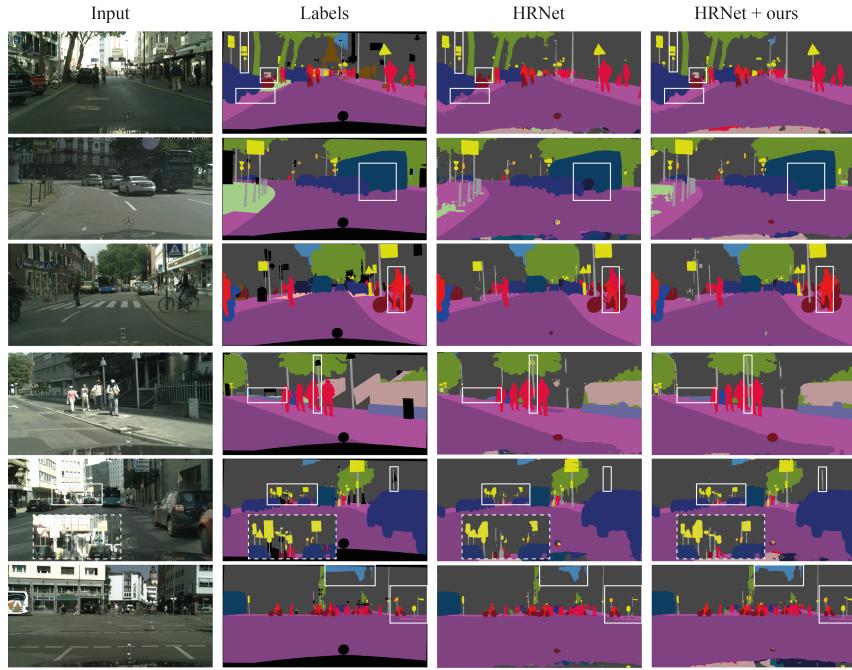


Fig. 7: Qualitative results on Cityscapes validation set: we present more qualitative results, comparing HRNet trained with CE to HRNet it trained with our multi- and cross-scale losses. White bounding boxes, outline some of the differences between the two models. Notably, the 2nd and 3rd rows depict cases where the baseline, misclassifies local segments of an object instance, namely a bus is partially recognized as "truck" and a bike rider is partially recognized as a simple pedestrian (i.e "person" in the dataset classes). Our model does not produce these inconsistencies in those cases, showcasing better ability to consider local-global interactions in recognizing and delineating an object instance. Other rows demonstrate examples where the model trained with our loss performs better than the baseline, at delineating small objects with fine details such as traffic signs or poles.

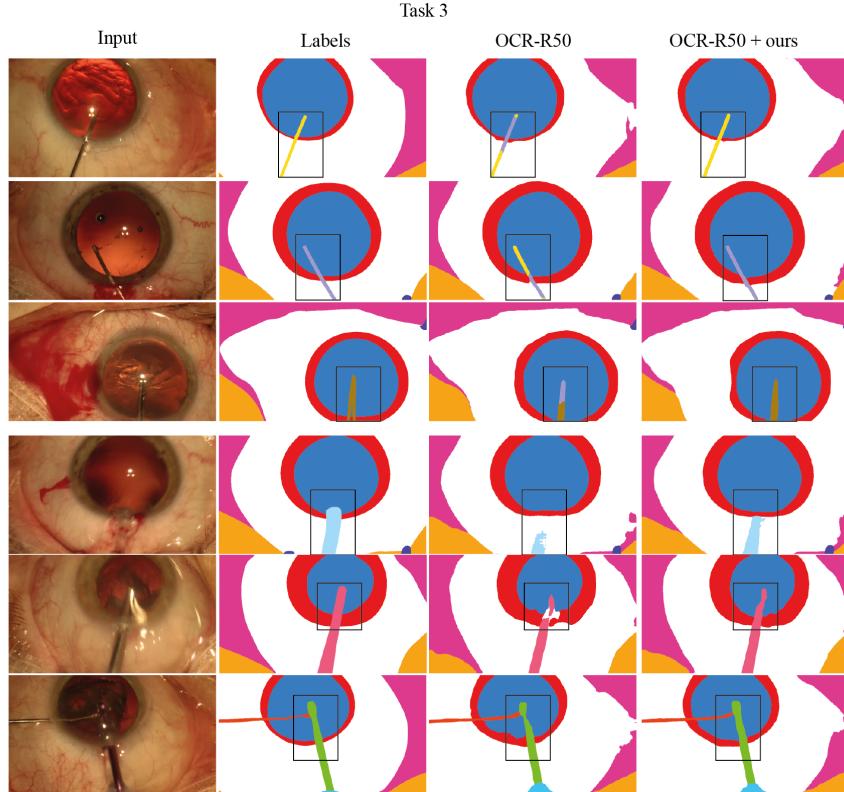


Fig. 8: Qualitative results on CaDIS validation set for task 3: We present a visual comparison of the baseline and the result of combining it with our multi- and cross-scale losses. Rows 1-3 demonstrate falsely recognized instrument classes by the baseline whereas our result accurately segments and classifies the tools. Notably, all 3 cases, correspond to tools that have very similar appearance but should be discriminated in task 3, that requires fine grained segmentation and classification. Further, rows 4-6 demonstrate, a barely humanly visible translucent tool and two blurry and specular images with tools, respectively. In all three cases our model achieves clearly more accurate delineation of the tools than the baseline, under challenging conditions.

Table 11: Training settings on **Pascal-Context**.

Network		Settings					
Model	Backbone	crop	lr	decay	w_d	Batch/steps	optim
OCRNet	HR48v2	512×512	10^{-3}	poly	10^{-4}	$16/160K$	SGD
HRNet	HR48v2	512×512	10^{-3}	poly	10^{-4}	$16/160K$	SGD

Table 12: Training settings on **CaDIS**.

Network		Settings					
Model	Backbone	crop	lr	decay	w_d	Batch/steps	optim
OCRNet	R50	540×960	10^{-4}	exp	-	$8/20K$	ADAM

References

1. Alonso, I., Sabater, A., Ferstl, D., Montesano, L., Murillo, A.C.: Semi-supervised semantic segmentation with pixel-level contrastive learning from a class-wise memory bank. In: Proceedings of the IEEE International Conference on Computer Vision (2021)
2. Bachman, P., Hjelm, R.D., Buchwalter, W.: Learning representations by maximizing mutual information across views (2019)
3. Berman, M., Triki, A.R., Blaschko, M.B.: The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
4. Chaitanya, K., Erdil, E., Karani, N., Konukoglu, E.: Contrastive learning of global and local features for medical image segmentation with limited annotations. Advances in Neural Information Processing Systems **33** (2020)
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs (2016)
6. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation (2017)
7. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. pp. 833–851. Springer International Publishing, Cham (2018)
8. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
9. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. CorR **abs/1604.01685** (2016), <http://arxiv.org/abs/1604.01685>
10. Grammatikopoulou, M., Flouty, E., Kadkhodamohammadi, A., Quellec, G., Chow, A., Nehme, J., Luengo, I., Stoyanov, D.: Cadis: Cataract dataset for image segmentation (2020)
11. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. arXiv preprint arXiv:1911.05722 (2019)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
13. Hénaff, O.J., Koppula, S., Alayrac, J.B., Oord, A.v.d., Vinyals, O., Carreira, J.: Efficient Visual Pretraining with Contrastive Detection. International Conference on Computer Vision (2021)
14. Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., Bengio, Y.: Learning deep representations by mutual information estimation and maximization. In: ICLR (2019)
15. Hu, H., Cui, J., Wang, L.: Region-aware contrastive learning for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 16291–16301 (October 2021)
16. Hwang, J.J., Yu, S.X., Shi, J., Collins, M.D., Yang, T.J., Zhang, X., Chen, L.C.: Segsort: Segmentation by discriminative sorting of segments. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 7334–7344 (2019)

17. Kalantidis, Y., Sariyildiz, M.B., Pion, N., Weinzaepfel, P., Larlus, D.: Hard negative mixing for contrastive learning. CoRR **abs/2010.01028** (2020), <https://arxiv.org/abs/2010.01028>
18. Ke, T.W., Hwang, J.J., Liu, Z., Yu, S.X.: Adaptive affinity fields for semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV) (September 2018)
19. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. arXiv preprint arXiv:2004.11362 (2020)
20. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 10012–10022 (October 2021)
21. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
22. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
23. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
24. van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. CoRR **abs/1807.03748** (2018), <http://arxiv.org/abs/1807.03748>
25. Pissas*, T., Ravasio*, C.S., Da Cruz, L., Bergeles, C.: Effective semantic segmentation in cataract surgery: What matters most? In: de Bruijne, M., Cattin, P.C., Cotin, S., Padoy, N., Speidel, S., Zheng, Y., Essert, C. (eds.) Medical Image Computing and Computer Assisted Intervention – MICCAI 2021. pp. 509–518. Springer International Publishing, Cham (2021)
26. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 12179–12188 (October 2021)
27. Saunshi, N., Plevrakis, O., Arora, S., Khodak, M., Khandeparkar, H.: A theoretical analysis of contrastive unsupervised representation learning. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 5628–5637. PMLR (09–15 Jun 2019), <https://proceedings.mlr.press/v97/saunshi19a.html>
28. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(4), 640–651 (2017). <https://doi.org/10.1109/TPAMI.2016.2572683>
29. Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Proceedings of the International Conference on Learning Representations (2015)
31. Strudel, R., Garcia, R., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7262–7272 (October 2021)

32. Strudel, R., Garcia, R., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7262–7272 (October 2021)
33. Takikawa, T., Acuna, D., Jampani, V., Fidler, S.: Gated-scnn: Gated shape cnns for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019)
34. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., Xiao, B.: Deep high-resolution representation learning for visual recognition. TPAMI (2019)
35. Wang, T., Isola, P.: Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 9929–9939. PMLR (13–18 Jul 2020), <https://proceedings.mlr.press/v119/wang20k.html>
36. Wang, W., Zhou, T., Yu, F., Dai, J., Konukoglu, E., Van Gool, L.: Exploring cross-image pixel contrast for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7303–7313 (October 2021)
37. Wang, X., Zhang, R., Shen, C., Kong, T., Li, L.: Dense contrastive learning for self-supervised visual pre-training. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR) (2021)
38. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
39. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. pp. 432–448. Springer International Publishing, Cham (2018)
40. Xiao, T., Wang, X., Efros, A.A., Darrell, T.: What should not be contrastive in contrastive learning. In: International Conference on Learning Representations (2021), <https://openreview.net/forum?id=CZ8Y3NzuVz0>
41. Xie, E., Ding, J., Wang, W., Zhan, X., Xu, H., Sun, P., Li, Z., Luo, P.: Detco: Unsupervised contrastive learning for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 8392–8401 (October 2021)
42. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. In: Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems (2021), <https://openreview.net/forum?id=0G18MI5TRL>
43. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: Computer Vision and Pattern Recognition (CVPR) (2017)
44. Yuan, Y., Chen, X., Wang, J.: Object-contextual representations for semantic segmentation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 173–190. Springer International Publishing, Cham (2020)
45. Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Zhang, Z., Lin, H., Sun, Y., He, T., Muller, J., Manmatha, R., Li, M., Smola, A.: Resnest: Split-attention networks. arXiv preprint arXiv:2004.08955 (2020)
46. Zhang, Y., He, R., Liu, Z., Lim, K.H., Bing, L.: An unsupervised sentence embedding method by mutual information maximization. EMNLP (2021)

47. Zhao, S., Wang, Y., Yang, Z., Cai, D.: Region mutual information loss for semantic segmentation. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019), <https://proceedings.neurips.cc/paper/2019/file/a67c8c9a961b4182688768dd9ba015fe-Paper.pdf>
48. Zhao, X., Vemulapalli, R., Mansfield, P.A., Gong, B., Green, B., Shapira, L., Wu, Y.: Contrastive learning for label efficient semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 10623–10633 (October 2021)
49. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., Zhang, L.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6881–6890 (June 2021)
50. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. International Journal of Computer Vision **127**, 302–321 (2018)