Factorize and Zoom - A Model Agnostic Framework for Improved Object-part Scene Segmentation

Rishubh Singh*, Pranav Gupta†‡, Pradeep Shenoy* and Ravi Kiran Sarvadevabhatla†§
* *Google Research India,* † *CVIT, IIIT Hyderabad*
* {*rishubh, shenoypradeep*}*@google.com,* ‡ *pranavmicro7@gmail.com,* § *ravi.kiran@iiit.ac.in*

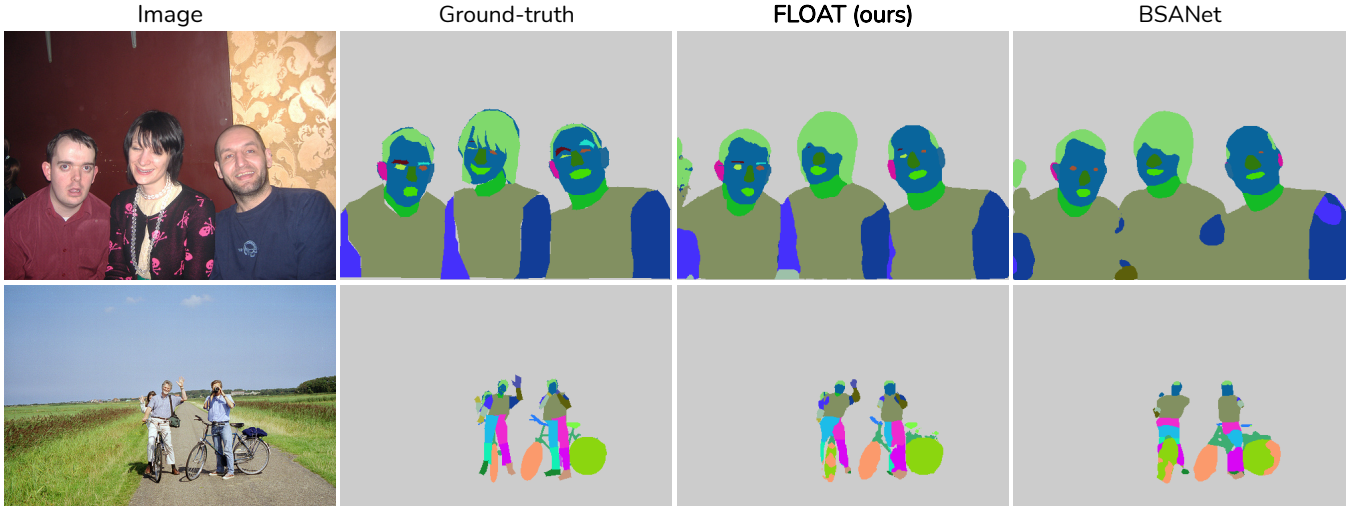

Figure 1: Multi-object multi-part semantic segmentation results for sample images from our expanded label space dataset, Pascal-Part-201. Compared to state of the art BSANet [57], FLOAT accurately segments tiny parts (e.g. `left eyebrow`, `right eyebrow` on faces in upper image) and handles scale variations better – note the size variations of `person` instances. Also, observe that FLOAT predicts directional attributes of parts (e.g. 'left'/'right') accurately – ['left'/'right']: see `eyebrow`, `eye`, `arm` in upper image and `leg` in lower image ; ['front'/'back']: see `wheel` parts of the `bicycle` (lower image).

***Abstract*—Multi-object multi-part scene parsing is a challenging task which requires detecting multiple object classes in a scene and segmenting the semantic parts within each object. In this paper, we propose FLOAT, a factorized label space framework for scalable multi-object multi-part parsing. Our framework involves independent dense prediction of object category and part attributes which increases scalability and reduces task complexity compared to the monolithic label space counterpart. In addition, we propose an inference-time 'zoom' refinement technique which significantly improves segmentation quality, especially for smaller objects/parts. Compared to state of the art, FLOAT obtains an absolute improvement of 2.0% for mean IOU (mIOU) and 4.8% for segmentation quality IOU (sqIOU) on the Pascal-Part-58 dataset. For the larger Pascal-Part-108 dataset, the improvements are 2.1% for mIOU and 3.9% for sqIOU. We incorporate previously excluded part attributes and other minor parts of the Pascal-Part dataset to create the most comprehensive and challenging version which we dub Pascal-Part-201. FLOAT obtains improvements of 8.6% for mIOU and 7.5% for sqIOU on the new dataset, demonstrating its parsing effectiveness across a challenging diversity of objects and parts. The code and new dataset will be made available.**

1. Introduction

Semantic scene parsing is a foundational image understanding problem in the vision community [23], [48], [49], [51], [53], [54], [59]. Typically, the goal is to segment objects and "stuff" regions (e.g. road, background) in the scene. Multi-object multi-part parsing is a significantly more challenging variant which requires *part*-level segmentation of each scene object [31], [39], [57]. Compared to traditional object-level segmentation, semantic representations infused with fine-grained part-level knowledge can provide richer information for downstream reasoning tasks including visual question answering [19], perceptual concept learning [5], shape modelling [1], [12] and many others [2], [8], [10], [21], [38], [52].

For part-based object segmentation, some existing approaches tackle the simpler problem of *single*-object part

parsing [14], [15], [16], [40], [41]. Although a few recent approaches have addressed multi-object multi-part parsing [31], [39], [57], they consider part labels to be independent and do not take advantage of intra/inter ontological relationships among objects and parts at label level. They also tend to perform poorly on smaller and infrequent parts/categories. To address these shortcomings, we propose FLOAT, a novel factorized label space framework for scalable multi-object multi-part parsing. Our approach is motivated by the following observations:

Observation #1: Object part names in datasets typically consist of a *root* component and *side* component(s). Many object categories contain parts with the same *root* component. For example, the *root* component of 'left front leg' found in `horse`, `cow` etc. and 'right leg' found in `person`, is leg. Therefore, parts can be grouped based on their *root* component.

The example also suggests that object categories whose instances contain shared category-level attributes (e.g. "living things that move") are likely to contain same *root* components (such as leg). Using this criterion, some object categories (e.g. `cow, person, bird`) can be grouped as 'animate'. Similarly, some categories (e.g. "rigid bodied") can be grouped as 'inanimate'. As with the 'animate' group, 'inanimate' group categories also share many *root* part components (e.g. 'wheel' in `aeroplane`, `bicycle`, `car`).

Observation #2: Similar to Observation #1, parts can also be grouped by *side* component – e.g. 'front' is a *side* component of 'front wheel' found in `bike` and 'left front leg' in `person`.

Factoring the object/part label space in terms of these groups ('animate', 'inanimate', 'side') greatly reduces the effective number of output labels. In turn, this increases scalability in terms of object categories and part cardinality. The design choice ('factoring') also enables efficient data sharing when learning semantic representations for grouped parts and improves performance for infrequent classes (see Fig. 1).

A second key feature of our framework is IZR, an *inference-time* segmentation refinement technique. IZR transforms 'zoomed in' versions of preliminary per-object label maps into refined counterparts which are finally composited back onto the segmentation canvas. Apart from the advantage of not requiring additional training, IZR is empirically superior to alternate inference-time schemes and significantly improves segmentation quality, especially for smaller objects/parts.

In existing works, results are reported on simplified, label-merged versions of the original dataset (Pascal-Part [8]). In our work, we incorporate previously excluded part attributes and other minor parts to create Pascal-Part-201, the most comprehensive and challenging version of Pascal-Part [8]. Along with the standard mean IOU (mIOU) and mAvg scores, we report sqIOU [20] and sqAvg – normalized segmentation quality measures which are less affected by spatial scale of objects and parts.

In summary, our contributions are the following:

- FLOAT, a novel factorized label space framework for scalable multi-object multi-part parsing (Sec. 3).
- IZR, an inference-time refinement technique which significantly improves segmentation quality especially for smaller objects/parts in the scene (Sec. 3.4).
- Pascal-Part-201, the most comprehensive and challenging version of the Pascal-Part [8] dataset (Sec. 4). Experimental evaluation demonstrates FLOAT's superior performance on Pascal-Part-201 relative to existing approaches (Sec. 5).

2. Related Work

Semantic segmentation is a broad area with intensive research. We do not attempt to summarize all approaches to enable focus on more directly relevant works. A common design pattern for semantic segmentation is the encoder-decoder setup [3], [6], [7], [55]. In particular, the baselines, existing approaches and our proposed approach all adopt the popular DeepLab architecture [6] for various components of the segmentation task pipeline.

Single-Object Multi-Part Parsing has been extensively explored. Existing approaches typically consider object category subsets such as persons [14], [15], [24], [25], [26], [28], [29], [35], [43], [44], [45], [56], animals [16], [40], [41] and vehicles [25], [27], [35], [37]. However, in this setting, most works assume a single object of interest per image.

Multi-object multi-part parsing is a relatively new and under studied problem [31], [39], [57]. The approaches of Zhao et al. [57] and Michieli et al. [31] tackle multi-object multi-part parsing by providing object-level feature guidance to the part segmentation network during optimization. Zhao et al. [57] additionally provides boundary-level awareness to features. Tan et al. [39] create a semantic co-ranking loss modelling intra and inter part relationships. Xiao et al. [46] introduce a composite dataset and an approach for predicting perceptual visual concepts in scenes. However, in contrast to our framework, these approaches report results on simplified (label-merged) versions of standard datasets and empirically exhibit inferior performance for smaller parts.

Factorization: In machine vision applications, early works such as Zheng et al. [58] used factorial Conditional Random Field models to separately predict object category, coarse object labels and object attributes such as shape, material and surface type. Other work involve jointly learning object and attribute-related information as a separable latent representation [34] or using graph networks [33]. Misra et al. [32] propose a factorization over global object attributes and object classifiers to enable compositionality. Other works extend this idea to inter-object relationships, e.g. noun-preposition-noun triplets [19], [22], [30]. In all these works, a simple *global* property of the object (e.g., material, texture, color, size, shape) is learnt jointly with the object category information. In their work on panoptic part segmentation, Geus et al. [9] conduct experiments involving two categories from Pascal-Part-58 with some parts grouped by semantic similarity. To our knowledge, we are the first

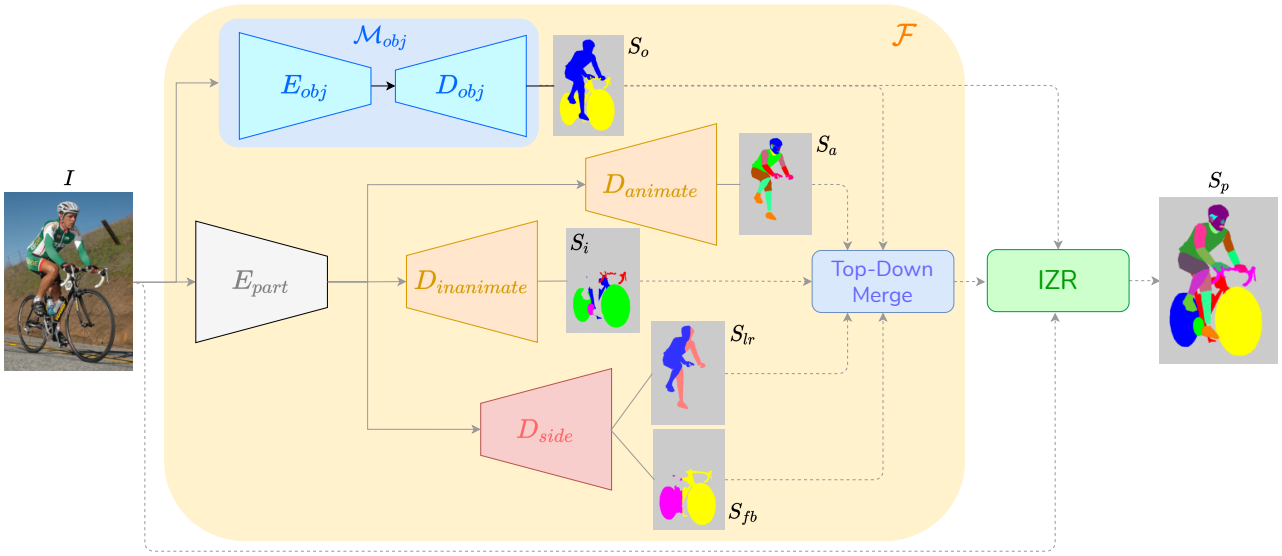

Figure 2: An overview diagram of our FLOAT framework (Sec. 3). Given an input image I, an object-level semantic segmentation network ($\mathcal{M}_{obj}$, in blue) generates object prediction map (S_o). Two decoders (in orange) produce object category grouped part-level prediction maps for 'animate' (S_a) and 'inanimate' objects (S_i) in the scene. Another decoder (in red) produces part-attribute grouped prediction maps for 'left-right' (S_{lr}) and 'front-back' (S_{fb}). At inference time (shown by dotted lines), outputs from the decoders are merged in a top-down manner. The resulting prediction is further refined using the IZR technique (see Fig. 3) to obtain the final segmentation map (S_p).

to show that *object parts* can be factorized across diverse object categories at scale, and that such factorization significantly improves segmentation performance, in resonance with theories of visual recognition [4], [18].

Zooming in on image regions using bounding boxes generated by attention maps [42] and reinforcement learning policies [11], [47] have been found to improve detection and segmentation. Other works use the technique on object instances for video interpolation [50] and on part instances for object parsing [43]. Porzi et al. [36] use zoomed in crops based on object classes for improving panoptic segmentation of high resolution images. Similar to the latter set of approaches, FLOAT also employs zooming in on object regions. However, our zoom-based refinement does not require any extra training and can be directly used during inference for improved performance.

3. Our framework (FLOAT)

As mentioned earlier, FLOAT's design leverages the shared-attribute groups that naturally exist within object categories ('animate', 'inanimate') and part attributes ('left', 'right', 'front', 'back') - see Fig. 2. The sections that follow describe how we operationalize the idea. Although our approach is general in nature, we use object categories and part names from the Pascal-Part dataset [8] for ease of understanding.

3.1. Relabeling images with factored labels

The original Pascal-Part dataset contains object and part level label maps. We re-label or partition these maps to obtain five new label groups as described below.

object: The label set for this group comprises unique object category labels. For example, S_o in Fig. 2 is a label map from this group containing `person` and `bicycle` objects.

animate: For this group, the label set comprises *root* components of part labels from the object categories `bird`, `cat`, `cow`, `cat`, `dog`, `horse`, `person`, `sheep`. Note that part labels are pooled across all object categories which share the part name and omit directional attribute information. For example, a single label leg covers all corresponding part instances from all objects in the 'animate' group. This can also be seen in S_a in Fig. 2 – the `left foot` and `right foot` of `person` are color-coded the same ('orange') and assigned the common label `foot`.

inanimate: The label set comprises *root* components of part labels from `aeroplane`, `bicycle`, `bottle`, `bus`, `car`, `motorbike`, `pottedplant`, `train`, `tv`. Note that (i) these categories are disjoint from the 'animate' group (see S_i in Fig. 2) (ii) the part label pooling mentioned for 'animate' is applicable here as well.

side: In this case, two disjoint label groups exist. One group comprises all part labels which have the words 'left' or 'right' in their name (e.g. `left hand`, `right wing`). Label map regions whose part labels contain 'left'/'right' are considered seed pixels for a flood-fill style procedure which produces corresponding 'left'/'right' label maps (e.g. S_{lr} in Fig. 2). The same procedure is used for label group comprises of part labels which have the words 'front' or 'back' in their name (see S_{fb} in Fig. 2).

Broadly, object parts from living things that move are in

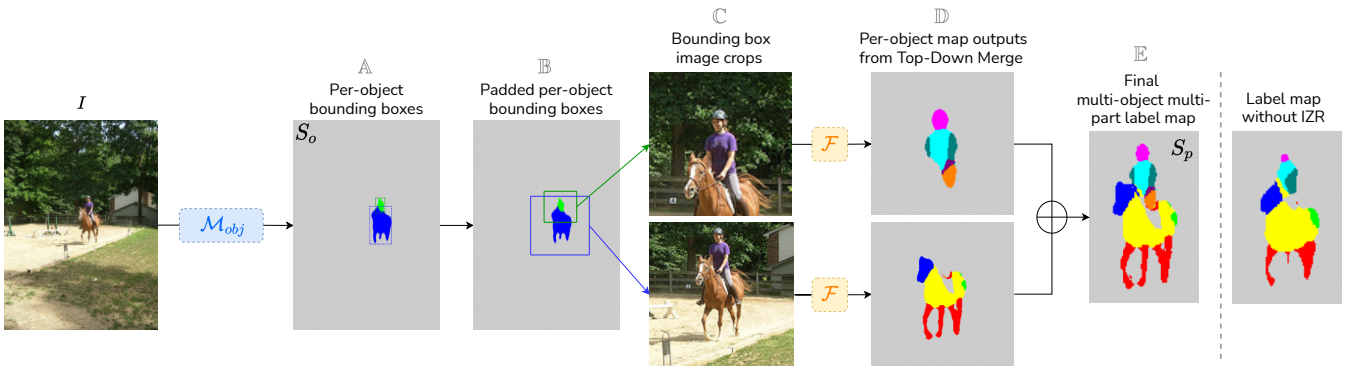

Figure 3: An overview of Inference-time Zoom Refinement (IZR) - Sec. 3.4. During inference, predictions from the object-level network $\mathcal{M}_{obj}$ are used to obtain padded bounding boxes for scene objects ($\mathbb{B}$). The corresponding object crops ($\mathbb{C}$) are processed by the factorized network ($\mathcal{F}$, Sec. 3). The resulting label maps ($\mathbb{D}$) are composited to generate S_p, the final refined part segmentation map ($\mathbb{E}$). Notice the improvement in segmentation quality relative to the part label map without IZR (included for comparison).

the 'animate' group while other parts, typically from rigidly shaped non-living things, are in the 'inanimate' group. As mentioned before, such grouping enables data-efficient representation learning for common parts (e.g. `torso` in 'animate' group). A similar reasoning holds for 'side' directional grouping ({'left', 'right'}, {'front','back'}).

3.2. Factorized semantic segmentation architecture

We configure the segmentation architecture to output the factorized label maps described in previous section. As Fig. 2 shows, we employ two semantic segmentation networks, one for object-level and other for part-level label maps. The object-level network ($\mathcal{M}_{obj}$) outputs the object prediction map (S_o). The part-level network consists of a shared encoder (E_{part}), and three decoders: the 'animate' decoder ($D_{animate}$) which outputs the 'animate' label map (S_a), the 'inanimate' decoder ($D_{inanimate}$) which outputs the 'inanimate' label map (S_i). The 'side' decoder (D_{side}) outputs the 'left/right' (S_{lr}) and 'front/back' (S_{fb}) label maps. The outputs from the object-level network (S_o) and part-level network (S_i, S_a, S_{lr}, S_{fb}) are merged at inference time. We describe this merging process next.

3.3. Top-Down Merge

To combine the factorized label maps output by segmentation architecture $\mathcal{F}$ (see Fig. 2), we adopt a top-down merging strategy. For each object (e.g. `bicycle`) in the object prediction map (S_o), we examine the labels of corresponding pixel locations in the part-level label maps. Depending on the type of object ('animate' or 'inanimate'), the corresponding label regions are copied to the scene-level prediction canvas. (e.g. for `bicycle`, the considered labels in S_i would be `wheel`, `chainwheel`, `handlebar`, `headlight`, `saddle`). Similarly, the object-level map's pixel locations are referenced from 'side' label maps ({'left','right'} - S_{lr}, {'front','back'} - S_{fb}). The corresponding label regions are copied to the scene prediction canvas.

In the next section, we describe how the resulting prediction map is refined using a per-object 'zooming' technique.

3.4. Inference-time Zoom Refinement (IZR)

The Inference-time Zoom Refinement (IZR) technique improves segmentation quality by 'zooming' into each scene object. As the first step, the input image I is processed by the object-level network $\mathcal{M}_{obj}$ to obtain object-level map (see $\mathbb{A}$ in Fig. 3). The bounding box corresponding to each object component is then padded so that the object is centered and aspect ratio is preserved ($\mathbb{B}$ in Fig. 3). Image crops corresponding to the padded bounding box extents are then obtained ($\mathbb{C}$). Note that the padding enables scene context to be included for each cropped object and also helps account for inaccuracies in the object map prediction. The cropped object images are then processed by FLOAT's factorized network $\mathcal{F}$ to obtain the corresponding part-level label maps ($\mathbb{D}$). These label maps are then composited to generate the final refined segmentation map ($\mathbb{E}$). In the next two sections, we describe the optimizer formulation for the networks in FLOAT and implementation details.

3.5. Optimization

We train the object model $\mathcal{M}_{obj}$ (Sec. 3.2) using the standard per-pixel cross-entropy loss. For training the part-level model, we use a combination of cross-entropy loss (L_{CE}) and graph matching loss (L_{GM}) [31]. The cross-entropy loss is applied to each of the 4 output part-level maps i.e. S_a, S_i, S_{lr}, S_{fb} (see Fig. 2).

The graph matching loss [31] captures proximity relationships between part pairs within the map and scores the matching of these pairs between the ground truth and the predicted map. The degree of proximity between a part pair

is represented by the number of pixels in one part situated T pixels or less from the other part, where T is an empirically set threshold. For efficiency, the pairwise proximity map is approximated by dilating each part mask by $\lceil T/2 \rceil$ and computing the intersecting region. The ground truth proximity map M^{GT} (and similarly predicted map M^{pred}) is formally defined as: $\tilde{m}^{GT}_{i,j} = |\{s \in \Phi(p^{GT}_i) \cap \Phi(p^{GT}_j)\}|$ where $\tilde{m}^{GT}_{i,j}$ is the proximity between the ith and jth parts, p_i, p_j are the respective part mask, s is a generic pixel, Φ is morphological 2D dilation operator and $|.|$ is the cardinality of the given set. A row-wise normalization is applied to the proximity matrix: $\boldsymbol{M}^{GT}_{[i,:]} = \tilde{\boldsymbol{M}}^{GT}_{[i,:]} / ||\tilde{\boldsymbol{M}}^{GT}_{[i,:]}||_2$. The graph matching loss L_{GM} is computed as the Frobenius norm between the two adjacency matrices: $\mathcal{L}_{GM} = ||\boldsymbol{M}^{GT} - \boldsymbol{M}^{pred}||_F$.

Additionally, for the 'animate' and 'inanimate' branches, a composite foreground-background binary cross-entropy loss serves as extra guidance. The loss for the part level network is a weighted combination of the losses for all part branches: $\mathcal{L}^{part} = \mathcal{L}^{anim} + \mathcal{L}^{inanim} + \mathcal{L}^{side}$, where $\mathcal{L}^{anim} = \mathcal{L}^{anim}_{CE} + \lambda_{GM}\mathcal{L}^{anim}_{GM}$.

3.6. Implementation and Training Details

For fair comparison with previous works [31], [39], [57], we employ the DeepLab-v3 [6] architecture with a ImageNet pre-trained ResNet-101 [17] as the encoder (backbone) and follow the same training scheme and augmentations. During training, images are randomly left-right flipped and scaled 0.5 to 2 times the original resolution with bilinear interpolation. The results at testing stage are reported at the original image resolution. The threshold T employed for proximity matrix (Sec. 3.5) is empirically set to 4. The model is trained for 40K steps with the base learning rate set to $7 \cdot 10^{-3}$ which is decreased with a polynomial decay rule with power 0.9. We employ weight decay regularization of 10^{-4}. We use a batch size of 16 images and use $\lambda_{GM} = 0.1$ for weighting graph matching loss relative to the cross-entropy loss. We use 2 NVIDIA A100 GPUs each with 40GB GPU memory to train our models, and for experiments.

4. Datasets and Evaluation Metrics

Pascal-Part: For experiments, we use the Pascal-Part [8] which is currently the largest multi-object multi-part parsing dataset. It contains 10,103 variable-sized images with pixel-level part annotations on the 20 Pascal VOC2010 [13] semantic object classes (plus the `background` class). We use the original split from Pascal-Part with 4998 images for training and 5105 images in the publicly provided validation set for testing.

Pascal-Part-58/108: For comparison with previous work, we use the datasets Pascal-Part-58 [57] and Pascal-Part-108 [31] which contain 58 and 108 part classes respectively. Both the Pascal-Part variants simplify the original semantic classes by grouping some parts together, and contain 58 and 108 part classes respectively. Pascal-Part-58 mostly contains large parts of objects such as `head`, `torso`, `leg` etc. for animals and `body`, `wheel` etc. for non-living objects. Pascal-Part-108 is more challenging and additionally contains relatively smaller parts (e.g. `eye`, `neck`, `foot` etc. for animals and `roof`, `door` etc. for non-living objects).

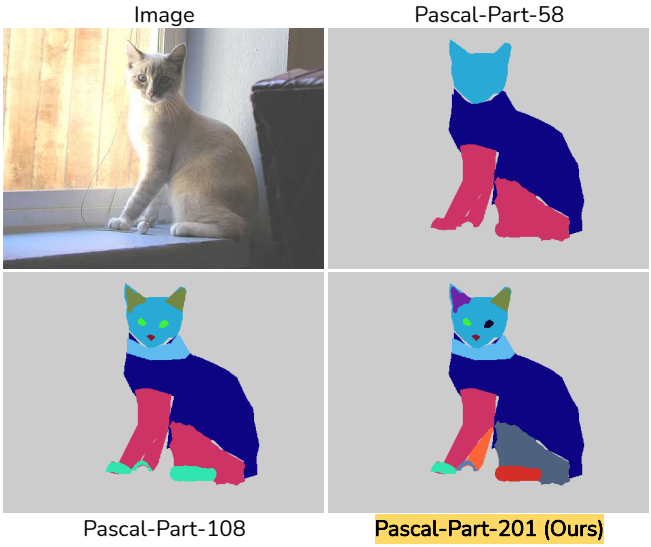

Figure 4: An illustration of labelling granularity in different versions of the Pascal-Part dataset. Pascal-Part-108 [31] adds smaller parts (e.g. `eyes`, `ears`) to Pascal-Part-58 [57]. Our newly introduced Pascal-Part-201 further adds directional information to parts as appropriate (e.g. {'left','right'} to `eyes`, `ears`; {'front','back'} to `legs`).

Pascal-Part-201: We incorporate part attributes ('left', 'right', 'front', 'back', 'upper', 'lower') and other minor parts (e.g. `eyebrow`) excluded in both the mentioned variants (58/108), to create the most comprehensive and challenging version of the dataset containing 201 parts which we dub Pascal-Part-201. We observed that the original part labelling scheme in Pascal-Part leaves out large chunks of an object's pixels unlabelled for the `bike`, `motorbike` and `tv` categories which lead to disconnected objects. To address this, we add a `body` part annotation for `bike`, `motorbike`, and a `frame` part for `tv`. An example illustrating the differences in part labelling and granularity of the Pascal-Part variants can be seen in Fig. 4.

4.1. Evaluation Metrics

For performance evaluation, we use two versions of Intersection over Union (IOU) metric. We first describe mIOU and mAvg, the standard segmentation quality metrics reported for the problem setting. We then describe balanced variants of these metrics – sqIOU and sqAvg.

mIOU: Let $Pred^j_p$ and GT^j_p be the prediction and ground truth respectively for the pth part in the jth image I_j. Suppose the dataset contains N images. The mIOU for the part ($mIOU_p$) is calculated as:

Model	bgr	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	TV	**mIOU**	**mAvg**
Baseline	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
GMNet [31]	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
BSANet [57]	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
FLOAT	**92.5**	**36.7**	**49.7**	**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	**37.1**	**46.9**

	bgr	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	TV	**sqIOU**	**sqAvg**
Baseline	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
GMNet [31]	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
BSANet [57]	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
FLOAT	**90.8**	**32.5**	**41.8**	**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**	**29.6**	**39.5**

TABLE 1: Category-wise results for Pascal-Part-201. FLOAT outperforms competing methods by large margins w.r.t mIOU (top) and sqIOU (bottom).

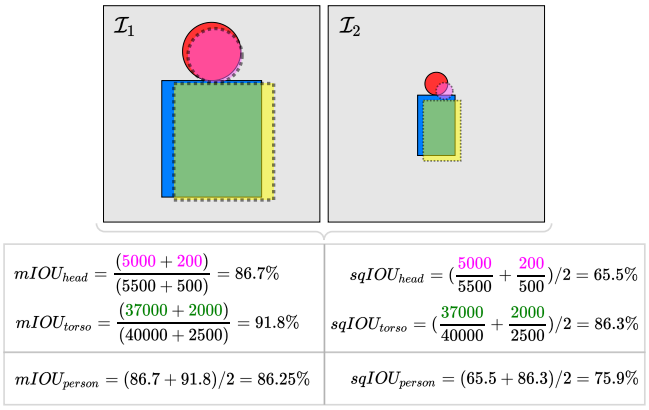

Figure 5: Toy example comparing mIOU and sqIOU with two images from `toy-person` category containing parts `head` and `torso` . 'Red' and 'blue' represent ground-truth, 'pink' and 'green' represent prediction overlap areas. mIOU fails to reflect the bad segmentation of head in image $\mathcal{I}_2$ while sqIOU is fairer.

$$mIOU_p = \frac{\sum_{j=1}^{N}(Pred_p^j \cap GT_p^j) \cdot \mathbb{I}[p \in I_j]}{\sum_{j=1}^{N}(Pred_p^j \cup GT_p^j) \cdot \mathbb{I}[p \in I_j]} \quad (1)$$

where $\mathbb{I}[.]$ is the indicator function (i.e. summation is performed only for images where part p is present). The mIOU for the dataset is then calculated as: $mIOU = \left(\sum_p mIOU_p\right)/N_p$, where N_p is the number of part categories (classes) in the dataset (58/108/201).

mAvg: The mIOU score for an object category is the average of its per-part scores, i.e. $mIOU_c = \left(\sum_p mIOU_p\right)/N_c$ where N_c is the number of unique part labels in object category c. Finally, mAvg is calculated as mAvg $= \left(\sum_c mIOU_c\right)/C$, where C is the number of object categories (21 for Pascal-Part datasets).

sqIOU: This is a modified version of Segmentation Quality (SQ) metric [20] tailored for semantic segmentation. The sqIOU for the part p is calculated as:

$$sqIOU_p = \sum_{j=1}^{N}\left(\underbrace{\frac{Pred_p^j \cap GT_p^j}{Pred_p^j \cup GT_p^j}}_{IOU(Pred,GT)_p^j} \cdot \mathbb{I}[p \in I_j]\right)/N \quad (2)$$

The calculation for sqIOU and sqAvg is similar to that of mIOU. Due to their formulation, mIOU and mAvg [31], [57] tend to be dominated by contributions from bigger[1] instances. In contrast, sqIOU and sqAvg weight parts of all sizes equally – compare Eqn. 1 and 2 and also see the toy example in Fig. 5. Therefore, sqIOU and sqAvg can be considered a more 'fair' measure for segmentation quality.

5. Experimental Results

For evaluation, we compare the performance of FLOAT with BSANet [57], GMNet [31] and CO-Rank [39]. As a baseline, we train a DeepLab-v3 [6] model with independently paired object category and associated part names (e.g. `cow left eye`, `cow right ear`) as labels. BSANet and CO-Rank report results on Pascal-Part-58 while GMNet additionally reports results on Pascal-Part-108. We report results on all variants of the Pascal-Part dataset, including our newly introduced Pascal-Part-201. To enable comparison, we train GMNet and BSANet on our dataset, Pascal-Part-201. For evaluation, we employ the mIOU, mAvg and sqIOU, sqAvg metrics described previously (Sec. 4.1). In addition, we analyze the relative contribution of various components in FLOAT via ablation studies.

5.1. Pascal-Part-201

Table 1 shows the category-wise and overall performance on Pascal-Part-201. Overall, we see that FLOAT

1. Informally, an instance is deemed "big" if it is among the largest instances for an object part category by area.

Method	Dataset	mIOU	mAvg	sqIOU	sqAvg
Baseline	58	54.3	55.4	46.0	48.4
BSANet [57]		58.2	58.9	49.3	51.5
GMNet [31]		59.0	61.8	49.4	54.3
CO-Rank [39]		60.7	60.6	-	-
FLOAT		**61.0**	**64.2**	**54.2**	**57.1**
Baseline	108	41.3	43.6	32.2	36.1
BSANet [57]		45.9	48.4	36.6	41.0
GMNet [31]		45.8	50.5	35.8	41.9
FLOAT		**48.0**	**53.0**	**40.5**	**45.6**

TABLE 2: Results on Pascal-Part-58, Pascal-Part-108: FLOAT outperforms the baseline and other existing methods on mIOU and with a significant gap on sqIOU. Missing CO-Rank entries are due to incomplete official codebase and missing details in the paper.

outperforms baselines and existing approaches by a significantly large margin. We obtain large gains of 10.8% on mIOU and 8.1% on sqIOU relative to the baseline. We outperform the next best method BSANet [57] by large margins of 8.6% on mIOU and 7.5% on sqIOU as well.

Empirically, we obtain significant sqIOU gains of 10%-30% on small parts – for e.g. `left/right eye`, `left/right ear`, `left/right horn` etc. of 'animate' categories such as `bird`, `cat`, `cow`. For 'inanimate' categories (e.g. `bus`, `car`, `aeroplane`), we obtain sqIOU improvements in the range of 5%-11% on small parts such as `front/back plate`, `left/right wing`. The performance improvement is also similarly substantial for most parts containing side components ('left/right' or 'front/back'). Result tables and additional details can be found in Supplementary.

5.2. Pascal-Part-58 and Pascal-Part-108

We also show results on previously proposed datasets Pascal-Part-58 [57] and Pascal-Part-108 [31]. As shown in Table 2, FLOAT framework achieves the best performance on both these datasets. In terms of mIOU, we outperform CO-Rank [39] by 0.3% on Pascal-Part-58 and GMNet [31] by 2.0%. In terms of sqIOU, we outperform other methods by large margins as well – 4.8% over GMNet and 4.9% over BSANet. A similar trend is seen for Pascal-Part-108 with large improvements of 2.1% on mIOU and 3.9% on sqIOU over the next best method BSANet [57].

Overall, the results across existing and challenging new variants of Pascal-Part dataset demonstrate the strengths of our factorized label space setup. In particular, the increasing gains with increasing dataset complexity demonstrates the superior scaling capacity of the FLOAT framework. Detailed part level and object level metrics for all the dataset variants are reported in the supplementary material.

5.3. Ablation Studies

We perform multiple experiments with ablative variant models of FLOAT to verify the effectiveness of our design choices. From the results in Table 3, we see that starting from baseline (first row in each dataset variant), systematically adding components of FLOAT pipeline noticeably improves segmentation quality. The gains are most apparent for Pascal-Part-201 dataset, particularly when factorized components are included. From the last two rows, we also see that IZR is a superior choice compared to Random Crop Zoom (RCZ) - a variant which uses random crops whose cardinality matches the number of objects in the scene. Some part names in the original Pascal-Part dataset [8] contain the side component 'upper/lower'. We attempted to train a FLOAT variant with these components as outputs of D_{side} decoder. However, the model failed to converge. We hypothesize this is due to the drastically smaller quantum of training data compared to other *side* attributes, i.e. 'left/right' and 'front/back'.

Method	Dataset	No Factorization	Object	Part	Anim/Inanim	Side	Inference Augmentation	mIOU	sqIOU
Baseline	58	✓				-		54.3	46.0
$\mathcal{M}_{obj} + \mathcal{M}_{part}$			✓	✓		-		60.7	51.5
$\mathcal{F}$			✓		✓	-		60.9	51.7
FLOAT			✓		✓	-	IZR	**61.0**	**54.2**
Baseline	108	✓				-		41.3	32.2
$\mathcal{M}_{obj} + \mathcal{M}_{part}$			✓	✓		-		46.1	36.7
$\mathcal{F}$			✓		✓	-		47.8	38.4
FLOAT			✓		✓	-	IZR	**48.0**	**40.5**
Baseline	201	✓						26.3	21.5
$\mathcal{M}_{obj} + \mathcal{M}_{part}$			✓	✓				29.1	22.8
$\mathcal{F} - D_{side}$			✓		✓			31.3	24.1
$\mathcal{F}$			✓		✓	✓		36.9	27.8
$\mathcal{F}$*			✓		✓	✓*		36.9	27.6
$\mathcal{F}$ + RCZ			✓		✓	✓	RCZ	36.6	28.0
FLOAT			✓		✓	✓	IZR	**37.1**	**29.6**

TABLE 3: Ablation study: Starting from baseline with no factorization at all, we see that systematically adding components of FLOAT pipeline noticeably improves segmentation quality. $\mathcal{M}_{part}$ is combined decoder for all part-level labels, **FLOAT** $= \mathcal{F} + IZR$ (see Fig. 2) is the proposed model. RCZ stands for Random Crop Zoom (see Sec. 5.3). The * indicates separate decoders for 'left/right' and 'front/back'. 'No factorization' – parts are labelled with concatenated category and associated part name. 'Object' – predicting object labels separately.

5.4. Qualitative Analysis

Fig. 6 shows qualitative comparisons of our framework with existing approaches on Pascal-Part-201, reflecting the improvements gains we observe for mIOU and sqIOU metrics (Table 1). FLOAT is visually superior at segmenting smaller object parts – notice the significantly improved segmentation for parts in object categories `person` (first row) and `cat` (second row). From the examples, we see that FLOAT is also better at learning directionality ('left/right',

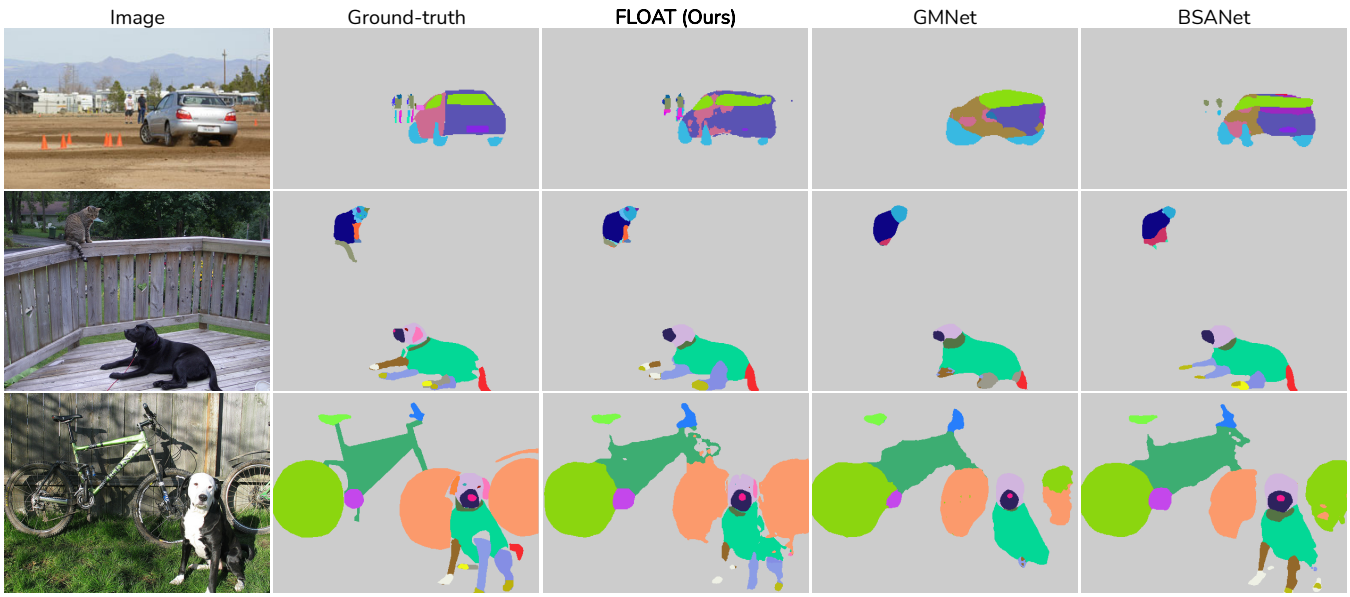

Figure 6: Qualitative comparison on Pascal-Part-201. We observe that FLOAT gets small objects parts – `person` in the upper image, `cat` in the middle image. FLOAT also gets the left-right and front-back correct – `leg(s)` of `dog` and `cat`, `side` of `car`, `wheel` of `bike`.

'front/back'). Similar improvements are evident from the examples provided in Figure 1. Some limitations of FLOAT include missing predictions for the smallest of parts (e.g. `eye` in people far from camera) and partial predictions for thin parts leading to disconnections.

6. Conclusion

FLOAT is a simple but effective framework for improving semantic segmentation performance in multi-object multi-part parsing. Our idea of *factorized label space* is a key contribution which fully takes advantage of label-level intra/inter ontological relationships among objects and parts. The factorization not only enables scalability in terms of both object categories and part labels, but also improves segmentation performance substantially. Another key contribution is our inference-time zoom. By focusing only on object-centric regions of interest, IZR efficiently enhances segmentation quality without requiring explicit object feature guidance or other modifications to the part network setup. Apart from our framework, we introduce a new variant of Pascal-Part called Pascal-Part-201 which constitutes the most challenging benchmark dataset for the problem. Our experimental evaluation, using fairer versions of existing measures, shows that FLOAT clearly outperforms existing state-of-the-art approaches for existing and newly introduced Pascal-Part variants. The gains from our framework increase with increased part and object dataset complexity, empirically supporting our assertion of FLOAT's scalability. Although presented in a 2D scene parsing setting, we expect ideas from FLOAT to be useful for the 3D scene parsing counterpart and in general, for scenarios with appropriately factorizable attributes.

References

[1] Panos Achlioptas, Judy Fan, Robert Hawkins, Noah Goodman, and Leonidas J Guibas. Shapeglot: Learning language for shape differentiation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8938–8947, 2019. 1

[2] Hossein Azizpour and Ivan Laptev. Object detection using strongly-supervised deformable part models. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 836–849, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 1

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017. 2

[4] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987. 3

[5] Andreea Bobu, Chris Paxton, Wei Yang, Balakumar Sundaralingam, Yu-Wei Chao, Maya Cakmak, and Dieter Fox. Learning perceptual concepts by bootstrapping from human queries. *CoRR*, abs/2111.05251, 2021. 1

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

[7] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. 06 2017. 2

[8] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. 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*, pages 1971–1978, 2014. 1, 2, 3, 5, 7

[9] Daan de Geus, Panagiotis Meletis, Chenyang Lu, Xiaoxiao Wen, and Gijs Dubbelman. Part-aware panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5485–5494, 2021. 2

[10] Jian Dong, Qiang Chen, Xiaohui Shen, Jianchao Yang, and Shuicheng Yan. Towards unified human parsing and pose estimation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages

843–850, 2014. 1

[11] Nanqing Dong, Michael Kampffmeyer, Xiaodan Liang, Zeya Wang, Wei Dai, and Eric Xing. Reinforced auto-zoom net: towards accurate and fast breast cancer segmentation in whole-slide images. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 317–325. Springer, 2018. 3

[12] Anastasia Dubrovina, Fei Xia, Panos Achlioptas, Mira Shalah, Raphaël Groscot, and Leonidas J Guibas. Composite shape modeling via latent space factorization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8140–8149, 2019. 1

[13] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 5

[14] Hao-Shu Fang, Guansong Lu, Xiaolin Fang, Jianwen Xie, Yu-Wing Tai, and Cewu Lu. Weakly and semi supervised human body part parsing via pose-guided knowledge transfer. *arXiv preprint arXiv:1805.04310*, 2018. 2

[15] Ke Gong, Xiaodan Liang, Yicheng Li, Yimin Chen, Ming Yang, and Liang Lin. Instance-level human parsing via part grouping network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 770–785, 2018. 2

[16] Hussein Haggag, Ahmed Abobakr, Mohammed Hossny, and Saeid Nahavandi. Semantic body parts segmentation for quadrupedal animals. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 000855–000860. IEEE, 2016. 2

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

[18] D.D. Hoffman and W.A. Richards. Parts of recognition. *Cognition*, 18(1):65–96, 1984. 3

[19] Yining Hong, Li Yi, Joshua B Tenenbaum, Antonio Torralba, and Chuang Gan. Ptr: A benchmark for part-based conceptual, relational, and physical reasoning. In *Advances In Neural Information Processing Systems*, 2021. 1, 2

[20] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019. 2, 6

[21] Jonathan Krause, Hailin Jin, Jianchao Yang, and Li Fei-Fei. Fine-grained recognition without part annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5546–5555, 2015. 1

[22] Tian Lan, Weilong Yang, Yang Wang, and Greg Mori. Image retrieval with structured object queries using latent ranking svm. In *European conference on computer vision*, pages 129–142. Springer, 2012. 2

[23] Xiangtai Li, Xia Li, Li Zhang, Guangliang Cheng, Jianping Shi, Zhouchen Lin, Shaohua Tan, and Yunhai Tong. Improving semantic segmentation via decoupled body and edge supervision. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 435–452. Springer, 2020. 1

[24] Xiaodan Liang, Ke Gong, Xiaohui Shen, and Liang Lin. 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. 2

[25] Xiaodan Liang, Xiaohui Shen, Jiashi Feng, Liang Lin, and Shuicheng Yan. Semantic object parsing with graph lstm. In *European Conference on Computer Vision*, pages 125–143. Springer, 2016. 2

[26] Xiaodan Liang, Chunyan Xu, Xiaohui Shen, Jianchao Yang, Si Liu, Jinhui Tang, Liang Lin, and Shuicheng Yan. Human parsing with contextualized convolutional neural network. In *Proceedings of the IEEE international conference on computer vision*, pages 1386–1394, 2015. 2

[27] Qing Liu, Adam Kortylewski, Zhishuai Zhang, Zizhang Li, Mengqi Guo, Qihao Liu, Xiaoding Yuan, Jiteng Mu, Weichao Qiu, and Alan Yuille. Cgpart: A part segmentation dataset based on 3d computer graphics models. *arXiv preprint arXiv:2103.14098*, 2021. 2

[28] Yunan Liu, Liang Zhao, Shanshan Zhang, and Jian Yang. Hybrid resolution network using edge guided region mutual information loss for human parsing. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1670–1678, 2020. 2

[29] Yawei Luo, Zhedong Zheng, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Macro-micro adversarial network for human parsing. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018. 2

[30] Mateusz Malinowski and Mario Fritz. A pooling approach to modelling spatial relations for image retrieval and annotation. *arXiv preprint arXiv:1411.5190*, 2014. 2

[31] Umberto Michieli, Edoardo Borsato, Luca Rossi, and Pietro Zanuttigh. Gmnet: Graph matching network for large scale part semantic segmentation in the wild. In *European Conference on Computer Vision*, pages 397–414. Springer, 2020. 1, 2, 4, 5, 6, 7

[32] Ishan Misra, Abhinav Gupta, and Martial Hebert. From red wine to red tomato: Composition with context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1792–1801, 2017. 2

[33] Muhammad Ferjad Naeem, Yongqin Xian, Federico Tombari, and Zeynep Akata. Learning graph embeddings for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 953–962, 2021. 2

[34] Tushar Nagarajan and Kristen Grauman. Attributes as operators: factorizing unseen attribute-object compositions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 169–185, 2018. 2

[35] Xuecheng Nie, Jiashi Feng, and Shuicheng Yan. Mutual learning to adapt for joint human parsing and pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 502–517, 2018. 2

[36] Lorenzo Porzi, Samuel Rota Bulo, and Peter Kontschieder. Improving panoptic segmentation at all scales. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7302–7311, 2021. 3

[37] Yafei Song, Xiaowu Chen, Jia Li, and Qinping Zhao. Embedding 3d geometric features for rigid object part segmentation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[38] Jian Sun and Jean Ponce. Learning discriminative part detectors for image classification and cosegmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 3400–3407, 2013. 1

[39] Xin Tan, Jiachen Xu, Zhou Ye, Jinkun Hao, and Lizhuang Ma. Confident semantic ranking loss for part parsing. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2021. 1, 2, 5, 6, 7

[40] Jianyu Wang and Alan L Yuille. Semantic part segmentation using compositional model combining shape and appearance. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1788–1797, 2015. 2

[41] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. Joint object and part segmentation using deep learned potentials. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1573–1581, 2015. 2

[42] Zhe Wang, Yanxin Yin, Jianping Shi, Wei Fang, Hongsheng Li, and Xiaogang Wang. Zoom-in-net: Deep mining lesions for diabetic retinopathy detection. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 267–275. Springer, 2017. 3

[43] Fangting Xia, Peng Wang, Liang-Chieh Chen, and Alan L Yuille. Zoom better to see clearer: Human and object parsing with hierarchical auto-zoom net. In *European Conference on Computer Vision*, pages 648–663. Springer, 2016. 2, 3

[44] Fangting Xia, Peng Wang, Xianjie Chen, and Alan L Yuille. Joint multi-person pose estimation and semantic part segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6769–6778, 2017. 2

[45] Fangting Xia, Jun Zhu, Peng Wang, and Alan Yuille. Pose-guided human parsing with deep learned features. *arXiv preprint arXiv:1508.03881*, 2015. 2

[46] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *European Conference on Computer Vision*. Springer, 2018. 2

[47] Jingtao Xu, Yali Li, and Shengjin Wang. Adazoom: Adaptive zoom network for multi-scale object detection in large scenes. *arXiv preprint arXiv:2106.10409*, 2021. 3

[48] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang.

Denseaspp for semantic segmentation in street scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3684–3692, 2018. 1

[49] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018. 1

[50] Liangzhe Yuan, Yibo Chen, Hantian Liu, Tao Kong, and Jianbo Shi. Zoom-in-to-check: Boosting video interpolation via instance-level discrimination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12183–12191, 2019. 3

[51] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 173–190. Springer, 2020. 1

[52] Ning Zhang, Jeff Donahue, Ross B. Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. *CoRR*, abs/1407.3867, 2014. 1

[53] Zhenli Zhang, Xiangyu Zhang, Chao Peng, Xiangyang Xue, and Jian Sun. Exfuse: Enhancing feature fusion for semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–284, 2018. 1

[54] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 405–420, 2018. 1

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

[56] Jian Zhao, Jianshu Li, Xuecheng Nie, Fang Zhao, Yunpeng Chen, Zhecan Wang, Jiashi Feng, and Shuicheng Yan. Self-supervised neural aggregation networks for human parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 7–15, 2017. 2

[57] Yifan Zhao, Jia Li, Yu Zhang, and Yonghong Tian. Multi-class part parsing with joint boundary-semantic awareness. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9177–9186, 2019. 1, 2, 5, 6, 7

[58] Shuai Zheng, Ming-Ming Cheng, Jonathan Warrell, Paul Sturgess, Vibhav Vineet, Carsten Rother, and Philip H. S. Torr. Dense semantic image segmentation with objects and attributes. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, Ohio, United States, 2014. 2

[59] 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. 1