International Journal of Computer Vision (2022) 130:2797–2821
https://doi.org/10.1007/s11263-022-01671-z

Edge-Aware Graph Matching Network for Part-Based Semantic Segmentation

Umberto Michieli[1] · **Pietro Zanuttigh**[1]

Received: 23 December 2021 / Accepted: 10 August 2022 / Published online: 3 September 2022

Abstract
Semantic segmentation of parts of objects is a marginally explored and challenging task in which multiple instances of objects and multiple parts within those objects must be recognized in an image. We introduce a novel approach (GMENet) for this task combining object-level context conditioning, part-level spatial relationships, and shape contour information. The first target is achieved by introducing a class-conditioning module that enforces class-level semantics when learning the part-level ones. Thus, intermediate-level features carry object-level prior to the decoding stage. To tackle part-level ambiguity and spatial relationships among parts we exploit an adjacency graph-based module that aims at matching the spatial relationships between parts in the ground truth and predicted maps. Last, we introduce an additional module to further leverage edges localization. Besides testing our framework on the already used Pascal-Part-58 and Pascal-Person-Part benchmarks, we further introduce two novel benchmarks for large-scale part parsing, i.e., a more challenging version of Pascal-Part with 108 classes and the ADE20K-Part benchmark with 544 parts. GMENet achieves state-of-the-art results in all the considered tasks and furthermore allows to improve object-level segmentation accuracy.

Keywords Part parsing · Semantic segmentation · Graph matching · Edge localization · Coarse-to-fine learning

1 Introduction

In recent years, many different approaches for semantic segmentation (Chen et al., 2018; Zhao et al., 2017a; Long et al., 2015) have been developed, however state-of-the-art methodologies focus on object-level semantic segmentation without addressing in any way the internal decomposition of the various object into their parts. The fine-grained decomposition into the different parts of each object in the image provides a richer representation for many challenging tasks, including pose estimation (Dong et al., 2014; Yang & Ramanan, 2011), category detection (Chen et al., 2014; Azizpour & Laptev, 2012; Zhang et al., 2014), fine-grained action detection (Wang et al., 2012) and image classification (Sun & Ponce, 2013; Krause et al., 2015). Even if in principle it is possible to train a generic semantic segmentation model using part-level annotations, current semantic segmentation approaches are not optimal for the task of distinguishing between different semantic parts, i.e., they do not account for the fact that corresponding parts in different semantic classes often share similar appearance and they typically capture limited local context. As exemplified in Fig. 1, accounting for the spatial relationships between semantic parts and for their interactions require additional provisions. As in the two examples reported in the figure, seeing only a limited portion of the image (e.g., the regions highlighted by the ovals) prevents a proper understanding of the overall scene and consists in a major source of errors during semantic segmentation. Thus, training a state-of-the-art semantic segmentation architecture by treating each part as an independent class without accounting for how they are arranged into the corresponding object, leads to sub-optimal performance as we show in Sect. 7.

While object-level semantic segmentation has been widely explored, part parsing has only been marginally addressed, and mostly in the context of a single specific type of object. The most explored task is human part parsing (Liang et al., 2015; Yamaguchi et al., 2012; Zhu et al., 2011; Eslami &

Communicated by Jifeng Dai.

✉ Pietro Zanuttigh
zanuttigh@dei.unipd.it

Umberto Michieli
michieli@dei.unipd.it

[1] Department of Information Engineering, University of Padova, 35131 Padua, Italy

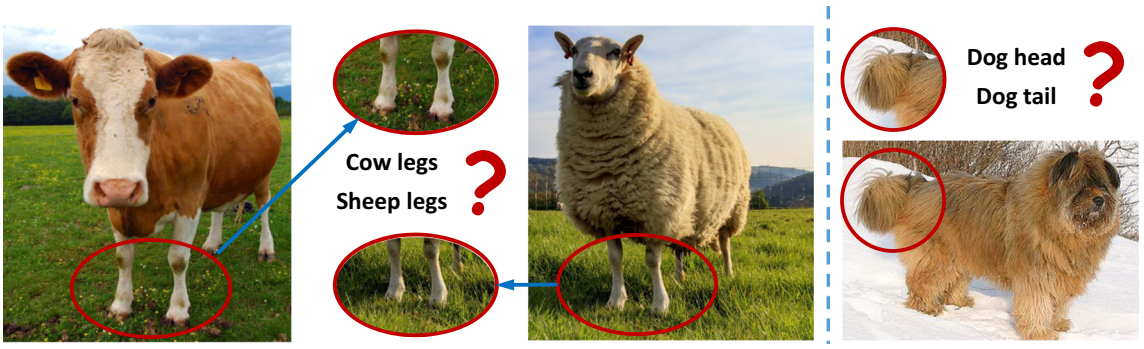

Fig. 1 Left: corresponding parts in different semantic classes often share similar appearance (*cow legs* and *sheep legs* are almost indistinguishable without larger object-level context). Right: sometimes semantic localization of parts within objects is challenging due to their similar appearance (the highlighted oval could be both *dog head* or *dog tail* without additional awareness of reciprocal spatial localization)

Williams, 2012), since it represents a key step for the pose estimation task. A few works explored the part parsing of cars (Song et al., 2017; Lu et al., 2014) or animals (Wang & Yuille, 2015; Wang et al., 2015; Haggag et al., 2016).

The only works dealing with the challenging scenario of multi-class part-based semantic segmentation where part-level as well as object-level ambiguities are present, are BSANet (Zhao et al., 2019) and the conference version of this work (Michieli et al., 2020). BSANet (Zhao et al., 2019) was the first work to investigate the problem of multi-object part parsing even if it addressed a single task with 58 parts. In the previous version of this work (Michieli et al., 2020) we introduced a more advanced learning architecture able to tackle a more challenging experimental setup with a large number of parts contained in the scenes (up to 108 parts).

The exploitation of object-level information in part parsing is strictly related to a very active research direction that is the transfer of previous knowledge, acquired on a different but related task, to a new setting. Different interpretations may exist to this regard. In the class-incremental learning task, the learned model is updated to perform a new task whilst preserving previous knowledge: many methods have been proposed for image classification (Dhar et al., 2019; Rebuffi et al., 2017; Li & Hoiem, 2018), object detection (Shmelkov et al., 2017) and semantic segmentation (Michieli & Zanuttigh, 2019; Cermelli et al., 2020; Michieli & Zanuttigh, 2021a; Douillard et al., 2021; Michieli & Zanuttigh, 2021b; Maracani et al., 2021). In semantic-level coarse-to-fine learning, previous knowledge acquired on a coarser task is exploited to perform a finer-grained segmentation task (Hariharan et al., 2015; Xia et al., 2017; Mel et al., 2020). Differently, in this work we analyze coarse-to-fine refinement at the spatial level, where part-level classes are hierarchically derived from the object-level classes (Wang et al., 2015; Xia et al., 2016; Zhao et al., 2019).

More precisely, we investigate the multi-object and multi-part parsing in the wild, where different semantic objects and multiple parts within each object are present in the scene. Current state-of-the-art architectures designed for classical object-level semantic segmentation, both based on the widely used encoder-decoder architecture (Long et al., 2015; Badrinarayanan et al., 2017; Zhao et al., 2017a; Chen et al., 2017, 2018) and on more recent vision transformers schemes (Dosovitskiy et al., 2021; Zheng et al., 2021; Liu et al., 2021; Strudel et al., 2021; Xie et al., 2021), face additional challenges when dealing with this task, as preliminarily shown in Zhao et al. (2019). In particular, the simultaneous appearance of multiple objects and the inter-class ambiguity may cause inaccurate edge localization and severe classification errors. For instance, animals often have homogeneous appearance due to furs on the whole body. Additionally, the appearance of some parts over multiple object classes may be visually similar, such as *cow legs* and *sheep legs* in Fig. 1. Standard semantic segmentation models heavily suffer from these aspects. In this work we propose an object-level conditioning module to address object-level ambiguity, acting as a guidance for part parsing within the object. An auxiliary reconstruction module matching the part-level predictions with the object-level ones further penalizes predictions of parts in regions occupied by an object to which the predicted parts do not belong. Thus, predicting a wrong part of the same object-level class is treated as a less severe type of error. To address part-level ambiguity, we design a graph-matching module able to preserve the relative spatial relationships between ground truth and predicted parts. Finally, contours are further preserved through a separate edge module.

When humans look at scenes, they tend to first locate the objects and then to refine the understanding by analyzing the parts composing the objects (Xia et al., 2016). Similarly, our class-conditioning approach refines part-level localization, by enforcing object-level semantics. In partic-

ular, object-level predictions of an off-the-shelf architecture serve as a conditioning term to guide the part-level decoding stage. The predictions are processed via an object-level semantic embedding CNN, whose features are concatenated with those produced by the encoder designated to tackle part-level recognition. Therefore, part-level extracted features are enriched with object-level information prior, bringing object-level awareness at the part-level decoding stage. Furthermore, we address part-level ambiguity via a graph-matching technique computed over the segmentation maps. In order to model vicinity among parts, we build a couple of weighted adjacency graphs from both the ground truth and the predicted segmentation maps, where the weight is given by the normalized number of adjacent pixels to represent the strength of connection between the parts. Then, an ad-hoc loss function enforces their similarity during training. Finally, to further improve the accuracy of contours of the part shapes, enhanced edge-related awareness is brought directly to the encoding layers via an auxiliary branch forcing the matching of the edge information with the ground truth one. These provisions allow the architecture (1) to discover the differences in appearance between different parts within a single object, (2) to avoid the ambiguity across similar object categories, and (3) to improve segmentation of classes close to the contours.

Compared to the conference version (Michieli et al., 2020), the main contributions of this paper are:

- we extended our approach (now called GMENet) to give more importance to edges localization, since the regions close to the edges account for most of the errors in the predicted segmentation map;
- we introduced an additional large-scale benchmark (ADE 20K-Part) containing 544 parts originating from 150 objects of the ADE20K dataset (Zhou et al., 2017);
- we included a new experimental evaluation on standard (object-level) semantic segmentation (on Pascal VOC2012 (Everingham et al., 2010) and on Cityscapes (Cordts et al., 2016)), showing how the provided methods could improve the results on both part-level and object-level semantic segmentation;
- we included more comparisons with existing state-of-the-art segmentation methods and more qualitative results on all the scenarios;
- our complete approach (GMENet) outperforms the previous version and achieves new state-of-the-art performance on part parsing, especially when multiple small parts are present.

2 Related Work

Semantic Segmentation is a long-standing problem towards detailed scene understanding (recent reviews on the topic are (Liu et al., 2019a; Guo et al., 2018)). The Fully Convolutional Network (FCN) framework (Long et al., 2015) firstly enabled accurate and end-to-end semantic segmentation, introducing the basic encoder-decoder model that has then been extended in many following works. Widely used approaches building on top of this scheme include SegNet (Badrinarayanan et al., 2017), PSPNet (Zhao et al., 2017a), DRN (Yu et al., 2017), UNet (Ronneberger et al., 2015) and BiSeNet (Yu et al., 2018, 2021). The DeepLab (Chen et al., 2017, 2018) family of architectures is typically regarded as the state-of-the-art encoder-decoder architecture for semantic segmentation. Following recent progresses in Natural Language Processing (Vaswani et al., 2017), several segmentation methods couple convolutional backbones with alternative aggregation schemes based on channel, spatial (Fu et al., 2019; Yuan et al., 2018; Huang et al., 2019, 2020) or point-wise (Fu et al., 2019) attention to better capture contextual information. Recently, Vision Transformers (ViT) (Dosovitskiy et al., 2021) introduced a convolution-free transformer architecture for downstream vision tasks. SETR (Zheng et al., 2021) uses a ViT backbone and a standard CNN decoder. Swin (Liu et al., 2021) and Twins (Chu et al., 2021) transformers compute self-attention within a local (windowed) region. The Segmenter work (Strudel et al., 2021) proposes a transformer encoder-decoder architecture. TopFormer (Zhang et al., 2022) exploits tokens from different scales and pools them efficiently. SegFormer (Xie et al., 2021) focus on removing positional encoding at the encoder and designs the decoder as a multilayer perceptron. TrSeg (Jin et al., 2021) adaptively captures multi-scale information with the dependencies on original contextual information.

Single-Object Part Parsing has been actively researched in the recent literature. However, the vast majority of previous works deals with images containing only the considered object, well-localized beforehand and without occlusions. Recent works tackled this problem for animals (Wang & Yuille, 2015), cars (Eslami & Williams, 2012; Lu et al., 2014; Song et al., 2017) and especially humans (Liang et al., 2015; Yamaguchi et al., 2012; Zhu et al., 2011; Eslami & Williams, 2012; Yin et al., 2021; He et al., 2021a; Li et al., 2017).

Standard deep networks regarding each semantic part as a separate class label are inadequate to solve the task. Therefore, some coarse-to-fine strategies have been proposed. Hariharan et al. (2015) propose to sequentially perform object detection, object segmentation and part segmentation with different architectures. However, there are some limitations, in particular the complexity of the training and the error propagation throughout the pipeline. An upgraded version of the framework has been presented in Xia et al. (2016), where

the same structure is employed for the three networks and an automatic adaptation to the size of the object is introduced. In Wang et al. (2015) a two-channels FCN is employed to jointly infer object and part segmentation for animals. However, it uses only a single-scale network not capturing small parts and uses a post-processing based on a fully connected CRF to explore the relationship between parts and body in order to make the final prediction. An attention mechanism that learns to softly weight the multi-scale features at each location is proposed in Chen et al. (2016).

Some approaches resort to structure-based methodologies, e.g., compositional, to model part relations (Wang & Yuille, 2015; Wang et al., 2015; Liang et al., 2018, 2017, 2016; Fang et al., 2018). Wang & Yuille (2015) propose a model to learn a mixture of compositional models under various poses and viewpoints for certain animal classes. In Liang et al. (2018) a self-supervised structure-sensitive learning approach is proposed to constrain human pose structures into parsing results. In (Liang et al., 2016, 2017) graph LSTMs are employed to refine the parsing results of initial over-segmented superpixel maps. Pose estimation is also useful for part parsing task (Xia et al., 2017; Nie et al., 2018; Fang et al., 2018; Liang et al., 2018; Zhao et al., 2017b). In Xia et al. (2017), the authors refine the segmentation maps by supervised pose estimation. In Nie et al. (2018) a mutual learning model is built for pose estimation and part segmentation. In Fang et al. (2018), the authors exploit anatomical similarity among humans to transfer the parsing results of a person to another person with similar pose. In Zhao et al. (2017b) multi-scale features aggregation at each pixel is combined with a self-supervised joint loss to further improve the feature discriminative capacity. Other approaches utilize tree-based structures to hierarchically partition the parts (Lu et al., 2014; Xia et al., 2015). Lu et al. (2014) propose a method based on tree-structured graphical models from different viewpoints combined with segment appearance consistency for part parsing. Xia et al. (2015) firstly generate part segment proposals and then infer the best ensemble of parts-segment through and-or graphs. Recently, some approaches have considered attention-based transformer architectures (Yuan et al., 2018) or part-aware panoptic segmentation schemes (de Geus et al., 2021).

Multi-Object and Multi-Part Parsing has been considered only recently in Zhao et al. (2019) and Michieli et al. (2020). Most of the previous techniques fails in addressing this task due to the presence of simultaneous objects in the scene that were not previously well-localized nor isolated. Zhao et al. (2019) tackle this task via a joint parsing framework with boundary and semantic awareness for enhanced part localization and object-level guidance. Part boundaries are detected at early stages of feature extraction and then used in an attention mechanism to emphasize the features along the boundaries at the decoding stage. An additional attention module is employed to perform channel selection and is supervised by a supplementary branch predicting the semantic object classes. In the conference version of this work (Michieli et al., 2020) we introduced a graph matching module and an object-level segmentation to aid learning of different parts in the scene.

3 Method

When humans look at images, they often firstly locate the objects and then they decompose them into the various parts using mainly two priors: (1) object-level information and (2) relative spatial relationships among parts. In our approach we replicate a similar behavior: an initial object-level prediction supports semantic parts parsing. Furthermore, part-level predictions are reinforced via a graph-matching strategy to match neighboring parts, and via an auxiliary edge branch to refine the contour edges.

An outlook of the proposed framework is shown in Fig. 2: there are two semantic segmentation networks $\mathcal{A}_o$, trained for the objects-level task, and $\mathcal{A}_p$, trained for the parts-level one. Furthermore, a semantic embedding network $\mathcal{S}$ takes care of processing and transferring the information of the first network to the second to address the object-level prior. This novel coarse-to-fine strategy is the main focus of this section. We account for the second clue by introducing an adjacency graph structure to mimic the spatial relationship between neighboring parts. This module will be detailed in Sect. 4. Finally, an edge module $\mathcal{E}_{ed}$ brings part-level contour awareness at the encoding stage, as detailed in Sect. 5.

The contributions from all these components compose the overall training objective of our framework, leading to the minimization of:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda_1 \mathcal{L}_{rec} + \lambda_2 \mathcal{L}_{GM} + \lambda_3 \mathcal{L}_{ed} \tag{1}$$

where $\mathcal{L}_{CE}$ is the standard cross entropy loss at part-level, $\mathcal{L}_{rec}$ forces the consistency of the object-level and part-level segmentations, $\mathcal{L}_{GM}$ is the contribution from the graph module capturing the spatial arrangement of parts and $\mathcal{L}_{ed}$ forces the proper contours localization. The hyper-parameters λ_1, λ_2 and λ_3 are used to control the relative contribution of the losses to the overall objective function.

Most state-of-the-art semantic segmentation networks have an encoder-decoder structure and they can be represented as the composition of an encoder and a decoder, i.e., as $\mathcal{A}_o = \{\mathcal{E}_o, \mathcal{D}_o\}$ for the object-level network and $\mathcal{A}_p = \{\mathcal{E}_p, \mathcal{D}_p\}$ for the part-level one, respectively. In our experiments, we employ the DeepLab-v3 (Chen et al., 2017) segmentation network with ResNet-101 (He et al., 2016) as encoder since it achieves state-of-the-art results; however, the proposed approach is network agnostic and can be added

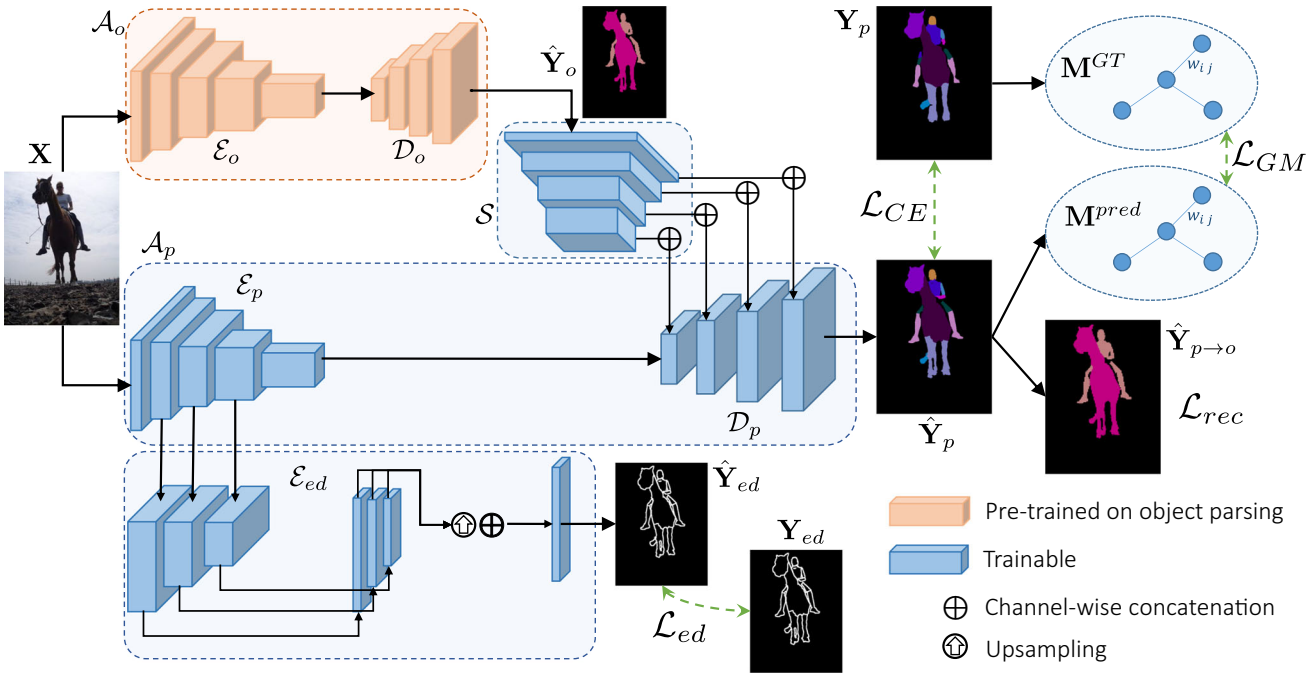

Fig. 2 Architecture of the proposed Graph Matching and Edge Network (GMENet) approach. A semantic embedding network $\mathcal{S}$ takes as input the object-level segmentation map and acts as high-level conditioning when learning the semantic segmentation of parts. On the right, a reconstruction loss function rearranges parts into objects and the graph matching module aligns the relative spatial relationships between ground truth and predicted parts. At the bottom, an auxiliary branch predicts precise edge localization even within parts of the same semantic object-level class to bring more contour-aware encoded feature maps

on top of any encoder-decoder segmentation network. The module $\mathcal{A}_o$ is pre-trained for object-level semantic segmentation and then its weights are frozen. $\mathcal{A}_o$ outputs object-level segmentation maps $\hat{\mathbf{Y}}_o$ which are employed by the part-level decoder $\mathcal{D}_p$, to correct the predictions of parts $\hat{\mathbf{Y}}_p$ which are similar across different objects. Indeed, we fed $\hat{\mathbf{Y}}_o$ to an object-level semantic embedding CNN, $\mathcal{S}$, formed by a cascade of 4 convolutional layers with stride 2, square kernel sizes of 7, 5, 3, 3, and channel sizes of 128, 256, 512, 1024.

The part-level semantic segmentation network $\mathcal{A}_p$ has the same encoder architecture of $\mathcal{A}_o$. Its decoder $\mathcal{D}_p$, instead, merges the features computed from the input sample (the output of $\mathcal{E}_p$) and the ones computed on the object-level predicted map, $\mathcal{S}(\hat{\mathbf{Y}}_o)$, via multiple channel-wise concatenations. More in detail, each layer of the decoder considers a different resolution and its feature maps are concatenated with the layer at corresponding resolution of $\mathcal{S}$. Therefore, the combination is performed at multiple resolutions in the feature space to achieve higher scale invariance as visible in Fig. 2.

Formally, given an input sample $\mathbf{X} \in \mathbb{R}^{W\times H}$, the concatenation between part- and object-level aware features is given by

$$\mathcal{F}_i(\mathbf{X}) = \mathcal{D}_{p,i}(\mathbf{X}) \oplus \mathcal{S}_{k+1-i}(\mathcal{A}_o(\mathbf{X})) \quad i = 1,\ldots,k \tag{2}$$

where $\mathcal{D}_{p,i}$ is the i-th decoding layer of the part segmentation network, $\mathcal{S}_i$ indicates the i-th layer of $\mathcal{S}$, k is the number of layers and it matches the number of upsampling stages of the decoder (e.g., $k = 4$ in the DeepLab-v3), $\mathcal{F}_i$ is the input of $\mathcal{D}_{p,i+1}$. We remark that the performed object-level segmentation can be different from the ground truth one, depending on the accuracy of the prediction. Therefore, errors from the predicted class in the object segmentation may propagate to the part-level prediction stage. To alleviate this circumstance, similarly to Xia et al. (2016), here we do not make early decisions and the channel-wise concatenation leaves the final labeling decision to the part-level decoder. We will analyze the impact of such aspect in the ablation studies (Sect. 7.6).

As anticipated in Eq. 1, the training of $\mathcal{A}_p$ and $\mathcal{S}$ (note that $\mathcal{A}_o$ is frozen after the initial training) is driven by multiple constraints. The first is a standard cross-entropy loss $\mathcal{L}_{CE}$ to learn the part-level semantic segmentation task:

$$\mathcal{L}_{CE} = \sum_{c_p=1}^{N_p} \mathbf{Y}_p[c_p] \cdot \log\left(\hat{\mathbf{Y}}_p[c_p]\right) \tag{3}$$

where $\mathbf{Y}_p$ is the one-hot encoded ground truth map, $\hat{\mathbf{Y}}_p$ is the predicted map, c_p is the part-level index and N_p is the number of parts.

The object-level semantic embedding network is further guided by a reconstruction module that rearranges parts into objects. This is achieved by means of a cross-entropy loss between the object-level one-hot encoded ground truth maps $\mathbf{Y}_o$ and the cumulative probabilities $\hat{\mathbf{Y}}_{p\to o}$ derived from the part-level predictions. Let us define l as the parts-to-objects mapping such that object j contains parts from index $l[j-1]+1$ to $l[j]$, thus we can write the summed probability as:

$$\hat{\mathbf{Y}}_{p\to o}[j] = \sum_{i=l[j-1]+1,\ldots,l[j]} \hat{\mathbf{Y}}_p[i] \qquad j = 1,\ldots,N_o \tag{4}$$

where N_o is the number of object-level classes and l is initialized as $l[0] = 0$. Then, we define the reconstruction loss as:

$$\mathcal{L}_{rec} = \sum_{c_o=1}^{N_o} \mathbf{Y}_o[c_o] \cdot \log\left(\hat{\mathbf{Y}}_{p\to o}[c_o]\right) \tag{5}$$

The auxiliary reconstruction function $\mathcal{L}_{rec}$ acts differently from the usual cross-entropy loss on the parts $\mathcal{L}_{CE}$. While $\mathcal{L}_{CE}$ penalizes wrong predictions of parts in all the portions of the image, $\mathcal{L}_{rec}$ only penalizes for part-level predictions located outside the respective object-level class. In other words, the event of predicting parts *outside* the respective object-level class is penalized by both $\mathcal{L}_{CE}$ and $\mathcal{L}_{rec}$. Instead, parts predicted *within* the object class are penalized exclusively by $\mathcal{L}_{CE}$, i.e., they are considered as a less severe type of error since, in this case, parts only need to be properly localized inside the object-level class.

4 Graph-Matching for Semantic Parts Localization

Moving from previous considerations, both providing global context information and disentangling relationships is useful to distinguish fine-grained parts. For instance, upper and lower arms share highly similar appearance. To differentiate between them, both global and reciprocal information, like the relationship with contiguous parts, provide an effective context prior. In the previous section, we have already addressed global context conditioning through the semantic embedding network and the reconstruction constraint. In this section we further enhance the accuracy of part parsing, tackling part-level ambiguity and localization via a graph-based module to match part-level spatial relationships between ground truth and predicted maps. More in detail, the graphs capture the adjacency relationships between each couple of parts, then we enforce the matching between the ground truth and predicted graphs through an additional loss term. Although graph matching is a well studied problem (Emmert-Streib et al., 2016; Livi & Rizzi, 2013), to the best of our knowledge it has not been applied to this task before, i.e., to constraint deep learning architectures for semantic segmentation. The only attempt to design a graph-matching loss is Das & Lee (2018), which deals with the completely different task of domain adaptation in image classification, and has a different interpretation of the graph, that measures the similitude between the source and the target domains.

An overview of the proposed graph matching module is shown in Fig. 3. We represent the graphs using a couple of (square) weighted adjacency matrices of size N_p:

$$\tilde{\mathbf{M}}^{GT} = \left\{\tilde{m}^{GT}_{i,j}\right\}_{\substack{i=1,\ldots,N_p \\ j=1,\ldots,N_p}} \tag{6}$$

$$\tilde{\mathbf{M}}^{pred} = \left\{\tilde{m}^{pred}_{i,j}\right\}_{\substack{i=1,\ldots,N_p \\ j=1,\ldots,N_p}} \tag{7}$$

The matrix $\tilde{\mathbf{M}}^{GT}$ contains the adjacency information computed over the ground truth maps, while $\tilde{\mathbf{M}}^{pred}$ contains the same information computed over the predicted maps. Each element of the matrices provides a measure of closeness between parts p_i and p_j in the respective segmentation map (either ground truth or predicted). Self-connections are clueless, hence $\tilde{m}^{GT}_{i,i} = \tilde{m}^{pred}_{i,i} = 0$ for $i = 1,\ldots,N_p$. The weight between couples of parts represents the strength of connection between them: each entry $\tilde{m}_{i,j}$ of the matrices depends on the length of the contour in common between the two parts. Additionally, to cope for imprecise parts localization implicitly present in the dataset, where some adjacent parts are separated by thin background regions, the entries $\tilde{m}_{i,j}$ are the counts of pixels belonging to one part with a distance less or equal than $2T$ from a sample belonging to the other part. In other words, $\tilde{m}^{GT}_{i,j}$ represents the number of pixels in p_i whose distance from a pixel in p_j is less than $2T$. Empirically, we set $T = 2$ pixels. Since the matrix $\tilde{\mathbf{M}}^{pred}$ needs to be recomputed at each training step, we approximate this operation by dilating the two masks of T and computing the size of the intersecting region. Formally, defining with $p^{GT}_i := \mathbf{Y}_p[i]$ the i-th part in the ground truth map $\mathbf{Y}_p$, we compute the intersecting area between parts i and j as:

$$\tilde{m}^{GT}_{i,j} = \sum \Phi_T\left(p^{GT}_i\right) \cdot \Phi_T\left(p^{GT}_j\right), \tag{8}$$

where $\Phi_T(\cdot)$ is the morphological 2D dilation operator parametrized by T. The sum is computed over the spatial locations and the multiplication sign computes the intersection between the two parts. Then, we obtain a matrix of *proximity ratios* by applying a row-wise L2 normalization $\mathbf{M}^{GT}_{[i,j]} = \tilde{\mathbf{M}}^{GT}_{[i,j]} / ||\tilde{\mathbf{M}}^{GT}_{[i,:]}||_2$ to measure the flow from the considered part i to all the others (indexed by j).

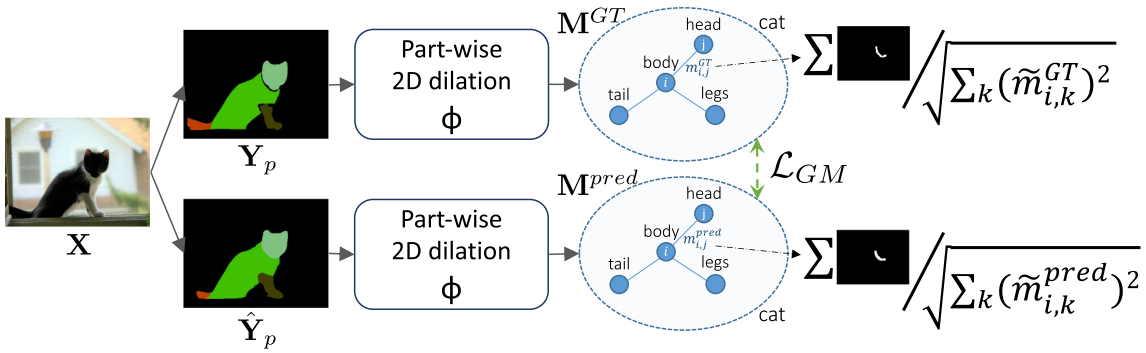

Fig. 3 Overview of the graph-matching module. The morphological dilation over the parts is needed to account for imprecise part localization in the ground truth maps (e.g., *cat head* and *cat body* would be considered as detached without the dilation)

According to the above definition, non-adjacent parts will have 0 as entry. Similarly, we compute the adjacency matrix over the predicted segmentation map $\mathbf{M}^{pred}$ by substituting p_i^{GT} with $p_i^{pred} := \hat{\mathbf{Y}}_p[i]$.

Finally, we define the Graph-Matching loss as the Frobenius norm between the two adjacency matrices:

$$\mathcal{L}_{GM} = ||\mathbf{M}^{GT} - \mathbf{M}^{pred}||_F \tag{9}$$

The aim of this loss function is to faithfully maintain the reciprocal relationships between parts. On one hand, disjoint parts are enforced to be predicted as disjoint; on the other hand, neighboring parts are enforced to be predicted as neighboring and to match the proximity ratios (i.e., the same contour length).

5 Edge Module for Parts Contour Detection

As we have already observed, the contours of the classes are the most difficult regions to be properly detected. Precisely understanding the location of the boundary between classes is an extremely non-trivial task due to many factors: sometimes, the edge region is set as unlabeled by the dataset itself; other times, images present blurring or zooming artifacts; some other times, especially in case of parts, there is a labeling ambiguity on where the boundary is located between two adjacent parts within the same object. Examples of this last aspect are reported in Fig. 4 for parts of the horse class.

We appreciate how the boundary between adjacent parts can be ambiguous as the upper part of the thigh, delimited by a yellow marker, is sometimes labeled as *leg* (top left image) or as *body* (all other images) or the *mane*, delimited by a pink marker, is sometimes included as part of the head (top right corner of top right image) or as *body* (all other cases). Finally, we observe that inaccurate labeling of parts, and therefore of edges, are implicitly present in the dataset (light blue marker).

In light of such issues, computer vision algorithms have always faced increased difficulty in determining precise segmented regions. Therefore, edge localization attempts have started to emerge either by building separate branches (Zhang & Pang, 2020; Han et al., 2020; Li et al., 2020b; Ruan et al., 2019; Li et al., 2020a) or with attention schemes (Zhao et al., 2019; Zhang et al., 2019). Indeed, standard segmentation models fail to properly localize edges even in simple cases where adjacent classes belong to different object classes. We can appreciate this from Fig. 5, where we show the entropy maps of the softmax probabilities $\hat{\mathbf{Y}}$, i.e., $H(\hat{\mathbf{Y}})$, with $H(\cdot)$ being the pixel-wise Shannon entropy (Wan et al., 2019; Vu et al., 2019; Michieli & Ozay, 2021). Low entropy (dark blue) indicates a peaked distribution which is the reflection of high confidence of the network on its prediction, and vice-versa. Ideally, the entropy should be low for every pixel. However, as we can observe even from object-level predictions, the contours of objects have high entropy due to uncertainty on the precise edge localization of the objects. The issue even worsen when considering predictions over the parts, which consist of within-objects edges whose precise localization is even more challenging.

In this work, we tackle this problem by building an auxiliary branch to bring more edge-related awareness at the encoding layers of the segmentation architecture, as we anticipated in Fig. 2. This module can help both recognition among different object-level classes and recognition of within-object parts, as we will discuss in Sect. 7.

The edge module is designed by taking the three central blocks of our encoder (i.e., a ResNet-101) and giving same importance to each edge feature map independently of the different resolutions by setting the same number of channels (i.e., applying three 1×1 convolutional layers with 256 channels). Then, we compute three different edge maps (via three 3×3 convolutional layers with 2 channels) which are later upsampled, concatenated and combined into a single edge map (via a final 1×1 convolutional layer with 2 output channels). The complete flow is shown in the bottom part of Fig. 2.

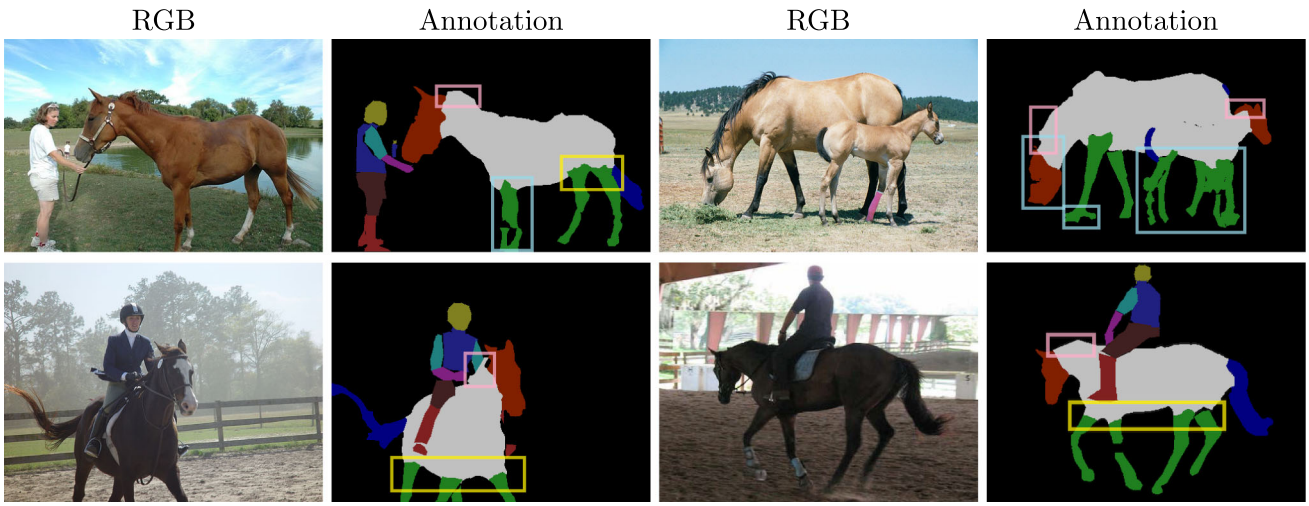

Fig. 4 Precise edge localization is even harder due to imprecise or inconsistent labeling across the dataset (*best viewed in colors*)

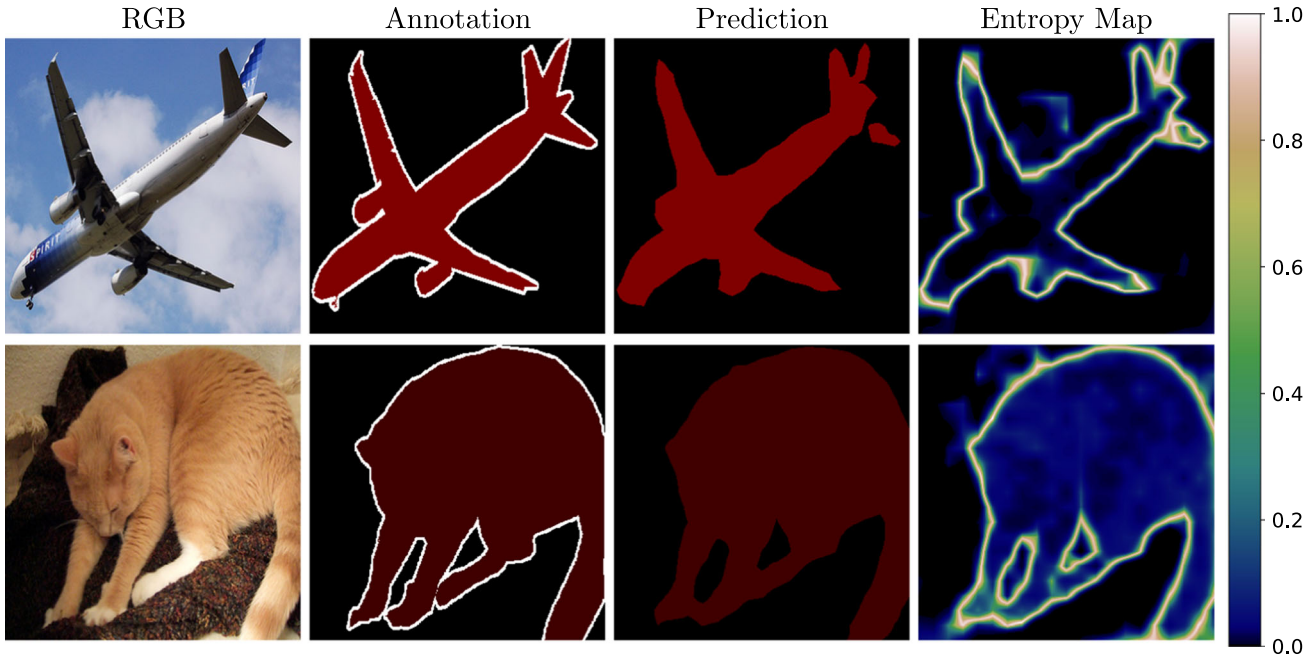

Fig. 5 Entropy maps computed from predicted softmax probabilites show greater uncertainty close to the edges even in simple cases where adjacent classes belong to different object-level classes (*best viewed in colors*)

We train this module to produce output edge maps close to the ground truth ones (which have been obtained from the segmentation maps) using a weighted binary cross entropy loss. The objective is weighted according to the inverse frequency of the two classes (respectively, *edge* and *not edge*) to handle the intrinsic class imbalance (there are much more non-edge pixels than edge ones, i.e., edges only account for a small percentage of the pixels in the training set). More formally, the edge loss is defined by:

$$\mathcal{L}_{ed} = \sum_{c=0}^{1} \omega_c \mathbf{Y}_{ed}[c] \cdot \log\left(\hat{\mathbf{Y}}_{ed}[c]\right) \tag{10}$$

where $\mathbf{Y}_{ed}$ and $\hat{\mathbf{Y}}_{ed}$ are the ground truth and predicted edge maps, ω_c represent the class weights that are set as equal to the inverse frequency of occurrences of the classes over the entire training set $\mathcal{T}$, i.e., for $c \in \{0, 1\}$:

$$\omega_c = \frac{|\{\mathbf{Y}^n_{ed} \text{ for } n = 1, \ldots, |\mathcal{T}|\}|}{|\{\mathbf{Y}^n_{ed}[c] \text{ for } n = 1, \ldots, |\mathcal{T}|\}|} \tag{11}$$

where n iterates over the training set samples $\mathcal{T}$ and $|\cdot|$ represents the cardinality operator of the enclosed set.

6 Training of the Deep Learning Architecture

6.1 Part Parsing Datasets with Multiple Objects

For the experimental evaluation of the proposed multi-class part-parsing framework we employ benchmarks exploiting the Pascal-Part (Chen et al., 2014) and the ADE20K (Zhou et al., 2017) datasets.

Pascal-Part. From the Pascal-Part we consider two different grouping of labels leading to two different benchmarks: namely, the Pascal-Part-58 and the Pascal-Part-108. Both of them contain a total of 10103 variable-sized images with pixel-level annotation of parts derived from the 20 Pascal VOC2010 (Everingham et al., 2010) semantic object classes (plus the *background* class). We use the split into train and test sets of Chen et al. (2014) that uses the *trainset* (4998 samples) for training and the *valset* (5105 samples) for testing. We consider two different sets of labels for this dataset. The *Pascal-Part-58* was proposed by Zhao et al. (2019), which is the first work dealing with the multi-class part parsing problem. In the Pascal-Part-58 the original semantic classes of the Pascal-Part have been grouped into 58 part classes in total. Additionally, we further test our method on an even more challenging scenario that we name *Pascal-Part-108* (Michieli et al., 2020). To build it, we consider the grouping rules proposed by Gonzalez-Garcia et al. (2018), that lead to a larger set of 108 parts (103 actual parts and 5 classes with a single part inside). We also tested our approach on the *Pascal-Person-Part* dataset, which is another subset of the Pascal-Part, containing multiple people within the same scene. This dataset is interesting since it is widely used even if it is designed for a slightly different task, i.e., it deals only with single-object part parsing. We follow the annotations from Chen et al. (2016); Xia et al. (2017), leading to 3533 (1716 for training and 1817 for testing) images of 7 human body parts.

ADE20K-Part. From the ADE20K dataset (Zhou et al., 2017) we followed the same rules proposed in Gonzalez-Garcia et al. (2018) yielding 544 part-level classes belonging to 150 object- and stuff-level categories. Therefore, this split, similarly to the recent released PartImageNet (He et al., 2021b) dataset, includes a high number of part-level classes in a multi-object benchmark, facilitating future research on large-scale semantic segmentation in the wild. We use the original split (Zhou et al., 2017) to divide between 20210 training samples and 2000 validation samples. A summary of statistics for this dataset is shown in Fig. 6. We verify that the label distribution follows a neat power-law rule, both at the pixel level (top plots) and at the image-level (bottom left plot), as many real-world data often show (Liu et al., 2019b; Kang et al., 2019; Cao et al., 2019). This translates in a heavily class imbalanced dataset with few frequent parts (with many pixels belonging to them and many images containing them) and many rare ones (with few pixels and/or contained in few images). Last, we show the occurrence of parts within each image in the bottom right plot of Fig. 6. We observe that the number of parts per each class ranges from 1 (i.e., no parts) to 36 (in case of the *building edifice* class), thus making the dataset highly imbalanced also with this regard. All together, ADE20K-Part is an extremely diverse, yet challenging, dataset with a large variety of visual domains (containing both indoor and outdoor scenes).

6.2 Training Details

The modules of this work are agnostic to the underlying network architecture and can be extended to other scenarios. We employ a DeepLab-v3 (Chen et al., 2017) architecture with ResNet101 (He et al., 2016) as the backbone for two main reasons: it allows a fair comparison with the only previous work in multi-object part parsing (Zhao et al., 2019) and it surpassed competing segmentation architectures in part segmentation. We follow the same training schemes of Chen et al. (2018, 2017) and Zhao et al. (2019) and we added our modules on top of the official TensorFlow (Abadi et al., 2016) implementation of DeepLab-v3 (Chen et al., 2017; Chen, 2020). We used the ResNet101 weights pre-trained on ImageNet (Deng et al., 2009) available at (Chen, 2020). To perform data augmentation images are randomly left-right flipped and rescaled with bilinear interpolation (from 0.5 to 2 times the original size). The results in the testing stage are reported at the original image resolution. The model is trained for N steps with the base learning rate set to $5 \cdot 10^{-3}$ and decreased with a polynomial decay rule with power 0.9 ($N = 50K$ for Pascal-Part, $N = 80K$ for ADE20K-Part). We employ weight decay regularization of 10^{-4}. The atrous rate in the Atrous Spatial Pyramid Poooling (ASPP) is set to (6, 12, 18) as in Chen et al. (2018); Zhao et al. (2019). We set $\lambda_1 = 10^{-3}$, $\lambda_2 = 10^{-1}$ and $\lambda_3 = 10^{-3}$ to balance the contributions of proposed modules and we found these parameters to perform well across different datasets. We use a batch size of 10 images for Pascal-Part datasets and a batch size of 5 for ADE20K-Part. All the compared approaches are trained for the same number of epochs. As evaluation metrics, we employ the mean Intersection over Union (mIoU) across all the parts, the average IoU for all the parts belonging to each single object, and the mean of these values (denoted as Avg). Notice that in this latter case, differently from the mIoU, each object has the same weight independently of the number of parts. As in previous approaches, pixel accuracy is not shown since it is strongly dependent on performance on (a few) large parts and does not capture the results in the segmentation of many small parts, which are the main target

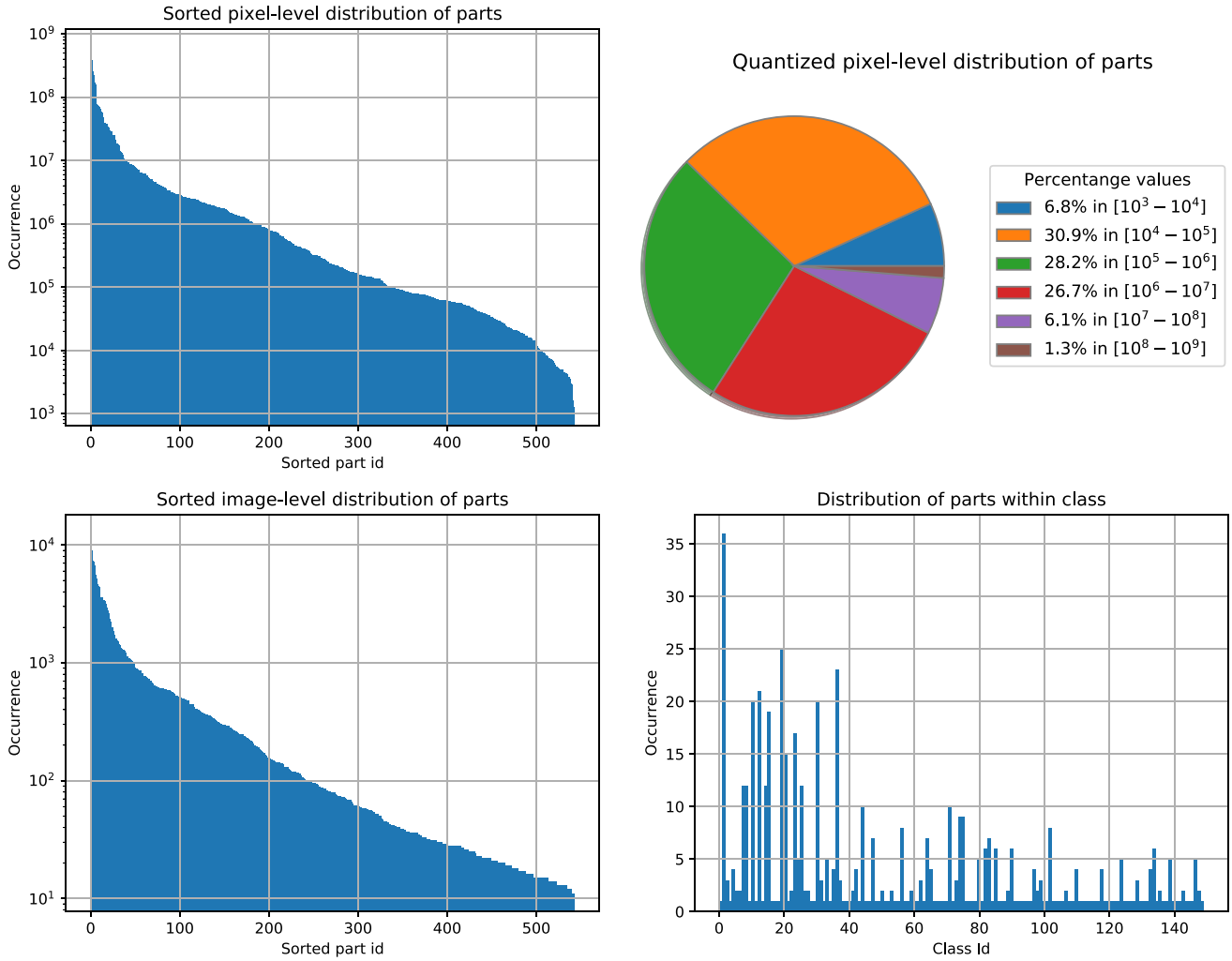

Fig. 6 Distributions of classes computed over the ADE20K-Part dataset. *Top left*: sorted pixel-level distribution of parts (i.e., the bars show many pixels belong to each part). *Top right*: quantized occurrences of the sorted pixel-level distribution of parts. *Bottom left*: sorted image-level distribution of parts (i.e., each bar shows how many images contain the corresponding part). *Bottom right*: distribution of parts within classes (i.e., the bars show how many parts each macro class has)

of our work (Zhao et al., 2019; Csurka et al., 2013). The part labels and the code are publicly available at https://lttm.dei.unipd.it/paper_data/GMENet.

7 Experimental Results

In this section we show the experimental results on the multi-class part-parsing task on the Pascal-Part-58 and 108 and on the ADE20K-Parts. Then, we move to analyse how the proposed module could boost the performance of standard training on object-level semantic segmentation. We also present some ablation studies to verify the effectiveness of the proposed methodologies.

7.1 Pascal-Part-58

To evaluate our framework we start from the scenario with 58 parts, i.e., the same experimental setting used in Zhao et al. (2019). In Table 1 we compare the proposed model with existing semantic segmentation schemes. As expected, standard semantic segmentation architectures such as FCN (Long et al., 2015), SegNet (Badrinarayanan et al., 2017), DeepLab (Chen et al., 2018), DRN-D 38 and DRN-D 105 (Yu et al., 2017) are unable to provide very good results on part parsing. We adopt as our baseline network the DeepLab-v3 architecture (Chen et al., 2017), that is the best performing among the compared convolutional-based approaches achieving 54.4% of mIoU. The recent Vision Transformers show promising results, often outperforming convolutional-based approaches. The first method specifically addressing

Table 1 Per-part-classes IoU results on the Pascal-Part-58 benchmark

Method	bgr	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	d. table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU	Avg	TrP
SegNet (Badrinarayanan et al. 2017)	85.4	13.7	40.7	11.3	21.7	10.7	36.7	26.3	28.5	16.6	8.9	16.6	24.2	18.8	44.7	35.4	16.1	17.3	15.7	41.3	26.1	24.4	26.5	***14.7***
FCN (Long et al., 2015)	87.0	33.9	51.5	37.7	47.0	45.3	50.8	39.1	45.2	29.4	31.2	32.5	42.4	42.2	58.2	40.3	38.3	43.4	35.7	66.7	44.2	42.3	44.9	***134.2***
DeepLab (Chen et al., 2018)	89.8	40.7	58.1	43.8	53.9	44.5	62.1	45.1	52.3	36.6	41.9	38.7	49.5	53.9	66.1	49.0	45.3	45.3	40.5	76.8	56.5	49.9	51.9	***65.1***
DRN-D 38 (Yu et al., 2017)	89.8	42.8	59.0	43.2	46.5	43.4	65.2	48.5	56.7	31.5	42.8	31.8	50.4	55.9	66.9	54.3	34.1	45.7	38.3	70.4	51.3	50.0	50.9	***26.5***
DRN-D 105 (Yu et al., 2017)	90.4	47.9	59.7	48.8	46.4	48.7	65.5	49.3	59.3	33.0	44.6	31.3	53.6	57.3	66.3	58.3	43.1	44.5	38.5	71.0	54.6	53.0	52.9	***54.8***
DANet (Fu et al., 2019)	89.7	37.4	45.7	37.8	56.7	32.1	44.2	40.9	53.1	30.3	45.6	23.2	50.2	59.0	60.8	53.5	48.1	36.9	40.4	78.8	50.3	47.1	48.3	***68.9***
CCNet (Huang et al., 2020)	89.4	39.0	43.3	39.3	50.1	27.3	46.4	44.7	47.0	31.2	39.6	21.1	43.2	50.0	58.4	53.1	44.3	37.4	17.2	65.4	46.6	44.5	44.5	***68.9***
Swin Tiny (Liu et al., 2021)	90.5	43.5	60.3	47.6	63.1	34.3	64.6	49.4	53.9	36.0	44.6	43.7	51.8	59.6	67.9	50.8	54.3	44.8	44.0	80.2	48.2	52.0	54.0	***59.9***
Swin Small (Liu et al., 2021)	91.6	47.8	64.0	51.4	64.3	48.1	66.9	52.9	63.4	41.9	49.9	47.5	59.9	64.0	69.4	55.7	59.0	47.6	46.3	81.8	62.1	57.1	58.8	***81.2***
Segmenter ViT-B (Strudel et al., 2021)	91.9	40.1	64.3	53.1	66.9	37.8	63.4	50.2	58.9	43.6	49.9	46.6	56.2	65.3	69.6	54.9	62.2	48.4	48.0	82.8	61.1	55.5	57.9	***70.1***
SegFormer B3 (Xie et al., 2021)	91.7	45.0	60.9	53.9	66.3	47.5	65.7	49.2	59.1	36.2	52.7	44.2	57.2	62.8	67.5	52.4	56.1	47.4	38.6	84.2	61.8	55.5	57.2	***44.6***
ViT B-16 (Dosovitskiy et al., 2021)	92.1	47.9	61.2	51.9	67.1	53.8	64.7	52.0	60.1	43.9	51.0	51.7	58.1	63.4	69.8	55.9	62.5	47.3	49.9	85.6	62.2	57.2	59.6	***144.1***
BSANet (Zhao et al., 2019)	91.6	**50.0**	65.7	**54.8**	60.2	49.2	70.1	**53.5**	**63.8**	36.5	52.8	43.7	58.3*	**66.0**	71.6*	**58.4**	55.0	49.6	43.1	82.2	61.4	58.2	58.9*	***74.3***
Baseline (Chen et al., 2017)	91.1	45.7	63.2	49.0	54.4	49.8	67.6	49.2	59.8	35.4	47.6	43.0	54.4	62.0	68.0	55.0	48.9	45.9	43.2	79.6	57.7	54.4	55.7	***70.2***
GMNet (Michieli et al., 2020)	**92.7**	46.7	66.4	52.0	70.0	55.7	71.1	52.2	63.2	51.4	54.8	51.3	59.6	64.4	**73.9**	56.2	56.2	53.6	**56.1**	85.0	65.6	59.0	61.8	***73.8***
GMENet (ours)	92.6	46.5	**66.6**	52.2	**70.7**	**55.8**	**71.6**	52.7	**63.8**	**51.6**	**55.5**	**51.5**	**59.9**	64.8	73.7	57.2	**56.5**	**54.2**	55.8	**85.8**	**66.4**	**59.6**	**62.2**	***74.7***

Best among methods with the same base model of our approach (i.e., DeepLab-v3) in bold, best overall underlined. Avg, average per-object-class mIoU; TrP, number of trainable parameters (M)
*values different from Zhao et al. (2019) since they were wrongly reported in the paper

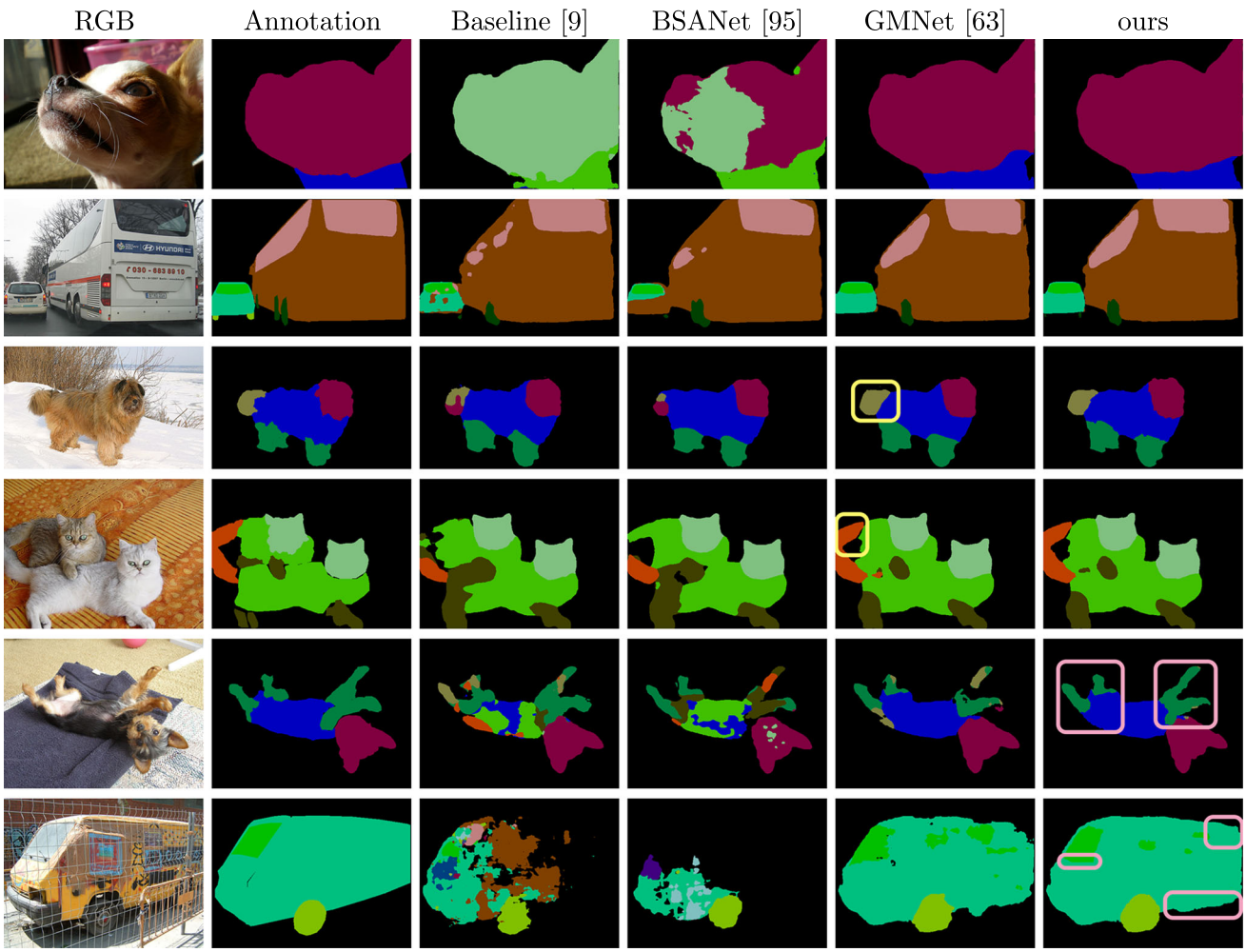

Fig. 7 Segmentation results from the Pascal-Part-58 dataset (*best viewed in colors*). The examples in the first two rows mainly show the effect of the semantic embedding network, the ones in rows 3 and 4 mainly show the effect of the graph matching module (yellow boxes), the last two rows mainly show the improvement due to the edge module (pink boxes)

part-based semantic segmentation is BSANet (Zhao et al., 2019), which achieves a mIoU of 58.2% gaining about 4% compared to the baseline. Our previous work, GMNet (Michieli et al., 2020), combining both object-level semantic embedding and graph matching achieves a higher accuracy of 59.0% of mIoU. Finally, the edge refinement module introduced in this extension of the work (GMENet) further improves the final mIoU score to 59.6%, significantly outperforming every competing method. In particular, GMENet shows the highest per-part-classes IoU result on 11 out of 21 classes. Compared to GMNet, our full method with the edge module improves the performances in terms of per-part-class IoU on 17 out of 21 classes, demonstrating the robustness of the improvement. The gain is experienced on both classes with many parts (like *cow*, *dog* and *sheep*) and with no or few parts (like *boat*, *bottle*, *chair*, *dining table* and *tv*). Therefore, we remark that the 1.1% relative improvement from 59.0% of GMNet to 59.6% of our approach is spread among the different categories, contributing to the overall accuracy improvement. As it is possible to see from the last column of Table 1, the number of trainable parameters of our approach is comparable to recent competing architectures. On the other side, we remark that the total number of parameters of our approach is higher than the baseline model due to the frozen object-level segmentation network, as we discuss in detail at the end of Sect. 7.6.

We report qualitative results in Fig. 7, where we appreciate the effects of the three main contributions of our work: the semantic embedding, the graph matching and the edge-aware modules.

First, the object-level semantic embedding network brings useful additional information prior to the part-level decoding stage, thus enriching the extracted features to be discriminative of the object. We can appreciate this aspect from the first

two rows. In the first row, the baseline completely misleads a dog with a cat (light green is the *cat head* while green is *cat torso*). The *dog head* is partially recovered by BSANet (amaranth corresponds to the correct labeling). Our method, instead, is able to accurately detect and segment the dog parts (*dog head* in amaranth and *dog torso* in blue) thanks to object-level priors coming from the semantic embedding module. The second image confirms these findings: the baseline confuses car parts (green corresponds to *window*, aquamarine to *body* and light green to *wheel*) with bus parts (pink is the *window*, brown the *body* and dark green the *wheel*) and also BSANet made the same errors. Our method, instead, can recognize the correct object-level class and the respective parts (excluding the *car wheels* that are very small and challenging), while also improving the segmentation of the *bus window*.

Furthermore, the graph-matching module helps in the mutual localization of parts within the same object-level class. The effect of the graph-matching module is more evident in the third and fourth row, as highlighted by the yellow boxes in the second-last column. In the third image, we can verify how both the baseline and BSANet are not able to correctly place the *dog tail* (in yellow) misleading it with the *dog head* (in red), recall the example in Fig. 1. Thanks to the graph-matching module, our approach can disambiguate these parts and correctly exploit their spatial relationship with respect to the *dog body*. In the fourth image, both the baseline and BSANet tend to overestimate the presence of the *cat legs* (in dark green) and they miss one *cat tail*. The constraints on the relative position among the various parts enforced by the graph-matching module allow our approach to properly recognize the *cat tail* and to improve the estimate of the *cat legs*.

Finally, the auxiliary edge branch successfully improves boundary localization with respect to GMNet, as we can observe from the last two rows and highlighted by the pink boxes in the last column. In the fifth row all competing approaches are unable to properly detect the edges of all the parts, while GMENet greatly improves contour localization, as can be seen in the *dog legs* and *dog body*. Similarly, in the last row GMENet better shapes the contours of the *car window*, *wheel* and *body*.

7.2 Pascal-Part-108

To further verify the robustness and the scalability of the proposed methodology we perform a second set of experiments using an even larger number of parts. The results on the Pascal-Part-108 benchmark are reported in Table 2. As expected, since the task is more complex with respect to the previous scenario with an almost double number of parts, we can immediately verify a drop in the overall performance for all methods. However, we can appreciate that our framework better scales to larger numbers of parts and is able to largely surpass both the baseline and Zhao et al. (2019). It achieves a mIoU of 45.8%, outperforming the baseline by 5% and the other compared standard segmentation networks by an even larger margin. GMENet shows a remarkable relative gain with respect to the main competitor (Zhao et al., 2019) of 6.8%, while it shows a relative increase of 1.1% with respect to the previous version (GMNet). In this scenario, indeed, most of the previous considerations holds and are even more evident from the results. In particular, GMENet shows the highest per-part-classes IoU result on 7 out of 21 classes. Compared to the previous version (GMNet), our full method ranks first in terms of per-part-class IoU on 15 out of 21 classes, demonstrating a robust performance improvement. Once more, the gain in accuracy is stable across the various classes and parts: the proposed framework significantly wins by large margins on almost every per-object-class mIoU.

Thanks to the object-level semantic embedding network our model is able to accuratey segment all the objects with few or no parts inside, such as *boat, bottle, chair, plant* and *sofa*. On these classes, the gain with respect to BSANet (Zhao et al., 2019) ranges from 6.4% for the *bottle* class to an impressive 14.3% on the *chair* class. On the other hand, thanks to the graph matching module, our framework is also able to correctly understand the spatial relationships between small parts, as for example the ones contained in *cat, cow, horse* and *sheep*. Although many objects are composed by tiny and difficult parts, the gain with respect to Zhao et al. (2019) is still significant and ranges between 1.1% on *horse* parts to 12.4% on *cow* ones.

Figure 8 shows some sample visual results and confirms the numerical evaluation. We can appreciate that the proposed method produces accurate segmentation maps both when a few elements are present or many parts coexist in the scene. In the first row we can verify the effectiveness of the object-level semantic embedding in conditioning part parsing. The baseline is not able to localize and segment the body and the neck of the sheep. On this sample BSANet (Zhao et al., 2019) achieves even worse segmentation and labeling performance. Both methods confuse the sheep with a dog (light blue denotes *dog head*, light purple *dog neck*, brown *dog muzzle* and yellow *dog torso*) or with a cat (purple denotes *cat torso*). Thanks to the object-level priors, our method is able to associate the correct label to each of the parts correctly identifying the sheep as the macro class. In the second row, the effect of the graph matching procedure is evident. The baseline approach tends to overestimate and badly localize the *cow horns* (in brown) and BSANet confuses the *cow horns* with the *cow ears* (in pink). Our approach, instead, achieves better results thanks to the graph matching module which allows for proper localization and contour shaping of the various parts. In the third row, a scenario with two object-level classes having no sub-parts is presented. Again, we can

Table 2 Per-part-classes IoU results on the Pascal-Part-108 benchmark

Method	bgr	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	d. table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU	Avg
SegNet (Badrinarayanan et al., 2017)	85.3	11.2	32.4	6.3	21.4	10.3	27.9	22.6	22.8	17.0	6.3	12.5	21.1	14.9	12.2	32.2	13.8	12.6	15.2	11.3	27.5	18.6	20.8
FCN (Long et al., 2015)	86.8	30.3	35.6	23.6	47.5	44.5	21.3	34.5	35.8	26.6	20.3	24.4	37.7	29.8	14.2	35.6	34.4	28.9	34.0	18.1	45.6	31.6	33.8
DeepLab (Chen et al., 2018)	90.2	38.3	35.4	29.4	57.0	41.5	27.0	40.1	45.5	36.6	33.3	35.2	41.1	48.8	19.5	40.6	46.0	23.7	40.8	17.5	70.0	35.7	40.8
DRN-D 38 (Yu et al., 2017)	90.0	58.4	40.3	35.5	47.4	44.7	33.0	39.9	45.5	30.4	36.9	31.3	46.1	43.5	21.7	48.9	38.3	31.9	35.7	17.9	61.7	39.1	41.9
DRN-D 105 (Yu et al., 2017)	89.7	58.4	40.3	32.5	39.3	33.1	33.8	41.6	46.3	30.5	38.6	30.6	44.5	46.7	24.0	53.8	39.5	30.7	33.3	14.8	58.6	39.5	41.0
DANet (Fu et al., 2019)	89.7	37.7	56.9	38.0	54.1	30.5	36.8	34.2	52.2	32.8	37.2	6.1	47.4	52.9	62.2	52.0	45.5	37.9	35.7	77.8	47.4	44.6	45.9
CCNet (Huang et al., 2020)	88.1	21.6	21.3	22.1	18.2	17.8	17.0	32.3	38.2	22.8	17.3	10.0	39.2	24.1	14.5	48.9	25.3	14.0	2.2	3.4	27.1	26.2	25.0
Swin Tiny (Liu et al., 2021)	91.1	45.1	44.0	39.8	66.1	46.4	33.3	41.5	48.2	38.3	37.1	45.7	46.9	46.3	22.3	51.6	52.9	29.3	41.3	17.9	60.2	41.3	45.0
Swin Small (Liu et al., 2021)	91.6	47.8	40.0	44.3	65.9	54.3	34.1	43.2	50.7	38.9	36.5	46.3	51.6	49.1	25.8	53.1	57.4	29.5	47.7	15.7	70.8	43.1	47.3
Segmenter ViT-B (Strudel et al., 2021)	92.3	45.4	38.7	44.0	66.4	47.5	29.0	42.6	50.9	43.6	41.2	49.3	48.8	51.1	24.4	52.6	60.6	31.6	51.2	15.7	74.6	43.9	47.7
SegFormer B3 (Xie et al., 2021)	92.4	48.3	47.5	42.9	69.4	58.3	37.3	47.3	51.6	41.5	47.0	49.3	52.0	51.4	24.9	54.9	60.9	40.9	47.5	23.6	71.4	45.7	50.5
ViT B-16 (Dosovitskiy et al., 2021)	92.2	47.9	42.3	41.2	69.6	54.4	33.1	43.8	48.4	46.2	41.7	46.7	47.8	49.9	22.8	51.4	62.6	30.3	46.0	20.1	70.6	43.1	48.0
BSANet (Zhao et al., 2019)	91.6	45.3	40.9	41.0	61.4	48.9	32.2	43.3	50.7	34.1	39.4	45.9	52.1	50.0	23.1	52.4	50.6	37.8	44.5	20.7	66.3	42.9	46.3
Baseline (Chen et al., 2017)	90.9	41.9	44.5	35.3	53.7	47.0	34.1	42.3	49.2	35.4	39.8	33.0	48.2	48.8	23.2	50.4	43.6	35.4	39.2	20.7	60.8	41.3	43.7
GMNet (Michieli et al., 2020)	92.7	48.0	46.2	39.3	69.2	**56.0**	37.0	45.3	52.6	**49.1**	50.6	**50.6**	52.0	**51.5**	24.8	52.6	56.0	40.1	**53.9**	21.6	**70.7**	45.8	50.5
GMENet (ours)	**92.9**	**48.9**	**47.3**	**40.2**	**69.6**	55.3	**37.8**	**46.7**	**53.3**	48.4	**51.8**	50.1	**52.3**	51.1	**27.4**	**54.2**	**57.8**	**41.5**	53.4	**24.3**	70.3	**46.3**	**51.2**

Best among methods with the same base model of our approach in bold, best overall underlined
Avg, average per-object-class mIoU

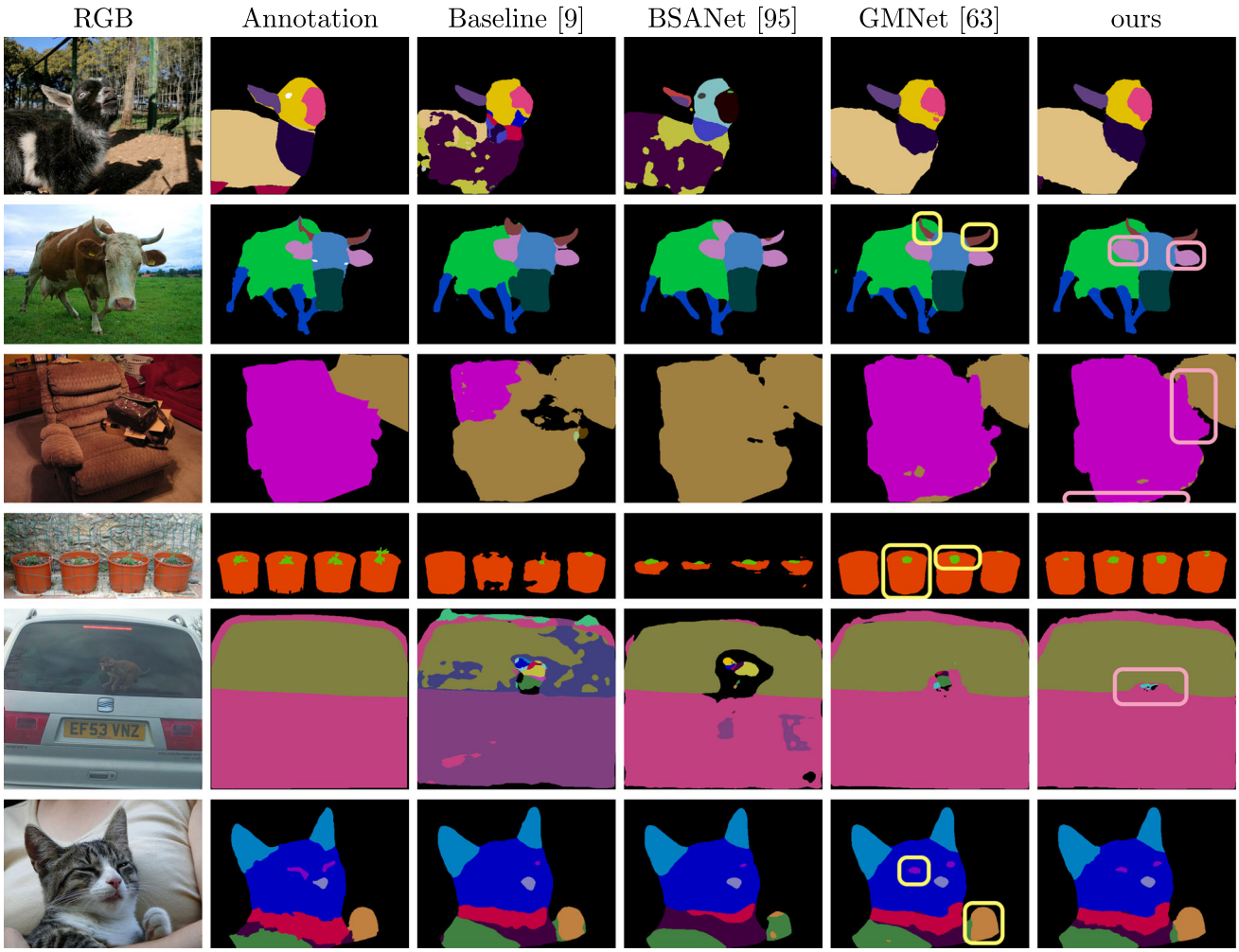

Fig. 8 Segmentation results from the Pascal-Part-108 dataset (*best viewed in colors*). The effect of the graph matching module is highlighted via yellow boxes, whilst the effect of the edge module is highlighted via pink boxes

check how our approach is able to discriminate between *chair* (in pink) and *sofa* (in light brown). Then, we can appreciate how the two parts of the *potted plant* in the fourth row and the parts of the *cat* (such as *eyes* and *paws*) in the last row are correctly segmented by our approach. This is due to the semantic embedding module for what concerns object identification and to the graph matching one for the localization of small parts. Finally, the edge module allows to further improve contour localization as can be seen from the separation between *cow horns* and *ears* in the second row, from the *armchair* edge in the third row and from the separation of the *vehicle body* and *window* in the fifth row.

7.3 ADE20K-Part

We also introduce in this work a large-scale multi-object part parsing dataset built from the ADE20K dataset (Zhou et al., 2017) (see Sect. 6).

Table 3 IoU results on the ADE20K-Part benchmark

	Baseline (Chen et al., 2017)	BSANet (Zhao et al., 2019)	GMNet (Michieli et al., 2020)	GMENet (ours)
mIoU	8.9	9.7	10.6	**12.9**
Avg	17.6	19.6	21.3	**23.6**
mIoUs wo $p : \text{IoU}_p = 0$ in baseline	23.4	25.1	27.5	**31.3**
mIoUs of single parts from classes	23.1	26.1	28.4	**30.4**
mIoUs of non-single parts from classes	6.0	6.6	7.2	**9.5**

"mIoUs wo $p : \text{IoU}_p = 0$ in baseline": mIoU computed over parts that have IoU greater than 0% in the baseline approach. "mIoUs of single parts from classes": mIoU computed only over classes which are not splitted into parts (i.e., only one part from the macro class). "mIoUs of non-single parts from classes": mIoU computed only over parts which are not single parts of the macro class. Avg: average per-object-class mIoU

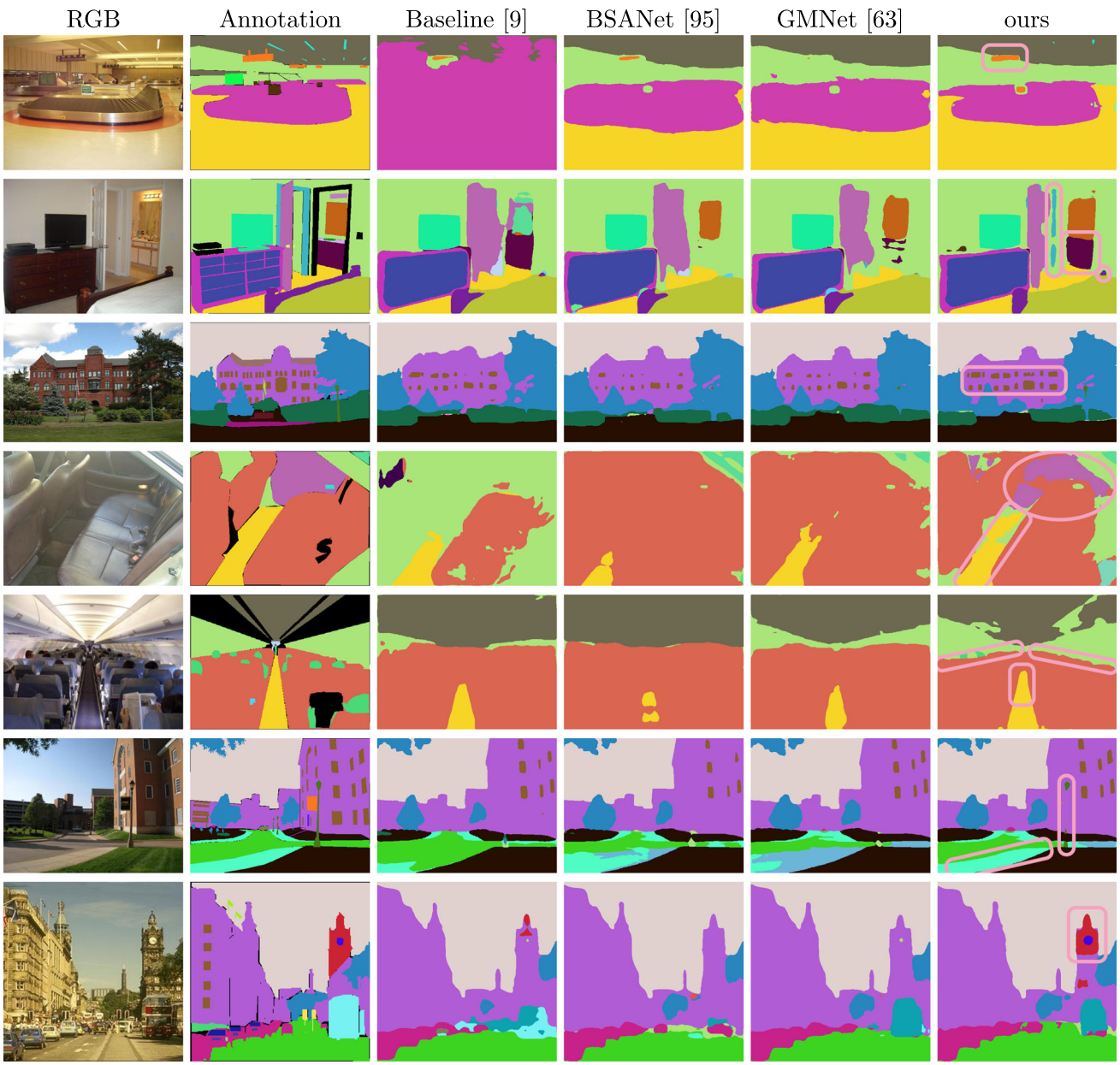

Fig. 9 Segmentation results from the ADE20K-Part dataset (*best viewed in colors*). The effect of the edge module is highlighted via pink boxes

In Table 3 we report the quantitative results for each competing approach. BSANet (Zhao et al., 2019) could improve the final accuracy results obtained with the baseline approach. However, in this dataset, where a much larger number of edges exists with respect to previous benchmarks, both GMNet (Michieli et al., 2020) and the edge-related constraint are able to greatly improve the results compared to BSANet, which achieves 9.7%. Indeed, GMNet raised the mIoU from 8.9 to 10.6%, with respect to the baseline, while our full approach reaches a mIoU of 12.9% (with a relative gain of 44.2% with respect to the baseline). The average per-object-class mIoU is dramatically higher than the mIoU, almost doubling it. The high number of extremely difficult under-represented classes, having IoU close to zero, significantly hinders the mIoU results. Indeed, removing from the computation all such parts having IoU equal to 0% for the baseline approach, the mIoU results significantly raised reaching the values reported in the third row of Table 3, i.e., 31.3% for our approach.

To evaluate the effect on accuracy on actual parts of classes (i.e., multiple parts inherited from the same macro class) and on classes not spilitted into parts (i.e., only one part from the macro class) we computed the average IoU results of all the approaches for the two types of classes, respectively in

fourth and fifth row of Table 3. Our approach could especially improve in presence of multiple parts from each macro class: in fact GMENet raises the mIoU from 6.0 to 9.5% (a relative gain of 53.5% with respect to the baseline) when computing the mIoU of non-single parts from classes; while it raises the mIoU from 23.1 to 30.4% (relative gain of 31.3% with respect to the baseline) when computing the mIoU of single parts from classes.

Some qualitative results are shown in Fig. 9 for both indoor and outdoor scenes. While the resulting mIoUs appear to be quite low, the visual quality on most of the images is quite satisfactory as many parts penalizing the mIoU scores are rare throughout the dataset. To guide the reader, we highlight the improvements due to the edge module via pink boxes. In the first row, the baseline approach is not able to correctly segment the scene; our approach, instead, is able to produce much more reliable segmentation maps of the *baggage carousel* and the *screen* with respect to the competitors. In the second row, edges of the *bed frame* and the *drawer* are more clearly segmented by our method. In the third row, GMENet is the only approach able to reliably segment the *windows* of the building. Furthermore, In the fourth and fifth rows the *seats* and the *aisle* in between are better identified by our approach. In the sixth row, GMENet can properly segment the *road*, the *sidewalk* and the *street light*. Finally, in the last row our method could recognize the *bell tower* with the respective *clock*.

7.4 Single-Object Part Parsing

To further validate our approach, we verify that GMENet could also target the simpler case of single-object part-parsing on the widely-used Pascal-Person-Part benchmark. The results are summarized in Table 4 where we compare with 12 state-of-the-art approaches. Some of them e.g., Nie et al. (2018), Xia et al. (2017) and Fang et al. (2018) make use of auxiliary pose annotation ground truth to improve part-level segmentation accuracy. We can appreciate that we can achieve competitive results improving the performance of the baseline DeepLab-v3 model (Chen et al., 2017), while using only pixel-level supervision. Furthermore, the main competitor BSANet-101 shows a lower accuracy, while employing the same backbone.

7.5 Object-Level Semantic Segmentation

Although the original target application is part parsing, we argue that the proposed strategy could be seamlessly applied to other scenarios where reciprocal relationship among categories are relevant. In this section, we investigate the effect of the proposed modules evaluated on standard object-level semantic segmentation benchmarks. The per-class and mean IoU results on the Pascal VOC2012 (Everingham et al., 2010) and on the Cityscapes (Cordts et al., 2016) datasets are reported in Tables 5 and 6, respectively. Graph matching alone improves the final mIoU by 0.3% on VOC2012 (raising the IoU on 14 out of 21 classes) and by 0.7% on Cityscapes, notice how the improvement is quite stable raising or matching the IoU on 14 out of 19 classes. Finally, edge refinement further leverages previous accuracy scores by 0.5% on the Pascal VOC2012 (with a total improvement of 0.8% with respect to the baseline) and by 1.2% on the Cityscapes (with a total of 1.9% with respect to the baseline).

Table 4 mIoU results on the Pascal-Person-Part benchmark

Method	Pose Annotation	mIoU
HAZN (Xia et al., 2016)	×	56.1
Attention (Chen et al., 2016)	×	56.4
LG-LSTM (Liang et al., 2016)	×	58.0
SS-JPPNet (Liang et al., 2018)	×	59.4
G-LSTM (Liang et al., 2016)	×	60.2
SS-NAN (Zhao et al., 2017b)	×	62.4
S-LSTM (Liang et al., 2017)	×	63.6
Joint (Xia et al., 2017)	✓	64.4
CDCL (Fang et al., 2018)	×	65.0
MuLA (Nie et al., 2018)	✓	65.1
BSANet-101 (Zhao et al., 2019)	×	67.4
WSHP (Fang et al., 2018)	✓	67.6
Baseline (Chen et al., 2017)	×	65.3
GMNet (Michieli et al., 2020)	×	67.5
GMENet	×	**68.4**

In this case, the improvements are generally smaller than in part-level semantic segmentation. An explanation for this is that some of the original assumptions are not as strong as before. In particular, in part-level segmentation the different parts of an object are always connected together in the same way (e.g., *person arm* is always connected to *person torso*), while in object-level segmentation the learned spatial relationships can be less consistent (e.g., a *car* typically runs on the *road* but could be parked on the *sidewalk* or over a grass field that correspond to the *vegetation* class). Therefore, the relationship between the objects are not so strong and fixed as the ones between the parts. Additionally, object-level classes are on average, by definition, larger than parts: therefore, there is a more limited set of edges across different classes to be exploited by the edge module. For such reasons, the gains derived from the graph and edge modules are reduced with respect to the original setting.

We report qualitative results on object-level semantic segmentation on the Pascal VOC2012 in Fig. 10, where we observe that our approaches better recognize the semantics of the objects (*table* in the second row, *sofa* in the third row, *cat* in the seventh row), while at the same time refining the

Table 5 Per-class and mean IoU (mIoU) on object-level semantic segmentation on the Pascal VOC2012 dataset

VOC2012	bgr	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	d. table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU
SegNet (Badrinarayanan et al., 2017)	92.5	74.2	57.0	56.2	40.2	46.3	81.3	72.8	77.0	27.8	53.5	30.6	65.3	63.3	62.3	75.5	27.9	44.8	35.4	61.2	53.4	57.1
FCN (Long et al., 2015)	90.0	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.1
DeepLab (Chen et al., 2018)	92.1	78.4	33.1	78.2	55.6	65.3	81.3	75.5	78.6	25.3	69.2	52.7	75.2	69.0	79.1	77.6	54.7	78.3	45.1	73.3	56.2	66.4
Baseline (Chen et al., 2017)	91.1	76.2	**72.7**	**83.4**	55.6	66.9	86.6	81.2	87.1	35.8	78.5	38.6	82.8	**80.6**	75.1	81.4	57.1	77.5	43.5	79.2	64.3	71.2
$\mathcal{L}_{GM}$	**91.5**	76.5	71.2	82.2	56.3	67.0	88.0	**82.3**	**87.8**	35.7	**79.8**	40.1	**83.8**	78.6	75.4	**81.5**	55.6	**80.2**	43.6	78.8	**64.8**	71.5
$\mathcal{L}_{GM} + \mathcal{L}_{ed}$	91.4	**78.0**	72.4	81.0	**57.6**	**70.1**	**88.6**	81.0	**87.8**	**36.2**	77.8	**42.3**	83.4	80.3	**77.5**	81.0	**59.8**	77.2	**44.5**	**80.0**	62.9	**72.0**

Best among methods with the same base model of our approach in bold, best overall underlined

Table 6 Per-class and mean IoU (mIoU) on object-level semantic segmentation on the Cityscapes dataset

Cityscapes	road	swalk	building	wall	fence	pole	t. light	t. sign	vegetat.	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
SegNet (Badrinarayanan et al., 2017)	96.4	73.2	84.0	28.4	29.0	35.7	39.8	45.1	87.0	63.8	91.8	62.8	42.8	89.3	38.1	43.1	44.1	35.8	51.9	57.0
FCN (Long et al., 2015)	97.0	75.4	87.3	37.4	39.0	35.1	47.7	53.3	89.3	66.1	92.5	69.5	46.0	90.8	41.9	52.9	50.1	46.5	58.4	61.9
DeepLab (Chen et al., 2018)	97.3	77.7	87.7	43.6	40.5	29.7	44.5	55.4	89.4	67.0	92.7	71.2	49.4	91.4	48.7	56.7	49.1	47.9	58.6	63.1
Baseline (Chen et al., 2017)	97.2	80.0	89.4	44.8	46.0	38.0	**41.6**	52.7	**88.3**	62.3	**93.2**	66.2	**47.0**	**90.7**	67.7	60.3	66.4	39.1	58.0	64.7
$\mathcal{L}_{GM}$	**97.3**	**80.8**	**89.5**	47.1	46.7	**38.2**	41.5	53.1	**88.3**	**63.2**	93.1	**66.4**	46.2	90.5	68.9	61.0	70.5	40.7	**58.9**	65.4
$\mathcal{L}_{GM} + \mathcal{L}_{ed}$	97.2	80.2	**89.5**	**48.3**	**47.8**	38.0	40.7	**53.7**	**88.3**	62.2	93.1	66.0	45.6	90.6	**69.0**	**71.6**	**79.2**	**45.5**	58.2	**66.6**

Best among methods with the same base model of our approach in bold, best overall underlined

contour shapes of the objects (*dog leg* in the first row, *bike* in the fourth row, *bottle* in the fifth row, and tail of *bird* in the sixth row). In all these samples, our proposed approach significantly outperforms both GMNet (Michieli et al., 2020) and the baseline (Chen et al., 2017).

Similarly, we show qualitative results on object-level Cityscapes sample scenes in Fig. 11. We observe that both our approaches improve the segmented maps compared to the baseline (*fence* in the second row, *traffic sign* in the last row). On the other side, our approach further improves compared to GMNet achieving both more accurate class semantics (*motorbike* in the third row, *bus* in the fourth row, *truck* in the sixth row) and more precise class contours (*person* and *poles* in the first row, *pole* in the fifth row).

7.6 Ablation Studies

In this section we conduct an accurate investigation of the effectiveness of the various modules of the proposed work on the Pascal-Part-58 dataset.

Impact of Modules In Table 7 we evaluate the individual impact of each module. Let us recall that the baseline architecture (i.e., the DeepLab-v3 network trained directly on the 58 parts with only the standard cross-entropy loss enabled) achieves a mIoU of 54.4%. The reconstruction loss on the object-level segmentation maps helps in preserving the object-level shapes rearranging parts into object-level classes and allows to improve the mIoU to 55.2%. The semantic embedding network $\mathcal{S}$ acts as a powerful class-conditioning module to retain object-level semantics when learning parts and allows to obtain a large performance gain. By combining it with the reconstruction loss we achieve a mIoU of 58.4%. The addition of the graph-matching procedure further boost the final accuracy to 59.0% of mIoU. To better understand the contribution of this module we also tried a simpler unweighted graph model whose entries are just binary values representing whether two parts are adjacent or not (denoted with $\mathcal{L}^u_{GM}$ in the table). This simplified graph leads to a mIoU of 58.7%, lacking some information about the closeness of adjacent parts. Finally, the edge branch brings contour awareness and further leverage the mIoU to 59.6%.

Design of $\mathcal{S}$ Then, we present a more accurate analysis of the impact of the semantic embedding module $\mathcal{S}$ the results are summarized in Table 8. First of all, the exploitation of the multiple concatenation between features computed by $\mathcal{S}$ and features of $\mathcal{D}_p$ at different resolutions allows object-level embedding at different scales and enhances the scale invariance. Concatenating only the output of $\mathcal{S}$ with the output of $\mathcal{E}_p$ (we refer to this approach with "single concatenation"), the final mIoU slightly decreases to 58.7%.

In order to evaluate the usefulness of exploiting features extracted from a CNN, we compared the proposed framework with a variation directly concatenating the output of $\mathcal{E}_p$ with the object-level predicted segmentation maps $\hat{\mathbf{Y}}_o$ after a proper rescaling ("without $\mathcal{S}$"). This approach leads to a quite low mIoU of 55.7%, thus outlining that the embedding network $\mathcal{S}$ is very effective and that simply stacking the object-level and part-level architectures is not sufficient for the considered task. Indeed, simply stacking the same architecture twice ("stacking DeepLab") without considering any additional provision, shows only minimal gain with respect to the baseline (54.8% versus 54.4%). Additionally, we considered also the option of directly feeding object-level features to the part parsing decoder, i.e., we tried to concatenate the output of $\mathcal{E}_o$ with the output of $\mathcal{E}_p$ and feed these features to $\mathcal{D}_p$ ("$\mathcal{E}_o$ conditioning"). Conditioning the part parsing with this approach does not bring in sufficient object-level indication and it leads to a mIoU of 55.7%, which is significantly lower than GMNet (59.0%) and GMENet (59.6%). Finally, to estimate an upper limit of the performance gain coming from the semantic embedding module we fed the object-level semantic embedding network $\mathcal{S}$ with object-level ground truth annotations $\mathbf{Y}_o$ ("with objects GT"), instead of the predictions $\hat{\mathbf{Y}}_o$ (notice that the network $\mathcal{A}_o$ has good performance but it is far from perfect accuracy, as it has 71.5% of mIoU at object-level). In this case, a mIoU of 65.6% is achieved, showing that there is still room for improvement.

Model complexity Finally, we analyze the computational complexity of the proposed method and find ways to relieve it from different perspectives. For benchmarking, we use a workstation with 32GB RAM, an Intel(R) Xeon(R) Gold 5118 CPU @ 2.30/3.20GHz processor, and one NVIDIA Titan RTX GPU card. We evaluate the statistics assuming input size of 513×513 and the frames per seconds (FPS) are computed as the average inference time across all the images of the test set. Table 9 summarizes the performed analyses. Our approach improves the final accuracy via an auxiliary off-the-shelf object-level pre-trained network which is not updated during training of the part-level branch. Therefore, the number of trainable parameters of our approach ($75M$) remains close to the number of trainable parameters of the baseline approach ($70M$): about $4M$ extra parameters are needed for the semantic embedding network $\mathcal{S}$ and about $1M$ for the edge module. However, GFLOPs and FPS are affected by the double inference pass we should carry on both branches, resulting in an roughly double computation time compared to the baseline (from 6.3 to 3.1 FPS). To alleviate the increased computational burden, however, some optimizations can be applied to the inference pipeline. On one side, the inference passes across the two ResNet architectures could be largely parallelized, bringing a 78% relative speedup from 3.1 FPS to 5.5 FPS, with no accuracy drop but with the assumption of a stronger hardware. On the other side, recent works (Jiang et al., 2019; Chang et al., 2019; Maracani et al., 2021) have highlighted that the encoder extracts general features from the input samples, which can be shared across

Fig. 10 Segmentation results on Pascal VOC2012 object-level semantic segmentation benchmark (*best viewed in colors*)

different tasks and datasets, and are later interpreted by the specific decoding architecture. Motivated by these findings, it is possible to share the encoders of the object-level and part-level architectures, i.e., freezing $\mathcal{E}_p$ and setting it equal to $\mathcal{E}_o$. Such approach results in a slight mIoU drop from 59.6 to 58.1% (in our scenario, the visual domain distributions are unchanged, further promoting features shareability), which remains significantly above the baseline mIoU of 54.4. This provision allows to reduce the number of trainable parameters by about a factor 7 and the GFLOPs by about a factor 2, thus increasing the FPS of 68% from 3.1 FPS to 5.2 on the same hardware, thus being a valuable option when computation time is the critical requirement. We remark, however, that the increase of the number of parameters alone is not sufficient to achieve the attained performance gain: as an example, the Deeplab-v3 model with the ResNet152 backbone requires 50% more training steps to reach good performances and achieves only a small gain of 0.4% with respect to the base-

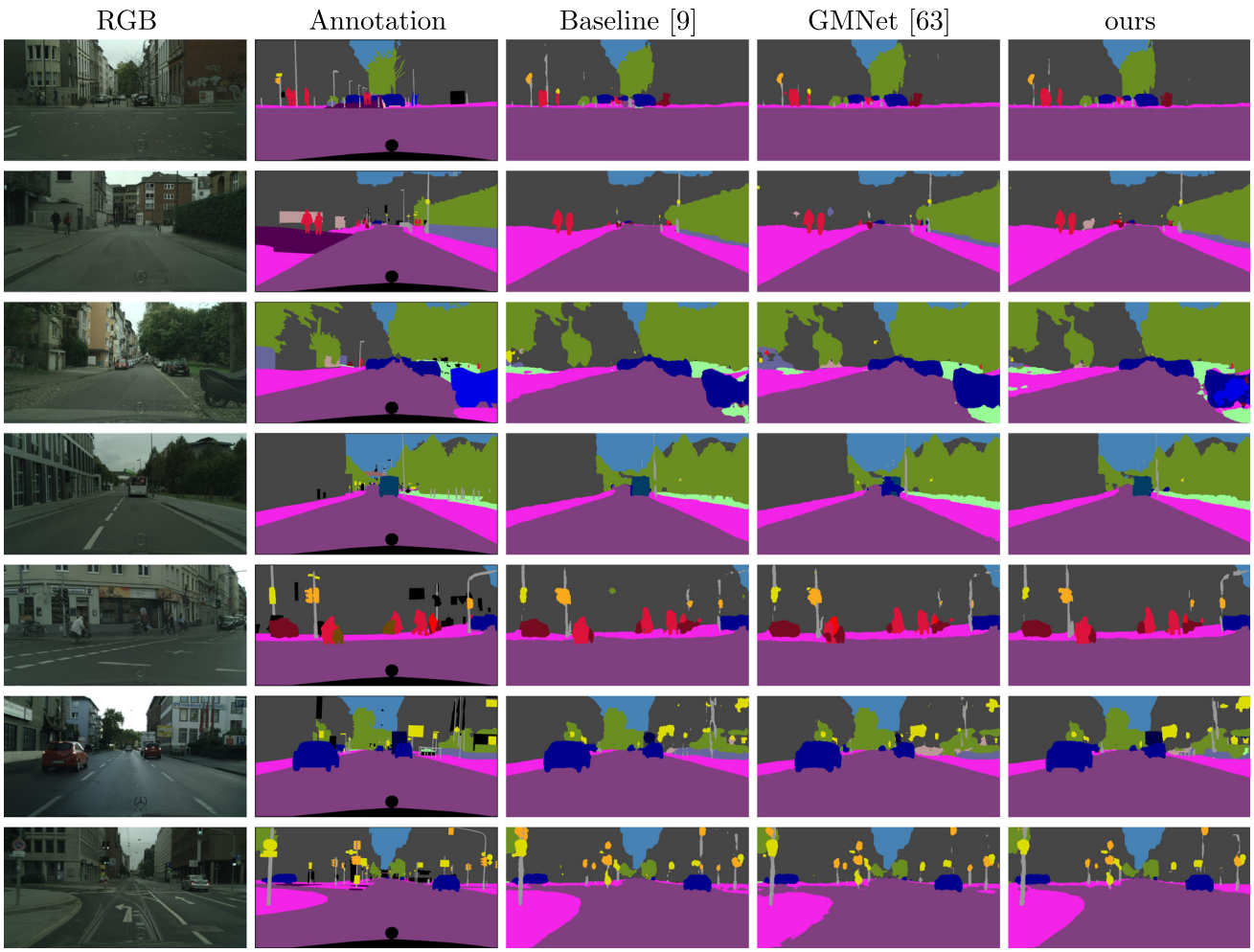

Fig. 11 Segmentation results on Cityscapes object-level semantic segmentation benchmark (*best viewed in colors*)

Table 7 mIoU ablation results on Pascal-Part-58

$\mathcal{L}_{CE}$	$\mathcal{L}_{rec}$	$\mathcal{S}$	$\mathcal{L}^{u}_{GM}$	$\mathcal{L}_{GM}$	$\mathcal{L}_{ed}$	mIoU
✓						54.4
✓	✓					55.2
✓	✓	✓				58.4
✓	✓	✓	✓			58.7
✓	✓	✓		✓		59.0
✓	✓	✓		✓	✓	**59.6**

$\mathcal{L}^{u}_{GM}$: graph matching with unweighted graph

Table 8 mIoU on Pascal-Part-58 with different configurations for the object-level semantic embedding

Method	mIoU
Single concatenation	58.7
Without $\mathcal{S}$	55.7
Stacking DeepLab	54.8
$\mathcal{E}_o$ conditioning	55.7
GMNet (Michieli et al., 2020)	59.0
GMENet (ours)	59.6
With objects GT	**65.6**

line, compared to more than 5% of our approach. Finally, we note that recent research directions in model compression, such as pruning (Liang et al., 2021) or quantization (Gholami et al., 2021), have shown remarkable results in terms of FPS increase and reduced model size, while maintaining the final precision practically unchanged. An extensive analyses on these aspects is left as a future study.

8 Conclusion

In this paper, we tackled the emerging task of multi-class semantic part segmentation. We propose a novel coarse-to-fine strategy where the features extracted from a semantic segmentation network are enriched with object-level semantics when learning part-level segmentation. Furthermore, an

Table 9 Model complexity analysis on Pascal-Part-58

Method	Trainable Params	GFLOPs	FPS	mIoU
Deeplab-v3 ResNet101 (Chen et al., 2017)	70M	132.2	6.3	54.4
Deeplab-v3 ResNet152 (Chen et al., 2017)	***85M***	***161.9***	***5.0***	***51.6***
Deeplab-v3 ResNet152[a] (Chen et al., 2017)	***85M***	***161.9***	***5.0***	***54.8***
GMNet (Michieli et al., 2020)	74M	259.8	3.1	59.0
GMENet (ours)	75M	263.2	3.1	59.6
GMENet parallel (ours)	75M	263.2	5.5	59.6
GMENet $\mathcal{E}$ shared (ours)	11M	181.6	5.3	58.1

[a]Trained for 50% more steps, up to convergence

adjacency graph-based module aims at matching the relative spatial relationships between ground truth and predicted parts which leads to large improvements particularly on small parts. Additionally, an auxiliary edge module improves the localization of class borders.

The proposed approach is able to achieve state-of-the-art results in the challenging task of multi-object part parsing. To prove it, we employ the Pascal-Part-58 benchmark and we propose the more challenging scenarios of Pascal-Part-108 and ADE20K-Part. Finally, we test our approach on classical object-level semantic segmentation benchmarks, showing that our techniques can generalize also to other scenarios.

Further research will aim at improving the graph model accounting for the relations between parts and to the exploitation of the proposed framework in more complex scenarios including domain adaption tasks or multiple levels of hierarchical part splitting. Additionally, the proposed strategies will be applied on top of recent vision transformers for semantic segmentation.

Funding Open access funding provided by Università degli Studi di Padova within the CRUI-CARE Agreement.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., ... Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI)* (pp. 265–283).

Azizpour, H., & Laptev, I. (2012). Object detection using strongly-supervised deformable part models. In *Proceedings of European conference on computer vision (ECCV)* (pp. 836–849). Springer.

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 39*(12), 2481–2495.

Cao, K., Wei, C., Gaidon, A., Arechiga, N., & Ma, T. (2019). Learning imbalanced datasets with label-distribution-aware margin loss. In *Neural information processing systems (NeurIPS)* (pp. 1567–1578).

Cermelli, F., Mancini, M., Bulo, S. R., Ricci, E., & Caputo, B. (2020). Modeling the background for incremental learning in semantic segmentation. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 9233–9242).

Chang, W. L., Wang, H. P., Peng, W. H., & Chiu, W. C. (2019). All about structure: Adapting structural information across domains for boosting semantic segmentation. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1900–1909).

Chen, L. C. (2020). DeepLab official TensorFlow implementation. https://github.com/tensorflow/models/tree/master/research/deeplab. Accessed 2020-03-01.

Chen, L. C., Yang, Y., Wang, J., Xu, W., & Yuille, A. L. (2016) Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3640–3649).

Chen, L. C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587.

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 40*(4), 834–848.

Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., & Yuille, A. (2014). Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1971–1978).

Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., & Shen, C. (2021). Twins: Revisiting the design of spatial attention in vision transformers. *Neural Information Processing Systems (NeurIPS), 34*, 9355–9366.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)*.

Csurka, G., Larlus, D., Perronnin, F., & Meylan, F. (2013). What is a good evaluation measure for semantic segmentation? In *Proceedings of British machine vision conference (BMVC)* (p. 2013).

Das, D., & Lee, C. G. (2018). Unsupervised domain adaptation using regularized hyper-graph matching. In *Proceedings of IEEE international conference on image processing (ICIP)* (pp. 3758–3762). IEEE.

de Geus, D., Meletis, P., Lu, C., Wen, X., & Dubbelman, G. (2021). Part-aware panoptic segmentation. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 5485–5494).

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 248–255). IEEE.

Dhar, P., Singh, R. V., Peng, K. C., Wu, Z., & Chellappa, R. (2019). Learning without memorizing. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 5138–5146).

Dong, J., Chen, Q., Shen, X., Yang, J. & Yan, S. (2014). Towards unified human parsing and pose estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 843–850).

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International conference on learning representations (ICLR)*.

Douillard, A., Chen, Y., Dapogny, A., & Cord, M. (2021). Plop: Learning without forgetting for continual semantic segmentation. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 4040–4050).

Emmert-Streib, F., Dehmer, M., & Shi, Y. (2016). Fifty years of graph matching, network alignment and network comparison. *Information Sciences, 346*, 180–197.

Eslami, S,. & Williams, C. (2012). A generative model for parts-based object segmentation. In *Neural information processing systems (NeurIPS)* (pp. 100–107).

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision (IJCV), 88*(2), 303–338.

Fang, H. S., Lu, G., Fang, X., Xie, J., Tai, Y. W., & Lu, C. (2018). Weakly and semi supervised human body part parsing via pose-guided knowledge transfer. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)*.

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., & Lu, H. (2019). Dual attention network for scene segmentation. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3146–3154).

Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M. W., & Keutzer, K. (2021). A survey of quantization methods for efficient neural network inference. arXiv preprint arXiv:2103.13630

Gonzalez-Garcia, A., Modolo, D., & Ferrari, V. (2018). Do semantic parts emerge in convolutional neural networks? *International Journal of Computer Vision (IJCV), 126*(5), 476–494.

Guo, Y., Liu, Y., Georgiou, T., & Lew, M. S. (2018). A review of semantic segmentation using deep neural networks. *International Journal of Multimedia Information Retrieval, 7*(2), 87–93.

Haggag, H., Abobakr, A., Hossny, M., & Nahavandi, S. (2016). Semantic body parts segmentation for quadrupedal animals. In *2016 IEEE international conference on systems, man, and cybernetics (SMC)* (pp. 000855–000860).

Han, H. Y., Chen, Y. C., Hsiao, P. Y., & Fu, L. C. (2020). Using channel-wise attention for deep CNN based real-time semantic segmentation with class-aware edge information. *IEEE Transactions on Intelligent Transportation Systems, 22*(2), 1041–1051.

Hariharan, B., Arbeláez, P., Girshick, R., & Malik, J. (2015). Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 447–456).

He, H., Zhang, J., Zhuang, B., Cai, J., & Tao, D. (2021a). End-to-end one-shot human parsing. arXiv preprint arXiv:2105.01241.

He, J., Yang, S., Yang, S., Kortylewski, A., Yuan, X., Chen, J. N., Liu, S., Yang, C. & Yuille, A. (2021b). Partimagenet: A large, high-quality dataset of parts. arXiv preprint arXiv:2112.00933.

He, K., Zhang, X., Ren, S., & Sun, J.(2016). Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778).

Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., & Liu, W. (2019). Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of international conference on computer vision (ICCV)* (pp. 603–612).

Huang, Z., Wang, X., Wei, Y., et al. (2020). Ccnet: Criss-cross attention for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

Jiang, H., Sun, D., Jampani, V., Lv, Z., Learned-Miller, E., & Kautz, J. (2019). SENSE: A shared encoder network for scene-flow estimation. In *Proceedings of international conference on computer vision (ICCV)* (pp. 3195–3204).

Jin, Y., Han, D., & Ko, H. (2021). Trseg: Transformer for semantic segmentation. *Pattern Recognition Letters, 148*, 29–35.

Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., & Kalantidis, Y. (2019). Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations (ICLR)*.

Krause, J., Jin, H., Yang, J., & Fei-Fei, L. (2015). Fine-grained recognition without part annotations. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 5546–5555).

Li, J., Zhao, J., Wei, Y., Lang, C., Li, Y., Sim, T., Yan, S. & Feng, J. (2017). Multiple-human parsing in the wild. arXiv preprint arXiv:1705.07206.

Li, P., Xu, Y., Wei, Y., & Yang, Y. (2020a). Self-correction for human parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

Li, X., Li, X., Zhang, L., Cheng, G., Shi, J., Lin, Z., Tan, S. & Tong, Y. (2020b). Improving semantic segmentation via decoupled body and edge supervision. In *Proceedings of European conference on computer vision (ECCV)* (pp. 435–452). Springer.

Li, Z., & Hoiem, D. (2018). Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 40*(12), 2935–2947.

Liang, T., Glossner, J., Wang, L., Shi, S., & Zhang, X. (2021). Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing, 461*, 370–403.

Liang, X., Liu, S., Shen, X., Yang, J., Liu, L., Dong, J., Lin, L., & Yan, S. (2015). Deep human parsing with active template regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 37*(12), 2402–2414.

Liang, X., Shen, X., Feng, J., Lin, L., & Yan, S. (2016). Semantic object parsing with graph lstm. In *Proceedings of European Conference on Computer Vision (ECCV)* (pp. 125–143). Springer.

Liang, X., Lin, L., Shen, X., Feng, J., Yan, S., & Xing, E. P. (2017). Interpretable structure-evolving lstm. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1010–1019).

Liang, X., Gong, K., Shen, X., & Lin, L. (2018). Look into person: Joint body parsing & pose estimation network and a new benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 41*(4), 871–885.

Liu, X., Deng, Z., & Yang, Y. (2019a). Recent progress in semantic image segmentation. *Artificial Intelligence Review, 52*(2), 1089–1106.

Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., & Yu, S. X. (2019b). Large-scale long-tailed recognition in an open world. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2537–2546).

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of international conference on computer vision (ICCV)* (pp. 10012–10022).

Livi, L., & Rizzi, A. (2013). The graph matching problem. *Pattern Analysis and Applications, 16*(3), 253–283.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3431–3440).

Lu, W., Lian, X., & Yuille, A. (2014). Parsing semantic parts of cars using graphical models and segment appearance consistency. In *Proceedings of British Machine Vision Conference (BMVC)*.

Maracani, A., Michieli, U., Toldo, M., & Zanuttigh, P. (2021). Recall: Replay-based continual learning in semantic segmentation. In *Proceedings of International Conference on Computer Vision (ICCV)* (pp. 7026–7035).

Mel, M., Michieli, U., & Zanuttigh, P. (2020). Incremental and multi-task learning strategies for coarse-to-fine semantic segmentation. *Technologies, 8*(1), 1.

Michieli, U., & Ozay, M. (2021). Prototype guided federated learning of visual feature representations. arXiv preprint arXiv:2105.08982.

Michieli, U., & Zanuttigh, P. (2019). Incremental learning techniques for semantic segmentation. In *Proceedings of IEEE conference on computer vision and pattern recognition workshops (CVPRW)*.

Michieli, U., & Zanuttigh, P. (2021a). Knowledge distillation for incremental learning in semantic segmentation. *Computer Vision and Image Understanding, 205*, 103167.

Michieli, U., & Zanuttigh, P. (2021b). Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1114–1124).

Michieli, U., Borsato, E., Rossi, L., & Zanuttigh, P. (2020). Gmnet: Graph matching network for large scale part semantic segmentation in the wild. In *Proceedings of European conference on computer vision (ECCV)* (pp. 397–414). Springer.

Nie, X., Feng, J., & Yan, S. (2018). Mutual learning to adapt for joint human parsing and pose estimation. In *Proceedings of European conference on computer vision (ECCV)* (pp. 502–517).

Rebuffi, S. A., Kolesnikov, A., Sperl, G., & Lampert, C. H. (2017). icarl: Incremental classifier and representation learning. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2001–2010).

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention* (pp. 234–241). Springer.

Ruan, T., Liu, T., Huang, Z., Wei, Y., Wei, S., & Zhao, Y. (2019). Devil in the details: Towards accurate single and multiple human parsing. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)* (pp. 4814–4821).

Shmelkov, K., Schmid, C., & Alahari, K. (2017). Incremental learning of object detectors without catastrophic forgetting. In *Proceedings of international conference on computer vision (ICCV)* (pp. 3400–3409).

Song, Y., Chen, X., Li, J., & Zhao, Q. (2017). Embedding 3d geometric features for rigid object part segmentation. In *Proceedings of international conference on computer vision (ICCV)* (pp. 580–588).

Strudel, R., Garcia, R., Laptev, I., & Schmid, C. (2021). Segmenter: Transformer for semantic segmentation. In *Proceedings of international conference on computer vision (ICCV)* (pp. 7262–7272).

Sun, J., & Ponce, J. (2013). Learning discriminative part detectors for image classification and cosegmentation. In *Proceedings of international conference on computer vision (ICCV)* (pp. 3400–3407).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L.,& Polosukhin, I. (2017). Attention is all you need. *Neural Information Processing Systems (NeurIPS)* 30

Vu, T. H., Jain, H., Bucher, M., Cord, M., & Pérez, P. (2019). Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2517–2526).

Wan, W., Chen, J., Li, T., Huang, Y., Tian, J., Yu, C., & Xue, Y. (2019). Information entropy based feature pooling for convolutional neural networks. In *Proceedings of international conference on computer vision (ICCV)* (pp. 3405–3414).

Wang, J., & Yuille, A. L. (2015). Semantic part segmentation using compositional model combining shape and appearance. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1788–1797).

Wang, P., Shen, X., Lin, Z., Cohen, S., Price, B. & Yuille, A. L. (2015). Joint object and part segmentation using deep learned potentials. In *Proceedings of international conference on computer vision (ICCV)* (pp. 1573–1581).

Wang, Y., Tran, D., Liao, Z., & Forsyth, D. (2012). Discriminative hierarchical part-based models for human parsing and action recognition. *Journal of Machine Learning Research, 13*, 3075–3102.

Xia, F., Zhu, J., Wang, P., & Yuille, A.(2015). Pose-guided human parsing with deep learned features. arXiv preprint arXiv:1508.03881.

Xia, F., Wang, P., Chen, L. C., & Yuille, A. L. (2016). Zoom better to see clearer: Human and object parsing with hierarchical auto-zoom net. In *Proceedings of European conference on computer vision (ECCV)* (pp. 648–663). Springer.

Xia, F., Wang, P., Chen, X., & Yuille, A. L. (2017). Joint multi-person pose estimation and semantic part segmentation. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 6769–6778).

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (2021). Segformer: Simple and efficient design for semantic segmentation with transformers. In *Neural information processing systems (NeurIPS)*.

Yamaguchi, K., Kiapour, M. H., Ortiz, L. E., & Berg, T. L. (2012). Parsing clothing in fashion photographs. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3570–3577).

Yang, Y., & Ramanan, D. (2011). Articulated pose estimation with flexible mixtures-of-parts. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1385–1392).

Yin, J., Liu, W., Xing, W., & Xiao, Y. (2021). Class-level aware network for human parsing. In *International conference on computing, networks and internet of things* (pp. 1–6).

Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., & Sang, N. (2018). Bisenet: Bilateral segmentation network for real-time semantic segmenta-

tion. In *Proceedings of European conference on computer vision (ECCV)* (pp. 325–341).

Yu, C., Gao, C., Wang, J., Yu, G., Shen, C., & Sang, N. (2021). Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision (IJCV), 129*(11), 3051–3068.

Yu, F., Koltun, V., & Funkhouser, T. (2017). Dilated residual networks. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)*.

Yuan, Y., Huang, L., Guo, J., Zhang, C., Chen, X., & Wang, J. (2018). Ocnet: Object context network for scene parsing. arXiv preprint arXiv:1809.00916.

Zhang, N., Donahue, J., Girshick, R., & Darrell, T. (2014). Part-based r-cnns for fine-grained category detection. In *Proceedings of European Conference on Computer Vision (ECCV)* (pp. 834–849). Springer.

Zhang, W., Huang, Z., Luo, G., Chen, T., Wang, X., Liu, W., Yu, G., & Shen, C. (2022) Topformer: Token pyramid transformer for mobile semantic segmentation. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)*.

Zhang, Z., & Pang, Y. (2020). Cgnet: Cross-guidance network for semantic segmentation. *Science China Information Sciences, 63*(2), 1–16.

Zhang, Z., Fu, H., Dai, H., Shen, J., Pang, Y., & Shao, L. (2019). Et-net: A generic edge-attention guidance network for medical image segmentation. In *International conference on medical image computing and computer-assisted intervention* (pp. 442–450). Springer.

Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017a). Pyramid scene parsing network. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2881–2890).

Zhao, J., Li, J., Nie, X., Zhao, F., Chen, Y., Wang, Z., Feng, J. & Yan, S. (2017b). Self-supervised neural aggregation networks for human parsing. In *Proceedings of IEEE conference on computer vision and pattern recognition workshops (CVPRW)* (pp. 7–15).

Zhao, Y., Li, J., Zhang, Y., & Tian, Y. (2019). Multi-class part parsing with joint boundary-semantic awareness. In *Proceedings of international conference on computer vision (ICCV)* (pp. 9177–9186).

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P. H., & Zhang, L. (2021) Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 6881–6890).

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., & Torralba, A. (2017). Scene parsing through ade20k dataset. In *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 633–641).

Zhu, L. L., Chen, Y., Lin, C., & Yuille, A. (2011). Max margin learning of hierarchical configural deformable templates (hcdts) for efficient object parsing and pose estimation. *International Journal of Computer Vision (IJCV), 93*(1), 1–21.