

ProtoSeg: Interpretable Semantic Segmentation with Prototypical Parts

Mikołaj Sacha¹

Dawid Rymarczyk^{1,2}

Łukasz Struski¹

Jacek Tabor¹

Bartosz Zieliński^{1,3}

¹ Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków, Poland

² Ardigen, Kraków, Poland ³ IDEAS NCBR, Warsaw, Poland

{mikolaj.sacha; dawid.rymarczyk}@doctoral.uj.edu.pl

{lukasz.struski; jacek.tabor; bartosz.zielinski}@uj.edu.pl

Abstract

We introduce ProtoSeg, a novel model for interpretable semantic image segmentation, which constructs its predictions using similar patches from the training set. To achieve accuracy comparable to baseline methods, we adapt the mechanism of prototypical parts and introduce a diversity loss function that increases the variety of prototypes within each class. We show that ProtoSeg discovers semantic concepts, in contrast to standard segmentation models. Experiments conducted on Pascal VOC and Cityscapes datasets confirm the precision and transparency of the presented method.

1. Introduction

Semantic segmentation is an essential component in many visual understanding systems. However, while deep learning-based models have achieved promising performance on challenging benchmarks [49], their decisions remain unclear due to lack of explanation [59]. This issue may appear particularly problematic in critical applications, such as medical imaging or autonomous driving.

Most of the eXplainable Artificial Intelligence (XAI) approaches focus on classification or regression task [8, 11, 23, 65, 78]. Therefore, interpretable segmentation is still considered an open question [60], with only a few initial works on the crossroad of XAI and segmentation. One of them is the Symbolic Semantic (S2) framework [64] where together with segmentation, the model generates a symbolic sentence derived from a categorical distribution. Another approach [71] generalizes the Grad-CAM method [65] to the problem of segmentation. However, both methods have significant disadvantages. The former requires a predefined vocabulary of symbolic words, while the latter can be unreliable and introduce additional bias to the results [3].

In this paper, we introduce ProtoSeg, an interpretable

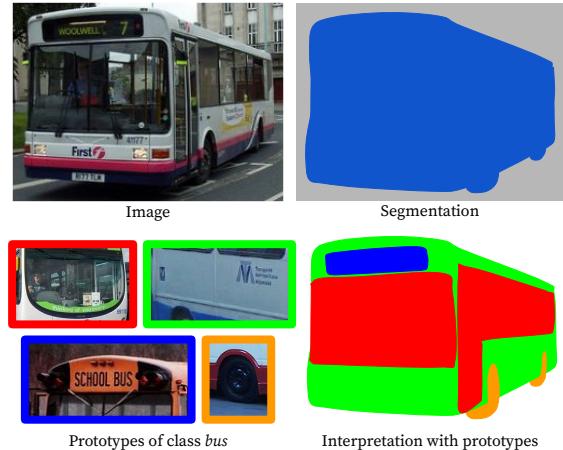


Figure 1: In contrast to existing methods, ProtoSeg provides an interpretation of resulted segmentation. For this purpose, it operates on patches selected from a training set (prototypes) corresponding to parts of the segmented objects. For a bus, prototypes can correspond to windows or wheels, represented by red and orange colors, respectively.

semantic segmentation method based on prototypical parts [11]. While the standard approaches return only the class probability for each input pixel, ProtoSeg learns prototypes for each class and uses them to generate and explain segmentation with patches (cases) from the training set. As we present in Figure 1 and Section 5, the main goal is to focus prototypes of the same class on different semantic concepts. For this purpose, we introduce a novel diversity loss function that increases the variety of prototypes for each class (see Figure 2). Such application of case-based methodology significantly increases the interpretability of the segmentation model. Moreover, in contrast to previous methods, it does not require additional effort from the users to provide explanations.

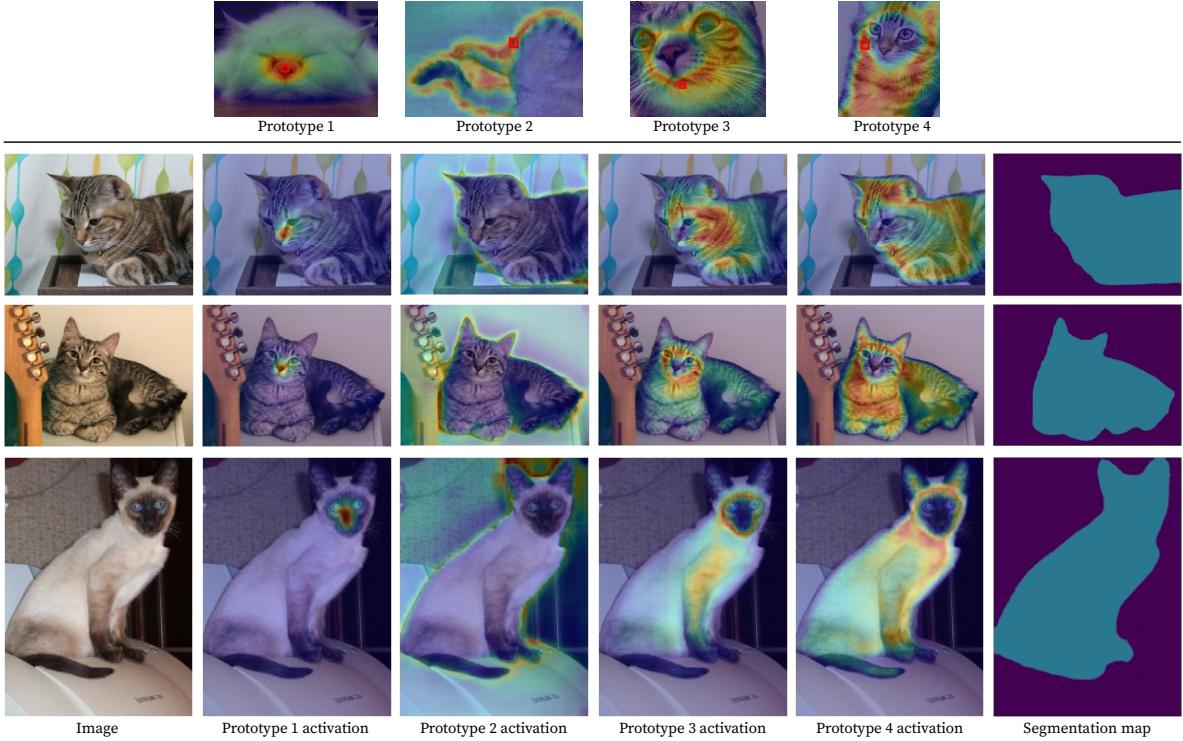


Figure 2: Prototype activation maps generated by ProtoSeg for four prototypes from class *cat* (columns) and three sample images from PASCAL VOC 2012 (rows). Maps differ from each other, e.g. prototype 1 concentrates on the cat’s nose, while prototype 4 activates mostly on the cat’s neck. We see that ProtoSeg can derive semantic concepts using prototypical cases from the training dataset.

To show the effectiveness of ProtoSeg, we conduct experiments on three datasets: Pascal VOC 2012 [19], Cityscapes [16] and EM Segmentation Challenge [1]. The results indicate no significant decrease in performance between our interpretable model and the original black-box approaches like DeepLabv2 [13] or U-Net [58]. Additionally, we present ablation studies showing how diversity loss influences model performance and transparency. We made the code available. Our contributions can be therefore summarized as follows:

- we introduce a model that employs prototypical parts to provide interpretable semantic segmentation,
- we define a diversity loss function based on Jeffrey’s divergence that increases the variability of prototypes within each class,
- we show that ProtoSeg can be used with different backbone architectures and on various semantic segmentation tasks.

In the next section of this paper, we discuss related works, then in Section 3 we introduce ProtoSeg and the diversity loss function. Later, in Section 4, we describe the

experimental setup that uses DeepLab [13] model as a backbone, followed by the results in Section 5, in which we also present the extendability of ProtoSeg to different segmentation architectures. Finally, we conclude our work in Section 6.

2. Related works

Explainable artificial intelligence. Deep learning explanations can be obtained with two types of models: post hoc or self-explainable [59]. Post hoc approaches explain the reasoning process of black-box methods. They include a saliency map [48, 56, 65, 66, 68] that is a heatmap of essential image parts, Concept Activation Vectors (CAV) revealing the internal network state as user-friendly concepts [15, 24, 37, 40, 76], counterfactual examples [2, 26, 52, 54, 73], or analyzing the networks’ reaction to the image perturbation [8, 21, 22, 57]. Post hoc methods are convenient because they do not require any changes to the models’ architecture, but they may produce biased and fragile explanations [3]. For this reason, self-explainable models attract attention [5, 9] making the decision process more transparent. Recently, many researchers have focused on enhanc-

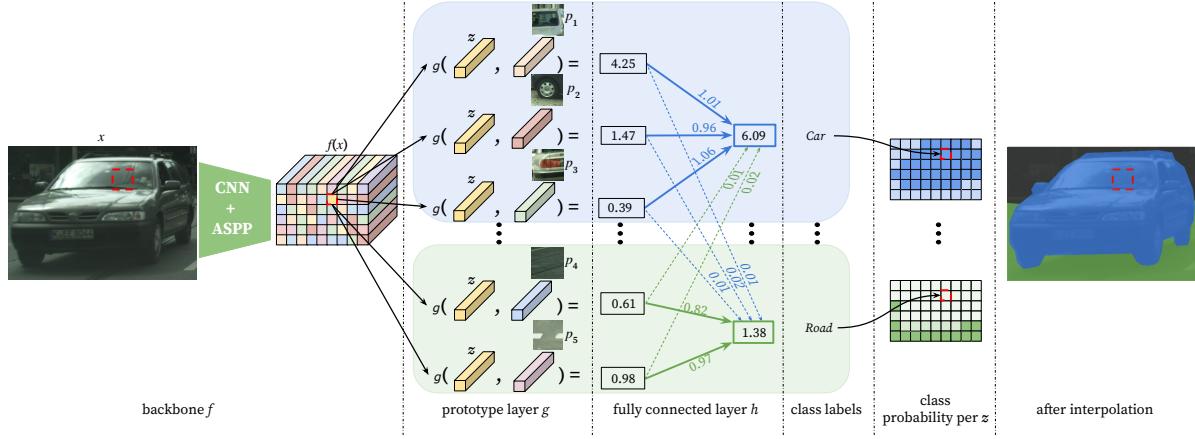


Figure 3: ProtoSeg consists of a backbone network f , prototype layer g , and a fully connected layer h . While the backbone network processes the image as a whole, the prototype and fully connected layers consider each z from feature map $f(x)$ separately. The final segmentation is obtained by interpolating the output map corresponding to class probability.

ing the concept of the prototypical parts introduced in ProtoPNet [11] to represent the networks’ activation patterns. The most prominent extensions include TesNet [72] and Deformable ProtoPNet [18] that exploits orthogonality in the prototype construction. ProtoPShare [63], ProtoTree [53], and ProtoPool [62] reduce the amount of prototypes used in the classification. Other methods consider hierarchical classification with prototypes [30], prototypical part transformation from the latent space to data space [41], and knowledge distillation technique from prototypes [36]. Moreover, prototype-based solutions are widely adopted in various applications, such as medical imaging [4, 7, 38, 61, 69], time-series analysis [23], graphs classification [78], and sequence learning [50]. In this paper, we adapt the prototype mechanism to the semantic segmentation task.

Semantic segmentation. Similarly to other tasks of computer vision, recent semantic segmentation methods base on deep architectures [25, 28, 47], especially convolutional neural networks [12, 32, 45, 79]. Moreover, they usually consist of two parts: an encoder pretrained on a classification task and a decoder network semantically projecting the activation features onto the pixel space. The most popular models include U-Net [58], which contain shortcuts between the down-sampling layer in the encoder and the corresponding up-sampling layer in the decoder that effectively capture fine-grained information. Several works use additional mechanisms (such as conditional random fields) at the network output to improve models’ performance [6, 10, 42]. Some approaches adapt superpixels [51, 67], Markov random field [46], or modules learning pixel affinities [44, 74] to obtain segmentation. Others employ contrastive learning [14, 34, 80] or multiple recep-

tive fields [77] to increase the segmentation quality. Moreover, with recent advancements in transformers architectures, models such as [27, 70, 75, 80] are used to obtain state-of-the-art results. Finally, Chen et al. [13] proposed DeepLab method that uses multiple techniques to improve the existing methods: atrous convolutions, atrous spatial pyramidal pooling, and conditional random fields. We provide an interpretable version of this method.

3. ProtoSeg

In this section, we first describe the architecture of our ProtoSeg method for interpretable semantic segmentation. Then, we provide information about the training procedure. Finally, we describe a novel regularization technique that increases the variety of prototypes within each class.

3.1. Architecture

Figure 3 illustrates the architecture of ProtoSeg, composed of a backbone network f , prototype layer g , and a fully connected layer h . Let $x \in \mathbb{R}^{H \times W \times 3}$ be an RGB image and feature map $f(x) \in \mathbb{R}^{H_d \times W_d \times D}$ be the output of the backbone network for this image. Moreover, let us consider $z \in \mathbb{R}^D$ as a point (or patch) from $f(x)$. Each z is passed to prototype layer g with M learnable prototypes $p_j \in \mathbb{R}^D$ to compute M similarity scores (prototype’s activations) using formula from [11]:

$$g(z, p_j) = \log \left(\frac{\|z - p_j\|_2^2 + 1}{\|z - p_j\|_2^2 + \epsilon} \right). \quad (1)$$

The M similarity scores computed for feature map point z are processed through a fully connected layer h with weight matrix $w_h \in \mathbb{R}^{M \times C}$ to produce probabilities of C classes.

As a result of processing all z from $f(x)$ through g and w_h , we acquire output map of shape $H_d \times W_d \times C$. To obtain the final segmentation, this map is interpolated to resolution $H \times W \times C$.

As the backbone network, we use DeepLab [13], a standard model for image segmentation that consists of ResNet-101 [31] pretrained on some large-scale computer vision task, followed by an Atrous Spatial Pyramid Pooling (ASPP) layer. In the prototype layer g , as in [11], each prototype is assigned to one of the C classes. We define \mathbf{P}_c as the set of all prototypes from class $c \in C$ and initialize $w_h^{(c,j)} = 1$ for all $p_j \in \mathbf{P}_c$ and $w_h^{(c,j)} = -0.5$ for all $p_j \notin \mathbf{P}_c$. This initialization steers the model towards producing high activation between feature map points and prototypes of their predicted class, while lowering their activation to prototypes from other classes. Similarly to [13], the model’s output is obtained differently in the inference and training phases. In inference, we use bilinear interpolation to match the size of the segmentation map with the input image size, while in training, we decrease the resolution of ground truth segmentation to fit the size of the output feature map.

3.2. Multi-step training procedure

We apply a multi-step training protocol from [11]. We start from the ResNet-101 layers pretrained on same large visual recognition task, while randomly initializing ASPP and prototype vectors and setting the weights w_h as described in Section 3.1. We start with a warmup phase, where we freeze ResNet-101 and w_h weights, training only ASPP and prototype layer. Then, we run a joint optimization process, where we train everything except w_h weights. Next, we execute prototype projection, which replaces prototypes with the representation of the nearest patch z from the training set. During this stage, we also remove duplicate prototypes that are projected onto same training patch. Then, in the tuning phase, we fine-tune the w_h weights of the last layer. Subsequently, we apply prototype pruning that removes non-class-specific prototypes using the pruning algorithm from [11]. Finally, we again fine-tune the last layer h .

3.3. Diverse prototypes of same class

We simplify the approach of [11] and remove the *cluster* and *separation* losses, which we find redundant for our method. Instead, we combine the standard cross-entropy loss with an additional component which enforces same-class prototypes to be activated in different image areas (see Figure 4b), resulting in optimal utilization of prototypes by the model (Figure 5). In this section we will describe in detail how we construct the additional loss term that improves the diversity of prototypes.

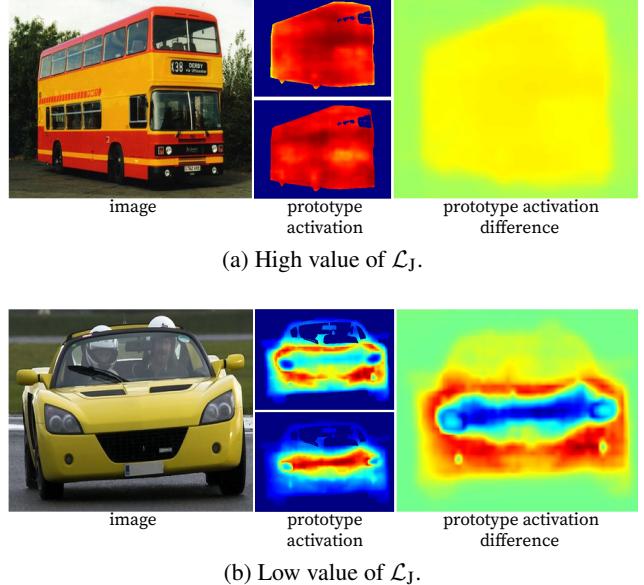


Figure 4: Comparison between high and low values of \mathcal{L}_J for the activation of two prototypes. \mathcal{L}_J has a high value if two prototypes of the same class activate in the same area (a). For this reason, we add \mathcal{L}_J as an additional component of the loss function to increase the variety of prototypes within each class (b).

Jeffrey’s similarity Firstly, we start by introducing a function which will encourage the diversity of prototypes. Let us recall that *Jeffrey’s divergence* [35] between two probability distributions U and V

$$\mathcal{D}_J(U, V) = \frac{1}{2}\mathcal{D}_{KL}(U\|V) + \frac{1}{2}\mathcal{D}_{KL}(V\|U) \quad (2)$$

is defined as the symmetrization of the Kullback-Leibler divergence. Clearly, $\mathcal{D}_J(U, V) = 0$ iff $U = V$, and large value of \mathcal{D}_J implies that the distributions concentrate on different regions. Now given a sequence of distributions U_1, \dots, U_l we introduce their *Jeffrey’s similarity* by the formula

$$\mathcal{S}_J(U_1, \dots, U_l) = \frac{1}{\binom{l}{2}} \sum_{i < j} \exp(-\mathcal{D}_J(U_i, U_j)) \quad (3)$$

Observe that Jeffrey’s similarity is permutation invariant, it attains values in the interval $[0, 1]$ and $\mathcal{S}_J(U_1, \dots, U_n) = 1$ iff $U_1 = \dots = U_n$. Moreover, if distributions U_i have pairwise disjoint supports then $\mathcal{S}_J(U_1, \dots, U_n) = 0$.

Prototype-class-image distance vector Let $p \in \mathbf{P}_c$ be some prototype assigned to class $c \in C$ and $Z = f(x) \in \mathbb{R}^{H_d \times W_d \times D}$ be the feature map of some image x after processing through the backbone f . Let also $Y_Z \in \mathbb{R}^{H_d \times W_d}$ be the ground-truth class labels per each feature map points

of Z . We define *prototype-class-image distance vector* between image feature map Z and prototype p as

$$v(Z, p) = \text{softmax}(\|z_{ij} - p\|^2 \mid z_{ij} \in Z : Y_{ij} = c). \quad (4)$$

The vector $v(Z, p)$ measures the relative activation of the prototype $p \in \mathbf{P}_c$ on parts of the image that are assigned to class c . We note that the length of the vector $v(Z, p)$ is equal to the number of points on the feature map in Z assigned to class c .

Prototype diversity loss We define the *prototype diversity loss* between the sequence of prototypes $\mathbf{P}_c = (p_1, \dots, p_k)$ from the same class $c \in C$ on image feature map Z as

$$\mathcal{L}_J(Z, \mathbf{P}_c) = \mathcal{S}_J(v(Z, p_1), \dots, v(Z, p_k)). \quad (5)$$

Note that $\mathcal{L}_J(Z, \mathbf{P}_c)$ measures the difference between distributions of prototype activation within a sequence \mathbf{P}_c on feature map points from Z assigned to their class. It is minimized by lowering the Jeffrey's similarity of their prototype-class-image distance vectors. Finally, we introduce the *total prototype diversity loss* for a feature map Z and the set of all prototypes \mathbf{P}

$$\mathcal{L}_J = \frac{1}{C} \sum_{c=1}^C \mathcal{L}_J(Z, \mathbf{P}_c). \quad (6)$$

The final loss during *warmup* and *joint training* is

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda_J \cdot \mathcal{L}_J, \quad (7)$$

where \mathcal{L}_{CE} is the cross entropy loss for pixel patch-wise classification and λ_J as a hyperparameter that controls the weight of diversity of same-class prototypes within their assigned class. Following the training protocol from [11], we add an additional L1-norm loss term on w_h weights during the *fine-tuning* phases, making the total loss in these phases equal to

$$\mathcal{L}_F = \mathcal{L} + \lambda_{L1} \cdot \sum_{c=1}^C \sum_{j: p_j \notin \mathbf{P}_c} |w_h^{(c,j)}|. \quad (8)$$

4. Experimental setup

In all experiments, we use DeepLab [13] with ResNet-101 [33] weights pretrained either on ImageNet [17] or COCO [43]. We assign 10 prototypes to each class, and set the prototype size to $D = 64$. We set the weights of loss terms to $\lambda_{L1} = 10^{-4}$ and $\lambda_J \in \{0, 0.25\}$. For input images, we employ augmentation techniques, such as random cropping, horizontal flipping and scaling images by a factor in range $[0.5, 1.5]$. We use batch size equal to 10

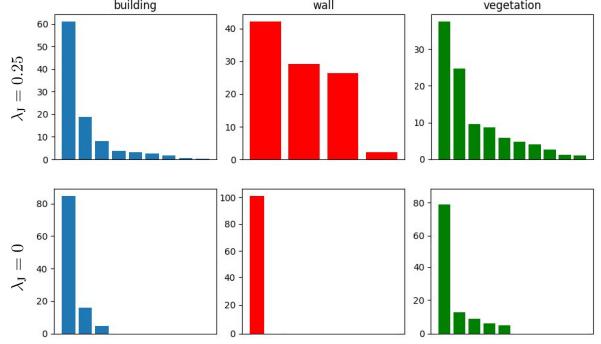


Figure 5: Histograms showing the assignment of feature map points to prototypes per class on Cityscapes. As the assignment, we understand finding the highest activated prototype for a given feature map point. In the top row, we present the model trained with the diversity loss \mathcal{L}_J , while at the bottom without \mathcal{L}_J . One can observe that the diversity loss increases the utilization of prototypes by the ProtoSeg.

and Adam [39] optimizer with weight decay $5 \cdot 10^{-4}$ and $\beta_1 = 0.9$, $\beta_2 = 0.999$. We freeze the batch normalization parameters during training to avoid noisy normalization statistics due to small batch size. In the warmup phase, we use a constant learning rate of $2.5 \cdot 10^{-4}$ and train for $3 \cdot 10^4$ steps. In the joint training phase, we start with learning rate of $2.5 \cdot 10^{-5}$ for ResNet-101 weights and $2.5 \cdot 10^{-4}$ for ASPP and prototype layers. We employ the polynomial learning rate policy [13] with $power = 0.9$, training for $3 \cdot 10^4$ steps. In both fine-tuning phases, we use a constant learning rate equal to 10^{-5} and train for 2000 steps.

We run experiments on a single NVidia GeForce RTX 2080 GPU. For both datasets, the whole training procedure takes up to 48 hours. The code is written using PyTorch [55] and Pytorch Lightning [20] libraries.

Pascal VOC 2021. We evaluate ProtoSeg on PASCAL VOC 2012 segmentation benchmark [19] that consists of 1464 train, 1449 validation, and 1446 test images with pixel-level labels from 21 distinct classes, including 20 foreground classes and a background class. We use the offline augmented train_aug dataset with 10582 images provided in [29] for model training. However, we use the non-augmented training set for the prototype projection phase. We employ Multi-Scale inputs with max fusion (MSC) [13] using scales 0.5, 0.75, and 1.0. We set the image resolution to 321×321 pixels during training and evaluate on full images resized to 513×513 pixels for inference.

Cityscapes. We also test ProtoSeg on Cityscapes [16], a large-scale image segmentation dataset that contains 2975 train, 500 validation, and 1525 test images of street scenes.

Dataset	Method	Pretraining	mIOU	
			val	test
Pascal	DeepLabv2	COCO	77.69	79.70
	ProtoSeg	COCO	67.98	68.71
	ProtoSeg	ImageNet	72.05	72.92
Cityscapes	DeepLabv2	COCO	71.40	70.40
	ProtoSeg	COCO	55.35	56.77
	ProtoSeg	ImageNet	67.23	67.04

Table 1: Performance of ProtoSeg and the baseline method on the validation and test sets of PASCAL VOC 2012 and Cityscapes. The interpretability comes with a decrease in the mIOU. However, it is compensated with our diversity loss.

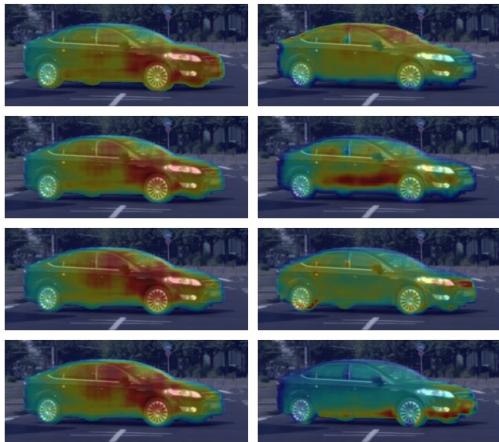


Figure 6: Activation maps of four prototypes from class *car* for ProtoSeg trained without (left column) and with \mathcal{L}_J (right column). In the former, the activations of different prototypes overlap, in contrast to the latter, where the diversity of prototypes increases.

Following the suggestions of dataset authors, we train and evaluate the model on 19 selected pixel classes and ignore the void class during training and evaluation. We use similar settings as those for PASCAL VOC 2012 with the following differences: lack of MSC following [13], training on random image crops of resolution 513×513 pixels, and evaluating on full images with the original resolution of 1024×2048 pixels.

5. Results

Table 1 presents the mean Intersection over Union (mIoU) scores obtained for the validation and test sets of

Dataset	λ_J	prototype overlap (mIOU)	mIOU
PASCAL	0.00	48.16	69.60
	0.25	26.59	72.05
Cityscapes	0.00	57.99	61.60
	0.25	24.09	67.23

Table 2: We analyze how the activation maps of two prototypes from the same class overlap each other. For this purpose, we binarize activation maps of all prototypes and calculate their mean IOU over all pairs of prototypes from the same class. The overlap is reduced by half after applying our \mathcal{L}_J , which numerically confirms increased variability of prototypes. We also report the segmentation mIOU score in the rightmost column.

PASCAL VOC 2012 and Cityscapes by the baseline method and ProtoSeg with $\lambda_J = 0.25$. We observe that the interpretability of ProtoSeg comes with a decrease in mIOU compared to the baseline method. It could be caused by the constraint introduced by prototypes and, in our opinion, can be improved with more extended hyperparameter search. In the Supplementary Materials we present an ablation study on ProtoSeg’s hyperparameters on Pascal VOC 2012, which shows the difficulty in bridging the gap to the baseline. ProtoSeg, in contrast to [13], yields better results with weights obtained from the model pretrained on ImageNet classification than COCO segmentation task. We hypothesize that prototypes learned on ImageNet representation can be more informative because they correspond to a more generic task of image classification, whereas representation after pretraining on COCO segmentation can focus on more task-specific features such as object borders. Finally, we did not apply CRF [13], which can further improve the accuracy.

Influence of prototype diversity loss. In Table 2 we compare the accuracy of ProtoSeg with and without applying \mathcal{L}_J . We observe that ProtoSeg achieves higher accuracy with $\lambda_J = 0.25$ than $\lambda_J = 0$. This could be attributed to the higher informativeness of diverse prototypes that leads to better generalization. To analyze this trend, we calculate an additional metric of prototypes overlapping, which we also present in Table 2. For this purpose, we binarize activation maps of all prototypes using 95th percentile and calculate the mean IOU of highly activated regions over all pairs of prototypes from the same class. This overlap is reduced by half after applying our \mathcal{L}_J . Hence, on average, for two prototypes from the same class, their highly activated regions have about 50% overlap when $\lambda_J = 0$ and only about 25% when $\lambda_J = 0.25$. Figure 6 presents activa-



Figure 7: Sample ProtoSeg segmentations on PASCAL VOC 2012 (left) and Cityscapes (right). ProtoSeg captures the overall object contours but may be inaccurate for fine-grained details. Note that pixels not considered in the evaluation are masked with black color in ground truth images.

Dataset	Training stage	Num prototypes	mIOU
PASCAL	warmup	210	25.65
	joint training	210	68.24
	projection	201	72.00
	pruning	133	72.05
Cityscapes	warmup	190	31.45
	joint training	190	65.38
	projection	188	67.24
	pruning	128	67.23

Table 3: Model performance after successive training stages for the validation set of PASCAL VOC 2012 and Cityscapes. The highest gain is achieved after joint training. However, the projection step is also beneficial. At the same time, pruning does not introduce significant performance improvement but removes around 30% of irrelevant prototypes.

tion maps of prototypes from class *car* for models trained with $\lambda_J \in \{0, 0.25\}$ on Cityscapes with ImageNet pretraining. We observe that the prototypes of the model trained with non-zero λ_J activate in semantically different regions, while the model with no diversity loss learns indistinguishable prototypes. To conclude, we observe that adding a non-zero \mathcal{L}_J increases the diversity of prototypes and allows for their interpretation as specific semantic object concepts.

Segmentation with interpretable prototypes. In Figure 7, we present examples of segmentation maps predicted by the models pretrained on ImageNet with $\lambda_J = 0.25$. Moreover, to exemplify that ProtoSeg finds semantically meaningful prototypes, in Figure 2 we draw activation maps of prototypes from class *cat* trained on PASCAL VOC 2012. We notice that the model learns prototypes representing the same semantic part concepts throughout different images. For instance, prototype 1 from Figure 2 activates on cat’s nose, and prototype 3 activates on outer rim of the cat’s mouth. However, some prototypes can carry low-level information; for example, prototype 2 activates around the edge of the cat. Activations of prototypes can also be used to segment an image into semantic concepts, as shown in Figure 8, where four prototypes of class *person* are activated on different object fragments and could be interpreted as pointing to specific parts of an object, such as legs, torso, or boundary between the outline of a person and background. We provide more examples showing the interpretability of prototypes in the Supplementary Materials.

Accuracy after different training stages. As described in Section 3.2, ProtoSeg employs a multi-stage procedure that affects model performance and the number of prototypes. In Table 3 we show mIOU scores of the models trained with ImageNet pretraining and $\lambda_J = 0.25$ after different training phases, as well as the number of unique prototypes. We see that the model needs joint training of all layers, including the backbone, to achieve satisfactory accuracy. We also note that projection and pruning phases have no negative effect on model performance, even though they substantially reduce the number of prototypes.

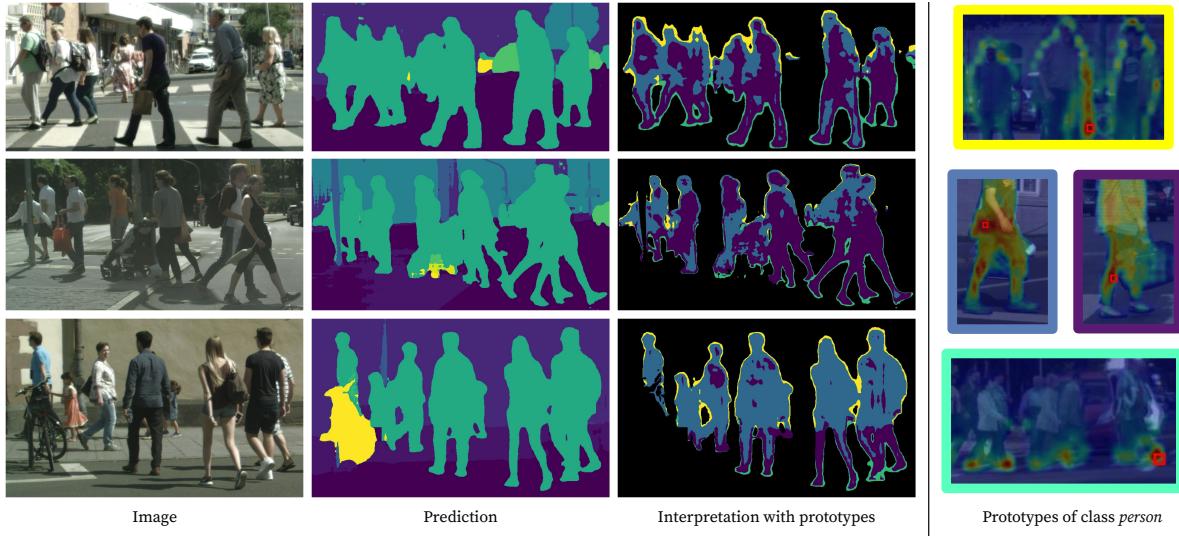


Figure 8: Sample segmentation (second column) of images (first column) and interpretation with four prototypes (third column) from class *person* obtained by ProtoSeg on Cityscapes. Interpretation with prototypes is acquired by assigning the prototype with the maximal activation to a considered pixel. Pictures in the right column show the four prototypes (activating on legs, torso or other fragments), and their frame colors correspond to the colors from the third column.

Model	mIOU	Pixel error
U-Net	78.74	0.0537
ProtoSeg (U-Net backbone)	76.58	0.0540

Table 4: Comparison between baseline U-Net and ProtoSeg with U-Net backbone on the EM segmentation challenge dataset. ProtoSeg achieves pixel error comparable to the baseline model while introducing interpretability of its predictions.

ProtoSeg with a different backbone. In this section, we show the adaptability of ProtoSeg to a different backbone model. We extend U-Net [58] with ProtoSeg and evaluate it on the EM segmentation challenge dataset from ISBI 2012 [1], which contains 30 pixel-labeled microscopy images of *Drosophila* larva. To perform the evaluation, we randomly divide the dataset into 20 training and 10 test samples. Our model achieves almost the same pixel error as U-Net (see Table 4), while introducing model transparency with prototypes. In the Supplementary Materials, we provide the details about the training of the U-Net-based methods and some prediction examples

6. Conclusions

In this work, we presented ProtoSeg, a model for semantic segmentation that constructs its decisions by referring to

prototypes found on the training set. Moreover, to increase the variability of prototypes within each class, we provide a novel diversity loss function. As presented in experiments conducted on various semantic segmentation datasets, we developed a method that allows for interpretation of obtained segmentation and achieves accuracy comparable to the baseline approaches.

The possible areas of future work include enhancing the precision of ProtoSeg and applying it on novel state-of-the-art segmentation architectures or more challenging segmentation tasks. We also see room for improvement in better prototype selection or sharing prototypes between classes.

Code availability. We made the code available at: <https://github.com/gmum/proto-segmentation>

Acknowledgements

The research of D. Rymarczyk was carried out within the research project “Bio-inspired artificial neural network” (grant no. POIR.04.04.00-00-14DE/18-00) within the Team-Net program of the Foundation for Polish Science co-financed by the European Union under the European Regional Development Fund. The work of M. Sacha, J. Tabor and B. Zieliński is supported by the National Centre of Science (Poland) Grant No. 2021/41/B/ST6/01370, and the work of Ł. Struski is supported by the National Centre of Science (Poland) Grant No. 2020/39/D/ST6/01332.

References

[1] EM segmentation challenge, ISBI 2012. <https://imagej.net/events/isbi-2012-segmentation-challenge>.

[2] Ehsan Abbasnejad, Damien Teney, Amin Parvaneh, Javen Shi, and Anton van den Hengel. Counterfactual vision and language learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10044–10054, 2020.

[3] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[4] Michael Anis Mihdi Afnan, Yanhe Liu, Vincent Conitzer, Cynthia Rudin, Abhishek Mishra, Julian Savulescu, and Mousoud Afnan. Interpretable, not black-box, artificial intelligence should be used for embryo selection. *Human Reproduction Open*, 2021.

[5] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[6] Anurag Arnab and Philip HS Torr. Bottom-up instance segmentation using deep higher-order crfs. *arXiv preprint arXiv:1609.02583*, 2016.

[7] Alina Jade Barnett, Fides Regina Schwartz, Chaofan Tao, Chaofan Chen, Yinhao Ren, Joseph Y Lo, and Cynthia Rudin. A case-based interpretable deep learning model for classification of mass lesions in digital mammography. *Nature Machine Intelligence*, 3(12):1061–1070, 2021.

[8] Dominika Basaj, Witold Oleszkiewicz, Igor Sieradzki, Michał Górszczak, B Rychalska, T Trzcinski, and B Zieliński. Explaining self-supervised image representations with visual probing. In *International Joint Conference on Artificial Intelligence*, 2021.

[9] Wieland Brendel and Matthias Bethge. Approximating CNNs with bag-of-local-features models works surprisingly well on imagenet. In *International Conference on Learning Representations*, 2019.

[10] Siddhartha Chandra and Iasonas Kokkinos. Fast, exact and multi-scale inference for semantic image segmentation with deep gaussian crfs. In *European conference on computer vision*, pages 402–418. Springer, 2016.

[11] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems*, pages 8930–8941, 2019.

[12] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[13] 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.

[14] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2613–2622, 2021.

[15] Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020.

[16] 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 *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[18] 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.

[19] 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>.

[20] William Falcon et al. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, 3, 2019.

[21] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2950–2958, 2019.

[22] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pages 3429–3437, 2017.

[23] 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.

[24] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[25] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

ference on computer vision and pattern recognition, pages 580–587, 2014.

[26] 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.

[27] Jiaqi Gu, Hyoukjun Kwon, Dilin Wang, Wei Ye, Meng Li, Yu-Hsin Chen, Liangzhen Lai, Vikas Chandra, and David Z Pan. Multi-scale high-resolution vision transformer for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12094–12103, 2022.

[28] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *European conference on computer vision*, pages 345–360. Springer, 2014.

[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, 2011.

[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.

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[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.

[34] Hanzhe Hu, Jinshi Cui, and Liwei Wang. Region-aware contrastive learning for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16291–16301, 2021.

[35] Harold Jeffreys. *The theory of probability*. OUP Oxford, 1998.

[36] Monish Keswani, Sriranjani Ramakrishnan, Nishant Reddy, and Vineeth N Balasubramanian. Proto2proto: Can you recognize the car, the way i do? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10233–10243, 2022.

[37] 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.

[38] Eunji Kim, Siwon Kim, Minji Seo, and Sungroh Yoon. Xprononet: Diagnosis in chest radiography with global and local explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15719–15728, 2021.

[39] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[40] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348. PMLR, 13–18 Jul 2020.

[41] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[42] Guosheng Lin, Chunhua Shen, Anton Van Den Hengel, and Ian Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3194–3203, 2016.

[43] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[44] 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.

[45] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015.

[46] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. In *Proceedings of the IEEE international conference on computer vision*, pages 1377–1385, 2015.

[47] 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.

[48] Diego Marcos, Sylvain Lobry, and Devis Tuia. Semantically interpretable activation maps: what-where-how explanations within cnns. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4207–4215. IEEE, 2019.

[49] Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[50] 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.

[51] Mohammadreza Mostajabi, Payman Yadollahpour, and Gregory Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3376–3385, 2015.

[52] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse

counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617, 2020.

[53] Meike Nauta et al. Neural prototype trees for interpretable fine-grained image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14933–14943, 2021.

[54] Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. Counterfactual vqa: A cause-effect look at language bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12700–12710, 2021.

[55] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[56] Sylvestre-Alvise Rebuffi, Ruth Fong, Xu Ji, and Andrea Vedaldi. There and back again: Revisiting backpropagation saliency methods. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8839–8848, 2020.

[57] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “why should i trust you?” explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[58] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[59] 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.

[60] Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16:1–85, 2022.

[61] Dawid Rymarczyk, Aneta Kaczyńska, Jarosław Kraus, Adam Pardyl, and Bartosz Zieliński. Protomil: Multiple instance learning with prototypical parts for fine-grained interpretability. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2022.

[62] Dawid Rymarczyk, Łukasz Struski, Michał Górszczak, Koryna Lewandowska, Jacek Tabor, and Bartosz Zieliński. Interpretable image classification with differentiable prototypes assignment. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[63] 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 International Conference on Knowledge Discovery & Data Mining*, pages 1420–1430, 2021.

[64] Alberto Santamaria-Pang, James Kubricht, Aritra Chowdhury, Chitresh Bhushan, and Peter Tu. Towards emergent language symbolic semantic segmentation and model interpretability. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 326–334. Springer, 2020.

[65] 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.

[66] Ramprasaath R Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Shalini Ghosh, Larry Heck, Dhruv Batra, and Devi Parikh. Taking a hint: Leveraging explanations to make vision and language models more grounded. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2591–2600, 2019.

[67] Abhishek Sharma, Oncel Tuzel, and David W Jacobs. Deep hierarchical parsing for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 530–538, 2015.

[68] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*. Citeseer, 2014.

[69] Gurmail Singh and Kin-Choong Yow. These do not look like those: An interpretable deep learning model for image recognition. *IEEE Access*, 9:41482–41493, 2021.

[70] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021.

[71] 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.

[72] Jiaqi Wang et al. Interpretable image recognition by constructing transparent embedding space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 895–904, 2021.

[73] Pei Wang and Nuno Vasconcelos. Scout: Self-aware discriminant counterfactual explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8981–8990, 2020.

[74] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[75] Enze Xie, Wenhui 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.

[76] Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-

aware concept-based explanations in deep neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20554–20565. Curran Associates, Inc., 2020.

- [77] Jianlong Yuan, Zelu Deng, Shu Wang, and Zhenbo Luo. Multi receptive field network for semantic segmentation. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1883–1892. IEEE, 2020.
- [78] Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Cheekong Lee. Protgnn: Towards self-explaining graph neural networks. 2022.
- [79] 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.
- [80] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021.