# OLAF: A Plug-and-Play Framework for Enhanced Multi-object Multi-part Scene Parsing

Pranav Gupta[1], Rishubh Singh[2,3], Pradeep Shenoy[3], and Ravi Kiran Sarvadevabhatla[1]

[1] IIIT Hyderabad
[2] Swiss Federal Institute of Technology (EPFL)
[3] Google Research
{pranav.gu@research.,ravi.kiran@}iiit.ac.in , rishubh.singh@epfl.ch, shenoypradeep@google.com

**Abstract.** Multi-object multi-part scene segmentation is a challenging task whose complexity scales exponentially with part granularity and number of scene objects. To address the task, we propose a plug-and-play approach termed OLAF. First, we augment the input (RGB) with channels containing object-based structural cues (fg/bg mask, boundary edge mask). We propose a weight adaptation technique which enables regular (RGB) pre-trained models to process the augmented (5-channel) input in a stable manner during optimization. In addition, we introduce an encoder module termed LDF to provide low-level dense feature guidance. This assists segmentation, particularly for smaller parts. OLAF enables significant mIoU gains of **3.3** (Pascal-Parts-58), **3.5** (Pascal-Parts-108) over the SOTA model. On the most challenging variant (Pascal-Parts-201), the gain is **4.0**. Experimentally, we show that OLAF's broad applicability enables gains across multiple architectures (CNN, U-Net, Transformer) and datasets. The code is available at olafseg.github.io

## 1 Introduction

Multi-object multi-part segmentation is a challenging task that involves simultaneously segmenting multiple objects in an image while also segmenting their individual parts. The task goes beyond conventional object segmentation [3,4,10, 33, 58, 71–73] and aims to enable multi-granular scene understanding. The availability of granular semantic detail is crucial for applications in robotics [47, 65], visual question answering [27], object interaction and modeling [1, 16] and other domains [2, 11, 15, 32] where understanding the scene in terms of objects and their constituent parts is crucial.

Related approaches primarily address simpler variants such as single-object part parsing [22, 36–38, 48, 61] or part parsing for objects with fewer or visibly larger parts [24]. Some recent methods [45, 55, 57, 74] have been developed to specifically tackle the more complex task of multi-object multi-part parsing. However, these suffer from three significant limitations:

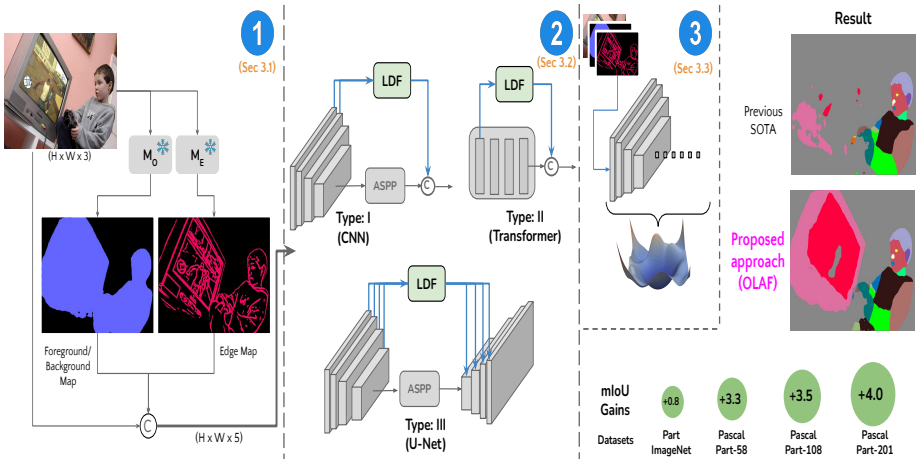


**Fig. 1:** The recipe for OLAF, our plug-and-play framework for enhanced multi-object multi-part scene parsing: 1 Augment RGB input with object-based channels (fg/bg, boundary edges) obtained from frozen pre-trained models ($M_O, M_E$) 2 Use Low-level Dense Feature guidance from segmentation encoder (LDF, shaded green) 3 Employ targeted weight adaptation for stable optimization with augmented input. We show that following this recipe leads to significant gains (up to **4.0** mIoU) across multiple architectures and across multiple challenging datasets.

*Limitation 1:* Foreground (union of object regions) is often incorrectly segmented, impacting the constituent part segmentation (Figure 2, first row). *Limitation 2:* Crucial boundary details between objects and parts are not captured accurately (Figure 2, second row). *Limitation 3*: Small and thin parts especially fail to be segmented (Figure 4, Figure 5).

To address *Limitation 1 & 2*, we first obtain a plausible boundary edge mask using a pre-trained network. We use another pre-trained network to obtain preliminary object segmentation and combine the object label channels to obtain a binary foreground mask. These masks are included as additional channels to constitute the 5-channel (3 RGB + 2 masks) input to the reference segmentation network (see 1 in Figure 1). The masks provide an inductive bias and guide the model to focus on relevant parts from the onset of training. We also propose a weight adaptation technique that enables the pre-trained segmentation encoder to process the new 5-channel input without destabilizing optimization (3 in Figure 1).

To address *Limitation 3* (i.e. small and thin parts), we introduce an encoder module termed Low-level Dense Feature (LDF) - see 2 in Figure 1. This module, in conjunction with augmented input representation, provides low-level dense feature guidance enabling better segmentation, especially for small/thin parts.

To summarize our contributions:

– **Input Augmentation**: We introduce an augmented 5-channel input representation with auxiliary channels containing object and boundary cues.

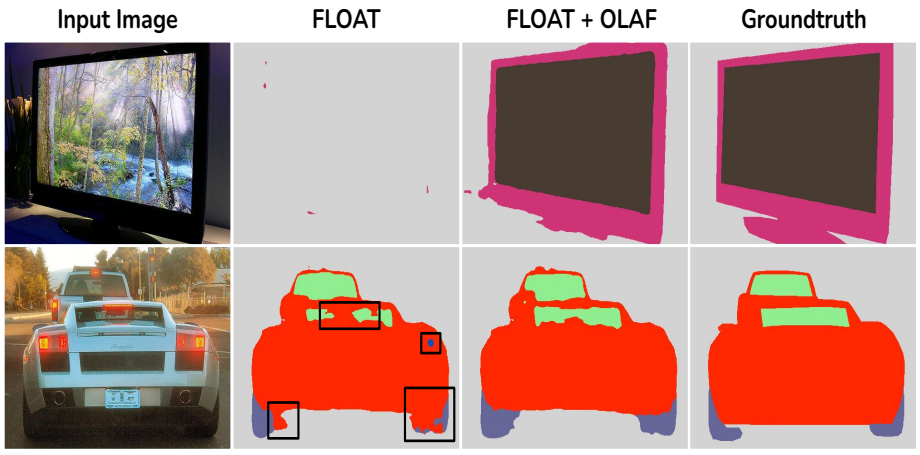


**Fig. 2:** The segmentation results for state-of-the-art approach FLOAT [55] and its limitations can be seen in the second column. In the first row, FLOAT completely fails to identify *TV Frame* and *TV Screen*. In the second row, FLOAT fails to capture the edge partition between *Car-Body*, *Car-Tire* and also between *Car-Body*, *Car-Window*. The third column shows results by incorporating our plug-and-play approach OLAF into FLOAT, leading to significantly improved object and part segmentation results.

- **Weight Adaptation Technique**: We introduce a targeted weight-adaptation training procedure that ensures stable optimization of pre-trained backbones on the augmented (5-channel) input.
- **Low-Level Dense Feature Guidance (LDF)**: We propose a generic encoder module called LDF which provides valuable low-level dense feature guidance, especially for small part segmentation.
- **Performance Boost**: OLAF achieves significant mIoU improvements, surpassing state-of-the-art by **3.3** on Pascal-Parts-58, **3.5** on Pascal-Parts-108, and **4.0** on Pascal-Parts-201.
- **Generalizability**: We show that OLAF enhances performance across multiple representative segmentation families (CNN, U-Net, Transformer) and multiple datasets (Pascal-Parts 58/108/201 and PartImageNet), suggesting broad applicability as a plug-and-play framework.

## 2 Related Work

**Single-Object Multi-Part Segmentation** considers a single object and its constituent parts. Most works explore segmentation of non-rigid object categories such as person [13, 19, 21, 35, 36, 38, 42, 49, 68], some animals [22, 59, 61] and rigid object categories (e.g. vehicles [37, 41, 48, 56]). Some works have also examined open-world part segmentation [50, 64]. However, the single object condition is restrictive and not representative of general scenes which contain multiple objects from distinct categories and associated occlusions.

**Multi-Object Multi-Part Segmentation** has recently got increased attention due to its complexity and importance in downstream applications. There have been multiple different approaches to tackle this hard problem [45,55,57,74]. The initial work of Zhao et al. [74] and the works of Micheli et al. [45, 46] leverage object and boundary awareness through auxiliary tasks and model design changes. The current state of the art approach (FLOAT [55]) employs label-space factorization to reduce the number of output heads. Typically, existing approaches do not attempt segmentation of very small/thin parts although annotations are available [55]. Beyond the popular datasets (Pascal-Parts-58), our approach enables improved performance for the harder variants (Pascal-Parts-108, Pascal-Parts-201).

**Object level guidance** and **Boundary/edge awareness** is typically present as an auxiliary task or in terms of guidance from an object network's features [5, 44–46, 50, 54, 74, 75]. For multi-object multi-part segmentation, Zhao et al. [74] add an auxiliary task of predicting object segments from the learned part segmentation representation. Michieli et al. [45] use the features from the last layer of an object segmentation network as guidance to the part segmentation decoder. In contrast, our work OLAF adds object segmentation and edge information directly as additional channels to the input which is observed to be more beneficial.

**Low-level Feature Guidance** has been used in previous works [8,23,51,70] to enhance the performance of segmentation by leveraging low-level visual (spatial) cues. While some works incorporate skip-connections [7, 29], others utilize downsampling strategies along with skip-connections to obtain sufficient receptive field for context capturing [51,70]. While this strategy generally works well, it may not be suitable for tasks such as part segmentation because the information in low-level features is too coarse. In particular, downsampling compromises details of small or thin parts. By contrast, our approach strives to efficiently exploit low-level cues in the most beneficial manner while also capturing the semantics of small or thin parts.

## 3 Methodology of OLAF

OLAF introduces structural modifications at two key aspects of the standard segmentation pipeline (see Fig. 3). The first change occurs at input stage, where we enrich RGB input with auxiliary channels containing object-based structural cues, including foreground/background masks and boundary edge masks. The second structural adjustment takes place within the encoder. We introduce a dedicated module termed LDF which provides low-level dense feature guidance to benefit the segmentation of smaller parts. In addition to these structural enhancements, we introduce a weight adaptation technique which ensures that pre-trained RGB (3-channel) backbones seamlessly adapt to augmented (5-channel) input during optimization. In this section, we provide a detailed explanation of these crucial components.

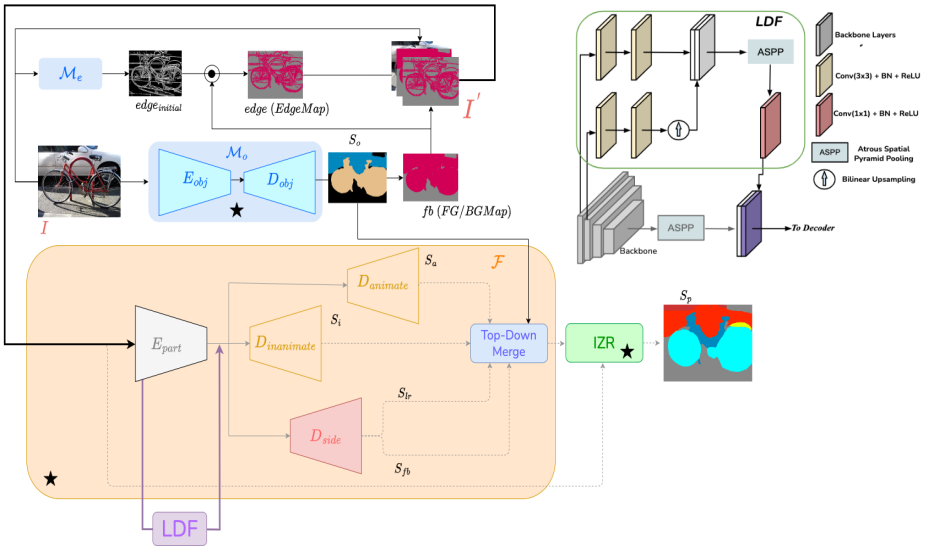


**Fig. 3:** Illustration of OLAF's architectural integration with FLOAT [55](Sec. 3). FLOAT's components are tagged with ★. The object masks from output $S_o$ of object segmentation network $\mathcal{M}_o$ are merged to obtain the foreground map $fg$. The output of edge generation network $\mathcal{M}_e$ is thresholded and filtered using $fg$ to obtain edge map $edge$. The obtained maps are stacked with input image $I$ to obtain the 5-channel input $I^{'}$ for the part segmentation network $\mathcal{F}$. The interface for LDF (Sec. 3.2) with encoder $E_{part}$ and its architecture (top right) are also shown. A similar integration of OLAF also exists for U-Net style and Transformer style architectures.

### 3.1 Foreground and Edge masks as boundary cues

The inclusion of foreground (union of object regions) guides the part predictions to lie within object boundaries. To obtain the corresponding channel mask, we use a state-of-the-art object segmentation network [7] to obtain object predictions. These are transformed into a binary foreground/background mask.

$$fb(x,y) = \begin{cases} 1, & \text{if } P(x,y) \in C \text{ and } P(x,y) \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

where $C$ denotes the predicted object class and pixel location is $(x, y)$.

Edges play a crucial role in delineating boundaries between different objects, parts and recognizing intricate details within the scene. To obtain a collection of such edges, we use the Holistically-Nested Edge Detection (HED) [67] model. The raw output $edge_{\text{initial}}$ from HED lies in the range $[0, 1]$ and contains edges from background as well. To filter out these background edges, we employ the fg/bg mask as follows:

$$edge = \mathbb{I}[edge_{initial} > 0] \odot fb$$

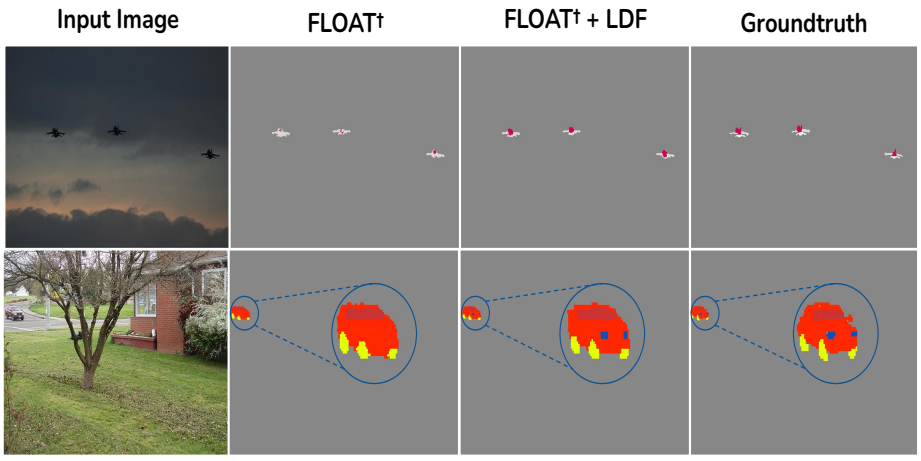


**Fig. 4:** LDF (Section 3.2) consistently improves the performance of small/thin parts. As shown in Row I, FLOAT [55] with LDF successfully predicts Aeroplane-Body while FLOAT fails to do so adequately. Similar results can be seen in Row II, where LDF successfully predicts Car-Light which FLOAT completely misses.

where *edge* represents foreground edges and $\odot$ denotes element-wise multiplication and $\mathbb{I}$ is the indicator function. Note that *edge* mask is binary, i.e. pixel location where there is an *edge* has pixel value 1 otherwise 0.

To prepare the input, we append the foreground and edge masks as separate channels to the input image $I$. More precisely, the original input image $I$ with dimensions $H \times W \times 3$ is transformed into a modified input $I'$ with dimensions $H \times W \times 5$ (see 1 in Figure 1).

Conventional segmentation approaches typically include auxiliary tasks to learn foreground/background [44] and edges during training [74]. However, directly including foreground/background and edges as part of the input can be thought of as a structural inductive bias for the task. These masks provide strong boundary cues throughout the optimization process. In addition, they eliminate the issue of irregular gradient flow arising from ad-hoc scaling of task-related losses [14] in existing (RGB input only) approaches.

### 3.2 Low-level Dense Feature Extractor (LDF)

Typically, encoders in segmentation architectures [3, 7, 63, 70, 73] process image representations in a downsampled feature space (often $1/8^{th}$ or $1/16^{th}$ size of input image). However, aggressive downsampling and intermediate pooling operations lead to the loss of fine details and small entity instances in an image. This effect is more pronounced for parts since they cover relatively smaller pixel areas compared to scene objects.

To address this issue, some architectures either employ skip-connections [7] or directly concatenate output from early backbone blocks with the decoder [28].

However, despite these methods allowing flow of low level features to the decoder, they still fail to segment small/thin parts. This is because information from early stages of the encoder is too coarse and lacks contextual information for accurate segmentation of such parts.

To overcome these challenges, we introduce the Low-level Dense Feature Extractor (LDF) module. LDF leverages early blocks of the backbone network, where low-level information associated with small/thin parts are more prominent. To capture dense features of these small/thin parts, LDF includes (a) convolutional layers to enhance the features extracted from the initial stages of the backbone (b) an upsampling layer to maintain consistent feature map size (c) Atrous Spatial Pyramid Pooling (ASPP) [7] to capture contextual information at multiple scales (see Figure 3). This enables the model to extract dense low-level features at various spatial resolutions and consider context at different scales, including context relevant to small/thin parts. LDF can be formalized as:

$$feat(x_1, x_2) = Conv_{3\times3}(x_1) \copyright UP(Conv_{3\times3}(x_2))$$
$$LDF(x_1, x_2) = Conv_{1\times1}(ASPP(feat(x_1, x_2)))$$

where $x_1$ and $x_2$ are the features from the first and second block of the backbone, $Conv_{3\times3}$ represents a $3 \times 3$ convolution with *stride = 1* and *padding = 1* to avoid dimension reduction, $Conv_{1\times1}$ represents $1 \times 1$ convolution. $ASPP(.)$ represents Atrous Spatial Pyramid Pooling [7] and $\copyright$ represents the concatenation operation. $UP(.)$ represents applying an Upsample Convolution layer which applies an upsampling step with a scale factor, followed by a $Conv_{1\times1}$, batch normalization, and a ReLU activation.

Similar to input channel augmentation (Section 3.1), LDF can also be viewed as imposing a structural inductive bias for smaller and thin parts. LDF enables noticeable improvements in performance for such parts (see Figure 4,Table 4).

### 3.3 Weight Adaptation

Standard pre-trained segmentation models support 3-channel RGB images as input. To efficiently adapt existing models for our modified 5 channel input, we employ a simple but effective technique. As the initial step, for filters in the first layer, we average their weights across the channel dimension. The result is used to initialize the weights of the filters corresponding to the two newly included input channels (i.e. fg/bg and edges). To ensure stable training, the optimization contains a warm-up phase consisting of $n_{warm}$ epochs. Compared to alternative schemes (Section 5.3), this approach prevents weight updates that lead to instability or divergence during the initial stages of training

## 4 Experiments

### 4.1 Datasets

**Pascal-Part Dataset Variants:** The Pascal-Parts dataset contains 4,998 training samples and 5,105 samples for testing [18]. The dataset contains part level

annotation for the 20 Pascal VOC2010 semantic object classes (including the background class). We follow Singh et al. [55] and evaluate OLAF on three variants of Pascal-Part, which are Pascal-Part-58, Pascal-Part-108 and Pascal-Part-201 in increasing order of complexity. Pascal-Part-58 contains 58 part classes, mostly focusing on larger parts of objects such as heads, torsos, legs for animals, and components such as body, wheels for non-living objects. Pascal-Part-108 is more challenging, featuring 108 part classes. It includes smaller parts such as eyes, necks, feet for animals, and roofs, doors for non-living objects. Pascal-Part-201 [55] is the most challenging version among the Pascal-Part dataset variants. It includes 201 part classes and introduces additional part attributes 'left', 'right', 'front', 'back', 'upper', 'lower', along with minor parts (e.g. 'eyebrows'), which are not present in the other variants (58/108).
**PartImageNet [25]:** This is a large-scale dataset containing 24,095 images. We follow the official training/validation split and evaluate performance on the publicly available validation set.

### 4.2 Evaluation Metrics

We use the standard mean Intersection over Union (mIoU) score as a performance measure. mIoU tends to be influenced more by the contributions of "larger" part instances. Therefore, we also report sqIoU [55], designed specifically for fairer balance between small and larger parts. For comparison with existing works [24], we report average pixel accuracy as a metric for PartImageNet [25].

### 4.3 Training Details

To show the effectiveness of our approach for multi-object multi-part scene parsing, we apply the recipe for OLAF (see Figure 1) on BSANet [74], GMNet [45], Deeplabv3 [7] and the current state-of-the-art FLOAT [55]. For training these models, we consider all the Pascal-Part dataset variants. We also apply OLAF to DeepLabV3+ [7], Segformer [66] trained on PartImageNet [25] dataset. We also experiment with different backbones - ResNet-50 [26], Swin Transformer [43] and MiT-B2 [66]. We apply the same hyperparameters, training strategy, augmentations and pretrained backbones used in the respective methods[4]. For weight adaptation (Section 3.3), $n_{warm}$ is set to 5. All the experiments are conducted on clusters with NVIDIA A100 GPUs.

## 5 Experimental Results

### 5.1 Pascal-Part-58, 108 and 201

Table 1 presents performance metrics for the most challenging variant of the Pascal-Part dataset, Pascal-Part-201. When applying OLAF to state-of-the-art model FLOAT [55], we observe significant improvements: a **4.0** increase in **mIoU**

[4] The training details can be accessed from the respective papers.

**Table 1:** Pascal-Part-201 results (mIoU/mAvg - top table and sqIoU/sqAvg - bottom table). "+ O": augmented with OLAF, "†": with ViT-H backbone.

| Model | bgr | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | TV | **mIOU** | **mAvg** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DeepLabv3 [7] | 91.0 | 31.6 | **47.7** | 24.3 | 56.7 | 46.4 | 31.0 | 36.7 | 24.2 | 35.6 | 17.5 | 38.6 | 27.3 | 20.7 | 38.0 | 26.9 | 50.8 | 13.3 | 42.1 | 14.7 | 57.6 | 26.3 | 36.8 |
| **Deeplabv3 + O** | **93.6** | **35.3** | 42.9 | **27.2** | **65.5** | **51.0** | **33.3** | **38.4** | **25.8** | **42.1** | **20.5** | **46.9** | **29.4** | **22.6** | **39.3** | **28.9** | **56.1** | **15.5** | **46.8** | **17.2** | **65.2** | **28.6** | **40.2** |
| GMNet [45] | 90.8 | 26.6 | 33.1 | 21.2 | 55.0 | 43.5 | 24.6 | 27.5 | 21.7 | 35.5 | 15.1 | 40.3 | 25.0 | 17.5 | 31.9 | 21.9 | 44.2 | 11.9 | 43.3 | 14.0 | 53.2 | 22.5 | 33.2 |
| **GMNet + O** | **93.1** | **28.5** | **35.9** | **23.5** | **64.5** | **47.0** | **26.7** | **29.5** | **26.4** | **41.9** | **19.0** | **47.9** | **29.1** | **20.3** | **34.0** | **26.3** | **49.8** | **16.9** | **54.8** | **15.9** | **60.5** | **26.0** | **37.7** |
| BSANet [74] | 91.2 | 34.6 | 41.7 | 27.9 | 61.2 | 51.7 | 34.1 | 38.1 | 26.1 | 35.4 | 24.0 | 43.6 | 28.4 | 23.0 | 37.4 | 27.7 | 54.7 | 14.3 | 40.4 | 17.8 | 59.4 | 28.5 | 38.7 |
| **BSANet + O** | **92.9** | **35.9** | **43.1** | **29.9** | **69.8** | **53.3** | **35.8** | **39.8** | **28.9** | **39.8** | **26.6** | **49.2** | **31.2** | **24.9** | **38.9** | **31.1** | **59.5** | **18.7** | **49.9** | **19.0** | **65.8** | **31.1** | **42.1** |
| FLOAT [55] | 92.5 | 36.7 | 42.6 | 34.4 | 75.3 | 51.4 | 35.8 | 42.0 | 37.8 | 59.6 | 35.5 | 58.2 | 41.0 | 34.0 | 40.2 | 40.8 | 52.2 | 28.5 | 69.0 | 15.1 | 56.1 | 36.9 | 46.6 |
| **FLOAT + O** | **93.3** | **38.7** | **45.1** | **37.4** | **76.7** | **54.1** | **38.7** | **46.2** | **41.4** | **59.7** | **38.9** | **58.3** | **43.1** | **35.2** | **42.3** | **44.6** | **61.0** | **31.5** | **69.1** | **16.4** | **69.3** | **39.8** | **49.6** |
| FLOAT† | 93.2 | 36.9 | 44.5 | 36.8 | 76.4 | 32.1 | 36.3 | 44.2 | 39.3 | 59.3 | 36.8 | 58.1 | 42.7 | 34.1 | 41.2 | 43.2 | 56.7 | 30.2 | 68.9 | 15.2 | 59.4 | 37.7 | 46.9 |
| **FLOAT† + O** | **93.7** | **39.9** | **45.6** | **38.3** | **77.1** | **55.3** | **39.3** | **47.9** | **42.2** | **60.7** | **39.4** | **59.6** | **44.2** | **37.3** | **44.2** | **47.1** | **63.5** | **32.2** | **69.6** | **16.9** | **70.9** | **40.9** | **50.7** |

| | bgr | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | TV | **sqIOU** | **sqAvg** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Deeplabv3 [7] | 89.6 | 28.9 | **39.3** | 17.1 | 57.4 | 32.3 | 27.1 | 26.0 | 20.5 | 39.8 | 14.8 | 34.7 | 22.7 | 17.2 | 31.5 | 19.2 | 34.9 | 10.8 | 52.6 | 14.4 | 53.8 | 21.5 | 32.6 |
| **Deeplabv3 + O** | **93.2** | **33.4** | 36.5 | **21.0** | **67.2** | **38.0** | **30.0** | **28.8** | **23.2** | **47.3** | **18.8** | **44.1** | **25.8** | **19.9** | **33.5** | **22.1** | **41.2** | **13.7** | **58.3** | **17.9** | **62.4** | **24.8** | **37.0** |
| GMNet [45] | 89.4 | 20.7 | 23.5 | 12.6 | 53.1 | 25.8 | 19.3 | 17.2 | 18.1 | 38.2 | 11.2 | 35.2 | 15.9 | 14.2 | 25.4 | 13.8 | 26.9 | 8.5 | 52.0 | 13.8 | 46.9 | 16.9 | 27.7 |
| **GMNet + O** | **91.4** | **23.8** | **27.3** | **16.5** | **63.6** | **30.2** | **23.7** | **20.9** | **22.7** | **45.7** | **15.8** | **43.6** | **23.0** | **17.4** | **28.5** | **19.1** | **33.5** | **14.9** | **64.5** | **17.3** | **56.0** | **21.4** | **33.3** |
| BSANet [74] | 89.9 | 30.7 | 33.5 | 18.6 | 60.2 | 31.2 | 29.2 | 26.4 | 21.2 | 37.8 | 17.5 | 38.0 | 22.3 | 17.8 | 31.2 | 18.2 | 33.6 | 10.8 | 47.2 | 17.5 | 55.4 | 22.1 | 32.8 |
| **BSANet + O** | **91.4** | **32.9** | **35.7** | **21.6** | **69.8** | **34.2** | **31.8** | **29.1** | **25.0** | **43.2** | **21.0** | **44.6** | **26.1** | **20.7** | **33.3** | **22.5** | **39.3** | **16.2** | **57.7** | **19.6** | **62.8** | **25.7** | **37.1** |
| FLOAT [55] | 90.8 | 32.5 | 35.8 | 24.5 | 63.9 | 36.1 | 30.4 | 29.9 | 33.0 | 50.8 | 28.1 | 47.6 | 35.6 | 26.1 | 33.6 | 29.9 | 34.5 | 20.6 | 69.0 | 13.6 | 56.8 | 29.5 | 39.2 |
| **FLOAT + O** | **91.5** | **34.9** | **39.0** | **27.2** | **65.1** | **39.1** | **34.6** | **34.0** | **36.3** | **50.9** | **32.5** | **47.7** | **37.7** | **30.0** | **35.9** | **33.1** | **38.7** | **22.9** | **69.2** | **13.8** | **65.2** | **32.5** | **41.9** |
| FLOAT† | 91.4 | 32.6 | 37.5 | 26.4 | 65.2 | 16.9 | 40.2 | 32.4 | 34.2 | 50.6 | 29.6 | 47.5 | 37.3 | 26.4 | 34.5 | 32.6 | 39.3 | 22.6 | 68.8 | 13.9 | 60.5 | 30.8 | 40.0 |
| **FLOAT† + O** | **92.3** | **36.4** | **39.7** | **28.5** | **65.6** | **40.6** | **35.5** | **35.7** | **37.4** | **52.2** | **33.4** | **49.3** | **38.9** | **32.3** | **37.9** | **35.8** | **41.6** | **23.7** | **69.7** | **14.4** | **66.9** | **34.3** | **43.2** |

**Table 2:** Pascal-Part-58&108 segmentation results. '+ O':augmented with our approach OLAF, '†': with ViT-H backbone.

| Method | 58 | | | | 108 | | | |
|---|---|---|---|---|---|---|---|---|
| | mIOU | mAvg | sqIOU | sqAvg | mIOU | mAvg | sqIOU | sqAvg |
| Deeplabv3 [7] | 54.3 | 55.4 | 46.0 | 48.4 | 41.3 | 43.6 | 32.2 | 36.1 |
| **Deeplabv3 + O** | **59.0** | **61.6** | **52.1** | **55.6** | **46.4** | **51.5** | **39.2** | **45.2** |
| BSANet [74] | 58.2 | 58.9 | 49.3 | 51.5 | 45.9 | 48.4 | 36.6 | 41.0 |
| **BSANet + O** | **59.8** | **61.7** | **51.9** | **55.3** | **47.1** | **50.3** | **38.7** | **43.5** |
| GMNet [45] | 59.0 | 61.8 | 49.4 | 54.3 | 45.8 | 50.5 | 35.8 | 41.9 |
| **GMNet + O** | **60.2** | **63.4** | **51.6** | **55.5** | **47.2** | **52.1** | **38.5** | **44.8** |
| HIPIE [62] | 63.8 | 67.1 | 57.2 | 60.7 | - | - | - | - |
| FLOAT [55] | 61.0 | 64.2 | 54.2 | 57.1 | 48.0 | 53.0 | 40.5 | 45.6 |
| **FLOAT + O** | **62.7** | **66.1** | **55.4** | **58.5** | **50.3** | **55.3** | **43.4** | **48.4** |
| FLOAT† | 62.1 | 65.5 | 55.8 | 58.9 | 48.9 | 54.2 | 41.6 | 46.7 |
| **FLOAT† + O** | **64.3** | **68** | **57.7** | **60.8** | **51.5** | **56.9** | **45.0** | **49.9** |

**Table 3:** Results on validation set of the PartImageNet [25] dataset. The backbone for each approach is specified separately. "+ O": augmented with our proposed method OLAF.

| Method | Backbone | mIOU | mAcc |
|---|---|---|---|
| Deeplabv3+ [9] | ResNet-50 [26] | 57.53 | 71.07 |
| Compositor [24] | ResNet-50 [26] | 61.44 | 73.41 |
| **Deeplabv3+ + O** | ResNet-50 [26] | **61.71** | **74.26** |
| Segformer [66] | MiT-B2 [66] | 60.52 | 71.62 |
| Compositor [24] | Swin-T [43] | 64.64 | 78.31 |
| **Segformer + O** | MiT-B2 [66] | **65.46** | **79.10** |

and a **4.8** increase in **sqIoU** compared to FLOAT. This improvement is particularly noteworthy given that (a) Pascal-Part-201 is characterized by numerous small parts (b) object categories in this dataset have subtle intra and inter-category part label variations (e.g. 'left front leg'/'right front leg' in `horse`, `cow` etc. and 'right leg' in `person`).

In both Pascal-Part-58 and Pascal-Part-108 (see Table 2), our approach consistently outperforms the baselines. Specifically, FLOAT with OLAF exhibits improvements of **3.3** in **mIoU** and **3.5** in **sqIoU**. Similarly, in Pascal-Part-108, FLOAT achieves substantial improvements: **3.5** in **mIoU** and **4.5** in **sqIoU**.

**Table 4:** Ablation experiments on PASCAL-Part-201 (Section 5.3). We use FLOAT$^\dagger$ as the base model.

| | LDF | Edge-Map | Fg/Bg-Map | mIoU | mAvg | sqIoU | sqAvg | mIoU$_{small}$ |
|---|---|---|---|---|---|---|---|---|
| Input Channel Presence and Architectural Changes | – | – | – | 37.7 | 46.9 | 30.8 | 40.0 | 24.0 |
| | ✓ | | | 38.8 | 48.1 | 31.8 | 41.0 | 25.7 |
| | | ✓ | | 38.9 | 48.2 | 32.2 | 41.3 | 24.5 |
| | | | ✓ | 39.1 | 48.3 | 32.0 | 41.2 | 24.6 |
| | | ✓ | ✓ | 39.2 | 48.3 | 32.2 | 41.4 | 24.8 |
| **OLAF** | ✓ | ✓ | ✓ | **40.9** | **50.5** | **34.3** | **43.2** | **26.9** |
| fg map baseline | Segment Anything (SAM) [31] | | | 40.5 | 50.2 | 34.0 | 42.8 | 26.4 |
| Edge Map baselines | EDTER [52] | | | 39.5 | 49.1 | 33.1 | 41.9 | 24.9 |
| | Canny [6] | | | 39.0 | 48.8 | 32.6 | 41.6 | 24.3 |
| Added depth map baselines | Marigold [30] | | | 40.7 | 50.2 | 34.2 | 43.1 | 25.1 |
| | Depth Anything [69] | | | 40.8 | 50.4 | 34.2 | 43.2 | 25.2 |
| Optimization (Input Layer Weight Adaptation) | Random-5 | | | 35.2 | 44.2 | 28.4 | 37.4 | 19.7 |
| | Average-RGB-5 [60] | | | 36.3 | 45.5 | 29.2 | 38.9 | 20.5 |
| | Adapt-n-Freeze [17] | | | 38.2 | 47.1 | 31.3 | 39.5 | 21.7 |
| | Random-2 | | | 40.2 | 49.6 | 33.0 | 41.6 | 24.3 |

### 5.2 PartImageNet

For PartImageNet [25] (Table 3), OLAF augmented DeepLabV3+ [9] outperforms both DeepLabV3+ and state-of-the-art Compositor [24] with large improvements in mean accuracy. The results suggest that performance improvement is prominent for more recent, modern backbones (Swin-T [43], MiT-B2 [66]). The results also suggest that OLAF's methodology generalizes well to enable gains across datasets and architectural frameworks.

### 5.3 Ablation Studies

We conduct extensive ablation experiments with current state-of-the-art model FLOAT [55] trained on Pascal-Part-201 (Table 4).

**Input channel presence and architectural changes:** Considering the LDF module alone, there is an initial improvement in mIoU from 37.7 to 38.8. The introduction of edge and foreground/background cues improves the mIoU to 39.2. However, the most substantial gains are observed when all components are combined, achieving an mIoU of 40.9.

Similar to the convention from COCO [39] for objects, we define 'small' parts as those smaller than $25 \times 25$ pixels by area. We report the corresponding measure ($mIoU_{small}$) in Table 4 (last column). Clearly, the inclusion of LDF enables noticeable gains for small parts compared to input channels' inclusion.

**Input channel baselines:** As a baseline for foreground map input channel, we used the foreground map obtained by combining object mask outputs from Segment Anything (SAM) [31] but this led to a slightly lower performance.

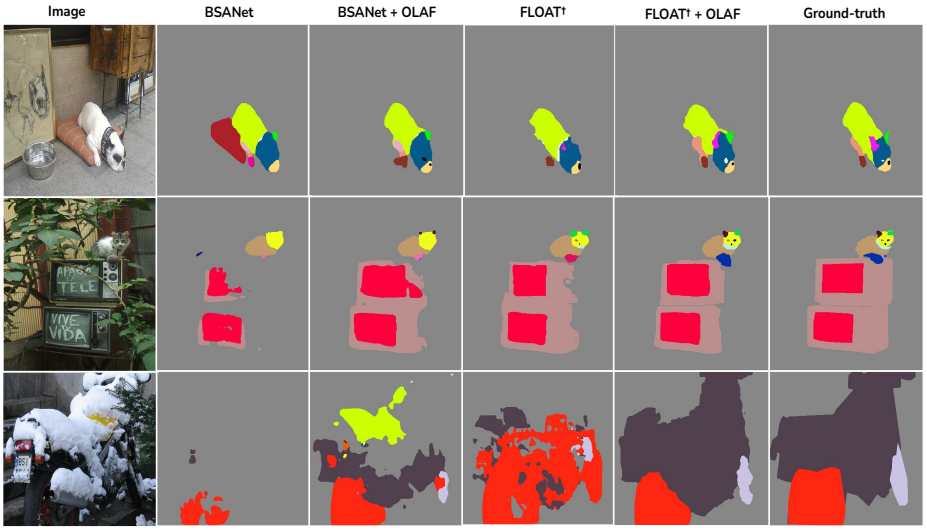


**Fig. 5:** Qualitative comparison on Pascal-Part-201. OLAF consistently improves the performance of previous methods (BSANet [74], FLOAT [55]). This is especially seen for small parts as shown in Row 1 (eyes, ears, nose and right-front-leg), Row 2 (eye, ears, nose, mouth and tail) and occluded parts as shown in Row 3 (parts of the `motorbike`).

Empirically, we observed that SAM does not accurately segment certain object categories such as `potted plant`, and tends to omit tiny parts of certain classes (e.g. tail of `airplane`, `bird`, `cat`, `cow`, `dog`, `horse`, `sheep`).

For edge map, we explored Canny [6] and EDTER [53] as alternatives. Compared to our default choice (HED [67]), these choices fail to strike the right balance in terms of boundary edge density – Canny maps contain too few semantically crucial edges while EDTER maps tend to be too dense. See Supplementary for examples illustrating these observations.

Intrigued by the performance enhancement capabilities of additional input channels, we considered adding depth maps obtained from monocular depth estimation approaches (Marigold [30], Depth Anything [69]) as additional input channels. In effect, this increased the number of channels to 6. The weight adaptation procedure was applied as described earlier (Sec. 3.3). But the inclusion of depth map did not lead to performance gains. A likely reason could be that depth cues are likely more useful for differentiating objects, particularly when the scene has a large depth of field. But for intra-object parts, depth might not vary much. Consequently, depth map might not be as effective for aiding part segmentation.

**Weight Adaptation:** To examine the effect of our weight adaptation procedure at input layer (Section 3.3), we conducted experiments with alternate schemes as described below.

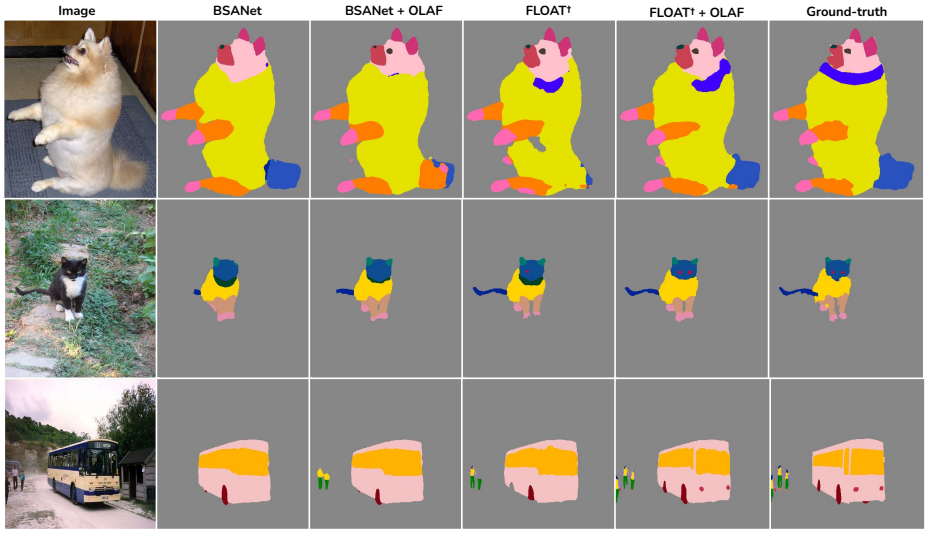


**Fig. 6:** Qualitative comparison on Pascal-Part-108. OLAF consistently improves the performance of previous methods (BSANet [74], FLOAT [55]). This is especially seen for small parts as shown in Row 1 (eyes and tail), Row 2 (eyes, ears, front-paw and tail) and Row 3 (wheel, headlight and very tiny parts of humans).

- *Random-5*: Retain weights of the entire backbone except for the input layer. The input layer dimension is reconfigured from 3 channel input to 5 channel input and weights for this layer's filters are initialized randomly.
- *Random-2*: Retain weights of the entire backbone, including those for the 3 channels in the original (RGB) input layer. The weights for two newly added channels in the input layer are initialized randomly.
- *Average-RGB-5* [60]: Average the channel-wise weights of the original (RGB) network's input layer. Initialize all 5 channels with this average.
- *Adapt-n-Freeze* [17]: First, include a convolution layer to match augmented input (5 filters) and then include $1 \times 1$ filter so that output (3 channels) is compatible with base (RGB) network. Freeze the base network and train the adapter layers (Conv2D, $1 \times 1$). Then unfreeze and train the entire layer augmented base network together.

To ensure fair and consistent comparison, all methods used a warm-up phase consisting of $n_{warm} = 5$ epochs. As shown in Table 4, our proposed approach provides the best performance. In practice, we found that *Random-5* method resulted in unstable learning. This instability arises from initializing the input layer with random weights, causing erratic gradient flow and impacting the pretrained weights of subsequent layers in the backbone. While *Random-2* had stable optimization, the performance suffers from erratic gradients due to random initialization. The other adaptation schemes (*Average-RGB-5* [60], *Adapt-n-Freeze* [17])

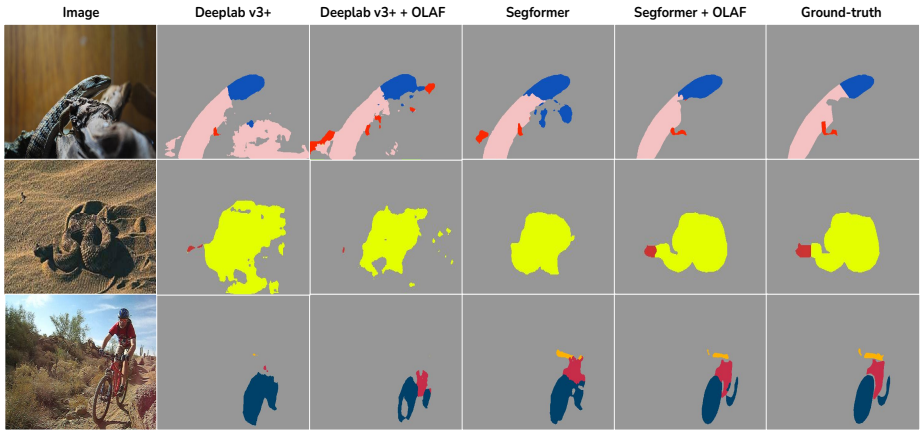


**Fig. 7:** Qualitative comparison for images from PartImageNet [25]. OLAF consistently improves the segmentation quality, especially for the harder small parts of objects.

also had similar instability issues. In contrast, our weight adaptation approach ensures stable optimization and distinctly improved performance.

Overall, the ablation experiments suggest that all the ingredients of our recipe — object-based channels, LDF, targeted weight adaptation — are crucial and contribute to the enhanced performance of OLAF.

### 5.4 Qualitative Results

As seen in Figure 5, OLAF consistently improves performance for BSANet [74] and FLOAT [55]. It particularly improves the segmentation of small objects (`cat` in Row 2), small parts as shown in Row 1 ('right front leg', 'right eye' and 'left/right ear' of `dog`) and Row 2 (parts in face region of `cat`) and occluded parts as shown in Row 3 (parts of `motorbike` especially the body). A similar trend can also be seen for Pascal-Part-108 (Figure 6) and PartImageNet dataset in Figure 7. A limitation of OLAF stems from its dependence on the additional input channels (Section 3.1). Poor-quality channel masks can affect segmentation results – see Supplementary for examples.

### 5.5 Computational Metrics

As Table 5 shows, inclusion of OLAF leads to a modest rise in trainable parameters for BSANet (20%), GMNet (8%), FLOAT (14%) and FLOAT$^\dagger$ (1.5%). The training time per epoch shows a slight uptick (5% - 10%) while inference time increases by 0.26s for state of the art network. Overall, there is a pragmatic balance between OLAF's performance gain and increase in computational demand.

**Table 5:** Compute metrics for various methods with inclusion of OLAF ('+ O') and without (baseline) on Pascal-Parts-58. The batch size is 16.

| Method | Trainable Param(M) | Train Time (mins/epoch) | Test Time (secs/image) |
|---|---|---|---|
| BSANet [74] | 63.9 | 31.3 | 0.46 |
| **BSANet + O** | 76.8 | 32.9 | 0.81 |
| GMNet [45] | 123.4 | 16.6 | 0.49 |
| **GMNet + O** | 133.9 | 17.4 | 0.77 |
| FLOAT [55] | 76.2 | 18.3 | 0.98 |
| **FLOAT + O** | 86.7 | 20.1 | 1.36 |
| FLOAT$^{\dagger}$ | 674.7 | 100.8 | 8.33 |
| **FLOAT$^{\dagger}$ + O** | 685.2 | 101.1 | 8.59 |

**Table 6:** FLOAT [55] with different train and inference time resolutions for Pascal-Parts-58. The underlined value is the default setup for FLOAT. OLAF's results are included for reference.

| Method | Train | Inference | mIoU |
|---|---|---|---|
| FLOAT [55] | 513 x 513 | 513 x 513 | 60.8 |
| | | 770 x 770 | 61.0 |
| | | 1024 x 1024 | 57.4 |
| FLOAT [55] | 770 x 770 | 770 x 770 | 60.5 |
| | | 1024 x 1024 | 57.6 |
| **FLOAT+O** | **513 × 513** | **770 × 770** | **62.7** |

### 5.6 Effect of input resolution

During inference, all baselines conventionally operate on a higher resolution input ($770 \times 770$) compared to the resolution during training ($513 \times 513$). We examined the effect of higher resolution during training and inference on FLOAT [55] (the SOTA baseline). The results (Table 6) show that (i) for inference, there is a limit to the gain achieved by increasing resolution (ii) training with higher (than default) resolution does not necessarily provide a stronger baseline model. Moreover, a higher resolution significantly increases run time and memory requirements. The findings reemphasize the effectiveness of OLAF's plug-and-play design for enhancing performance without requiring an increase in default input resolution.

## 6 Conclusion

OLAF is a broadly applicable plug-and-play approach for enhancing multi-object multi-part scene parsing. OLAF's recipe consists of (i) *augmenting RGB input with object-based channels* (fg/bg, boundary edges). This acts as a structural inductive bias and guides the model to focus on relevant parts throughout optimization (ii) *using lightweight yet efficient low-level dense feature guidance (LDF).* This acts as an inductive bias for small and thin parts. (iii) *targeted weight-adaptation* for stable optimization with augmented input.

Our approach shows the benefit of efficiently infusing targeted inductive biases into existing models. OLAF also addresses multiple limitations of existing methods. OLAF consistently improves segmentation performance, especially for small and thin parts, across a broad spectrum of challenging datasets and architectures. We expect OLAF's lightweight and modular enhancements to also benefit other computer vision tasks such as panoptic part segmentation [12,20,34,40].

## References


1. Achlioptas, P., Fan, J., Hawkins, R., Goodman, N., Guibas, L.J.: Shapeglot: Learning language for shape differentiation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8938–8947 (2019)
2. Azizpour, H., Laptev, I.: Object detection using strongly-supervised deformable part models. In: Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part I 12. pp. 836–849. Springer (2012)
3. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence **39**(12), 2481–2495 (2017)
4. Benenson, R., Popov, S., Ferrari, V.: Large-scale interactive object segmentation with human annotators. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11700–11709 (2019)
5. Cai, Y., Zhou, W., Zhang, L., Yu, L., Luo, T.: Dhfnet: Dual-decoding hierarchical fusion network for rgb-thermal semantic segmentation. The Visual Computer pp. 1–11 (2023)
6. Canny, J.: A computational approach to edge detection. IEEE Trans. Patt. Anal. and Mach. Intel. (6), 679–698 (1986)
7. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
8. Chen, L.C., Yang, Y., Wang, J., Xu, W., Yuille, A.L.: Attention to scale: Scale-aware semantic image segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3640–3649 (2016)
9. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 801–818 (2018)
10. Chen, M., Artières, T., Denoyer, L.: Unsupervised object segmentation by redrawing. Advances in neural information processing systems **32** (2019)
11. Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., Yuille, A.: Detect what you can: Detecting and representing objects using holistic models and body parts. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1971–1978 (2014)
12. Cheng, B., Collins, M.D., Zhu, Y., Liu, T., Huang, T.S., Adam, H., Chen, L.C.: Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12475–12485 (2020)
13. Cho, J.H., Krähenbühl, P., Ramanathan, V.: Partdistillation: Learning parts from instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7152–7161 (2023)
14. Dery, L.M., Dauphin, Y., Grangier, D.: Auxiliary task update decomposition: The good, the bad and the neutral. arXiv preprint arXiv:2108.11346 (2021)
15. Dong, J., Chen, Q., Shen, X., Yang, J., Yan, S.: Towards unified human parsing and pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 843–850 (2014)
16. Dubrovina, A., Xia, F., Achlioptas, P., Shalah, M., Groscot, R., Guibas, L.J.: Composite shape modeling via latent space factorization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8140–8149 (2019)

17. evadingban123: Computer vision discussion on reddit. https://www.reddit.com/r/computervision/comments/m6dno8/comment/gr65yvw/ (2023), accessed: March 29, 2023
18. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision **88**, 303–338 (2010)
19. Fang, H.S., Lu, G., Fang, X., Xie, J., Tai, Y.W., Lu, C.: Weakly and semi supervised human body part parsing via pose-guided knowledge transfer. arXiv preprint arXiv:1805.04310 (2018)
20. de Geus, D., Meletis, P., Lu, C., Wen, X., Dubbelman, G.: Part-aware panoptic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5485–5494 (2021)
21. Gong, K., Liang, X., Li, Y., Chen, Y., Yang, M., Lin, L.: Instance-level human parsing via part grouping network. In: Proceedings of the European conference on computer vision (ECCV). pp. 770–785 (2018)
22. Haggag, H., Abobakr, A., Hossny, M., Nahavandi, S.: Semantic body parts segmentation for quadrupedal animals. In: 2016 IEEE international conference on systems, man, and cybernetics (SMC). pp. 000855–000860. IEEE (2016)
23. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 447–456 (2015)
24. He, J., Chen, J., Lin, M.X., Yu, Q., Yuille, A.L.: Compositor: Bottom-up clustering and compositing for robust part and object segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11259–11268 (2023)
25. He, J., Yang, S., Yang, S., Kortylewski, A., Yuan, X., Chen, J.N., Liu, S., Yang, C., Yu, Q., Yuille, A.: Partimagenet: A large, high-quality dataset of parts. In: European Conference on Computer Vision. pp. 128–145. Springer (2022)
26. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
27. Hong, Y., Yi, L., Tenenbaum, J., Torralba, A., Gan, C.: Ptr: A benchmark for part-based conceptual, relational, and physical reasoning. Advances in Neural Information Processing Systems **34**, 17427–17440 (2021)
28. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
29. Huang, H., Lin, L., Tong, R., Hu, H., Zhang, Q., Iwamoto, Y., Han, X., Chen, Y.W., Wu, J.: Unet 3+: A full-scale connected unet for medical image segmentation. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 1055–1059. IEEE (2020)
30. Ke, B., Obukhov, A., Huang, S., Metzger, N., Daudt, R.C., Schindler, K.: Repurposing diffusion-based image generators for monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
31. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
32. Krause, J., Jin, H., Yang, J., Fei-Fei, L.: Fine-grained recognition without part annotations. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5546–5555 (2015)

33. Li, G., Xie, Y., Lin, L., Yu, Y.: Instance-level salient object segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2386–2395 (2017)
34. Li, X., Xu, S., Yang, Y., Cheng, G., Tong, Y., Tao, D.: Panoptic-partformer: Learning a unified model for panoptic part segmentation. In: European Conference on Computer Vision. pp. 729–747. Springer (2022)
35. Liang, X., Gong, K., Shen, X., Lin, L.: Look into person: Joint body parsing & pose estimation network and a new benchmark. IEEE transactions on pattern analysis and machine intelligence **41**(4), 871–885 (2018)
36. Liang, X., Shen, X., Feng, J., Lin, L., Yan, S.: Semantic object parsing with graph lstm. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. pp. 125–143. Springer (2016)
37. Liang, X., Shen, X., Xiang, D., Feng, J., Lin, L., Yan, S.: Semantic object parsing with local-global long short-term memory. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3185–3193 (2016)
38. Liang, X., Xu, C., Shen, X., Yang, J., Liu, S., Tang, J., Lin, L., Yan, S.: Human parsing with contextualized convolutional neural network. In: Proceedings of the IEEE international conference on computer vision. pp. 1386–1394 (2015)
39. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
40. Liu, H., Peng, C., Yu, C., Wang, J., Liu, X., Yu, G., Jiang, W.: An end-to-end network for panoptic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6172–6181 (2019)
41. Liu, Q., Kortylewski, A., Zhang, Z., Li, Z., Guo, M., Liu, Q., Yuan, X., Mu, J., Qiu, W., Yuille, A.: Learning part segmentation through unsupervised domain adaptation from synthetic vehicles. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19140–19151 (2022)
42. Liu, Y., Zhao, L., Zhang, S., Yang, J.: Hybrid resolution network using edge guided region mutual information loss for human parsing. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 1670–1678 (2020)
43. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)
44. Ma, A., Wang, J., Zhong, Y., Zheng, Z.: Factseg: Foreground activation-driven small object semantic segmentation in large-scale remote sensing imagery. IEEE Transactions on Geoscience and Remote Sensing **60**, 1–16 (2022). https://doi.org/10.1109/TGRS.2021.3097148
45. Michieli, U., Borsato, E., Rossi, L., Zanuttigh, P.: Gmnet: Graph matching network for large scale part semantic segmentation in the wild. In: European Conference on Computer Vision. pp. 397–414. Springer (2020)
46. Michieli, U., Zanuttigh, P.: Edge-aware graph matching network for part-based semantic segmentation. International Journal of Computer Vision **130**(11), 2797–2821 (2022)
47. Nagarajan, T., Feichtenhofer, C., Grauman, K.: Grounded human-object interaction hotspots from video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8688–8697 (2019)

48. Nie, X., Feng, J., Yan, S.: Mutual learning to adapt for joint human parsing and pose estimation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 502–517 (2018)
49. Nie, X., Feng, J., Yan, S.: Mutual learning to adapt for joint human parsing and pose estimation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 502–517 (2018)
50. Pan, T.Y., Liu, Q., Chao, W.L., Price, B.: Towards open-world segmentation of parts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15392–15401 (2023)
51. Pohlen, T., Hermans, A., Mathias, M., Leibe, B.: Full-resolution residual networks for semantic segmentation in street scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4151–4160 (2017)
52. Pu, M., Huang, Y., Liu, Y., Guan, Q., Ling, H.: Edter: Edge detection with transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1402–1412 (June 2022)
53. Pu, M., Huang, Y., Liu, Y., Guan, Q., Ling, H.: EDTER: Edge detection with transformer. In: CVPR. pp. 1402–1412 (June 2022)
54. Sauvalle, B., de La Fortelle, A.: Unsupervised multi-object segmentation using attention and soft-argmax. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3267–3276 (2023)
55. Singh, R., Gupta, P., Shenoy, P., Sarvadevabhatla, R.K.: FLOAT: Factorized learning of object attributes for improved multi-object multi-part scene parsing. In: CVPR. pp. 1445–1455 (June 2022)
56. Song, Y., Chen, X., Li, J., Zhao, Q.: Embedding 3d geometric features for rigid object part segmentation. In: Proceedings of the IEEE international conference on computer vision. pp. 580–588 (2017)
57. Tan, X., Xu, J., Ye, Z., Hao, J., Ma, L.: Confident semantic ranking loss for part parsing. In: 2021 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6 (2021). https://doi.org/10.1109/ICME51207.2021.9428332
58. Ventura, C., Bellver, M., Girbau, A., Salvador, A., Marques, F., Giro-i Nieto, X.: Rvos: End-to-end recurrent network for video object segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5277–5286 (2019)
59. Wang, J., Yuille, A.L.: Semantic part segmentation using compositional model combining shape and appearance. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1788–1797 (2015)
60. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: European conference on computer vision. pp. 20–36. Springer (2016)
61. Wang, P., Shen, X., Lin, Z., Cohen, S., Price, B., Yuille, A.L.: Joint object and part segmentation using deep learned potentials. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1573–1581 (2015)
62. Wang, X., Li, S., Kallidromitis, K., Kato, Y., Kozuka, K., Darrell, T.: Hierarchical open-vocabulary universal image segmentation (2023)
63. Wang, Y., Zhou, Q., Liu, J., Xiong, J., Gao, G., Wu, X., Latecki, L.J.: Lednet: A lightweight encoder-decoder network for real-time semantic segmentation. In: 2019 IEEE international conference on image processing (ICIP). pp. 1860–1864. IEEE (2019)
64. Wei, M., Yue, X., Zhang, W., Kong, S., Liu, X., Pang, J.: Ov-parts: Towards open-vocabulary part segmentation. In: Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2023)

65. Xiang, F., Qin, Y., Mo, K., Xia, Y., Zhu, H., Liu, F., Liu, M., Jiang, H., Yuan, Y., Wang, H., et al.: Sapien: A simulated part-based interactive environment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11097–11107 (2020)
66. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. Advances in Neural Information Processing Systems **34**, 12077–12090 (2021)
67. Xie, S., Tu, Z.: Holistically-nested edge detection. In: ICCV. pp. 1395–1403 (2015)
68. Yang, J., Wang, C., Li, Z., Wang, J., Zhang, R.: Semantic human parsing via scalable semantic transfer over multiple label domains. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19424–19433 (2023)
69. Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth anything: Unleashing the power of large-scale unlabeled data. In: CVPR (2024)
70. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: Bisenet: Bilateral segmentation network for real-time semantic segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 325–341 (2018)
71. Zhang, S., Liew, J.H., Wei, Y., Wei, S., Zhao, Y.: Interactive object segmentation with inside-outside guidance. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12234–12244 (2020)
72. Zhao, H., Qi, X., Shen, X., Shi, J., Jia, J.: Icnet for real-time semantic segmentation on high-resolution images. In: Proceedings of the European conference on computer vision (ECCV). pp. 405–420 (2018)
73. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2881–2890 (2017)
74. Zhao, Y., Li, J., Zhang, Y., Tian, Y.: Multi-class part parsing with joint boundary-semantic awareness. In: ICCV. pp. 9177–9186 (2019)
75. Zheng, Z., Zhong, Y., Wang, J., Ma, A.: Foreground-aware relation network for geospatial object segmentation in high spatial resolution remote sensing imagery. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4096–4105 (2020)