GMNet: Graph Matching Network for Large Scale Part Semantic Segmentation in the Wild

Umberto Michieli[0000−0003−2666−4342], Edoardo Borsato, Luca Rossi, and Pietro Zanuttigh[0000−0002−9502−2389]

Department of Information Engineering, University of Padova, Padova, Italy
{michieli, borsatoedo, rossiluc, zanuttigh}@dei.unipd.it

Abstract. The semantic segmentation of parts of objects in the wild is a challenging task in which multiple instances of objects and multiple parts within those objects must be detected in the scene. This problem remains nowadays very marginally explored, despite its fundamental importance towards detailed object understanding. In this work, we propose a novel framework combining higher object-level context conditioning and part-level spatial relationships to address the task. To tackle object-level ambiguity, a class-conditioning module is introduced to retain class-level semantics when learning parts-level semantics. In this way, mid-level features carry also this information prior to the decoding stage. To tackle part-level ambiguity and localization we propose a novel adjacency graph-based module that aims at matching the relative spatial relationships between ground truth and predicted parts. The experimental evaluation on the Pascal-Part dataset shows that we achieve state-of-the-art results on this task.

Keywords: Part Parsing, Semantic Segmentation, Graph Matching, Deep Learning

1 Introduction

Semantic segmentation is a wide research field and a huge number of approaches have been proposed for this task [5,49,30]. The segmentation and labeling of parts of objects can be regarded as a special case of semantic segmentation that focuses on parts decomposition. The information about parts provides a richer representation for many fine-grained tasks, such as pose estimation [12,47], category detection [8,2,48], fine-grained action detection [42] and image classification [39,22]. However, current approaches for semantic segmentation are not optimized to distinguish between different semantic parts since corresponding parts in different semantic classes often share similar appearance. Additionally, they only capture limited local context while the precise localization of semantic part layouts and their interactions requires a wider perspective of the image. Thus, it is not sufficient to take standard semantic segmentation methods and treat each part as an independent class. In the literature, object-level semantic segmentation has been extensively studied. Part parsing, instead, has only been marginally explored in the context of a few specific single-class objects, such as humans [26,46,52,14], cars [38,31] and animals [40,41,19]. Multi-class part-based semantic segmentation has only been considered in a recent work [51], due to the challenging scenario of part-level as well as object-level ambiguities. Here, we introduce an approach dealing with the semantic segmentation of an even larger set of parts and we demonstrate that the proposed methodology is able to deal with a large amount of parts contained in the scenes.

Nowadays, one of the most active research directions is the transfer of previous knowledge, acquired on a different but related task, to a new situation. Different interpretations may exist to this regard. In the class-incremental task, the learned model is updated to perform a new task whilst preserving previous capabilities: many methods have been proposed for image classification [11,36,23], object detection [37] and semantic segmentation [33,34]. Another aspect regards the coarse-to-fine refinement at the semantic level, in which previous knowledge acquired on a coarser task is exploited to perform a finer task [20,44,32]. In this paper, instead, we investigate the coarse-to-fine refinement at the spatial level, in which object-level classes are split into their respective parts [41,43,51].

More precisely, we investigate the multi-object and multi-part parsing in the wild, which simultaneously handles all semantic objects and parts within each object in the scene. Even strong recent baselines for semantic segmentation, such as FCN [30], SegNet [3], PSPNet [49] or Deeplab [5,6], face additional challenges when dealing with this task, as shown in [51]. In particular, the simultaneous appearance of multiple objects and the inter-class ambiguity may cause inaccurate boundary 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 very similar, such as cow legs and sheep legs. Current algorithms heavily suffer from these aspects. To address object-level ambiguity, we propose an object-level conditioning to serve as guidance for part parsing within the object. An auxiliary reconstruction module from parts to objects further penalize predictions of parts in regions occupied by an object which does not contain the predicted parts within it. At the same time, to tackle part-level ambiguity, we introduce a graph-matching module to preserve the relative spatial relationships between ground truth and predicted parts.

When people look at scenes, they tend to locate first the objects and then to refine them via semantic part parsing [43]. This is the same rationale for our class-conditioning approach, which consists of an approach to refine parts localization exploiting previous knowledge. In particular, the object-level predictions of the model serve as a conditioning term for the decoding stage on the part-level. The predictions are processed via an object-level semantic embedding Convolutional Neural Network (CNN) and its features are concatenated with the ones produced by the encoder of the part-level segmentation network. The extracted features are enriched with this type of information prior, guiding the output of the part-level decoding stage. We further propose to address part-level ambiguity via a novel graph-matching technique applied to the segmentation maps. A couple of adjacency graphs are built from neighboring parts both from the ground-truth and from the predicted segmentation maps. Such graphs are weighted with the normalized number of adjacent pixels to represent the strength of connection between the parts. Then, a novel loss function is designed to enforce their similarity. These provisions allow the architecture to discover the differences in appearance between different parts within a single object, and at the same time to avoid the ambiguity across similar object categories.

The main contributions of this paper can be summarized as follows:

- We tackle the challenging multi-class part parsing via an object-level semantic embedding network conditioning the part-level decoding stage.
- We introduce a novel graph-matching module to guide the learning process toward accurate relative localization of semantic parts.
- Our approach (GMNet) achieves new state-of-the-art performance on multi-object part parsing on the Pascal-Part dataset [8]. Moreover, it scales well to large sets of parts.

2 Related Work

Semantic Segmentation is one of the key tasks for automatic scene understanding. Current techniques are based on the Fully Convolutional Network (FCN) framework [30], which firstly enabled accurate and end-to-end semantic segmentation. Recent works based on FCN, such as SegNet [3], PSPNet [49] and Deeplab [6,5], are typically regarded as the state-of-the-art architectures for semantic segmentation. Some recent reviews on the topic are [28,18].

Single-Object Part Parsing has been actively investigated in the recent literature. However, most previous work assumes images containing only the considered object, well-localized beforehand and with no occlusions. Single-object parts parsing has been applied to animals [40], cars [14,31,38] and humans parsing [26,46,52,14]. Traditional deep neural network architectures may also be applied to part parsing regarding each semantic part as a separate class label. However, such strategies suffer from the high similar appearance between parts and from large scale variations of objects and parts. Some coarse-to-fine strategies have been proposed to tackle this issue. Hariharan et al. [20] 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 [43], where the same structure is employed for the three networks and an automatic adaptation to the size of the object is introduced. In [41] 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 a fully connected CRF is used as post-processing technique to explore the relationship between parts and body to perform the final prediction. In [7] an attention mechanism that learns to softly weight the multi-scale features at each pixel location is proposed.

Some approaches resort to structure-based methodologies, e.g. compositional, to model part relations [40,41,24,25,27,16]. Wang et al. [40] propose a model to learn a mixture of compositional models under various poses and viewpoints for certain animal classes. In [24] a self-supervised structure-sensitive learning approach is proposed to constrain human pose structures into parsing results. In [27,25] graph LSTMs are employed to refine the parsing results of initial over-segmented superpixel maps. Pose estimation is also useful for part parsing task [44,35,16,24,50]. In [44], the authors refine the segmentation maps by supervised pose estimation. In [35] a mutual learning model is built for pose estimation and part segmentation. In [16], the authors exploit anatomical similarity among humans to transfer the parsing results of a person to another person with similar pose. In [50] 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 approach to hierarchically partition the parts [31,45]. Lu et al. [31] propose a method based on tree-structured graphical models from different viewpoints combined with segment appearance consistency for part parsing. Xia et al. [45] firstly generate part segment proposals and then infer the best ensemble of parts-segment through and-or graphs.

Even though single-object part parsing has been extensively studied so far, **Multi-Object and Multi-Part Parsing** has been considered only recently [51]. In this setup, most previous techniques fail struggling with objects that were not previously well-localized, isolated and with no occlusions. Zhao et al. in [51] 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.

3 Method

When we look at images, we often firstly locate the objects and then the more detailed task of semantic part parsing is addressed using mainly two priors: (1) object-level information and (2) relative spatial relationships among parts. Following this rationale, the semantic parts parsing is supported by the information coming from an initial prediction of the coarse object-level set of classes and by a graph-matching strategy working at the parts-level.

An overview of our framework is shown in Figure 1. We employ two semantic segmentation networks $\mathcal{A}_o$ and $\mathcal{A}_p$ trained for the objects-level and parts-level task respectively, together with a semantic embedding network $\mathcal{S}$ transferring and processing the information of the first network to the second to address the object-level prior. This novel coarse-to-fine strategy to gain insights into parts detection will be the subject of this section. Furthermore, we account for the second prior exploiting an adjacency graph structure to mimic the spatial

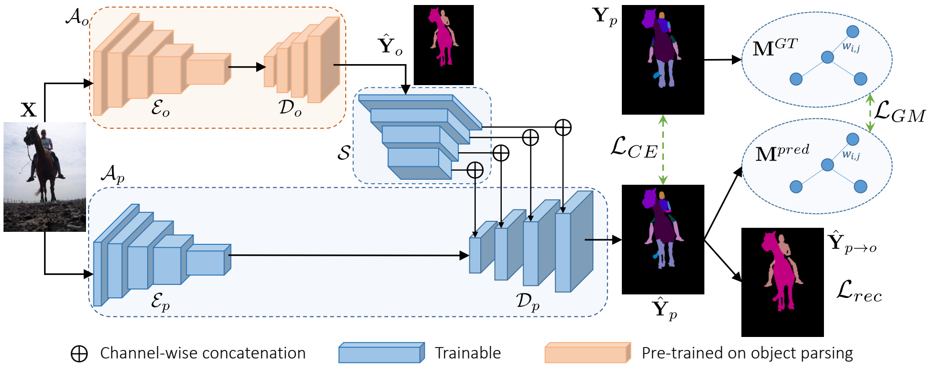

Fig. 1. Architecture of the proposed Graph Matching Network (GMNet) approach. A semantic embedding network takes as input the object-level segmentation map and acts as high level conditioning when learning the semantic segmentation of parts. On the right, a reconstruction loss function rearranges parts into objects and the graph matching module aligns the relative spatial relationships between ground truth and predicted parts.

relationship between neighboring parts to allow for a general overview of the semantic parts as described in Section 4.

The semantic segmentation networks have an autoencoder structure and can be written as the composition of an encoder and a decoder as $\mathcal{A}_o = \{\mathcal{E}_o, \mathcal{D}_o\}$ and $\mathcal{A}_p = \{\mathcal{E}_p, \mathcal{D}_p\}$ for the object-level and part-level networks, respectively. We employ the Deeplab-v3 [6] segmentation network with Resnet-101 [21] as encoder. The network $\mathcal{A}_o$ is trained using the object-level ground truth labels and then kept fixed. It extracts object-level segmentation maps which serve as a guidance for the decoder of the parts-level network $\mathcal{D}_p$, in order to avoid the ambiguity across similar object categories. We achieve this behavior by feeding the output maps of $\mathcal{A}_o$ to an object-level semantic embedding network. In this work, we used a CNN (denoted with $\mathcal{S}$) formed by a cascade of 4 convolutional layers with stride of 2, square kernel sizes of 7, 5, 3, 3, and channel sizes of 128, 256, 512, 1024.

The parts-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 on the RGB image and the ones computed on the object-level predicted map 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 $\mathcal{S}$. In this way, the combination is performed at multiple resolutions in the feature space to achieve higher scale invariance as shown in Figure 1.

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

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

where $\mathcal{D}_{p,i}$ is the i-th decoding layer of the part segmentation network, $\mathcal{S}_i$ denotes the i-th layer of $\mathcal{S}$, k is the number of layers and matches the number of upsampling stages of the decoder (e.g., $k = 4$ in the Deeplab-v3), $\mathcal{F}_i$ is the input of $\mathcal{D}_{p,i+1}$. Since the object-level segmentation is not perfect, in principle, errors from the predicted class in the object segmentation may propagate to the parts. To account for this, similarly to [43], here we do not make premature decisions but the channel-wise concatenation still leaves the final decision of the labeling task to the decoder.

The training of the proposed framework (i.e., of $\mathcal{A}_p$ and $\mathcal{S}$, while $\mathcal{A}_o$ is kept fixed after the initial training) is driven by multiple loss components. The first is a standard cross-entropy loss $\mathcal{L}_{CE}$ to learn the semantic segmentation of parts:

$$\mathcal{L}_{CE} = \sum_{c_p=1}^{N_p} \mathbf{Y}_p[c_p] \cdot \log\left(\hat{\mathbf{Y}}_p[c_p]\right) \tag{2}$$

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-class 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 done applying a cross-entropy loss between object-level one-hot encoded ground truth maps $\mathbf{Y}_o$ and the summed probability $\hat{\mathbf{Y}}_{p\to o}$ derived from the part-level prediction. More formally, defining l as the parts-to-objects mapping such that object j contains parts from index $l[j-1]+1$ to $l[j]$, we can write the summed probability as:

$$\hat{\mathbf{Y}}_{p\to o}[j] = \sum_{i=l[j-1]+1,\ldots,l[j]} \hat{\mathbf{Y}}_p[i] \qquad j = 1, \ldots, N_o \tag{3}$$

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

$$\mathcal{L}_{rec} = \sum_{c_o=1}^{N_o} \mathbf{Y}_o[c_o] \cdot \log\left(\hat{\mathbf{Y}}_{p\to o}[c_o]\right) \tag{4}$$

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 the losses. Instead, parts predicted within the object class are penalized only 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

Providing global context information and disentangling relationships is useful to distinguish fine-grained parts. For instance, upper and lower arms share highly

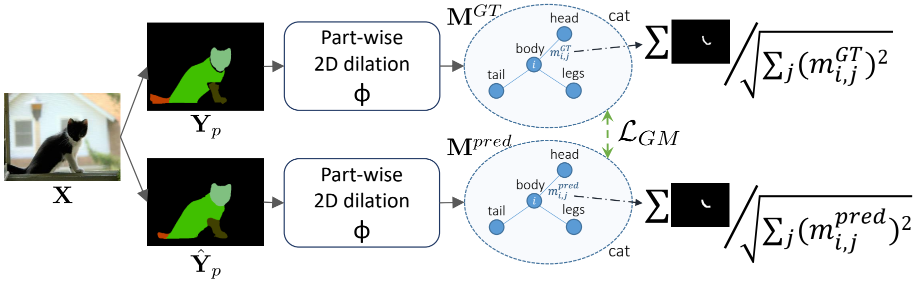

Fig. 2. Overview of the graph matching module. In this case, cat's head and body would be considered as detached without the proper morphological dilation over the parts.

similar appearance. To differentiate between them, global and reciprocal information, like the relationship with neighboring parts, provides an effective context prior. Hence, to further enhance the accuracy of part parsing, we tackle part-level ambiguity and localization by proposing a novel module based on an adjacency graph that matches the parts spatial relationships between ground truth and predicted parts. More in detail, the graphs capture the adjacency relationships between each couple of parts and then we enforce the matching between the ground truth and predicted graph through an additional loss term. Although graph matching is a very well studied problem [13,29], it has never been applied to this context before, i.e. as a loss function to drive deep learning architectures for semantic segmentation. The only other attempt to design a graph matching loss is [9], which however deals with a completely different task, i.e., domain adaptation in classification, and has a different interpretation of the graph, that measures the matching between the source and target domains.

An overview of this module is presented in Figure 2. Formally, we represent the graphs using two (square) weighted adjacency matrices of size N_p:

$$\tilde{\mathbf{M}}^{GT} = \left\{\tilde{m}^{GT}_{i,j}\right\}_{\substack{i=1,...,N_p \\ j=1,...,N_p}} \qquad\qquad \tilde{\mathbf{M}}^{pred} = \left\{\tilde{m}^{pred}_{i,j}\right\}_{\substack{i=1,...,N_p \\ j=1,...,N_p}} \tag{5}$$

The first matrix ($\tilde{\mathbf{M}}^{GT}$) contains the adjacency information computed on ground truth data, while the second ($\tilde{\mathbf{M}}^{pred}$) has the same information computed on the predicted segmentation maps. Each element of the matrices provide a measure of how close the two parts p_i and p_j are in the ground truth and in the predicted segmentation maps, respectively. We do not consider self-connections, hence $\tilde{m}^{GT}_{i,i} = \tilde{m}^{pred}_{i,i} = 0$ for $i = 1, ..., N_p$. To measure the closeness between couples of parts, that is a hint of the strength of connection between them, we consider weighted matrices where each entry $\tilde{m}_{i,j}$ depends on the length of the contour in common between them. Actually, to cope for some inaccuracies inside the dataset where some adjacent parts are separated by thin background regions, the entries of the matrices are the counts of pixels belonging to one part with a distance less or equal than T 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 T. We empirically set $T = 4$ 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 $\lceil T/2 \rceil$ and computing the intersecting region. Formally, defining with $p_i^{GT} = \mathbf{Y}_p[i]$ the mask of the i-th part in the ground truth map $\mathbf{Y}_p$, we have:

$$\tilde{m}^{GT}_{i,j} = \left|\left\{ s \in \Phi\left(p_i^{GT}\right) \cap \Phi\left(p_j^{GT}\right) \right\}\right| \tag{6}$$

Where s is a generic pixel, $\Phi(\cdot)$ is the morphological 2D dilation operation and $|\cdot|$ is the cardinality of the given set. We apply a row-wise L2 normalization and we obtain a matrix of *proximity ratios* $\mathbf{M}^{GT}_{[i,:]} = \tilde{\mathbf{M}}^{GT}_{[i,:]} / \|\tilde{\mathbf{M}}^{GT}_{[i,:]}\|_2$ that measures the flow from the considered part to all the others.

With this definition, non-adjacent parts have 0 as entry. The same approach is used for the adjacency matrix computed on the predicted segmentation map $\mathbf{M}^{pred}$ by substituting p_i^{GT} with $p_i^{pred} = \hat{\mathbf{Y}}_p[i]$.

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

$$\mathcal{L}_{GM} = ||\mathbf{M}^{GT} - \mathbf{M}^{pred}||_F \tag{7}$$

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.

Summarizing, the overall training objective of our framework is:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda_1 \mathcal{L}_{rec} + \lambda_2 \mathcal{L}_{GM} \tag{8}$$

where the hyper-parameters λ_1 and λ_2 are used to control the relative contribution of the three losses to the overall objective function.

5 Training of the Deep Learning Architecture

5.1 Multi-Part Dataset

For the experimental evaluation of the proposed multi-class part parsing framework we employed the Pascal-Part [8] dataset, which is currently the largest dataset for this purpose. It contains a total of 10103 variable-sized images with pixel-level parts annotation on the 20 Pascal VOC2010 [15] semantic object classes (plus the *background* class). We employ the original split from [8] with 4998 images in the *trainset* for training and 5105 images in the *valset* for testing. We consider two different sets of labels for this dataset. Firstly, following [51], which is the only work dealing with the multi-class part parsing problem, we grouped the original semantic classes into 58 part classes in total. Additionally, to further test our method on a even more challenging scenario, we consider the grouping rules proposed by [17] for part detection that, instead, leads to a larger set of 108 parts.

5.2 Training Details

The modules introduced in this work are agnostic to the underlying network architecture and can be extended to other scenarios. For fair comparison with [51] we employ a Deeplab-v3 [6] architecture with ResNet101 [21] as the backbone. We follow the same training schemes of [5,6,51] and we started from the official TensorFlow [1] implementation of the Deeplab-v3 [6,4]. The ResNet101 was pre-trained on ImageNet [10] and its weights are available at [4]. During training, images are randomly left-right flipped and scaled of a factor from 0.5 to 2 times the original resolution with bilinear interpolation. The results in the testing stage are reported at the original image resolution. The model is trained for $50K$ steps with the base learning rate set to $5 \cdot 10^{-3}$ and decreased with a polynomial decay rule with power 0.9. 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 [5,51]. We use a batch size of 10 images and we set $\lambda_1 = 10^{-3}$ and $\lambda_2 = 10^{-1}$ to balance part segmentation. For the evaluation metric, we employ the mean Intersection over Union (mIoU) since pixel accuracy is dominated by large regions and little sensitive to the segmentation on many small parts, that are instead the main target of this work. The code and the part labels are publicly available at `https://lttm.dei.unipd.it/paper_data/GMNet`.

6 Experimental Results

In this section we show the experimental results on the multi-class part parsing task in two different scenarios with 58 and 108 parts respectively. We also present some ablation studies to verify the effectiveness of the proposed methodologies.

6.1 Pascal-Part-58

To evaluate our framework we start from the scenario with 58 parts, i.e., the same experimental setting used in [51]. In Table 1 we compare the proposed model with existing semantic segmentation schemes. As evaluation criteria we

Table 1. IoU results on the Pascal-Part-58 benchmark. mIoU: mean per-part-class IoU. Avg: average per-object-class mIoU.

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[3]	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
FCN[30]	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
DeepLab[5]	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
BSANet[51]	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*
Baseline[6]	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
GMNet	**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**

*: values different from [51] since they were wrongly reported in the paper.

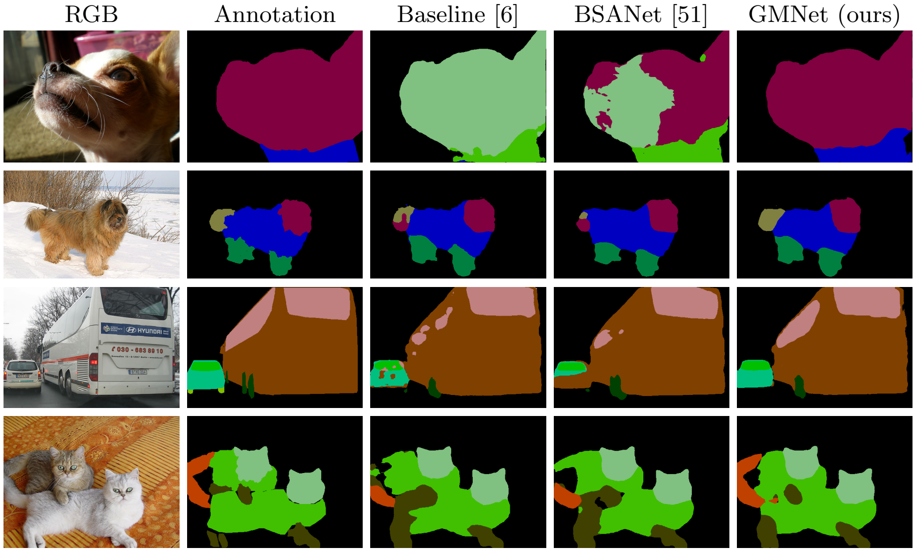

Fig. 3. Segmentation results from the Pascal-Part-58 dataset (*best viewed in colors*).

employ the mean IoU of all the parts (i.e., mIoU), the average IoU for all the parts belonging to each single object, and the mean of these values (denoted as Avg., i.e., in this case each object has the same weight independently of the number of parts). Part-level metrics are reported in the supplementary material. As expected, traditional semantic segmentation architectures such as FCN [30], SegNet [3] and DeepLab [5] are not able to perform a fully satisfactory part-parsing. We adopt as our baseline network the DeepLab-v3 architecture [6], that is the best performing among the compared standard approaches achieving 54.4% of mIoU. The proposed GMNet approach combining both the object-level semantic embedding and the graph matching module achieves a higher accuracy of 59.0% of mIoU, significantly outperforming all the other methods and in particular the baseline on every class with a gap of 4.6% of mIoU. The only other method specifically addressing part-based semantic segmentation is BSANet [51], which achieves a lower mIoU of 58.2%. Our method achieves higher results over most of the objects, both with many parts (like *cow*, *dog* and *sheep*) and with no or few parts (like *boat*, *bottle*, *chair*, *dining table* and *sofa*).

Some qualitative results are shown in Figure 3 while additional ones are in the supplementary material. The figure allows to appreciate the effects of the two main contributions, the semantic embedding and the graph matching modules.

From one side, the object-level semantic embedding network brings useful additional information prior to the part-level decoding stage, thus enriching the extracted features to be object discriminative. We can appreciate this aspect from the first and the third row. In the first row, the baseline completely misleads a dog with a cat (light green corresponds to *cat_head* and green to *cat_torso*).

BSANet is able to partially recover the *dog_head* (amaranth corresponds to the proper 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. A similar discussion can be done also on the third image, where 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 BSANet is not able to correct this error. GMNet, instead, can identify the correct object-level class and the respective parts, excluding the very small and challenging *car_wheels*, and at the same time can better segment the *bus_window*.

From the other side, 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 second and fourth row. In the second 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). Thanks to the graph matching module, GMNet can disambiguate between such 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 GMNet to properly segment and label the *cat_tail* and to partially correct the estimate of the *cat_legs*.

6.2 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. Even though we can immediately verify a drop in the overall performance, that is predictable being the task more complex with respect to the previous scenario with an almost double number of parts, we can appreciate that our framework is able to largely surpass both the baseline and [51]. It achieves a mIoU of 45.8%, outperforming the baseline by 4.5% and the other compared standard segmentation networks by an even larger margin. The gain with respect to the main competitor [51]

Table 2. IoU results on the Pascal-Part-108 benchmark. mIoU: mean per-part-class IoU. Avg: average per-object-class mIoU. †: re-trained on the Pascal-Part-108 dataset.

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[3]	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[30]	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[5]	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
BSANet† [51]	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 [6]	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	**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**

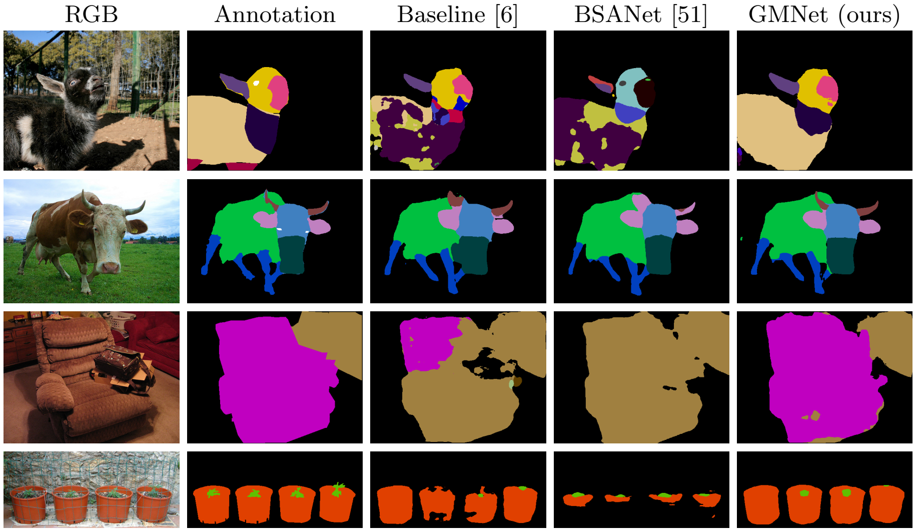

Fig. 4. Segmentation results from the Pascal-Part-108 dataset (*best viewed in colors*).

is remarkable with a gap of 2.9% of mIoU. In this scenario, indeed, most of the previous considerations holds and are even more evident from the results. 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. Also for this setup, further results regarding per-part metrics are reported in the supplementary material.

Thanks to the object-level semantic embedding network our model is able to obtain accurate segmentation of 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 [51] ranges from 5.4% for the *plant* class to an impressive 15% 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 objects are composed by tiny and difficult parts, the gain with respect to [51] is still significant and ranges between 1.5% on *horse* parts to 11.2% on *cow* ones.

The visual results for some sample scenes presented in Figure 4 confirm the numerical evaluation (additional samples are shown in the supplementary material). We can appreciate that the proposed method is able to compute accurate segmentation maps both when a few elements or many parts coexist in the scene. More in detail, 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. The BSANet approach [51] achieves even worse segmentation and labeling performance. Such methods mislead the sheep with a dog (in the figure 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, GMNet 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 more 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). GMNet, instead, achieves superior results thanks to the graph module which accounts 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 reported. Again, we can check how GMNet is able to discriminate between *chair* (in pink) and *sofa* (in light brown). Finally, in the last row we can appreciate how the two parts of the *potted plant* are correctly segmented by GMNet thanks to the semantic embedding module for what concerns object identification and to the graph matching strategy for what concerns small parts localization.

6.3 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.

We start by evaluating the individual impact of the modules and the performance analysis is shown in Table 3. 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: its combination with the reconstruction loss leads to 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 (column $\mathcal{L}_{GM}^{u}$ in the table). This simplified graph leads to a mIoU of 58.7%, lacking some information about the closeness of adjacent parts.

Then, we present a more accurate analysis of the impact of the semantic embedding module and the results are summarized in Table 4. 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 a simple stacking of architectures is not the best option for our task. Additionally, we considered also the option of directly

Table 3. mIoU ablation results on Pascal-Part-58. $\mathcal{L}_{GM}^{u}$: graph matching with unweighted graph.

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

Table 4. 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
$\mathcal{E}_o$ conditioning	55.7
GMNet	**59.0**
With objects GT	65.6

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 the complete proposed framework (59.0%). 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 introduces some errors, 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.

We conclude remarking that GMNet achieves almost always higher accuracy than the starting baseline, even if small and unstructured parts remain the most challenging to be detected. Furthermore, the gain depends also on the amount of spatial relationships that can be exploited.

7 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. Additionally, we designed a novel adjacency graph-based module that aims at matching the relative spatial relationships between ground truth and predicted parts which has shown large improvements particularly on small parts. Combining the proposed methodologies we were able to achieve state-of-the-art results in the challenging task of multi-object part parsing both at a moderate scale and at a larger one.

Further research will investigate the extension of the proposed modules to other scenarios. We will also consider the explicit embedding into the proposed framework of the edge information coming from part-level and object-level segmentation maps. Novel graph representations better capturing part relationships and different matching functions will be investigated.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI). pp. 265–283 (2016)
2. Azizpour, H., Laptev, I.: Object detection using strongly-supervised deformable part models. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 836–849. Springer (2012)
3. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **39**(12), 2481–2495 (2017)
4. Chen, L.C.: DeepLab official TensorFlow implementation (Accessed: 2020-03-01), `https://github.com/tensorflow/models/tree/master/research/deeplab`
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: 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 (2018)
6. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
7. Chen, L.C., Yang, Y., Wang, J., Xu, W., Yuille, A.L.: Attention to scale: Scale-aware semantic image segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3640–3649 (2016)
8. Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., Yuille, A.: Detect what you can: Detecting and representing objects using holistic models and body parts. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1971–1978 (2014)
9. Das, D., Lee, C.G.: Unsupervised domain adaptation using regularized hyper-graph matching. In: Proceedings of IEEE International Conference on Image Processing (ICIP). pp. 3758–3762. IEEE (2018)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 248–255. IEEE (2009)
11. Dhar, P., Singh, R.V., Peng, K.C., Wu, Z., Chellappa, R.: Learning without memorizing. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5138–5146 (2019)
12. Dong, J., Chen, Q., Shen, X., Yang, J., Yan, S.: Towards unified human parsing and pose estimation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 843–850 (2014)
13. Emmert-Streib, F., Dehmer, M., Shi, Y.: Fifty years of graph matching, network alignment and network comparison. Information Sciences **346**, 180–197 (2016)
14. Eslami, S., Williams, C.: A generative model for parts-based object segmentation. In: Neural Information Processing Systems (NeurIPS). pp. 100–107 (2012)
15. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision (IJCV) **88**(2), 303–338 (2010)
16. Fang, H.S., Lu, G., Fang, X., Xie, J., Tai, Y.W., Lu, C.: Weakly and semi supervised human body part parsing via pose-guided knowledge transfer. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)

17. Gonzalez-Garcia, A., Modolo, D., Ferrari, V.: Do semantic parts emerge in convolutional neural networks? International Journal of Computer Vision (IJCV) **126**(5), 476–494 (2018)
18. Guo, Y., Liu, Y., Georgiou, T., Lew, M.S.: A review of semantic segmentation using deep neural networks. International journal of multimedia information retrieval **7**(2), 87–93 (2018)
19. Haggag, H., Abobakr, A., Hossny, M., Nahavandi, S.: Semantic body parts segmentation for quadrupedal animals. In: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 000855–000860 (2016)
20. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 447–456 (2015)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
22. Krause, J., Jin, H., Yang, J., Fei-Fei, L.: Fine-grained recognition without part annotations. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5546–5555 (2015)
23. Li, Z., Hoiem, D.: Learning without forgetting. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **40**(12), 2935–2947 (2018)
24. Liang, X., Gong, K., Shen, X., Lin, L.: Look into person: Joint body parsing & pose estimation network and a new benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **41**(4), 871–885 (2018)
25. Liang, X., Lin, L., Shen, X., Feng, J., Yan, S., Xing, E.P.: Interpretable structure-evolving lstm. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1010–1019 (2017)
26. Liang, X., Liu, S., Shen, X., Yang, J., Liu, L., Dong, J., Lin, L., Yan, S.: Deep human parsing with active template regression. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **37**(12), 2402–2414 (2015)
27. Liang, X., Shen, X., Feng, J., Lin, L., Yan, S.: Semantic object parsing with graph lstm. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 125–143. Springer (2016)
28. Liu, X., Deng, Z., Yang, Y.: Recent progress in semantic image segmentation. Artificial Intelligence Review **52**(2), 1089–1106 (2019)
29. Livi, L., Rizzi, A.: The graph matching problem. Pattern Analysis and Applications **16**(3), 253–283 (2013)
30. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3431–3440 (2015)
31. Lu, W., Lian, X., Yuille, A.: Parsing semantic parts of cars using graphical models and segment appearance consistency. arXiv preprint arXiv:1406.2375 (2014)
32. Mel, M., Michieli, U., Zanuttigh, P.: Incremental and multi-task learning strategies for coarse-to-fine semantic segmentation. Technologies **8**(1), 1 (2020)
33. Michieli, U., Zanuttigh, P.: Incremental learning techniques for semantic segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2019)
34. Michieli, U., Zanuttigh, P.: Knowledge distillation for incremental learning in semantic segmentation. arXiv preprint arXiv:1911.03462 (2020)
35. Nie, X., Feng, J., Yan, S.: Mutual learning to adapt for joint human parsing and pose estimation. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 502–517 (2018)

36. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2001–2010 (2017)
37. Shmelkov, K., Schmid, C., Alahari, K.: Incremental learning of object detectors without catastrophic forgetting. In: Proceedings of International Conference on Computer Vision (ICCV). pp. 3400–3409 (2017)
38. Song, Y., Chen, X., Li, J., Zhao, Q.: Embedding 3d geometric features for rigid object part segmentation. In: Proceedings of International Conference on Computer Vision (ICCV). pp. 580–588 (2017)
39. Sun, J., Ponce, J.: Learning discriminative part detectors for image classification and cosegmentation. In: Proceedings of International Conference on Computer Vision (ICCV). pp. 3400–3407 (2013)
40. Wang, J., Yuille, A.L.: 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 (2015)
41. Wang, P., Shen, X., Lin, Z., Cohen, S., Price, B., Yuille, A.L.: Joint object and part segmentation using deep learned potentials. In: Proceedings of International Conference on Computer Vision (ICCV). pp. 1573–1581 (2015)
42. Wang, Y., Tran, D., Liao, Z., Forsyth, D.: Discriminative hierarchical part-based models for human parsing and action recognition. Journal of Machine Learning Research **13**(Oct), 3075–3102 (2012)
43. Xia, F., Wang, P., Chen, L.C., Yuille, A.L.: 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 (2016)
44. Xia, F., Wang, P., Chen, X., Yuille, A.L.: Joint multi-person pose estimation and semantic part segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6769–6778 (2017)
45. Xia, F., Zhu, J., Wang, P., Yuille, A.: Pose-guided human parsing with deep learned features. arXiv preprint arXiv:1508.03881 (2015)
46. Yamaguchi, K., Kiapour, M.H., Ortiz, L.E., Berg, T.L.: Parsing clothing in fashion photographs. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3570–3577. IEEE (2012)
47. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1385–1392 (2011)
48. Zhang, N., Donahue, J., Girshick, R., Darrell, T.: Part-based r-cnns for fine-grained category detection. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 834–849. Springer (2014)
49. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2881–2890 (2017)
50. Zhao, J., Li, J., Nie, X., Zhao, F., Chen, Y., Wang, Z., Feng, J., Yan, S.: Self-supervised neural aggregation networks for human parsing. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 7–15 (2017)
51. Zhao, Y., Li, J., Zhang, Y., Tian, Y.: Multi-class part parsing with joint boundary-semantic awareness. In: Proceedings of International Conference on Computer Vision (ICCV). pp. 9177–9186 (2019)
52. Zhu, L.L., Chen, Y., Lin, C., Yuille, A.: 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 (2011)

GMNet: Graph Matching Network for Large Scale Part Semantic Segmentation in the Wild
Supplementary Material

Umberto Michieli[0000−0003−2666−4342], Edoardo Borsato, Luca Rossi, and Pietro Zanuttigh[0000−0002−9502−2389]

Department of Information Engineering, University of Padova, Padova, Italy
{michieli, borsatoedo, rossiluc, zanuttigh}@dei.unipd.it

In this document we show some further experimental results. In particular we report the Intersection over Union (IoU) and Pixel Accuracy (PA) for each per-part-class and some averaged metrics as mean IoU (mIoU), mean PA (mPA) and mean Class Accuracy (mCA) [2]. The results are reported for both the Pascal-Part-58 and the Pascal-Part-108 datasets. Finally, some additional visual results on both datasets are presented.

1 Additional Results on Pascal-Part-58

We start by analyzing the per-part-class IoU and PA on the Pascal-Part-58 dataset. The results are shown in Table 1, where it is possible to see that the proposed method (GMNet) outperforms the baseline [1] approach on almost every part both considering the per-part-IoU and the per-part-PA. With respect to BSANet [3], GMNet can produce clearly higher results on 15 objects out of 21 (such as *bottle*, *bus*, *dog*, *sheep*,...) and can produce comparable results on 2 objects (i.e., on *car* ad *cat*).

We can further verify the ranking of the compared methods analyzing the average metrics reported in Table 2. Here, we can appreciate how GMNet is able to outperform both the baseline and BSANet robustly on all the most widely used metrics for semantic segmentation.

Then, we proceed to analyze some additional qualitative results as reported in Figure 1. The effects of the two main components of our work, namely the object-level semantic embedding network $\mathcal{S}$ and the graph matching module, are clearly visible in the images. The effect of the semantic embedding network is evident in the last 5 rows, where object-level conditioning helps the part-level decoder to accurately segment and label the parts. For instance, in the last row both the baseline and BSANet mislead the dog's parts with cat's parts, while GMNet is able to avoid this error. In row 6, BSANet confuses cow's parts with sheep's parts. In row 7, the baseline confuses sheep's parts with cat's ones and BSANet with dog's parts. GMNet is able to correctly deal with these situations thanks to the object-level guidance.

Table 1. Per-part IoU and PA on the Pascal-Part-58 dataset.

Parts Name	**Baseline**		**BSANet**		**GMNet**		Parts Name	**Baseline**		**BSANet**		**GMNet**	
	IoU	**PA**	**IoU**	**PA**	**IoU**	**PA**		**IoU**	**PA**	**IoU**	**PA**	**IoU**	**PA**
background	91.1	96.3	91.6	96.7	**92.7**	**96.9**	cow tail	0.0	0.0	7.9	8.1	**8.1**	**8.4**
aeroplane body	66.6	79.8	**70.0**	**81.4**	69.6	81.2	cow leg	46.1	62.3	53.4	**67.5**	**53.5**	67.2
aeroplane engine	25.7	31.4	**29.1**	**33.8**	25.7	31.2	cow torso	69.9	83.5	73.5	85.9	**77.1**	**87.8**
aeroplane wing	33.5	48.2	**38.3**	**49.1**	34.2	46.4	dining table	43.0	55.4	43.7	54.8	**51.3**	**62.6**
aeroplane stern	57.1	68.2	**59.2**	**72.5**	57.2	70.8	dog head	78.7	88.3	82.5	91.4	**85.0**	**92.7**
aeroplane wheel	45.4	53.3	**53.2**	**62.5**	46.8	53.3	dog leg	48.1	59.9	53.8	63.0	**53.8**	**64.8**
bike wheel	78.0	88.1	78.0	**88.6**	**81.3**	88.5	dog tail	27.1	39.4	31.3	38.0	**31.4**	**41.5**
bike body	48.4	61.2	**53.4**	**68.4**	51.5	64.2	dog torso	63.7	76.8	65.7	79.7	**68.0**	**81.2**
bird head	64.6	72.7	**74.0**	**80.2**	71.1	79.3	horse head	74.7	81.7	**76.6**	**83.3**	73.9	80.5
bird wing	35.1	45.5	**39.7**	**53.2**	38.6	52.9	horse tail	47.0	60.4	**51.0**	59.9	50.4	**62.2**
bird leg	29.3	37.6	**34.8**	**42.6**	28.7	35.4	horse leg	55.9	70.9	**61.6**	**75.8**	59.3	72.9
bird torso	66.9	83.1	**70.9**	**84.4**	69.5	83.1	horse torso	70.3	84.2	**74.9**	86.6	73.9	**87.4**
boat	54.4	64.8	60.2	69.6	**70.0**	**78.5**	mbike wheel	70.9	82.5	71.6	82.1	**73.5**	**84.0**
bottle cap	30.7	35.4	29.8	35.0	**33.9**	**42.5**	mbike body	65.1	80.9	71.5	87.7	**74.3**	**87.8**
bottle body	68.8	78.5	68.6	74.8	**77.6**	**86.1**	person head	83.5	91.6	**85.0**	**92.3**	84.7	91.8
bus window	72.7	83.7	74.8	85.9	**75.4**	**86.1**	person torso	65.9	80.6	**68.2**	**82.7**	67.0	82.3
bus wheel	55.3	66.3	57.1	70.1	**58.1**	**72.1**	person larm	46.9	60.0	**52.0**	**65.6**	48.6	62.8
bus body	74.8	88.2	78.3	88.7	**79.9**	**89.8**	person uarm	51.5	65.8	**54.4**	**68.2**	52.4	66.9
car window	62.6	73.9	**68.1**	**78.2**	64.8	77.5	person lleg	38.6	51.5	**43.5**	**54.6**	40.2	51.5
car wheel	64.8	78.1	68.5	79.7	**70.3**	**79.8**	person uleg	43.8	60.0	**47.4**	**63.5**	44.5	59.9
car light	46.2	54.3	**53.7**	**61.7**	48.4	56.0	pplant pot	45.3	61.0	53.5	64.8	**56.0**	**69.1**
car plate	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	pplant plant	52.4	62.1	**56.6**	65.8	56.4	**66.4**
car body	72.1	86.4	77.0	**88.4**	**77.6**	88.2	sheep head	60.9	69.3	65.4	71.3	**70.8**	**79.0**
cat head	80.2	90.4	83.7	**92.3**	**83.8**	91.6	sheep leg	8.6	11.1	11.7	16.5	**14.3**	**20.2**
cat leg	48.6	**61.2**	**50.1**	58.6	49.4	59.1	sheep torso	68.3	84.4	71.6	86.1	**75.6**	**88.7**
cat tail	40.2	51.3	**48.8**	55.6	46.0	**56.7**	sofa	43.2	58.8	43.1	57.4	**56.1**	**65.0**
cat torso	70.3	85.7	72.6	**88.0**	**73.8**	87.6	train	79.6	86.1	82.2	90.2	**85.0**	**92.0**
chair	35.4	43.3	36.5	42.7	**51.4**	**63.9**	tv screen	69.5	76.0	73.1	78.6	**77.0**	**84.3**
cow head	74.3	85.6	76.4	86.0	**80.7**	**87.8**	tv frame	45.9	56.9	49.8	60.9	**54.1**	**67.4**

Table 2. Comparison in terms of mIoU, mCA and mPA on Pascal-Part-58.

Method	mIoU	mPA	mCA
Baseline [1]	54.45	89.86	65.42
BSANet [3]	58.15	90.76	68.12
GMNet	**59.04**	**91.55**	**69.22**

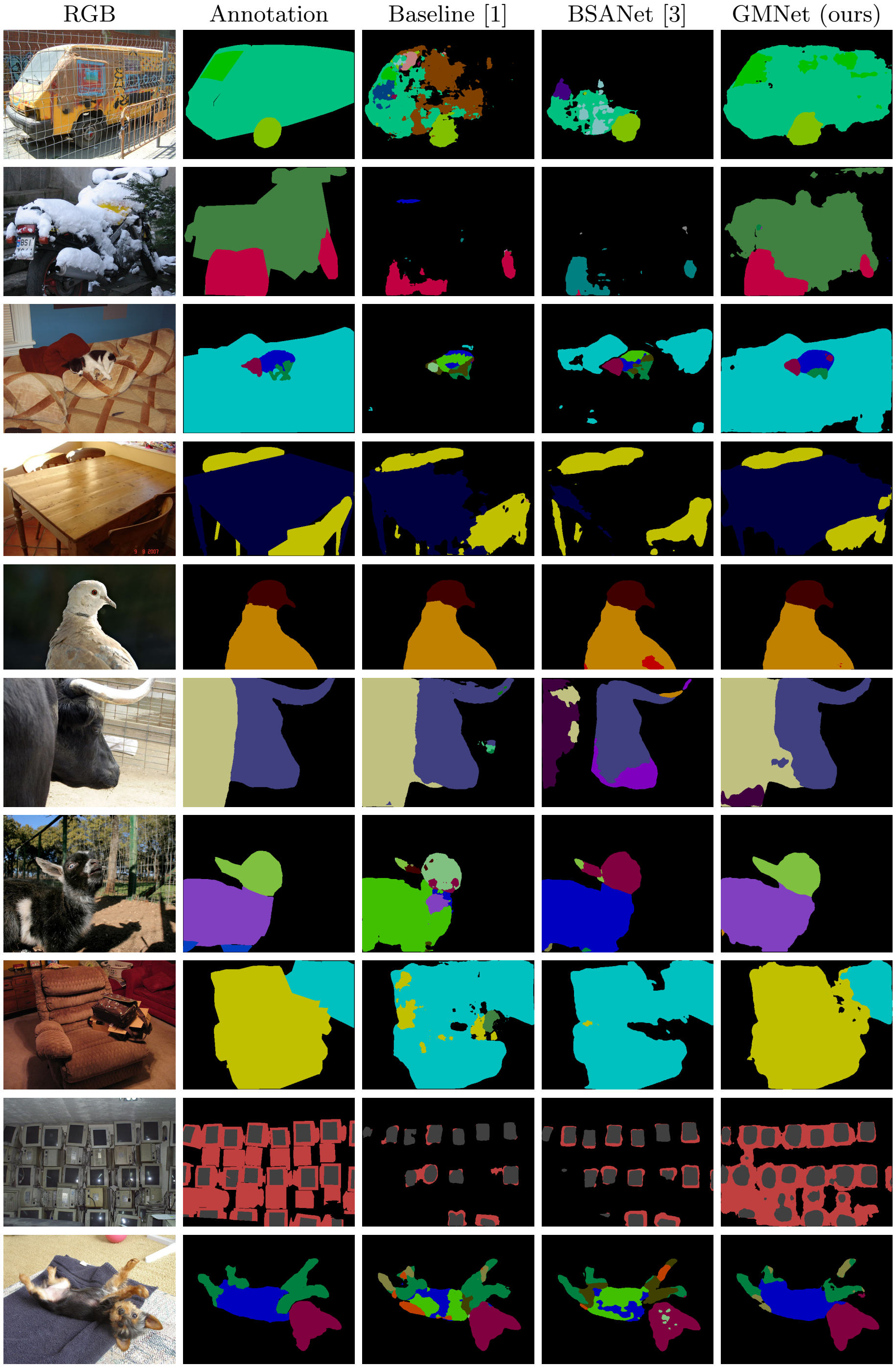

Fig. 1. Qualitative results on some sample scenes on the Pascal-Part-58 dataset (*best viewed in colors*).

The effect of the graph matching module is more appreciable on small parts. For example, we can verify its efficacy in the third row and in the last row. In row 3, both the baseline and BSANet mislead the dog's parts with cat's ones and also their localization is highly imprecise. From one hand, the semantic embedding network corrects the first issue, while the second (i.e., bad localization) is addressed by graph matching. In the last row, graph matching between different reciprocal spatial relationship among parts helps to correctly place the dog's parts.

Moreover, in very challenging images (such as rows 1 to 4), where both the baseline and BSANet partially or completely miss some classes, our method generate superior quality segmentation maps. For instance, in row 1 a vehicle behind a metal grid is being correctly identified and quite well localized in all its parts thanks to the semantic embedding module and to graph matching. The combination of the two modules is also helpful in row 2, where a motorbike covered with snow is being well recognized by our framework. In row 3 we identify the sofa and in row 4 the table, with higher accuracy than the compared methods.

2 Additional Results on Pascal-Part-108

In this section we present some additional results for the Pascal-Part-108 dataset. The per-part-IoU and per-part-PA are reported in Table 3, where we can notice that the gap between the proposed framework and the compared methods is significantly larger than for the Pascal-Part-58 dataset. GMNet achieves higher accuracy than the competitors on almost all the parts. In particular, our framework is able to outperform BSANet [3] in 19 out of 21 object-level classes both with many parts within them (such as *aeroplane*, *bus*, *cat*, *dog*, *person*, *sheep*,...) and with no or few parts within them (such as *boat*, *bottle*, *chair*, *sofa*, *tv*,..).

The mean accuracy results are shown in Table 4 where we can verify that our method clearly outperforms both the baseline [1] and BSANet [3] on all the most popular metrics used to evaluate semantic segmentation architectures. Hence, we prove the robustness of our framework to different evaluation criteria and to different datasets. Additionally, we argue that the proposed framework is able to scale well to even larger sets of parts.

Then, in Figure 2 we report some additional qualitative results. The effect of the object-level semantic embedding network is particularly evident in the first 4 rows. In row 1, a challenging image is presented where both the baseline and BSANet are not able to correctly identify the table. In rows 2 and 3, GMNet generates cleaner segmentation maps exploiting object-level priors which help to disambiguate between cars and buses. In row 4, the baseline and BSANet predict cat's parts in spite of horse's parts which are partially identified by our method.

Table 3. Per-part IoU and PA on the Pascal-Part-108 dataset.

Parts Name	**Baseline**		**BSANet**		**GMNet**		Parts Name	**Baseline**		**BSANet**		**GMNet**	
	IoU	**PA**	**IoU**	**PA**	**IoU**	**PA**		**IoU**	**PA**	**IoU**	**PA**	**IoU**	**PA**
background	90.9	**97.2**	91.6	97.1	**92.7**	97.0	dining table	33.0	40.2	45.9	59.7	**50.6**	**62.3**
aero body	61.9	72.3	**68.2**	77.6	61.9	**82.6**	dog head	60.5	75.5	63.8	78.2	**64.0**	**78.9**
aero stern	53.2	68.4	54.2	65.3	**57.4**	**71.0**	dog reye	50.1	61.4	54.1	61.4	**54.7**	**64.7**
aero rwing	28.9	39.8	33.1	**46.5**	**34.3**	46.0	dog rear	54.0	69.4	**57.2**	73.4	56.8	**73.9**
aero engine	24.7	29.0	26.5	32.0	**27.2**	**32.6**	dog nose	63.5	75.0	**66.3**	74.3	66.0	**76.8**
aero wheel	40.9	46.8	44.5	49.6	**51.5**	**61.3**	dog torso	58.4	74.6	62.3	78.4	**63.2**	**79.1**
bike fwheel	78.4	85.7	75.3	86.7	**80.2**	**87.8**	dog neck	27.1	35.4	26.2	30.8	**28.1**	**35.5**
bike saddle	34.1	39.8	31.0	31.9	**38.0**	**43.2**	dog rfleg	39.2	50.6	42.4	53.5	**43.7**	**55.8**
bike handlebar	**23.3**	**26.1**	20.6	22.8	22.4	25.9	dog rfpaw	39.4	47.9	**44.2**	51.7	43.7	**52.9**
bike chainwheel	42.3	50.4	36.5	41.6	**44.1**	**57.0**	dog tail	24.7	37.8	**34.9**	**42.3**	30.8	41.4
birds head	51.5	61.3	**66.4**	**78.0**	65.3	77.7	dog muzzle	65.1	76.1	**69.4**	**82.3**	68.9	80.4
birds beak	40.4	49.5	**47.1**	**54.6**	44.3	54.0	horse head	54.4	67.0	**57.1**	**68.9**	55.9	68.3
birds torso	61.7	77.9	**65.2**	79.4	64.8	**82.6**	horse rear	49.7	58.1	51.1	56.5	**52.2**	**65.6**
birds neck	27.5	32.2	**39.1**	**50.1**	28.4	35.7	horse muzzle	61.3	68.7	**65.2**	**74.0**	62.9	69.5
birds rwing	35.9	50.4	**39.3**	**53.7**	37.2	50.1	horse torso	56.7	75.9	59.5	75.9	**60.7**	**84.3**
birds rleg	23.5	28.6	**26.5**	32.2	23.8	**32.8**	horse neck	42.1	51.3	**49.6**	**64.8**	47.2	55.8
birds rfoot	13.9	16.3	11.6	12.7	**17.7**	**22.5**	horse rfuleg	54.1	68.5	**57.0**	**71.8**	56.4	70.9
birds tail	28.1	39.2	**33.0**	44.1	32.5	**46.1**	horse tail	48.1	63.5	47.6	54.5	**51.4**	**64.4**
boat	53.7	60.3	61.4	71.5	**69.2**	**77.8**	horse rfho	24.1	31.4	12.9	13.7	**25.3**	**32.7**
bottle cap	30.4	35.0	26.2	30.0	**33.4**	**40.0**	mbike fwheel	69.6	78.9	69.3	80.4	**73.6**	**83.3**
bottle body	63.7	69.5	71.5	78.3	**78.7**	**88.3**	mbike hbar	**0.0**	**0.0**	0.0	0.0	0.0	0.0
bus rightside	70.8	85.3	73.0	83.7	**75.7**	**88.4**	mbike saddle	0.0	0.0	0.0	0.0	**0.8**	**0.8**
bus roofside	7.5	7.7	0.3	0.3	**13.5**	**14.4**	mbike hlight	25.8	**32.8**	10.6	11.2	**28.5**	32.4
bus mirror	2.1	2.2	0.3	0.3	**6.6**	**7.6**	person head	68.2	81.9	**69.7**	82.2	69.3	**82.7**
bus fliplate	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	person reye	35.1	39.3	**41.3**	**46.3**	38.7	43.9
bus door	**40.1**	51.2	37.2	**53.2**	38.1	47.3	person rear	37.4	46.0	**41.9**	49.4	41.4	**51.5**
bus wheel	**54.8**	65.5	53.1	63.9	56.7	**69.4**	person nose	53.0	62.1	54.3	63.1	**56.7**	**67.5**
bus headlight	**25.6**	28.3	19.9	20.8	30.4	**34.2**	person mouth	48.9	56.9	49.5	54.9	**51.3**	**60.8**
bus window	71.8	85.2	73.5	86.4	**74.6**	**87.4**	person hair	70.8	83.3	**72.3**	**85.9**	71.8	83.9
car rightside	64.0	78.0	67.9	81.2	**70.5**	**84.5**	person torso	63.4	79.1	64.3	78.3	**65.2**	**80.9**
car roofside	21.0	25.4	16.1	17.6	**22.3**	**26.6**	person neck	49.7	63.8	50.9	65.1	**51.2**	**65.3**
car fliplate	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	person ruarm	54.7	68.6	55.7	70.2	**57.4**	**71.3**
car door	41.4	52.5	39.6	49.0	**42.3**	**53.5**	person rhand	43.0	55.4	**47.4**	**57.6**	44.1	56.8
car wheel	65.8	74.5	64.0	76.6	**70.2**	**80.0**	person ruleg	50.8	66.0	52.3	67.1	**53.0**	**67.9**
car headlight	42.9	48.4	**49.4**	**59.7**	46.4	54.4	person rfoot	29.8	38.9	28.9	32.4	**31.3**	**39.8**
car window	61.0	75.5	**66.5**	**82.4**	65.0	79.0	pplant pot	43.6	54.5	50.6	58.9	**56.0**	**69.0**
cat head	73.9	87.3	75.6	88.5	**77.5**	**88.5**	pplant plant	42.9	48.8	55.5	**68.7**	**56.6**	66.6
cat reye	58.8	69.0	62.0	71.1	**62.8**	**71.8**	sheep head	45.6	56.9	47.0	58.0	**54.0**	**66.9**
cat rear	65.5	77.7	66.8	77.1	**67.1**	**78.8**	sheep rear	43.2	53.0	**47.7**	56.6	45.3	**58.2**
cat nose	40.3	49.1	41.2	45.8	**46.3**	**56.2**	sheep muzzle	58.2	67.0	61.1	72.4	**64.9**	**74.7**
cat torso	64.2	81.4	66.8	84.2	**68.7**	**86.0**	sheep rhorn	3.0	3.6	0.0	0.0	**5.4**	**6.6**
cat neck	22.8	33.8	19.8	25.0	24.4	**34.1**	sheep torso	62.6	78.0	66.4	83.6	**68.8**	**86.3**
cat rfleg	36.5	48.5	38.5	49.2	**39.1**	**50.0**	sheep neck	26.9	38.1	25.3	**41.2**	**30.3**	41.0
cat rfpaw	40.6	50.2	**43.4**	**51.5**	41.7	50.7	sheep rfuleg	8.6	10.6	**17.4**	**24.5**	11.7	14.7
cat tail	40.2	52.2	42.6	49.5	**45.8**	**57.0**	sheep tail	6.7	7.4	1.1	1.1	**9.1**	**11.5**
chair	35.4	42.3	34.1	38.4	**49.1**	**60.4**	sofa	39.2	50.7	44.5	56.9	**53.9**	**66.1**
cow head	51.2	65.5	58.2	74.2	**63.8**	**74.9**	train head	5.3	**6.4**	**5.6**	6.4	4.5	5.3
cow rear	51.2	68.5	53.0	72.9	**60.0**	**75.1**	train hrightside	61.9	77.3	**63.5**	**84.0**	60.8	83.1
cow muzzle	61.2	77.6	67.2	81.9	**74.9**	**86.7**	train hroofside	**23.0**	**28.0**	13.7	17.0	21.1	26.3
cow rhorn	28.8	35.0	10.1	10.2	**44.0**	**50.6**	train headlight	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
cow torso	63.4	78.6	69.9	85.8	**73.2**	**87.2**	train coach	28.6	33.6	**42.0**	**47.8**	31.4	37.9
cow neck	9.5	12.7	7.3	7.9	**20.3**	**25.9**	train crightside	15.6	24.5	**19.0**	30.6	14.9	**33.8**
cow rfuleg	46.5	60.0	49.7	61.4	**54.8**	**70.7**	train croofside	10.8	11.9	1.0	1.0	**18.1**	**22.6**
cow tail	6.5	7.3	0.1	0.1	**13.6**	**14.9**	tv screen	60.8	71.3	66.3	79.5	**70.7**	**82.9**

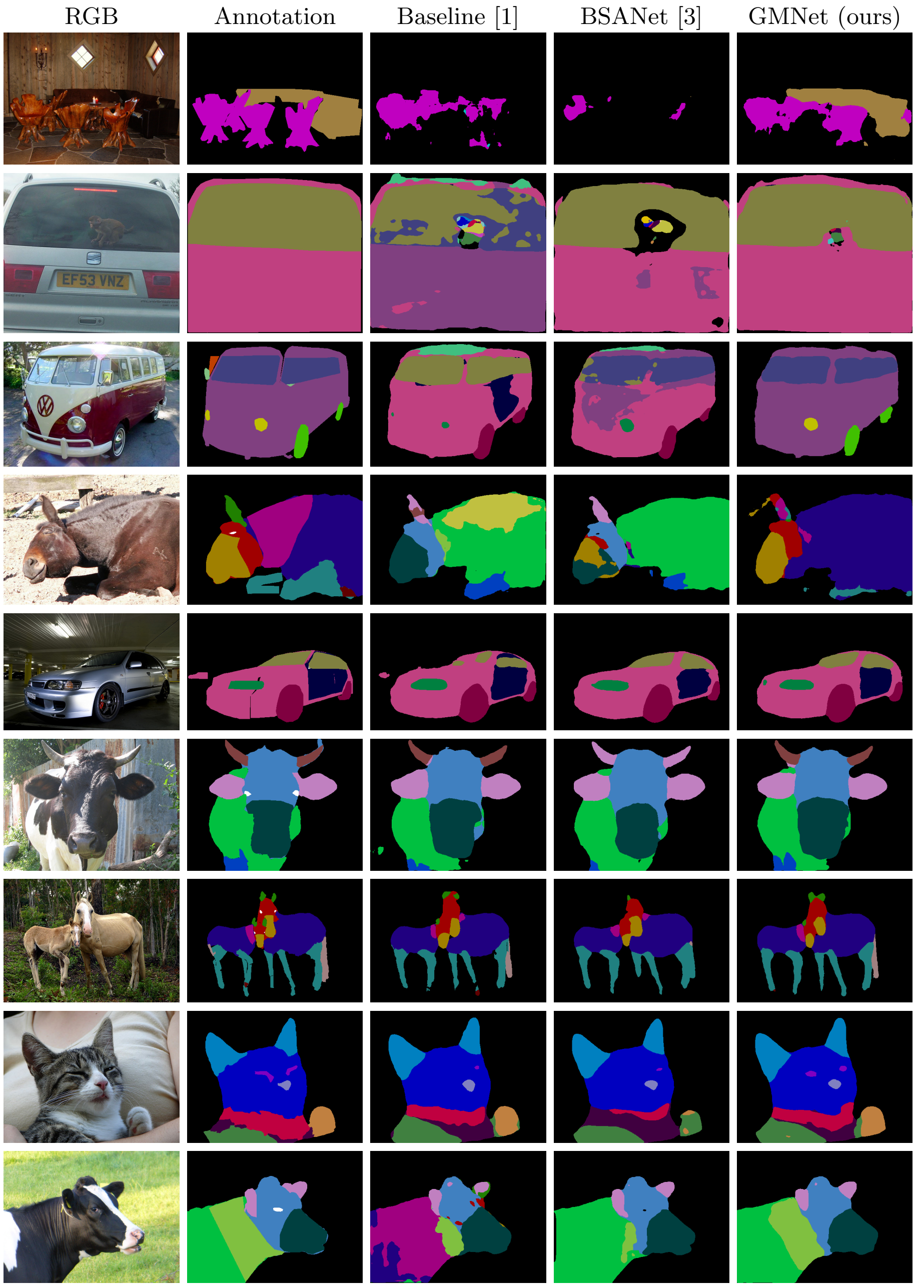

Fig. 2. Qualitative results on sample scenes on the Pascal-Part-108 dataset (*best viewed in colors*).

Table 4. Comparison in terms of mIoU, mCA and mPA on Pascal-Part-108.

Method	mIoU	mPA	mCA
Baseline [1]	41.36	88.57	50.51
BSANet [3]	42.95	89.52	51.71
GMNet	**45.80**	**90.32**	**55.68**

The graph matching module is much more effective on this dataset because contains many small-sized parts. We can verify this from the sixth to the last row. In row 6, the *cow_horns* and *cow_body* are badly localized and labelled both by the baseline and by BSANet. However, the graph matching component on the reciprocal spatial relationship between these parts and the others guides the network to properly localize and label such parts. In row 7 our framework is able to well localize horse's parts and especially the challenging *horse_tail* part. In the second-last row, GMNet correctly identifies difficult cat's parts such as *cat_eyes* and *cat_paws* thanks to the graph matching module. In the last row, the semantic embedding module allows our method to identify the cow and, at the same time, the graph matching module allows to correctly localize the spatial relations among all the parts.

References

1. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
2. Csurka, G., Larlus, D., Perronnin, F., Meylan, F.: What is a good evaluation measure for semantic segmentation? In: Proceedings of British Machine Vision Conference (BMVC). vol. 27, p. 2013 (2013)
3. Zhao, Y., Li, J., Zhang, Y., Tian, Y.: Multi-class part parsing with joint boundary-semantic awareness. In: Proceedings of International Conference on Computer Vision (ICCV). pp. 9177–9186 (2019)