This WACV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Multi-Scale Grouped Prototypes for Interpretable Semantic Segmentation

Hugo Porta[1] Emanuele Dalsasso[1] Diego Marcos[2,3] Devis Tuia[1]
[1]EPFL [2]Inria [3]University of Montpellier
{hugo.porta, emanuele.dalsasso, devis.tuia}@epfl.ch
diego.marcos@inria.fr

Abstract

Prototypical part learning is emerging as a promising approach for making semantic segmentation interpretable. The model selects real patches seen during training as prototypes and constructs the dense prediction map based on the similarity between parts of the test image and the prototypes. This improves interpretability since the user can inspect the link between the predicted output and the patterns learned by the model in terms of prototypical information. In this paper, we propose a method for interpretable semantic segmentation that leverages multi-scale image representation for prototypical part learning. First, we introduce a prototype layer that explicitly learns diverse prototypical parts at several scales, leading to multi-scale representations in the prototype activation output. Then, we propose a sparse grouping mechanism that produces multi-scale sparse groups of these scale-specific prototypical parts. This provides a deeper understanding of the interactions between multi-scale object representations while enhancing the interpretability of the segmentation model. The experiments conducted on Pascal VOC, Cityscapes, and ADE20K demonstrate that the proposed method increases model sparsity, improves interpretability over existing prototype-based methods, and narrows the performance gap with the non-interpretable counterpart models. Code is available at github.com/eceo-epfl/ScaleProtoSeg.

1. Introduction

In the last few years, deep learning-based semantic segmentation has seen rapid adoption in numerous fields, from industrial use cases such as autonomous driving [22, 74, 81] to the environmental sciences [5, 48, 65]. This expansion was driven by its increasing performance on a multitude of tasks and benchmarks, often coinciding with an increase in model complexity [11, 64, 86, 89], which favors un-interpretable black-box models, to the detriment of their explainability.

The models' lack of interpretability is particularly harmful in high-stakes applications [68, 82] and sometimes prevents the wider adoption of deep learning models in applied fields involving high-stakes decision making [26, 63, 76]. Indeed, deep learning models can rely on spurious cues, such as Clever-Hans predictors, often symptoms of data contamination [2, 49]. This is detrimental in real use cases due to the potential lack of generalization and opaqueness of the decision process. Issues in the generalization of deep learning models are also illustrated by adversarial examples, which can be engineered via small, imperceptible perturbations of images [47] or be inherent in natural images [34].

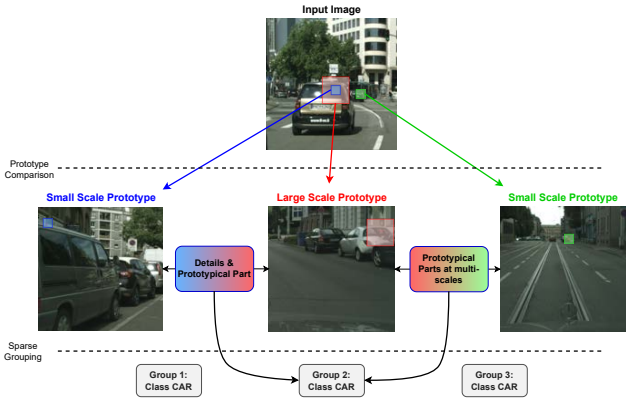


Figure 1. ScaleProtoSeg learns scale-specific prototypes at multiple scales and a sparse prototype grouping to extract patterns referring to different levels of details or scales.

The field of eXplainable Artificial Intelligence (XAI) aims at alleviating the risks associated with the lack of model interpretability by presenting some aspects of the models' decision process into a form understandable to humans [20]. To produce explainable results, the model design often requires some simplifications that reduce the performance of the original, more complex black-box counterpart [53]. Most methods focus on classification and regression tasks [3, 27, 44, 46, 72], including prototype-based approaches [8]. A few works aim at making semantic segmentation models interpretable [35, 70, 71, 83]. However, no approach considers questions about the relationship between prototypes across semantic classes and scales, despite instances of objects appearing at different positions in the

image or at multiple range of distances. Accounting for these redundancies allows for learning more diverse prototypes for a given object across scales, in turn leading to more explicit interpretability [57].

In this paper, we tackle these gaps and propose a method for multi-scale interpretable semantic segmentation, ScaleProtoSeg (see Figure 1), that *(i)* explicitly learns prototypes at several scales and *(ii)* groups the scale-specific prototypes thanks to a sparse grouping mechanism that provides information on the interaction between prototypical parts at multiple scales, while reducing the number of active prototypes contributing to the decision. Multi-scale prototype learning disentangles the information at different scales so that the model and the users have access to different levels of contextual information in the interpretable decision process. The sparse grouping mechanism allows a transparent understanding of the interaction of scale-specific representations such as object details and parts or similar parts at multi-scale (see Figure 1), while maintaining parts correspondences via regularization, avoiding altogether prototype pruning. We are the first to jointly leverage multi-scale representation and prototype learning in an interpretable semantic segmentation model.

We test our methods on three semantic segmentation benchmarks: Pascal VOC 2012 [21], Cityscapes [14], and ADE20K [100], by considering DeepLabv2 [9] as our base model architecture. We first show that multi-scale prototype learning improves the performance of single-scale prototype-based interpretable semantic segmentation methods (with similar amounts of prototypes) across all considered benchmarks. Moreover, thanks to the sparse grouping mechanism, we demonstrate that constraining the decision process to a small group of prototypes per class enforces interpretability while retaining competitive performance. The contributions of our paper can be summarized as follows:

- we propose a multi-scale prototype layer that enforces the model to focus on the prototypical parts' representations at multiple scales;
- we define a grouping procedure that learns sparse combinations of the scale-specific prototypes across all scales and increases the interpretability of the decision process;
- not only we show the superiority of our ScaleProtoSeg method in three popular datasets in semantic segmentation over the prototype-based method [70], but also highlight its improved interpretability measured in terms of *stability*, *consistency* and *sparsity*.

2. Related works

Semantic segmentation. Fully Convolutional Networks (FCN) [55] are widely used in semantic segmentation methods. They are based on an encoder-decoder architecture, where the encoder extracts discriminative features from the input image and the decoder converts the learned semantic representation into per-pixel predictions. Following FCN, researchers focused on improving different aspects of the semantic segmentation methods such as enlarging the model receptive field while limiting the parameters increase [10,94,96], specifying boundary information [17,78,93], or providing contextual information [32, 52, 95]. Some methods proposed specific modules learning pixel affinities or attention [23,38,54,97] to allow the network to base its prediction also on similar pixels that do not lie in the direct vicinity of the pixel at hand. Furthermore, there has been a growing interest in how multi-scale information could be extracted. A common pattern in decoder architectures is the concatenation of scale-specific feature maps at multiple scales [64, 67, 89]. For instance, Chen et al. [9] introduce atrous spatial pyramid pooling (ASPP) to learn in parallel a multi-scale representation from a single-scale feature map. We focus on providing an interpretable version of this model relying on its widely employed decoder multi-scale architecture.

Explainable artificial intelligence. XAI methods can be split into *post-hoc* vs *by-design* approaches. Post-hoc methods aim to explain black-box models after training by using an auxiliary method to generate explanations, while by-design approaches enforce interpretability in the model itself. Our proposed method falls into the latter category. Some examples of interpretable by-design approaches are concept bottleneck models [43, 46, 57, 62]; attention modules [66, 73, 98, 99] which point to critical parts of each input sample; generalized additive models [3, 31]; and prototype learning introduced in [8, 51], where part of an encoded input image is compared to a set of class-specific prototypes, represented by training samples. Several extensions followed the original paper [8], aiming at enforcing orthogonality in the prototype construction [19, 84], reducing the number of prototypes [69], or even leveraging label taxonomy via a hierarchical structure [30]. Prototype learning can be extended to other tasks beyond image classification such as sequence learning [60] and time series analysis [25] and, most recently, semantic segmentation [70]. In this work, we rely on prototype learning for interpretable semantic segmentation.

Interpretable semantic segmentation. Among the few existing methods tackling this problem, extensions of Grad-CAM to semantic segmentation have been explored [35,83] for post-hoc methods. By-design approaches can provide explainable results by leveraging symbolic language [71], through the use of a semantic bottleneck [56] or by exploiting the attention mechanism [28]. Prototypical parts learn-

ing was proposed recently in ProtoSeg [70] for interpretable semantic segmentation, extending the classification-based method proposed in [8]. In this work, we aim to investigate multi-scale prototype learning to break the performance/interpretability trade-off present in ProtoSeg. This is achieved by explicitly leveraging the multi-scale nature of semantic segmentation representations (via scale-specific prototypes) and by encouraging sparsity (via sparse grouping across prototypes and scales).

Prototypes and semantic segmentation. Prototype learning based on parametric or non-parametric prototypes has seen an increase of interest in image classification [59, 92], few-shot and zero-shot [42, 75, 90], unsupervised [88] and self-supervised learning [50]. For semantic segmentation, parametric prototypes are used to represent unique class embeddings, both in an explicit [12, 40, 77] or implicit way [101]. Non-parametric prototypes were also leveraged, first in the few-shot learning context [18, 85] and then in more classic semantic segmentation models [39, 87, 101]. The methods in [87, 101] are the closest to our proposition: both compute multiple class-specific prototypes via online clustering and leverage metric learning losses to enforce a compact embedding space. The pixel-wise classification is then performed via nearest-prototype assignment. Despite those similarities, those works compute prototypes in the final embedding space, while we focus on enhanced interpretability via i) prototypical parts through specific regularization and ii) multi-scale object representation. These core differences limit the applicability of direct performance comparisons. Finally, [79, 80] leverage non-parametric prototypes for modality and domain alignment for multi-modal semantic segmentation (e.g. video and point clouds), but do not explore issues related to interpretability or multi-scale prototype learning as our proposed method.

3. Method

In this section, we present our multi-scale grouped prototypes method for interpretable semantic segmentation: ScaleProtoSeg (Figure 2). We first introduce the multi-scale prototype architecture (Section 3.1). Then we describe the proposed grouping mechanism to extract sparse groups of prototypes across scales (Section 3.2). Lastly, we detail the multi-stage training procedure used to learn both prototypes and groups (Section 3.3).

3.1. Multi-scale prototype learning

Our model architecture for multi-scale prototype learning is presented in Figure 2: **Stage 1**. It is composed of a backbone network f, a multi-scale prototype layer g_{proto}, and a linear layer h_{proto}. For an input RGB image $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$, f outputs a multi-scale feature map $f(\mathbf{x}) \in \mathbb{R}^{H_r \times W_r \times S \times d}$, representing scale-specific features at S scales, each one with d dimensions. The scalar r is a size reduction factor. In practice, we modify the ASPP layer from [9] to concatenate the scale-specific feature maps instead of summing them. At each scale $s \in \mathcal{S}$, let $\mathbf{z}_s \in \mathbb{R}^d$ be a vector from the scale-specific feature map $f_s(\mathbf{x})$. As illustrated in Figure 2, this vector represents the features extracted from an area of the input image corresponding to the receptive field at scale s. The multi-scale prototype layer g_{proto} is composed of M learnable scale-dependent prototypes, where the m_{th} scale-dependent prototype (with $\mathrm{m} \in [1, \dots, M]$) is described by the vector $\mathbf{p}_{s,\mathrm{m}} \in \mathbb{R}^d$, which is randomly initialized and learned through gradient descent. For each feature vector $\mathbf{z}_s$, the prototypes' activations are computed following ProtoPNet [8]:

$$g_{\text{proto}}(\mathbf{z}_s, \mathbf{p}_{s,\mathrm{m}}) = \log\left(\frac{\|\mathbf{z}_s - \mathbf{p}_{s,\mathrm{m}}\|_2^2 + 1}{\|\mathbf{z}_s - \mathbf{p}_{s,\mathrm{m}}\|_2^2 + \epsilon}\right) \quad (1)$$

with $\epsilon \ll 1$ a constant for numerical stability. Then, for each scale-specific feature vector $\mathbf{z}_s$, the M activation scores $[g_{\text{proto}}(\mathbf{z}_s, \mathbf{p}_{s,1}), \dots, g_{\text{proto}}(\mathbf{z}_s, \mathbf{p}_{s,M})]$ are concatenated across the S scales indexed by s. The linear layer h_{proto}, with weight matrix $\mathbf{w}_{h_{\text{proto}}} \in \mathbb{R}^{C \times M \cdot S}$, learns a mapping from those prototype activations to the C output classes probabilities. The classification probabilities map, obtained after processing all feature vectors $\mathbf{z} = [\mathbf{z}_1, \dots, \mathbf{z}_S]$ in parallel, is of dimension $H_r \times W_r \times C$. To upscale it to the original image resolution and produce the final segmentation map, the classification probability is linearly interpolated.

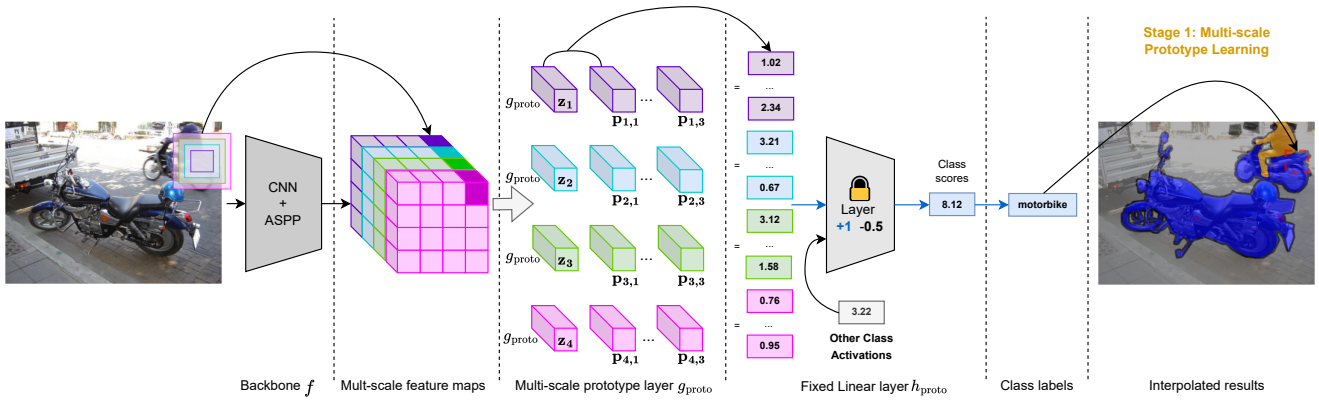
For semantic segmentation, the objective is to learn prototypes that are assigned to a specific class $c \in \mathcal{C}$. This is enforced by initializing the weights $\mathbf{w}_{h_{\text{proto}}}$ of the linear layer h_{proto} following [8], and replicating the same assignment to a specific class across all scales, such as with:

$$w_{h_{\text{proto}}}^{(c,\mathrm{m})} = \begin{cases} 1 & \text{if } \mathbf{p}_{\mathrm{m}} \in P_c \\ -0.5 & \text{otherwise} \end{cases}$$

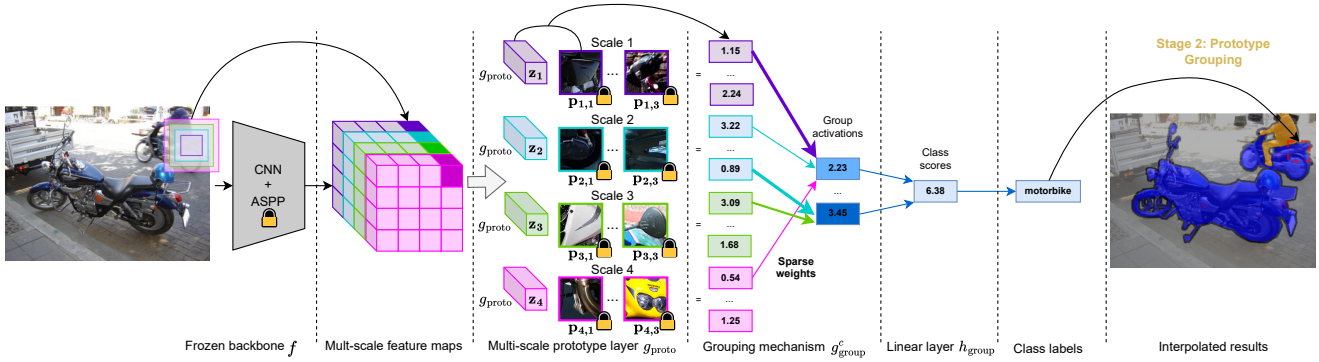
where P_c is the set of prototypes that we assign to class $c \in \mathcal{C}$, across all scales and $\mathrm{m} \in [1, \dots, |P_c|]$. Those weights are frozen for the majority of the training procedure (see Section 3.3) to maintain the steering of the prototypes toward class-specific patterns.

3.2. Prototype grouping

Our model architecture for prototype grouping is presented in Figure 2: **Stage 2**. Once the scale-specific prototypes $\mathbf{p}_{s,\mathrm{m}}$ are learned through the multi-scale learning stage as described in Section 3.1, we group them into sparse groups across scales. For this purpose, we use class-specific grouping functions: $g_c(\mathbf{z}) = g_{\text{group}}^c(g_{\text{proto}}(\mathbf{z}, P_c))$, which group prototypes assigned to the same class and compute



(a) **Stage 1:** Multi-scale prototype learning.



(b) **Stage 2:** Prototype grouping.

Figure 2. Overall architecture of ScaleProtoSeg. Each color in the feature maps and following layers corresponds to a specific scale ($S = 4$ and $M = 3$ in this illustration).

the groups' activations $g_c(\mathbf{z}) \in \mathbb{R}^{N_c}$, where N_c is the number of groups per class. Following [57], those functions are parametrized by sparse non-negative weight matrices $\mathbf{w}_{g,c} \in [0,1]^{N_c \times |P_c|}$, where each row is constrained on the probability simplex: $w_{g,c}^{(\mathrm{n,m})} \geq 0$ and $\sum_{\mathrm{m}} w_{g,c}^{(\mathrm{n,m})} = 1, \forall \mathrm{n} \in [1, \ldots, N_c]$. The output group activation for the group $g_{c,\mathrm{n}}(\mathbf{z})$ that combines the prototypes $\mathbf{p}_{\mathrm{m}} \in P_c$ at multiple scales is computed as follows:

$$g_{c,\mathrm{n}}(\mathbf{z}) = \prod_{\mathrm{m}=1,\ldots,|P_c|} g_{\text{proto}}(\mathbf{z}, \mathbf{p}_{\mathrm{m}})^{w_{g,c}^{(\mathrm{n,m})}}. \qquad (2)$$

The projection on the simplex and the weighted geometric mean are used to learn sparse weight matrices, as group activations can be high only if all prototypes in the group are strongly activated. The linear layer h_{group} is adapted to accommodate groups through a weight matrix $\mathbf{w}_{h_{\text{group}}} \in \mathbb{R}^{C \times N}$, where $N = N_c \times C$ is the total number of groups, each one containing a sparse combination of the set of $M \times S$ prototypes. The layer h_{group} follows the same initialization process as in Section 3.1, but with group assignment.

3.3. Multi-stage training procedure

In order to train ScaleProtoSeg, we resort to a two-stage training procedure: first, we learn the multi-scale prototypes and project them to the training set patches, without any grouping. Then we learn the prototypes grouping functions with the prototypes fixed. The two stages are illustrated in Figure 2 and detailed below.

Stage 1: Multi-scale prototype learning. For the multi-scale prototype learning we apply a training procedure similar to [8, 70], which consists of three steps. First, in a warm-up step, the ASPP [9] and the scale-specific prototypes are trained while freezing the rest of the backbone f and the last linear layer h_{proto}. Second, we run a joint optimization stage where all the model is trained except the last linear layer h_{proto}. Third, for all scales $s \in \mathcal{S}$, the prototypes are projected to their nearest vector $\mathbf{z}_s$ from the training set, and duplicates are removed. The fine-tuning stage from [70] is not necessary for ScaleProtoSeg, as we replace the last layer h_{proto} with h_{group} in the second stage below. Moreover, contrary to [8, 70], we do not need to run their pruning algorithm, as the sparse grouping mechanism will also enforce a natural decrease in the number of prototypes used.

In all those steps, we apply as a regularization loss the diversity loss [70], which aims to prevent the prototypes from activating the same region of an object. To enforce diversity among scale-specific prototypes, the diversity loss is evaluated independently at each scale $s \in \mathcal{S}$. Indeed, the multi-scale prototype layer aims to represent similar parts across scales with different contextual information. For each class $c \in \mathcal{C}$ and scale $s \in \mathcal{S}$, we note $P_{s,c}$ as the set of prototypes assigned to c and s. Furthermore, for the scale-specific feature map $f_s(\mathbf{x})$ we note $y_{\mathbf{z}} \in \{1, \dots, C\}$ as the ground truth labels for each vector $\mathbf{z}$ across all scales.
First, for the diversity loss, we define $v(f_s(\mathbf{x}), \mathbf{p}_{s,\mathrm{m}})$ as the softmax vector of distances between each vector assigned to a class c, and a prototype $\mathbf{p}_{s,\mathrm{m}} \in P_{s,c}$:

$$\begin{aligned} v(f_s(\mathbf{x}), \mathbf{p}_{s,\mathrm{m}}) =& \text{softmax}(\|\mathbf{z}_s - \mathbf{p}_{s,\mathrm{m}}\|^2 : \\ & \forall \mathbf{z}_s \in f_s(\mathbf{x}), y_{\mathbf{z}} = c) \end{aligned} \tag{3}$$

Next we use the Jeffrey similarity $S_J(U_1, \dots, U_l)$ as a measure of similarity between distributions $(U_1, \dots, U_l)$, following ProtoSeg [70]:

$$S_J(U_1, \dots, U_l) = \frac{1}{\binom{l}{2}} \sum_{i<j} \exp(-D_J(U_i, U_j)) \tag{4}$$

with $D_J(U, V)$ the Jeffrey's divergence defined in [41]. The diversity loss is then computed as follows for a given class c and scale s:

$$\begin{aligned} L_J(f_s(\mathbf{x}), P_{s,c}) =& S_J(v(f_s(\mathbf{x}), \mathbf{p}_{s,1}), \dots, \\ & v(f_s(\mathbf{x}), \mathbf{p}_{s,M})) \end{aligned} \tag{5}$$

and the total loss across all classes and scales is:

$$L_J = \frac{1}{C \cdot S} \sum_{c \in \mathcal{C}} \sum_{s \in \mathcal{S}} L_J(f_s(\mathbf{x}), P_{s,c}) \tag{6}$$

The final loss term for the multi-scale prototype learning (**Stage 1**) becomes:

$$L_{\text{proto}} = L_{\text{CE}} + \lambda_J \cdot L_J \tag{7}$$

with L_{CE} the per patch cross-entropy loss and λ_J a hyper-parameter controlling the weight of the regularization.

Stage 2: Prototype grouping mechanism. The multi-scale prototype grouping mechanism is applied to the learned scale-specific prototypes after they are projected to the training set patches. We proceed with two steps: we first run a warm-up step to train the class-specific grouping functions g^c_{group} while keeping the rest of the model frozen, namely the backbone f and the prototypes, so that we maintain interpretability. Then, in the second step, we run a joint training phase where we finetune both the grouping functions and the last linear layer h_{group}. During the learning of the groups, we apply a sparsity regularization term promoting that only a limited subset of prototypes are active for a given group. For this purpose, we define an entropic loss term, L_{ent}, on the weight matrices $\mathbf{w}_{g,c} \in [0,1]^{N \times |P_c|}$. Indeed, as we are projecting each row of those weight matrices $\mathbf{w}^{(\mathrm{n},:)}_{g,c}$ on the probability simplex it is possible to directly compute and minimize information theory measures such as the entropy of those rows, to enforce a sparse combination of prototypes among each group. The entropy loss is computed as follows:

$$L_{\text{ent}} = \frac{1}{C \cdot N_c} \sum_{c \in \mathcal{C}} \sum_{\mathrm{n}=1}^{N_c} \sum_{\mathrm{m}=1}^{M} -w^{(\mathrm{n},\mathrm{m})}_{g,c} \log(w^{(\mathrm{n},\mathrm{m})}_{g,c}) \tag{8}$$

Then, in the second step, we fine-tune also the linear layer h_{group}. To enforce more sparsity, following the training protocol from [8], we apply an L1-norm loss term on h_{group} solely on the weights $w^{c,\mathrm{n}}_{h_{\text{group}}}$ where the group g_{n} is not in the set of groups G_c assigned to class c. The total loss term for the grouping mechanism (Figure 2: **Stage 2**) becomes:

$$L_{\text{proto}} = L_{\text{CE}} + \lambda_{\text{ent}} \cdot L_{\text{ent}} + \lambda_{\text{L1}} \cdot \sum_{c \in \mathcal{C}} \sum_{\mathrm{n}: g_{\mathrm{n}} \notin G_c} |w^{(c,\mathrm{n})}_{h_{\text{group}}}| \tag{9}$$

where λ_{ent} and λ_{L1} are hyperparameters controlling the weight of the regularization terms. Once the whole architecture is trained, a final post-processing stage is done: the weights of the grouping function matrices $\mathbf{w}_{g,c}$ below a certain threshold α are set to 0. This enforces even more sparsity within the groups.

4. Experiments

4.1. Experimental setup

In all the experiments presented in the section 4.2, we use DeepLabv2 [9] with ResNet-101 [33] pre-trained on ImageNet as the backbone. For the multi-scale prototype learning, we assign $M = 3$ scale-dependent prototypes per scale to each class and $S = 4$ scales, so 12 prototypes per class in total. Moreover, we set the number of groups per class to $N_c = 3$ for the grouping mechanism. We evaluate ScaleProtoSeg on **(i)** Pascal VOC 2012 [21] (made of 1464 train, 1449 validation, and 1446 test images with 21 classes; the Pascal VOC training set is extended to 10582 images following [29]), **(ii)** Cityscapes [14] (composed of 2975 train, 500 validation, and 1525 test images of street scenes, with 19 classes) and **(iii)** ADE20K [100] (a scene-parsing dataset with 150 fine-grained semantic classes split in $\sim$ 20000 training and 2000 validation images; for the training of ProtoSeg [70] on ADE20K we extend the number of prototypes to 12 for direct comparison with ScaleProtoSeg). A detailed description of the experimental setup is available in the supplementary materials in Section 6.

4.2. Results and discussion

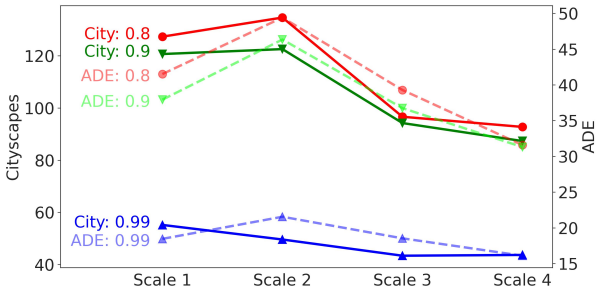
Method performance. In Table 1, we present the performance of our interpretable semantic segmentation method ScaleProtoSeg. Due to the well-known lack of faithfulness of saliency-based methods [1, 68] and the difficulty to compare against other by-design methods not based on prototypes (different benchmark datasets, different backbones and/or lack of public code repositories) [56, 71], we compare ScaleProtoSeg against ProtoSeg [70] (the closest previous methods in the literature aimed at prototype-based interpretability for semantic segmentation) and the non-interpretable counterpart DeepLabv2 [9]. It is worth mentioning that, as interpretability comes at the price of constrained and regularized training (namely through prototype projection to real training samples, the use of the diversity loss, and the sparse grouping mechanism), an interpretable model aims to close the gap with the non-interpretable counterpart, which acts as an upper-bound. In this regard, ScaleProtoSeg showcases a substantial improvement over ProtoSeg in terms of mIoU for Cityscapes and ADE20K, namely of $\sim 1.5\%$ and 4%. For ADE20K, the proposed ScaleProtoSeg goes beyond its original goal of enabling interpretability by surpassing the performance of its black-box counterpart. The Cityscapes and ADE20K datasets present more natural images with numerous objects at various scales, unlike in Pascal, where our method provides less of an advantage (results are on par with those of ProtoSeg). Indeed, we hypothesize that learning explicitly scale-specific prototypes at multiple scales is more advantageous in segmentation tasks with a large depth of field. In the supplementary materials, we further demonstrate the transferability of ScaleProtoSeg (a) to the medical domain, (b) to another segmentation architecture and (c) to a larger benchmark dataset (COCO-Stuff) in Section 7 and 8.

This observation is confirmed by the results presented in Table 2, which showcases the high improvement brought by the projection step of ScaleProtoSeg, performed during the proposed multi-scale prototype learning (see Figure 2: **Stage 1**). This improvement comes with an increase of solely two prototypes per class from 190 to 228 prototypes on Pascal VOC and 210 (201 after deduplication) to 252 on Cityscapes. Furthermore, in Table 2 we observe that, through the grouping mechanism and thresholding, we reduce the total number of prototypes used by our model by 51.3%, 48.0%, and 37.1% for Cityscapes, Pascal, and ADE20K respectively, compared to 32.6%, 36.7%, and 35.5% for ProtoSeg after deduplication and pruning. As a result, ScaleProtoSeg uses fewer prototypes than ProtoSeg despite a higher initial count except for ADE20K (see Section 4.1). The proposed grouping mechanism (see Figure 2: **Stage 2**) trades better interpretability and simplified decision process (interaction of only 3 groups per class) with only 0.4% to 0.8% mIoU loss.

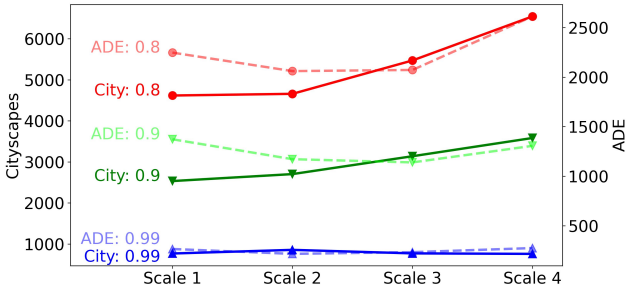
	Cityscapes mIoU		Pascal mIoU		ADE20K mIoU
Method	val	test	val	test	val
DeepLabv2 [9]	71.40	70.40	77.69	79.70	34.00
ProtoSeg [70]	67.54±0.22	67.04	**71.98**±0.11	**72.92**	29.67±0.23
ScaleProtoSeg	**68.97**±0.25	**68.52**	71.80±0.38	72.35	**34.18**±0.18

Table 1. ScaleProtoSeg mIoU performance on Pascal VOC, Cityscapes, and ADE20K. On the validation sets, we report the results over 3 runs, while for the test sets the results are based on [70] and our best ScaleProtoSeg validation run.

In the supplementary materials, we detail the capability of controlling the sparsity-performance trade-off via the thresholding of the grouping weights matrices and the entropy regularization in Section 9 and 10.



(a) Per-scale average number of connected components



(b) Per-scale average pixel area of the connected components

Figure 3. Analysis of the binarized prototype activations at multiple percentile thresholds $p_{th} \in \{0.8, 0.9, 0.99\}$ on Cityscapes and ADE20K validation sets.

Multi-scale prototypes analysis. After the prototype projection at the end of the multi-scale prototype learning stage, different patterns of prototype activations emerge across scales. In Figure 3, we present a quantitative analysis of the prototype activations scale-specific patterns on Cityscapes and ADE20K. For this purpose, we first binarize the activation maps of the multi-scale prototype layer from ScaleProtoSeg on the validation set of both datasets: we do so with multiple percentiles $p_{th} \in \{0.8, 0.9, 0.99\}$. Then we measure the number of connected components in the binarized maps as well as their average area in number

Dataset	Cityscapes				Pascal				ADE20K			
Method	ProtoSeg		ScaleProtoSeg		ProtoSeg		ScaleProtoSeg		ProtoSeg		ScaleProtoSeg	
Step	project	pruned	project	grouped	project	pruned	project	grouped	project	pruned	project	grouped
Prototypes	190	128	228	**111**	201	133	252	**131**	1756	1161	1800	**1132**
mIoU	67.24	67.23	**70.01**	69.22	72.00	72.05	**72.94**	72.26	30.89	29.97	**34.72**	34.32

Table 2. Methods performance at different training steps based on the results from [70] and our best ScaleProtoSeg run. The number of prototypes indicated in the *project* step (during **Stage 1**) is after the removal of the duplicates. The number of prototypes in the *grouped* step (during **Stage 2**) corresponds to a threshold of 0.05 on the group matrices.

of pixels. The average results per scale across all scale-specific prototypes are displayed in Figure 3.

We compute the connected components with the standard algorithm from *opencv* with an 8-connectivity relation between non-zero elements. We observe an overall decrease in the number of connected components when going from *Scale 1* to *Scale 4* across all thresholds and datasets and an increase in the average pixel area of the detected components (especially for ADE20K, which is less focused on images with large depth of field). In practice, this can be interpreted as a consequence of the lower field of view of the feature maps specific to *Scale 1* compared to those of *Scale 4* in the ASPP [9], where each scale-specific feature map vector represents a smaller receptive field on the image with less context. As such, the prototypes specific to *Scale 1* can be activated by smaller-scale prototypical parts and texture components. This analysis reflects the advantage of learning explicitly scale-specific prototypes across multiple scales: not only ScaleProtoSeg shows a performance improvement (see Table 2), but it also learns diverse object representations across scales, highlighting the multi-scale nature of semantic segmentation. In the supplementary material, we also analyzed if the prototypes present equivariance properties across scales in Section 11.

Quantitative analysis of interpretability. The quantitative evaluation of interpretability is an intrinsic problem in the field of XAI [1, 20] and especially in the context of per-design methods such as prototypical-parts learning [45, 91]. Several studies proposed human-based evaluations of explainable methods for classification tasks [13, 15]. However, scaling those evaluation scenarios to semantic segmentation raises concerns due to the complexity of the necessary human feedback compared to classification. We propose to quantitatively assess the degree of interpretability in terms of *consistency*, *stability*, and *sparsity*. The first two metrics have been proposed in [37] to measure the interpretability of a classification model. We leverage the part-annotations extension of Pascal and Cityscapes [16, 58] covering respectively 16 and 5 classes, and propose an extension of these metrics to the task of semantic segmentation. Given an input image $\mathbf{x}$ and a prototype $\mathbf{p}_\mathrm{m}$, the prototype activation $g_\mathrm{proto}(\mathbf{z}, \mathbf{p}_\mathrm{m}),\ \forall \mathbf{z} \in f(\mathbf{x})$ is binarized using multiple percentile thresholds: $\{70^{th}, 80^{th}, 90^{th}\}$ instead of a fixed size bounding box, as multiple objects of the same class can be present in $\mathbf{x}$. We then compute the average consistency and stability scores across the multiple thresholds on the part-annotated validation sets for 3 runs per method, similarly to [37]. We also introduce a global sparsity metric that measures the number of active prototypes per class after thresholding the last linear layer absolute values with $\tau_{th} = 0.005$, which is similar to the sparsity ratio used as a measure of interpretations' compactness in [61]. The measured sparsity is linked to several properties of interpretability: transparency [24], understandability [13, 15], and simplicity [60]. Despite this metrics not being exhaustive, our assessment shows that ScaleProtoSeg displays overall enhanced interpretability over ProtoSeg (see Table 3 and 4).

	Pascal		
Methods:	Consistency ↑	Stability ↑	Sparsity ↓
ProtoSeg	35.05 ±1.44	73.45 ±0.45	157.67 ±3.40
ScaleProtoSeg	**38.78** ±1.68	**76.30** ±0.26	**23.34** ±5.22

Table 3. Consistency, stability, and sparsity scores on Pascal part-annotated sets.

	Cityscapes		
Methods:	Consistency ↑	Stability ↑	Sparsity ↓
ProtoSeg	**34.48** ±1.74	27.00 ±1.58	134 ±2.45
ScaleProtoSeg	31.11 ±1.66	**32.54** ±2.36	**12.91** ±1.26

Table 4. Consistency, stability, and sparsity scores on Cityscapes part-annotated sets.

Qualitative analysis of grouping mechanism interpretability. In terms of interpretability, the groups can be first represented through specific visualization for each sample, as shown in Figure 4, where the images on the graph display the prototype activations for an input sample. Those activations are grouped per scale and provide local explanations for the model. Moreover, the edges in those visualizations are static for all samples assigned to the class *car* in Cityscapes, and so provide global explanations on the model behavior. In Figure 4, we can easily observe, despite the large number of prototypes compared to the model mean of 3.39, that different patterns emerge for each group. The first group focuses mainly on a prototype in *Scale 4* activated on the bottom part of the cars and another similar at *Scale 2*. The second group focuses on a prototype at *Scale 1*, which activates the top of the cars with another one similar at *Scale 3*. Lastly, the third group focuses on mul-

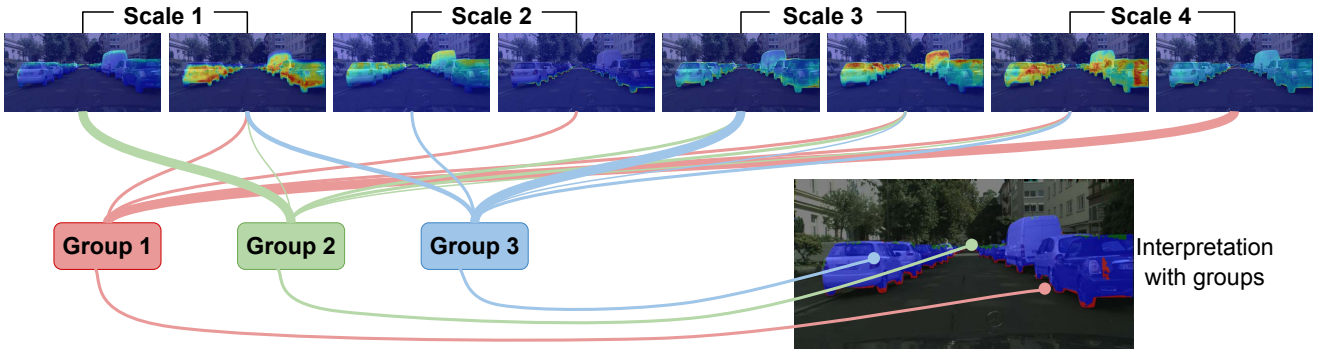


Figure 4. ScaleProtoSeg provides the interpretation of a segmentation through the analysis of groups of prototypes. For the example of the class *car* on Cityscapes, 2 prototypes per scale (whose activations are displayed **at the top** of the figure) are used by the model across the 3 learned groups shown **at the bottom right**. For this class, groups correspond to the bottom part, the main part or the upper part of the car.

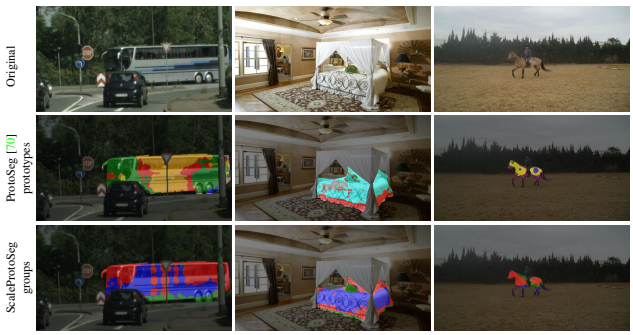


Figure 5. Model prototype and group assignments for the class *bus*, *bed*, and *horse* on Cityscapes, ADE20K, and Pascal.

tiple parts of the cars and, will activate on the main body of them. Groups 1 and 2, by combining similar prototypes across two scales, show transparently the use of multi-scale information in the model. Figure 4 demonstrates that the grouping mechanism in our method can be representative of both local and global interpretability. The example of group assignments for the class *car* illustrates the identified parts in the group visualization. Interestingly, the second group seems more activated on cars further in the depth field, as its main activated prototype is from *Scale 1* with a smaller field of view. This clearly illustrates the role of scale in the identification of prototypical parts.

In Figure 5, we show the final decision process based on the prototype assignments for ProtoSeg (row 2) and the group assignments for ScaleProtoSeg (row 3). The ProtoSeg outputs were computed based on a rerun of the model after pruning. In all three examples of Figure 5, our groups identify one or more prototypical parts each, for the class *bus*, *bed*, and *horse*. For instance, in the bus image, the first group in red corresponds to both the front and back top of the bus, a pattern that can be also seen in the other two images. Moreover, we see through those examples that, due to the limited number of groups ($N_c = 3$) in ScaleProtoSeg, the interpretation of the final decision process is more constrained compared to the baseline method ProtoSeg. Indeed, even if we consider the prototype interactions leading to the group and their assignment, our method still uses fewer prototypes compared to ProtoSeg, as described in Table 2. Lastly, as detailed in the supplementary Section 12, our method enforces stronger sparsity regularization on the final classification layer inducing a smaller computation overhead on DeepLabv2 [9] compared to ProtoSeg [70].

Overall, through the sparsity of the groups, their simple visualizations, their small number, and the sparse final linear layer, we advocate that our method enables the user to investigate more transparently the semantic segmentation model while improving upon the state-of-the-art ProtoSeg.

5. Conclusion

In this paper, we present an interpretable semantic segmentation model, ScaleProtoSeg, that leverages multi-scale representations for prototype learning and introduces a novel grouping mechanism to learn prototype interactions across scales. We showed that our model results are particularly advantageous in complex datasets presenting many objects at different depths: Cityscapes and ADE20K. Moreover, through our analysis, we evaluated ScaleProtoSeg interpretability across 3 quantitative metrics: *consistency*, *stability*, and *sparsity* and inspected the different patterns learned by the prototypes across scales. This showcases the potential of multi-scale prototype learning to provide a deeper understanding of the effect of scales on object representations. Lastly, our novel grouping mechanism provides a clear representation of the final decision process: it shows how the model learns to group the prototypes across scales by limiting the number of active groups, allowing for their easy and sparse visualization for all images.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018. 6, 7

[2] Julius Adebayo, Michael Muelly, Harold Abelson, and Been Kim. Post hoc explanations may be ineffective for detecting unknown spurious correlation. In *International conference on learning representations*, 2021. 1

[3] Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *Advances in neural information processing systems*, 34:4699–4711, 2021. 1, 2

[4] I Arganda-Carreras, HS Seung, A Cardona, and J Schindelin. Segmentation of neuronal structures in em stacks challenge-isbi 2012, 2012. 13, 14

[5] Miguel A Belenguer-Plomer, Mihai A Tanase, Angel Fernandez-Carrillo, and Emilio Chuvieco. Burned area detection and mapping using sentinel-1 backscatter coefficient and thermal anomalies. *Remote Sensing of Environment*, 233:111345, 2019. 1

[6] Andrea Bontempelli, Stefano Teso, Katya Tentori, Fausto Giunchiglia, and Andrea Passerini. Concept-level debugging of part-prototype networks. *arXiv preprint arXiv:2205.15769*, 2022. 18

[7] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018. 14

[8] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019. 1, 2, 3, 4, 5

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 2, 3, 4, 5, 6, 7, 8, 13, 15

[10] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 2

[11] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 1

[12] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021. 3

[13] Julien Colin, Thomas Fel, Rémi Cadène, and Thomas Serre. What i cannot predict, i do not understand: A human-centered evaluation framework for explainability methods. *Advances in neural information processing systems*, 35:2832–2845, 2022. 7

[14] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 2, 5, 13

[15] Omid Davoodi, Shayan Mohammadizadehsamakosh, and Majid Komeili. On the interpretability of part-prototype based classifiers: a human centric analysis. *Scientific Reports*, 13(1):23088, 2023. 7

[16] Daan de Geus, Panagiotis Meletis, Chenyang Lu, Xiaoxiao Wen, and Gijs Dubbelman. Part-aware panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 7, 13

[17] Henghui Ding, Xudong Jiang, Ai Qun Liu, Nadia Magnenat Thalmann, and Gang Wang. Boundary-aware feature propagation for scene segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6819–6829, 2019. 2

[18] Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, volume 3, 2018. 3

[19] Jon Donnelly, Alina Jade Barnett, and Chaofan Chen. Deformable protopnet: An interpretable image classifier using deformable prototypes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10265–10275, 2022. 2

[20] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017. 1, 7

[21] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 2, 5, 13

[22] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2020. 1

[23] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3146–3154, 2019. 2

[24] Srishti Gautam, Ahcene Boubekki, Stine Hansen, Suaiba Salahuddin, Robert Jenssen, Marina Höhne, and Michael Kampffmeyer. Protovae: A trustworthy self-explainable prototypical variational model. *Advances in Neural Information Processing Systems*, 35:17940–17952, 2022. 7

[25] Alan H Gee, Diego Garcia-Olano, Joydeep Ghosh, and David Paydarfar. Explaining deep classification of time-series data with learned prototypes. In *CEUR workshop*

proceedings, volume 2429, page 15. NIH Public Access, 2019. 2

[26] Marzyeh Ghassemi, Luke Oakden-Rayner, and Andrew L Beam. The false hope of current approaches to explainable artificial intelligence in health care. *The Lancet Digital Health*, 3(11):e745–e750, 2021. 1

[27] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. PMLR, 2019. 1

[28] Ran Gu, Guotai Wang, Tao Song, Rui Huang, Michael Aertsen, Jan Deprest, Sébastien Ourselin, Tom Vercauteren, and Shaoting Zhang. Ca-net: Comprehensive attention convolutional neural networks for explainable medical image segmentation. *IEEE transactions on medical imaging*, 40(2):699–711, 2020. 2

[29] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *2011 international conference on computer vision*, pages 991–998. IEEE, 2011. 5

[30] Peter Hase, Chaofan Chen, Oscar Li, and Cynthia Rudin. Interpretable image recognition with hierarchical prototypes. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pages 32–40, 2019. 2

[31] Trevor Hastie and Robert Tibshirani. Generalized additive models: some applications. *Journal of the American Statistical Association*, 82(398):371–386, 1987. 2

[32] Junjun He, Zhongying Deng, Lei Zhou, Yali Wang, and Yu Qiao. Adaptive pyramid context network for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7519–7528, 2019. 2

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[34] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021. 1

[35] Lukas Hoyer, Mauricio Munoz, Prateek Katiyar, Anna Khoreva, and Volker Fischer. Grid saliency for context explanations of semantic segmentation. *Advances in neural information processing systems*, 32, 2019. 1, 2

[36] Qin Huang, Chunyang Xia, Chihao Wu, Siyang Li, Ye Wang, Yuhang Song, and C-C Jay Kuo. Semantic segmentation with reverse attention. *arXiv preprint arXiv:1707.06426*, 2017. 13

[37] Qihan Huang, Mengqi Xue, Wenqi Huang, Haofei Zhang, Jie Song, Yongcheng Jing, and Mingli Song. Evaluation and improvement of interpretability for self-explainable part-prototype networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2011–2020, 2023. 7, 13

[38] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 603–612, 2019. 2

[39] Jyh-Jing Hwang, Stella X Yu, Jianbo Shi, Maxwell D Collins, Tien-Ju Yang, Xiao Zhang, and Liang-Chieh Chen. Segsort: Segmentation by discriminative sorting of segments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7334–7344, 2019. 3

[40] Shipra Jain, Danda Pani Paudel, Martin Danelljan, and Luc Van Gool. Scaling semantic segmentation beyond 1k classes on a single gpu. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7426–7436, 2021. 3

[41] Harold Jeffreys. *The theory of probability*. OuP Oxford, 1998. 5

[42] Saumya Jetley, Bernardino Romera-Paredes, Sadeep Jayasumana, and Philip Torr. Prototypical priors: From improving classification to zero-shot learning. *arXiv preprint arXiv:1512.01192*, 2015. 3

[43] Jeya Vikranth Jeyakumar, Luke Dickens, Luis Garcia, Yu-Hsi Cheng, Diego Ramirez Echavarria, Joseph Noor, Alessandra Russo, Lance Kaplan, Erik Blasch, and Mani Srivastava. Automatic concept extraction for concept bottleneck-based video classification. *arXiv preprint arXiv:2206.10129*, 2022. 2

[44] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018. 1

[45] Sunnie SY Kim, Nicole Meister, Vikram V Ramaswamy, Ruth Fong, and Olga Russakovsky. Hive: evaluating the human interpretability of visual explanations. *arXiv preprint arXiv:2112.03184*, 2021. 7

[46] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020. 1, 2

[47] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018. 1

[48] Nataliia Kussul, Mykola Lavreniuk, Sergii Skakun, and Andrii Shelestov. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5):778–782, 2017. 1

[49] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1096, 2019. 1

[50] Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020. 3

[51] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceed-*

ings of the AAAI Conference on Artificial Intelligence, volume 32, 2018. 2

[52] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017. 2

[53] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2020. 1

[54] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. *Advances in Neural Information Processing Systems*, 30, 2017. 2

[55] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2

[56] Max Losch, Mario Fritz, and Bernt Schiele. Interpretability beyond classification output: Semantic bottleneck networks. *arXiv preprint arXiv:1907.10882*, 2019. 2, 6

[57] Diego Marcos, Ruth Fong, Sylvain Lobry, Rémi Flamary, Nicolas Courty, and Devis Tuia. Contextual semantic interpretability. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 2, 4

[58] Panagiotis Meletis, Xiaoxiao Wen, Chenyang Lu, Daan de Geus, and Gijs Dubbelman. Cityscapes-panoptic-parts and pascal-panoptic-parts datasets for scene understanding. 7, 13

[59] Pascal Mettes, Elise Van der Pol, and Cees Snoek. Hyperspherical prototype networks. *Advances in neural information processing systems*, 32, 2019. 3

[60] Yao Ming, Panpan Xu, Huamin Qu, and Liu Ren. Interpretable and steerable sequence learning via prototypes. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 903–913, 2019. 2, 7

[61] Meike Nauta, Jörg Schlötterer, Maurice Van Keulen, and Christin Seifert. Pip-net: Patch-based intuitive prototypes for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2744–2753, 2023. 7

[62] Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. *arXiv preprint arXiv:2304.06129*, 2023. 2

[63] Jeremy Petch, Shuang Di, and Walter Nelson. Opening the black box: the promise and limitations of explainable machine learning in cardiology. *Canadian Journal of Cardiology*, 38(2):204–213, 2022. 1

[64] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 1, 2

[65] Dmitry Rashkovetsky, Florian Mauracher, Martin Langer, and Michael Schmitt. Wildfire detection from multisensor satellite imagery using deep semantic segmentation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:7001–7016, 2021. 1

[66] Mattia Rigotti, Christoph Miksovic, Ioana Giurgiu, Thomas Gschwind, and Paolo Scotton. Attention-based interpretability with concept transformers. In *International Conference on Learning Representations*, 2021. 2

[67] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 2, 13

[68] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019. 1, 6

[69] Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zieliński. Protopshare: Prototypical parts sharing for similarity discovery in interpretable image classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1420–1430, 2021. 2

[70] Mikołaj Sacha, Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zieliński. Protoseg: Interpretable semantic segmentation with prototypical parts. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1481–1492, 2023. 1, 2, 3, 4, 5, 6, 7, 8, 13, 16, 19, 20

[71] Alberto Santamaria-Pang, James Kubricht, Aritra Chowdhury, Chitresh Bhushan, and Peter Tu. Towards emergent language symbolic semantic segmentation and model interpretability. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part I 23*, pages 326–334. Springer, 2020. 1, 2, 6

[72] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 1

[73] Sofia Serrano and Noah A Smith. Is attention interpretable? *arXiv preprint arXiv:1906.03731*, 2019. 2

[74] Mennatullah Siam, Mostafa Gamal, Moemen Abdel-Razek, Senthil Yogamani, Martin Jagersand, and Hong Zhang. A comparative study of real-time semantic segmentation for autonomous driving. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 587–597, 2018. 1

[75] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017. 3

[76] Gregor Stiglic, Primoz Kocbek, Nino Fijacko, Marinka Zitnik, Katrien Verbert, and Leona Cilar. Interpretability of machine learning-based prediction models in healthcare. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(5):e1379, 2020. 1

[77] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmenta-

tion. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7262–7272, 2021. 3

[78] Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5229–5238, 2019. 2

[79] Pin Tang, Hai-Ming Xu, and Chao Ma. Prototransfer: Cross-modal prototype transfer for point cloud segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3337–3347, 2023. 3

[80] Yin Tang, Tao Chen, Xiruo Jiang, Yazhou Yao, Guo-Sen Xie, and Heng-Tao Shen. Holistic prototype attention network for few-shot vos. *arXiv preprint arXiv:2307.07933*, 2023. 3

[81] Michael Treml, José Arjona-Medina, Thomas Unterthiner, Rupesh Durgesh, Felix Friedmann, Peter Schuberth, Andreas Mayr, Martin Heusel, Markus Hofmarcher, Michael Widrich, et al. Speeding up semantic segmentation for autonomous driving. 2016. 1

[82] Devis Tuia, Ribana Roscher, Jan Dirk Wegner, Nathan Jacobs, Xiaoxiang Zhu, and Gustau Camps-Valls. Toward a collective agenda on ai for earth science data analysis. *IEEE Geoscience and Remote Sensing Magazine*, 9(2):88–104, 2021. 1

[83] Kira Vinogradova, Alexandr Dibrov, and Gene Myers. Towards interpretable semantic segmentation via gradient-weighted class activation mapping (student abstract). In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13943–13944, 2020. 1, 2

[84] Jiaqi Wang, Huafeng Liu, Xinyue Wang, and Liping Jing. Interpretable image recognition by constructing transparent embedding space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 895–904, 2021. 2

[85] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *proceedings of the IEEE/CVF international conference on computer vision*, pages 9197–9206, 2019. 3

[86] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14408–14419, 2023. 1

[87] Wenguan Wang, Cheng Han, Tianfei Zhou, and Dongfang Liu. Visual recognition with deep nearest centroids. *arXiv preprint arXiv:2209.07383*, 2022. 3

[88] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018. 3

[89] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 1, 2

[90] Wenjia Xu, Yongqin Xian, Jiuniu Wang, Bernt Schiele, and Zeynep Akata. Attribute prototype network for zero-shot learning. *Advances in Neural Information Processing Systems*, 33:21969–21980, 2020. 3

[91] Romain Xu-Darme, Georges Quénot, Zakaria Chihani, and Marie-Christine Rousset. Sanity checks and improvements for patch visualisation in prototype-based image classification. *arXiv preprint arXiv:2302.08508*, 2023. 7

[92] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3474–3482, 2018. 3

[93] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1857–1866, 2018. 2

[94] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 2

[95] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7151–7160, 2018. 2

[96] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 2

[97] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *Proceedings of the European conference on computer vision (ECCV)*, pages 267–283, 2018. 2

[98] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 5209–5217, 2017. 2

[99] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 2

[100] 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*, pages 633–641, 2017. 2, 5, 13

[101] Tianfei Zhou, Wenguan Wang, Ender Konukoglu, and Luc Van Gool. Rethinking semantic segmentation: A prototype view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2582–2593, 2022. 3