Embedding-Interaction based Panoptic Segmentation

Anonymous ICCV submission

Paper ID 8435

Abstract

Existing panoptic segmentation methods usually contain two independent branches: Since the two branches mostly employ different structures, simply aggregating features from each branch may lead to a sub-optimal solution due to the lack of modeling strong interaction between tasks or branches. The situation has motivated us to focus on the intrinsic logical and functional connection between the semantic embedding and the instance embedding to improve the quality of panoptic segmentation. In this paper, we present our Embedding Interaction Network (EINet) to explicitly model the interaction between these two types of embedding. Specifically, EINet adopts an encoder-decoder backbone with the proposed interaction mechanism to explicitly regularize semantic embedding and instance embedding and to learn center embedding of each semantic category in the dataset as well as centroid embedding of each instance, where instance segmentation is simplified by measuring the distance between pixel features in the embedding space and learned instance centroid embedding. As a result, EINet achieves superior results on Cityscapes and COCO datasets. The results clearly show the feasibility of joint optimization for panoptic segmentation through a designed embedding interaction mechanism without using too many intermediate results. Project page at https://github.com/Wastoon/EIPSNet.

1. Introduction

Panoptic segmentation [18] unifies semantic segmentation [5, 27] and instance segmentation [7, 39], and it attracted a lot of research attentions recently. Panoptic segmentation aims to give each pixel in the scene a single semantic category and instance id, separating semantic category in *stuff* and instance in *thing*. A general method of panoptic segmentation is to perform semantic segmentation and instance segmentation separately, and then transform the results into a panoptic segmentation format by post-processing through hand-crafted heuristic rules [10, 18] or non-parametric module [38, 6, 25].

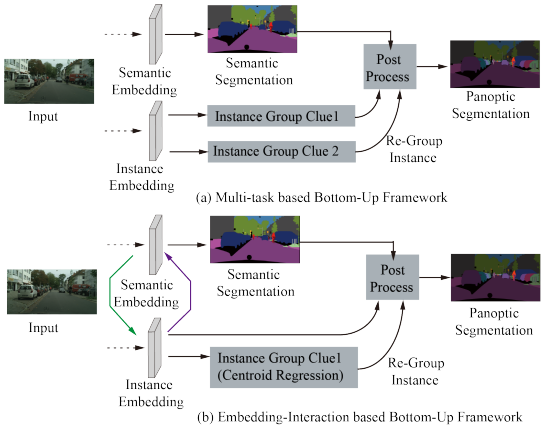


Figure 1. Compared with classical bottom-up framework, which often utilize multiple instance group clues to re-group instance segmentation from semantic segmentation in Fig.1(a), the proposed EINet in Fig.1(b) learns discriminative instance embedding through explicitly constraint to re-group instance segmentation, which reduces the number of subtasks to be optimized, and relieves the pressure on multiple tasks to obtain the optimal solution simultaneously.

Most panoptic segmentation approaches can be divided into two paradigms, the top-down [17, 25, 38, 6, 32] and bottom-up [35, 7, 39, 13] approaches respectively. Top-down panoptic segmentation methods usually employs region-based mask (*e.g.* MaskRCNN [15]) for dense proposals of instances as an auxiliary task. On the other side, bottom-up panoptic segmentation schemes have different representations for instances, such that panoptic segmentation strongly depends on the results of semantic segmentation branch and other designed auxiliary tasks for grouping instances.

The quality of panoptic segmentation can benefit from jointly optimizing both semantic segmentation and instance segmentation by allowing multiple tasks to interact between and benefit from each other [23, 25]. For example, top-down methods [37, 6, 38] consider the common components between semantic categories and instances, and many of them try to model the internal relationship between semantics and instances by sharing or concatenating features from the two branches without any explicit

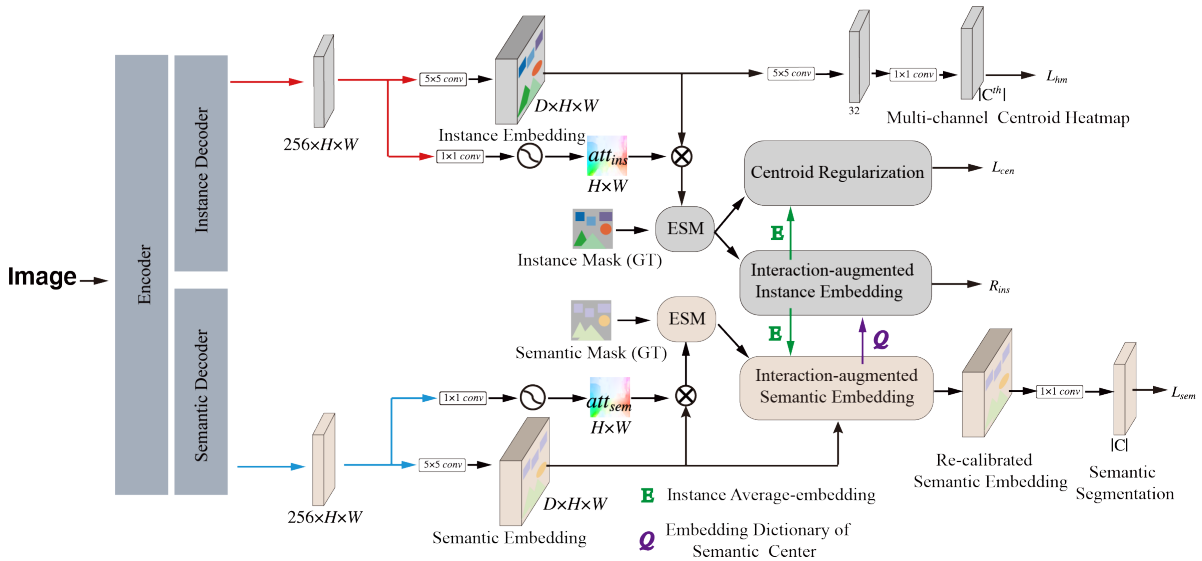


Figure 2. Whole training pipeline of EINet. $|C|$: number of semantic categories. $|C^{th}|$: number of *thing* class. **ESM**: extract and summary module, whose inputs are embedding and ground truth mask, output a extracted embedding tensor. Details of ESM is shown in Fig.3. R_{ins} is regularization item to enhance instance embedding through interaction with semantic embedding. L_{sem} is to optimize semantic segmentation with interaction-augmented semantic embedding. L_{cen} is centroid embedding regularization item to compress representation of instance to its centroid position. L_{hm} is used to regress category-known heatmap proposed in prior work. D is dimension of embedding. att_{ins} and att_{sem} are attention map for instance and semantic category. Additionally, we omit the process of learning instance embedding and semantic embedding based on the metric learning in Fig.2.

constraints. Compared to the top-down methods, bottom-up [7, 39, 13] approaches usually employ multiple auxillary tasks for panoptic segmentation. Newell *et al.* [31] and Li *et al.* [23] think that such framework makes the bottom-up approaches be prone to capture separate objectives in loss functions of multiple sub-tasks, whose local minima for the sub-tasks that do not optimize the joint criterion. Therefore, we want to use instance embedding as a clue to re-group instances to reduce subtasks that need to be optimized. To enhance the instance embedding, we propose the Embedding Interaction Network (EINet) for panoptic segmentation, which has explicit constraint between semantic embedding and instance embedding. Our focus is to make semantic embedding more category-discriminative and instance embedding more individual-discriminative. Specifically, we map from an original image space to a high-dimensional embedding space and learn category encoded semantic embedding as well as instance ID encoded instance embedding through an end-to-end training process. Semantic embedding and instance embedding are associated with each other through the proposed embedding interaction mechanism for joint optimization, and EINet can perform panoptic segmentation without too many intermediate tasks under bottom-up framework.

In summary, we include in the paper three major contributions:

- The EINet that explicitly associates semantic segmentation and instance segmentation together through the proposed interactive mechanism;
- A dynamicaly constructed instance dictionary that enables querying of instance centroid embedding in the dictionary for quick panoptic segmentation;
- New instance assembly clue for bottom-up panoptic segmentation without too many intermediate results. Extensive experiments on COCO [4] and Cityscapes [9] public benchmarks prove the effectiveness.

2. Related Works

According to the framework of panoptic segmentation, we divide it into two major groups: bottom-up and top-down.

Top-down. Top-down panoptic segmentation approaches mostly rely on region-based proposal to generate overlapping instance segmentation results, and then apply heuristic post-processing fusion methods to solve the problem of pixel overlapping. For example, AUNet [25] applies an attention module to guide the fusion of thing and stuff, and UPSNet [38] uses a parameter-free panoptic head and an additional unknown category channel to resolve instance mask overlap. BANet [6] solves instance pixel conflict by measuring the cosine similarity between pixel RGB value and the average RGB value of the individual instance. SOGNet [40] proposes to construct a relationship matrix between instances according to objects category, geometric and appearance features, and encode the occlusion relationship through a relationship matrix.

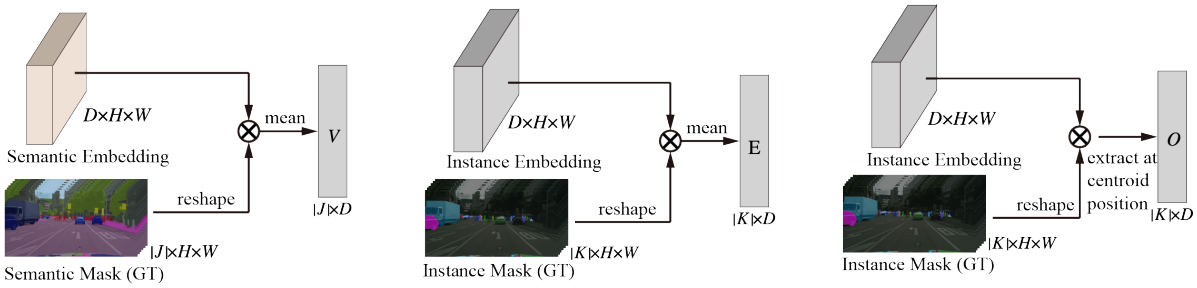


Figure 3. Three scenarios of Extract and Summary Module (ESM). (a) is used to extract $|J|$ semantic embeddings, where J is semantic category set of current input. (b) is used to extract $|K|$ instance embeddings, where K is instance ID set of current input. (c) applies K centroid positions of instances in current input to extract $|K|$ instance-centroid embeddings.

Bottom-up. The bottom-up panoptic segmentation method is region-proposal-free but cuts instances from semantic segmentation results of *thing* through designed instance pixel assembly clues, like pixel-pair affinity pyramid used in SSAP [13], regressed pixel offsets to instance centroid in PanopticDeeplab [7, 39], the edge map of the instance in Kirillov *et al.* [19], diatance to instance's boundary in DWT [2]. These methods [20, 2, 26, 8, 44, 30, 19, 34] leverage the designed intermediate results to assemble final instance segmentation. Some researchers argue that the results obtained with such intermediate tasks are sub-optimal. Newell *et al.* [31] optimizes at the high-dimensional embedding level, intervenes instance features in advance through discriminative loss function [3, 12], and uses embedding to assemble instance pixels. On the basis of this, we enhance semantic embedding and instance embedding by constraining interaction process between these two embeddings. Additionally, we compress representation of whole instance to its centroid position by centroid embedding regularization. As a result, our method quickly and efficiently assembles instance segmentation from semantic segmentation by measuring the distance between pixel instance embedding and centroid representation without any other cluster methods [1, 42], used in [3, 8, 29] which greatly simplifies the framework of panoptic segmentation. In order to obtain instance centroid, we need to model instance as a point.

Object As a Point. Modeling an object as a point has been widely used in object detection and tracking as well as panoptic segmentation. Cornernet [21] detects the paired corner and groups them by embedding. Zhou *et al.* [45] and Duan *et al.* [11] directly model object as a point for object detection. FairMot [43] represents pedestrians as points for multi-object tracking. DeeperLab [39] uses corners of instance bounding-box and instance centroid to group instance pixels. PanopticDeeplab [7] generates semantic segmentation result and cut the *thing* pixels by learning offset to the detected instance centroid. Following the same idea of instance modeling, we represent instances as points to obtain centroid embedding, which is employed to group *thing* pixels into different instances. Different from class-agnostic center representation, we represent the instance as a point with adaptive-scale [30, 45] and known-category.

Embedding Augmentation. Because of the obvious commonalities between instance segmentation and corresponding semantic segmentation, effectively modeling the internal interaction between the two has attracted wide attention from researchers. TASCNet [23] obtains better panoptic segmentation results by maintaining regroup-consistency between *thing* and *stuff*. BGRNet [37] enhances the embedding of instance branches and semantic branches by constructing bidirectional graphs. BANet [6] employs designed embedding aggregation policy to model the intrinsic interaction between semantic segmentation and instance segmentation. In order to make two branches of panoptic segmentation compatible with each other, Auto-Panoptic [41] unifies panoptic segmentation pipeline based on the prevailing one-shot Network Architecture Search paradigm. Different from the above methods, the interactive mechanism of instance embedding and semantic embedding we proposed has explicit constraints on embedding level, ensuring the commonality and individuality of instance embedding and semantic embedding.

3. Embedding Interaction Panoptic Segmentation

In this section, we reviewed preliminary of segmentation based on metric learning in background and then we will intruduce proposed embedding interaction network (EINet).

3.1. Problem Setting

Given a dataset, we assume the class set of *stuff* and *thing* on whole dataset are C^{st} and C^{th} and the total semantic category of dataset is $|C| = |C^{st}| + |C^{th}|$. Given an input image $P \in \mathbb{R}^{H\times W\times 3}$, the set of instances is K and the set of semantic category is $J \subset C$. We obtain semantic embedding $S \in \mathbb{R}^{H\times W\times D}$ and instance embedding $I \in \mathbb{R}^{H\times W\times D}$ of current input through multi-branch network. Through extract and summary module (ESM) in Fig.3, we have instance average-embedding $E \in \mathbb{R}^{|K|\times D}$

and semantic average-embedding $V \in \mathbb{R}^{|J|\times D}$. We learns a dictionary of semantic center $Q \in \mathbb{R}^{|C|\times D}$ after training with embedding interaction, which is initialized by semantic average-embedding V and is maintained on the whole dataset and covers *stuff* and *thing*. Additionally, we represent $O \in \mathbb{R}^{|K|\times D}$ as dictionary of instance centroid embedding, which is learned from E to represent instance. D is the dimension of embedding.

3.2. Background

Embedding Regularization with Metric Learning. Similiar to [16, 3], we have a quick definition of semantic center regularization and instance average-embedding regularization $\mathcal{R}_{reg}$ as following:

$$\begin{aligned}\mathcal{R}_{reg} = &\sum_{j}^{J\subset C}\frac{1}{N_j}[\sum_{y_i=j} D_e(S_i,Q_j) - \sum_{y_i\neq j} D_e(S_i,Q_j) + m]_+ \\ &+ \frac{1}{|K|}\frac{1}{|K|-1}\sum_{\substack{k}}^{K}\sum_{\substack{m\\k\neq m}}^{K}[2\sigma_d - ||E_k - E_m||]_+^2, \\ &+ \frac{1}{|K|}\sum_{k}^{K}\frac{1}{N_k}\sum_{t_i=k}[||I_i - E_k|| - \sigma_v]_+^2\end{aligned} \tag{1}$$

where $S_i \in \mathbb{R}^D$ represents semantic embedding of pixel i, Q_j is semantic center embedding of category j queried from dictionary of semantic center Q. $D_e(a,b) = ||\phi(a) - \phi(b)||_2$, where $\phi(\cdot)$ is L2 normalization. y_i is semantic category of pixel i, N_j is the numbers of pixel belonging to category j, t_i is instance ID of pixel i, and N_k is the number of pixel belonging to instance k. J and K are semantic category set and instance ID set of current input. I_i denotes instance embedding of pixel i, and E_k, E_m are instance average-embedding of instance k and m. $\{m, \sigma_v, \sigma_d\}$ is similiar to [16, 3], which is set as $\{5, 0.5, 1.5\}$.

Task-based Loss Function. When we obtain semantic embedding S, we use weighted bootstrapped cross entropy loss $\mathcal{L}_{sem}$ which is proposed in [7, 39] to optimize semantic segmentation tasks. The difference is that our semantic embedding S has been calibrated by Eq.4 below. Similiar to Object-as-point [45], we use $\mathcal{L}_{hm}$ to optimize $|C^{th}|$ channel adaptive-scale centroid heatmap with obtained instance embedding I, while we augmented instance embedding through subsequent regularization in Eq.7.

3.3. Embedding Interaction Network

Although we have already performed semantic segmentation and instance segmentation by modelling semantic center Q and instance average-embedding E in Eq. 1 and Eq. 2 separately, we were motivated by the naturally intrinsic relationship (the commonality and specificity of appearance, the subordination of category) between semantic embedding and instance embedding. Thus we propose the following embedding interaction mechanism to share or expand the association between semantic embedding and instance embedding.

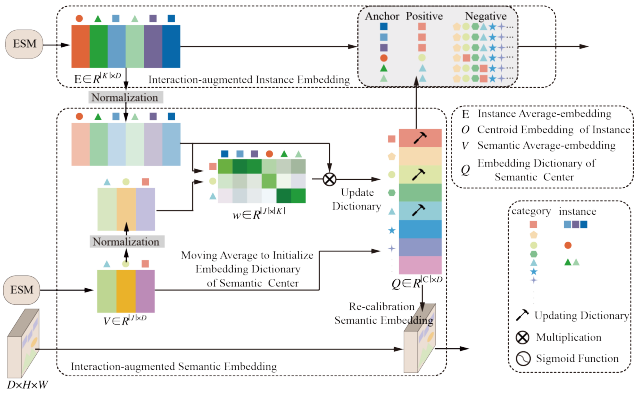


Figure 4. Details of the embedding interaction process. ω is matrix of correlation coefficient between normalized E and normalized V. Initializing with V, embedding dictionary of semantic center Q is maintained on the whole dataset, and Q is updated by the interaction with instance embedding.

Interaction-augmented Semantic Embedding. Semantic average-embedding V is short of shape prior for semantic pixels, while instance average-embedding E contains more appearance feature. We propose to augment sementic embedding with instance embedding to generalize semantic center embedding Q of the whole dataset. Given instance average-embedding dictionary E and semantic average-embedding V of current input, the correlation coefficient between E_k and V_j is calculated as follows:

$$\omega_{k,j} = \frac{\exp(D_e(E_k, V_j))}{\sum_j^C \exp(D_e(E_k, V_j))}. \tag{2}$$

Then update method for interaction-augmented semantic embedding is defined as follows:

$$Q_j = (1-\mu)\sum_{k}^{K}\omega_{k,j}^T \cdot \phi(E_k) + \mu Q_j, \tag{3}$$

where μ represents acceptance of semantic center for instance specificity, and ϕ denotes L2 normalization. We set μ as 0.9 in our implementation.

After enhancing semantic center embedding Q with instance average-embedding E, we re-calibrate Semantic Embedding S with updated semantic center in Q by:

$$S_i = \begin{cases} Q_j, & if \quad y_i = j \; and \; j \subset C^{th}, \\ S_i, & j \subset C^{st}, \end{cases} \tag{4}$$

where y_i is semantic category of pixel i. With calibrated semantic-embedding S, we optimize semantic segmentation results with general loss function $\mathcal{L}_{sem}$ in Sec.3.2, and we can not only optimize semantic segmentation results but also the dictionary of semantic center Q in the end-to-end training process.

Interaction-augmented Instance Embedding. Multiple instances with the same semantic category together constitute the semantic segmentation of the category, and the consistency between semantic segmentation results assembled by instance as well as semantic segmentation results directly obtained by semantic branch will benefit instance segmentation. Because this kind of consistency takes into account both local (instance level) and global (semantic level) information.

The normalized semantic embedding is distributed on the unit hypersphere, while the instance embedding is not normalized. We apply cosine similarity to measure the distance between instance average-embedding E_k of instance k and corresponding semantic center $Q_{g(k)}$. Specifically, we require the cosine similarity between instance average-embedding and corresponding semantic center to be as small as possible, and the cosine angle with other semantic center to be as large as possible. Given semantic center Q and instance average-embedding E of current input, positive and negative samples of instance k are defined as follows:

$$d_p^k = D_c(E_k, \phi(Q_{g_{(k)}})),$$
$$d_n^{k,j} = \begin{cases} D_c(E_k, \phi(Q_j)), & j \neq g_{(k)}, \\ 1, & j = g_{(k)}, \end{cases} \tag{5}$$

where $g_{(k)}$ is semantic category of instance k, $D_c(a, b)$ represents calibrated cosine similarity of vectors, whose range is $[0, 1]$ and 1 means two vectors point to the same.

Given a pair of positive sample and negative sample, per instance-semantic consistency is calculated as:

$$\ell_{k,j} = -log(1 - \frac{[d_n^{k,j} - d_p^k + \alpha]_+}{1 + \alpha}) \tag{6}$$

where soft margin α is set as 0.15 in our experiments, which is similar to [14]. Therefore, instance average-embedding is augmented by following regularization:

$$\mathcal{R}_{ins} = \frac{1}{|K|} \sum_{k}^{K} \sum_{j}^{|J|} \ell_{k,j}. \tag{7}$$

Instance Centroid Embedding Learning. We have model the instance as a scale-adaptive point through $\mathcal{L}_{hm}$ in Sec.3.2, so we can locate the centroid location of instance with the help of output heatmap, and then extract instance centroid embedding through ESM module in Fig.3.(c) to represent the whole instance. Although instance average-embedding E is already the weighted-aggregation of whole pixel embedding for each instance, we still need squeeze the instance average-embedding to instance centroid. Because we don't have mask of instance in inference process, we can't calculate E in inference process. Therefore, if we want to extract instance embedding in inference process, we need learn instance centroid embedding as following:

$$\mathcal{L}_{cen} = \frac{1}{|K|} \sum_{k}^{K} ||E_k - O_k||^2, \tag{8}$$

where instance average-embedding E is fetched by ESM module in Fig.3.(a). O_k is extracted by centroid location of instance k from instance embedding I, which construct the dictionary of instance centroid embedding O for following inference process. If we denote mask of instance k is M_k, then E_k is calculated by $E_k = sigmoid(M_k \bigotimes att_{ins}) \bigodot (M_k \bigotimes I)$, where att_{ins} is attention map for instance, $\bigotimes$ represents element-wise multiply and $\bigodot$ is element-wise multiply as well as plus.

3.4. Inference Process

Generate Semantic Segmentation Results. We can obtain semantic segmentation results through semantic segmentation branch for all pixels. N^{st} and N^{th} represent category set of *stuff* and *thing* of current input.

Split Instance Segmentation Results. In inference process, we obtained augmented instance-embedding I and C^{th}-channel adaptive-scale centroid heatmap.

We first perform a maxpooling-based non-maximum suppression (NMS) [45] and peak search with a hard threshold on adaptive-scale centroid heatmap, which aims to locate centroid of instance on each channel.

After that, we keep top-k instance location on each channel with high confidence scores and perform de-conflict for pixel through $\{l : \{i|top_k(p_i > threshold)\} \cap \{i|y_i = l\}\}$, where p_i is confidence score of pixel i and y_i represents semantic category of pixel i. $l \in I^{th}$ represents a *thing* semantic category, where I^{th} denotes category set of *thing* for current input, which obtained by the channel information of adaptive-scale centroid heatmap. In our implementation, we use maxpooling with kernel size 8, hard-threshold 0.3, and k=30.

Then, we extract $|I^{th}| \times$ k instance centroid embeddings from learned instance-embedding I, which construct the dictionary of instance centroid embedding O.

Finally, we split instance segmentation results from semantic segmentation results, whose semantic category is in set $S^{th} = N^{th} \cap I^{th}$. Specific, given pixel set $\{i|y_i = j\}$ with $j \in S^{th}$ semantic category and sub-dictionary $\{\{\texttt{key} : O_k\}|y_{key} = j\}$, the instance ID of pixel in set $\{i|y_i = j\}$ can be obtained through:

$$ID = \underset{k}{argmin} \, ||I_i - O_k||_2, \tag{9}$$

where y_{key} represents semantic category of instance key.

4. Experiments

Cityscapes [9]. This dataset consists of 2975, 500, and 1525 traffic-related images for training, validation, and testing, respectively. The label includes 11 *stuff* classes and 8 *thing* classes with instance level annotation.

COCO [4]. The datasets contains 118K, 5K, and 20K images for training, validation and testing, respectively. It contains 80 *thing* classes and 53 *stuff* classes totally.

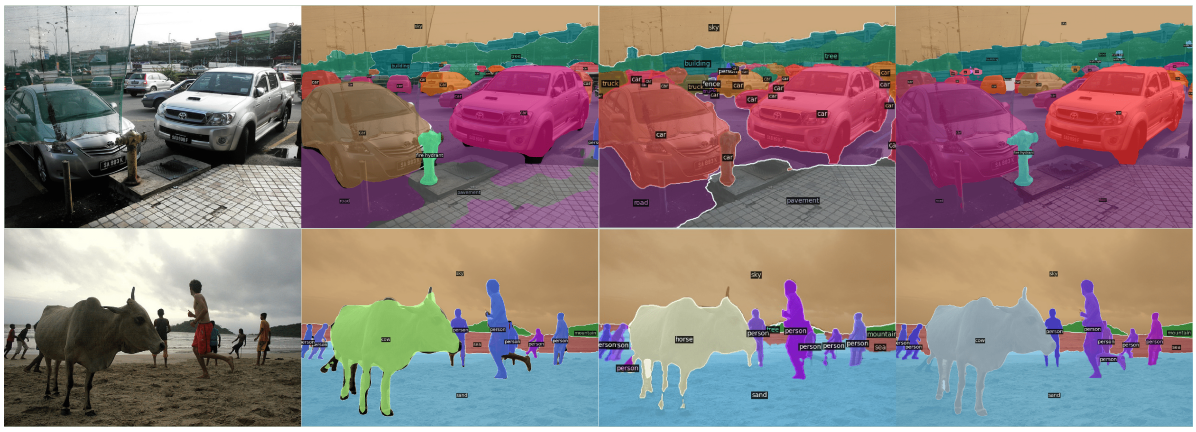


Figure 5. Visualization of panoptic segmentation on COCO with ResNet50. Left to right: input image, results by Detectron2[Panoptic-FPN] [36, 17], results by PanopticDeeplab [7], and EINet (**ours**). More qualitative analysis results of different datasets can be found in the supplement material.

Mapillary Vista [28]. A large scale traffic related dataset containing 18K, 2K and 5K images for training, validation and testing respectively. It contains 37 *thing* classes and 28 *stuff* classes, with image resolutions ranging from 1024×768 to more than 4000×6000.

Experimental Setup. All our models were trained using PyTorch on 8 Tesla V100 GPUs. In training process, we adopt a similiar training protocol as in [7]. Particularly, we perform random scale data, random gaussian blur and random CLAHE data augmentations during training. We train our model on Cityscapes dataset with whole image (*i.e.*, crop image equal to 1025×2049) with batch size 16, and we resize the images in COCO dataset to 800 pixels at the longest side and train the models with crop size 800×800 with batch size 32. Additionally, we use CosineAnnealingWarmRestarts with $T_0 = 5$, $T_{mult} = 2$ learning rate policy and we apply AdamW optimizer with an initial learning rate of 0.001. We set the training iteraction to 120K, 450K for Cityscapes and COCO, resetively. In evaluation process, because PQ is sensitive to noise segmentation, we re-assign to 'VOID' label all 'stuff' segments whose areas are smaller than a threshold given in [7]. When calculating AP for evaluation of instance segmentation, we also employ $Score(Objectness) \times Score(SemanticClass)$ to rank predictions, where $Score(Objectness)$ is objectness probability obtained from the centroid of class-known and adaptive-scale heatmap (do sigmoid channel-by-channel). And $Score\ (SemanticClass)$ is obtained from the average of semantic segmentation predictions within the predicted mask region, which is the same as [7]. We also adopt test time augmentation (TTA) like multi-scale inference (scales equal to 0.5,0.75,1,1.25,1.5,1.75,2 for Cityscapes and 0.5,0.75,1,1.25,1.5 for COCO) and horizontal flipped inputs, to further improve the performance. Our setting with best performance is to apply Xception-71 as backbone, which provides comparable results with other state-of-the-art methods. And the last 50K of 450K iteration are used to fine-tune the network using images with 1025×1025 resolution on COCO dataset. For the setup in ablation experimental groups, unless specified, the ResNet-50 with a shorter training period is employed.

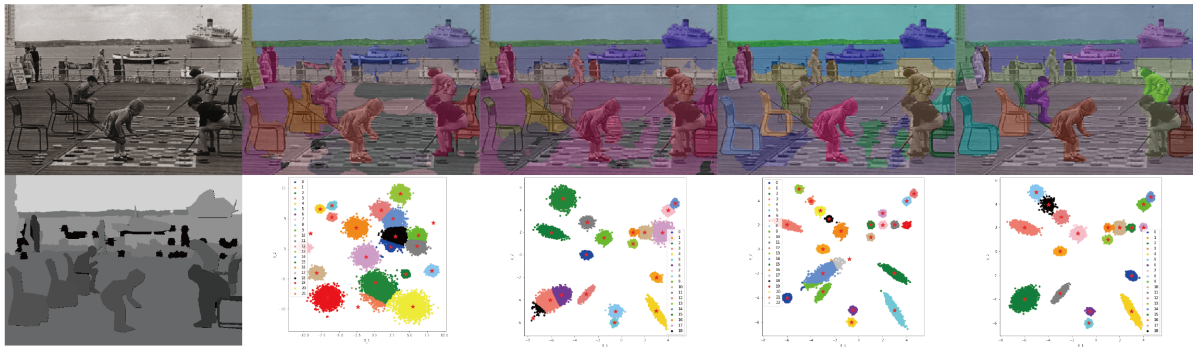
Training Objective. In proposed embedding interaction mechanism, we have regularization $\mathcal{R}_{ins}$ to augment instance embedding, and we have L_{cen} to learn instance centroid embedding. We also apply augmented semantic embedding to perform semantic segmentation, which is optimized through $\mathcal{L}_{reg}$. Additionally, Sec.3.2 provides $\mathcal{R}_{reg}$ to learn discriminative instance and semantic embedding, and provides $\mathcal{L}_{hm}$ to learn category-known and adaptive-scale heatmap for instances in *thing*. Therefore, the training objective is calculated as:

$$L = \lambda_e \mathcal{R}_{ins} + \lambda_c \mathcal{L}_{cen} + \lambda_s \mathcal{L}_{sem} + \lambda_r \mathcal{R}_{reg} + \lambda_h \mathcal{L}_{hm}. \tag{10}$$

For the training objective in Eq.10, we set $\{\lambda_e, \lambda_c, \lambda_r\}$ as $\{0.01, 10, 0.01\}$. Similiar to [7], we set λ_{sem}= 3 for pixels belonging to an area smaller than 4096 and λ_{sem}=1 elsewhere. And we set λ_h=200 for scale-aware center heatmap regression [45] to make sure the losses are in the similar magnitude.

4.1. Results

COCO ValSet. As visualization in Fig. 4, compared with the top-down method Detectron2 [36] and bottom-up method PanopticDeeplab [7], our EINet performed much better on foreground and background segmentation. In Tab. 1, it is seen that with left-right flip and multi-scale inference, EINet obtained the best bottom-up panoptic segmentation score with 43.6%PQ. With left-right flip and multi-scale data augmentation, EINet performed better than the best bottom-up method Panoptic Deeplab by 3.5%PQ, and the performance was comparabble to many top-down methods.



(a) (b) (c) (d) (e)

Figure 6. Visualization results of panoptic segmentation and learned embedding on COCO dataset. From top to bottom:(a) input, panoptic segmentation ground truth. (b) Panoptic segmentation of EINet only with **CenReg**, mean-shift clustering results of learned embedding. (c) Panoptic segmentation of EINet with **CenReg** and **SemAug**, mean-shift clustering results of learned embedding. (d) Panoptic segmentation of EINet with **CenReg** and **InsAug**, mean-shift clustering results of learned embedding.(e) Panoptic segmentation of EINet with **CenReg**, **InsAug** and **SemAug**, mean-shift clustering results of learned embedding.

Method	Backbone	Flip	M.S.	PQ	PQ^{Th}	PQ^{St}	PQ[test]
AUNet [25]	ResNet50			39.6	49.1	25.2	46.5
PanopticFPN [17]	ResNet101-FPN			40.3	47.5	29.5	40.9
AdaptIS [33]	ResNeXt101	✓		42.3	49.2	31.8	42.8
UPSNet [38]	ResNet50		✓	43.2	49.1	34.1	46.6
SOGNet [40]	ResNet50			43.7	50.6	33.2	47.8
BGRNet [37]	ResNet50			43.2	49.8	33.4	_
BANet [6]	ResNet50-FPN*	✓	✓	43.0	50.5	31.8	47.3
OCFusion [22]	ResNeXt101*		✓	46.3	53.5	35.4	46.7
DeeperLab [39]	Xception-71			33.8	_	_	34.3
SSAP [13]	ResNet101	✓	✓	36.5	_	_	36.9
PCV [35]	ResNet50	✓	✓	37.5	40.0	33.7	37.7
PanopticDeeplab [7]	Xception-71			39.7	43.9	33.2	_
PanopticDeeplab [7]	Xception-71	✓	✓	41.2	44.9	35.7	41.4
EINet(ours)	Xception-71			43.6	49.6	34.5	43.7
EINet(ours)	Xception-71	✓	✓	44.7	50.7	35.7	45.3

Table 1. COCO dataset. **Flip**: Applying horizontal flipped inputs. **M.S.**: Multi-scale inputs. $*$ uses deformable convolution. PQ[test]: PQ(%) in COCO test-dev dataset.

COCO TestSet. It can be seen from from Tab. 1 that the proposed EINet outperformed the best bottom-up method Panoptic Deeplab by 3.9%PQ, which is comparable to most proposal-based methods without heavier backbone, deformable convolution layers or even longer training schedule.

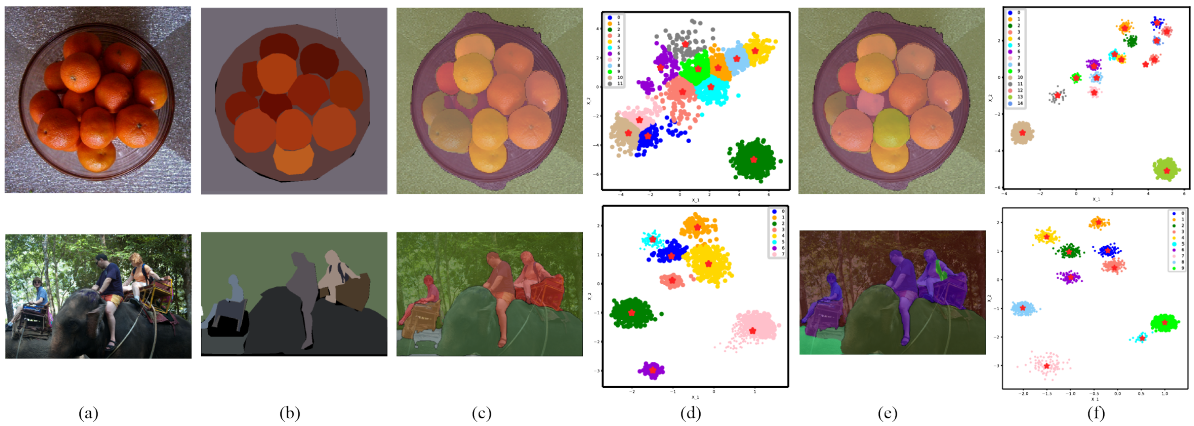
Cityscapes. We report our results on the Cityscapes validation dataset in Tab. 2. With multi-scale inputs and horizontal flip, EINet outperformed the best bottom-up approach, Panoptic Deeplab [7], by 0.8%PQ, 0.6%AP, and 0.5%mIOU. Additionally, EINet outperformed the best region proposal-based approach, AdaptIS [33], by 1.8%PQ, 0.1%AP and 2.0%mIOU, while EINet applies a much lighter backbone.

4.2. Ablation Study

Effect of Explicit Embedding Interaction. We conducted ablation studies on Cityscapes validation dataset, as shown in Tab. 3. Without interaction between instance embedding and semantic embedding, our baseline obtained low PQ and AP. After adding the interaction, PQ, AP, and mIOU improved by 2.9%, 6.1%, and 1.5% respectively. In Fig.5, we also perform a dimension reduction and clustering analysis of the learned embeddings in COCO dataset. The results in Fig.5 show that the explicit embedding interaction constraint effectively increases the inter-class distance of the embeddings and reduces the inter-class variance of the

Method	Backbone	Flip	M.S.	PQ	PQ^{Th}	PQ^{St}	AP	mIOU
w/o Extra Data								
Kirillov *etal.* [18]	Res50+X101			61.2	54.0	66.4	36.4	_
Li *etal.* [24]	Res101			53.8	42.5	62.1	_	79.8
DWT [2]	VGG16			_	_	_	21.2	_
SGN [26]	VGG16			_	_	_	29.2	_
PanopticFPN [17]	Res101FPN			58.1	52.0	62.5	33.0	75.7
UPSNet [38]	ResNet50		✓	60.1	55.0	63.7	33.3	76.8
AUNet [25]	Res101FPN			59.0	54.8	62.1	34.4	75.6
AdaptIS [33]	ResXt101			62.0	58.7	64.4	36.3	79.2
SOGNet [40]	Res50			60.0	56.7	62.5	_	_
Seamless [32]	Res50			60.2	55.6	63.6	33.3	74.9
OCFusion [22]	Res50*		✓	60.2	54.0	64.7	_	_
DeeperLab [39]	Xception71			56.5	_	_	_	_
SSAP [13]	Res101	✓	✓	61.1	55.0	65.5	37.3	_
PCV [35]	Res50	✓	✓	54.2	47.8	58.9	_	74.1
PanopticDeeplab [7]	Xception71			63.0	_	_	30.3	80.5
PanopticDeeplab [7]	Xception71	✓	✓	64.1	_	_	38.5	81.5
EINet(ours)	ResNet50			63.0	56.9	67.4	35.2	79.9
EINet(ours)	ResNet50	✓	✓	63.8	57.0	68.7	36.4	81.2
EINet(ours)	Xception-71			64.7	57.4	70.0	39.1	81.7
EINet(ours)	Xception-71	✓	✓	**65.6**	**57.5**	**71.5**	**39.5**	**82.6**
w/ Extra Data								
TASCNet [23]	Res50		✓	60.4	56.1	63.3	39.1	78.7
UPSNet [38]	Res101		✓	61.8	57.6	64.8	39.0	79.2
PanopticDeeplab [7]	Xception71			65.3	_	_	38.8	82.5
PanopticDeeplab [7]	Xception71	✓	✓	67.0	_	_	42.5	83.1
EINet(ours)+MV	ResNet50			64.2	58.5	68.3	38.6	81.5
EINet(ours)+MV	ResNet50	✓	✓	65.4	58.9	70.1	39.2	81.9
EINet(ours)+MV	Xception-71			66.7	60.3	71.4	41.7	83.1
EINet(ours)+MV	Xception-71	✓	✓	**67.8**	**60.5**	**73.1**	**43.1**	**83.6**

Table 2. Cityscapes val dataset. **Flip**: Applying horizontal flipped inputs. **M.S.**: Multi-scale inputs. $*$uses deformable convolution. MV means extra dataset Mapillary Vistas [28].



(a) (b) (c) (d) (e) (f)

Figure 7. The effect of centroid regularization. (a) is input image. (b) is panoptic ground truth of input. (c) is output of EINet **without** centroid regularization. (d) is clustering of learned embedding **without** centroid regularization. (e) is output of EINet **with** centroid. (f) is clustering of learned embedding **with** centroid regularization.

semantic embeddings as well as the variance of the intra-instance embeddings. Although the clustering centers of instance individuals with the same semantic category are different, they have nearly the same orientation, and this orientation reflects the embedding direction of the semantic category to which these instance individuals belong.

InsAug	SemAug	CenReg	PQ(%)	AP(%)	mIOU(%)
			57.4	22.3	77.0
✓			57.5	24.5	77.2
✓		✓	57.9	28.1	77.4
		✓	58.4	27.5	77.4
	✓		60.5	27.2	78.4
✓	✓		60.3	28.4	78.5
	✓	✓	60.7	31.1	78.6
✓	✓	✓	**61.9**	**33.5**	**79.4**

Table 3. Ablaton studies on Cityscaes val dataset with ResNet50 and Embedding$\in \mathbb{R}^{128}$. **InsAug**: Interaction-augmented constriant for instance embedding (Eq. 7). **CenReg**: Instance centroid representation regularization (Eq. 8). **SemAug**: Augmenting semantic embedding through interaction with instance (Eq. 2,3,4).

Effect of Centroid Regularization. The effect of centroid embedding regularization is also recorded in Tab.3. When the constraint of centroid regularization for instance embedding was applied, PQ, AP, and mIOU further increased by 1.6%, 5.1%, and 0.9%, respectively. From the results of embedding clustering in Fig.6, the use of instance centroid regularization effectively reduces the variance of pixel embedding within instances and makes the pixel embedding of different instances more distinguishable.

Dimension of Embedding. When the corresponding regularization constraints and the interaction mechanisms were applied to semantic embedding and instance embedding, it is observed that in general, PQ, AP, and mIOU increase with the dimension of embedding. We obtained the best results with an embedding dimension of 256.

Acceptance Rate of Instances Average-embedding. Parameter μ in Eq. (3) represents the acceptance rate of each semantic center to all instance embeddings in that category. It can be seen from Tab. 5 that, the best performance was achieved when μ=0.9, and improving instance embedding can clearly augment semantic center embedding.

D_emb	PQ(%)	SQ(%)	RQ(%)	AP(%)	mIOU(%)
32	51.9	77.8	64.5	9.82	78.4
64	57.3	79.2	71.0	27.7	76.9
96	59.1	79.7	73.0	28.9	78.2
128	61.9	81.5	74.9	33.5	79.4
256	**63.0**	**82.9**	**76.2**	**35.2**	**79.9**
384	62.7	82.9	75.8	34.7	79.8

Table 4. Analysis of instance Embedding and semantic embedding dimension on Cityscapes val dataset with InsAug, SemAug and CenReg. **D_emb**: Dimension of embedding.

μ	PQ(%)	PQ^{Th}(%)	PQ^{St}(%)	AP(%)	mIOU(%)
0.6	55.2	52.9	56.9	22.6	76.4
0.8	58.1	53.1	61.8	31.1	77.9
0.85	62.7	56.6	67.1	34.4	79.7
0.9	**63.0**	**56.9**	**67.4**	**35.2**	**79.9**
0.99	61.5	55.4	65.9	33.1	79.4
0.999	61.6	55.9	65.7	33.4	79.5

Table 5. Analysis of instance average-embedding acceptance rate on Cityscapes val dataset with Embedding $\in \mathbb{R}^{256}$.

5. Conclusion

The proposed EINet jointly optimize instance embedding and semantic embedding through the designed interaction mechanism in an end-to-end way, which models the intrinsic interaction between tightly coupled two branches of panoptic segmentation. Through a maintained panoptic embedding dictionary, the panoptic segmentation can be completed efficiently by the embedding distance metric. EINet inherits the simplicity of the bottom-up panoptic segmentation framework and does not need other auxiliary tasks except for regression of instance centroid.

References

[1] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. *SIGMOD Rec.*, 28(2):49–60, June 1999. 3

[2] Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3, 7

[3] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function, 2017. 3, 4

[4] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 5

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834—848, April 2018. 1

[6] Yifeng Chen, Guangchen Lin, Songyuan Li, Bourahla Omar, Yiming Wu, Fangfang Wang, Junyi Feng, Mingliang Xu, and Xi Li. Banet: Bidirectional aggregation network with occlusion handling for panoptic segmentation. 2020. 1, 2, 3, 7

[7] Bowen Cheng, Maxwell D. Collins, Yukun Zhu, Ting Liu, Thomas S. Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *CVPR2020*. 1, 2, 3, 4, 6, 7

[8] R. Cipolla, Y. Gal, and A. Kendall. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7482–7491, 2018. 3

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2, 5

[10] Daan de Geus, Panagiotis Meletis, and Gijs Dubbelman. Panoptic segmentation with a joint semantic and instance segmentation network, 2019. 1

[11] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection, 2019. 3

[12] Alireza Fathi, Zbigniew Wojna, Vivek Rathod, Peng Wang, Hyun Oh Song, Sergio Guadarrama, and Kevin P. Murphy. Semantic instance segmentation via deep metric learning, 2017. 3

[13] Naiyu Gao, Yanhu Shan, Yupei Wang, Xin Zhao, Yinan Yu, Ming Yang, and Kaiqi Huang. Ssap: Single-shot instance segmentation with affinity pyramid. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1, 2, 3, 7

[14] Naiyu Gao, Yanhu Shan, X. Zhao, and Kai-Qi Huang. Learning category- and instance-aware pixel embedding for fast panoptic segmentation. *ArXiv*, abs/2009.13342, 2020. 5

[15] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1

[16] Tong He, Dong Gong, Zhi Tian, and Chunhua Shen. Learning and memorizing representative prototypes for 3d point cloud semantic and instance segmentation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 564–580, Cham, 2020. Springer International Publishing. 4

[17] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollar. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 6, 7

[18] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollar. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 7

[19] Alexander Kirillov, Evgeny Levinkov, Bjoern Andres, Bogdan Savchynskyy, and Carsten Rother. Instancecut: From edges to instances with multicut. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[20] Victor Kulikov, Victor Yurchenko, and Victor Lempitsky. Instance segmentation by deep coloring, 2018. 3

[21] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. *International Journal of Computer Vision*, 2018. 3

[22] Justin Lazarow, Kwonjoon Lee, Kunyu Shi, and Zhuowen Tu. Learning instance occlusion for panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 7

[23] Jie Li, Allan Raventos, Arjun Bhargava, Takaaki Tagawa, and Adrien Gaidon. Learning to fuse things and stuff, 2019. 1, 2, 3, 7

[24] Qizhu Li, Anurag Arnab, and Philip H.S. Torr. Weakly- and semi-supervised panoptic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 7

[25] Yanwei Li, Xinze Chen, Zheng Zhu, Lingxi Xie, Guan Huang, Dalong Du, and Xingang Wang. Attention-guided unified network for panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2, 7

[26] Shu Liu, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. Sgn: Sequential grouping networks for instance segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 3, 7

[27] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1

[28] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE*

International Conference on Computer Vision (ICCV), Oct 2017. 6, 7

[29] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Fast scene understanding for autonomous driving, 2017. 3

[30] Davy Neven, Bert De Brabandere, Marc Proesmans, and Luc Van Gool. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[31] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2277–2287. Curran Associates, Inc., 2017. 2, 3

[32] Lorenzo Porzi, Samuel Rota Bulò, Aleksander Colovic, and Peter Kontschieder. Seamless scene segmentation, 2019. 1, 7

[33] Konstantin Sofiiuk, Olga Barinova, and Anton Konushin. Adaptis: Adaptive instance selection network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 7

[34] J. Uhrig, E. Rehder, B. Fröhlich, U. Franke, and T. Brox. Box2pix: Single-shot instance segmentation by assigning pixels to object boxes. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 292–299, 2018. 3

[35] Haochen Wang, Ruotian Luo, Michael Maire, and Greg Shakhnarovich. Pixel consensus voting for panoptic segmentation. 2020. 1, 7

[36] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 6

[37] Yangxin Wu, Gengwei Zhang, Yiming Gao, Xiajun Deng, Ke Gong, Xiaodan Liang, and Liang Lin. Bidirectional graph reasoning network for panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 3, 7

[38] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2, 7

[39] Tien-Ju Yang, Maxwell D. Collins, Yukun Zhu, Jyh-Jing Hwang, Ting Liu, Xiao Zhang, Vivienne Sze, George Papandreou, and Liang-Chieh Chen. Deeperlab: Single-shot image parser, 2019. 1, 2, 3, 4, 7

[40] Yibo Yang, Hongyang Li, Xia Li, Qijie Zhao, Jianlong Wu, and Zhouchen Lin. Sognet: Scene overlap graph network for panoptic segmentation. 2019. 2, 7

[41] Hang Xu Xiaodan Liang Yangxin Wu, Gengwei Zhang and Liang Lin. Auto-panoptic: Cooperative multi-componentarchitecture search for panoptic segmentation. In *Advances in Neural Information Processing Systems 33*. Curran Associates, Inc., 2020. 3

[42] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995. 3

[43] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *arXiv preprint arXiv:2004.01888*, 2020. 3

[44] Ziyu Zhang, Alexander G. Schwing, Sanja Fidler, and Raquel Urtasun. Monocular object instance segmentation and depth ordering with cnns. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015. 3

[45] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points, 2019. 3, 4, 5, 6