The Semantic Mutex Watershed for Efficient Bottom-Up Semantic Instance Segmentation

Steffen Wolf[1⋆], Yuyan Li[1⋆], Constantin Pape[1,2], Alberto Bailoni[1], Anna Kreshuk[2], and Fred A Hamprecht[1]

[1] HCI/IWR, Heidelberg University, Germany
<firstname>.<lastname>@iwr.uni-heidelberg.de
[2] EMBL, Heidelberg, Germany
<firstname>.<lastname>@embl.de

Abstract Semantic instance segmentation is the task of simultaneously partitioning an image into distinct segments while associating each pixel with a class label. In commonly used pipelines, segmentation and label assignment are solved separately since joint optimization is computationally expensive. We propose a greedy algorithm for joint graph partitioning and labeling derived from the efficient Mutex Watershed partitioning algorithm. It optimizes an objective function closely related to the Asymmetric Multiway Cut objective and empirically shows efficient scaling behavior. Due to the algorithm's efficiency it can operate directly on pixels without prior over-segmentation of the image into superpixels. We evaluate the performance on the Cityscapes dataset (2D urban scenes) and on a 3D microscopy volume. In urban scenes, the proposed algorithm combined with current deep neural networks outperforms the strong baseline of 'Panoptic Feature Pyramid Networks' by Kirillov *et al.* (2019). In the 3D electron microscopy images, we show explicitly that our joint formulation outperforms a separate optimization of the partitioning and labeling problems.

1 Introduction

Image segmentation literature distinguishes *semantic segmentation* - associating each pixel with a class label - and *instance segmentation*, i.e. detecting and segmenting individual objects while ignoring the background. The joint task of simultaneously assigning a class label to each pixel and grouping pixels to instances has been addressed under different names, including semantic instance segmentation, scene parsing [42], image parsing [43], holistic scene understanding [47] or instance-separating semantic segmentation [29]. Recently, a new metric and evaluation approach to such problems has been introduced under the name of *panoptic segmentation* [19].

From a graph theory perspective, semantic instance segmentation corresponds to the simultaneous partitioning and labeling of a graph. Most greedy

⋆ Authors contributed equally

graph partitioning algorithms are defined on graphs encoding attractive interactions only. Clusters are then formed through agglomeration or division until a user-defined termination criterion is met (often a threshold or a desired number of clusters). These algorithms perform pure instance segmentation. The semantic labels for the segmented instances need to be generated independently.

If repulsive - as well as attractive - forces are defined between the nodes of the graph, partitioning can be formulated as a Multicut problem [2]. In this formulation clusters emerge naturally without the need for a termination criterion. Furthermore, the Multicut problem can be extended to include the labeling of the graph, delivering a semantic instance segmentation from a joint optimization of partitioning and labeling [24].

We propose to solve the joint partitioning and labeling problem by an efficient algorithm which we term Semantic Mutex Watershed (SMWS), inspired by the Mutex Watershed [44]. In more detail, in this contribution we:

- propose a fast algorithm for joint graph partitioning and labeling
- prove that the algorithm (exactly) minimizes an objective function closely related to the Asymmetric Multiway Cut objective
- demonstrate competitive performance on natural and biological images.

2 Related Work

Semantic segmentation. State-of-the-art semantic segmentation algorithms are based on convolutional neural networks (CNNs) which are trained end-to-end. The networks commonly follow the design principles of image classification networks (*e.g.* [16,40,23]), replacing the fully connected layers at the end with convolutional layers to form a fully convolutional network [32]. This architecture can be further extended to include encoder-decoder paths [39], dilated or atrous convolutions [49,5] and pyramid pooling modules [6,50].

Instance segmentation. Many instance segmentation methods use a detection or a region proposal framework as their basis; object segmentation masks are then predicted inside region proposals. A cascade of multiple networks is employed by [11], each solving a specific subtask to find the instance labeling. Mask-RCNN [15] builds on the bounding box prediction capabilities of Faster-RCNN [38] to simultaneously produce masks and class predictions. An extension of this method with an additional semantic segmentation branch has been proposed in [18] as a single network for semantic instance segmentation.

In contrast to the region-based methods, proposal-free algorithms often start with a pixel-wise representation which is then clustered into instances [48,21,12]. Alternatively, the distance transform of instance masks can be predicted and clustered by thresholding [3].

Graph-based segmentation. Graph-based methods, used independently or in combination with machine learning on pixels, form another popular basis for image segmentation algorithms [13]. In this case, the graph is built from pixels or

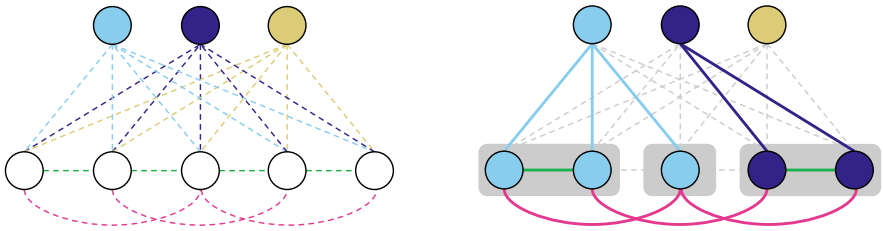

Figure 1: *Left:* An example of an extended graph. Nodes on the top are terminal nodes whereby each color represents a label class. The associated semantic edges are colored correspondingly. The internal nodes are on the bottom with attractive (green) and repulsive (red) edges between them. *Right:* Semantic instance segmentation. Edges that are part of the active set are shown in bold. Clusters are depicted in grey. Note that two adjacent nodes with the same label are not necessarily clustered together.

superpixels of the image and the instance segmentation problem becomes a graph partitioning problem. When the number of instances is not known in advance and repulsive interactions are present between the graph nodes, graph partitioning can in turn be formulated as a Multicut or correlation clustering problem [2]. This NP-hard problem can be solved reasonably fast for small problem sizes with integer linear programming solvers [1] or approximate algorithms [36,4]. A modified Multicut objective is introduced by [44] together with the Mutex Watershed - an efficient clustering algorithm for its optimization.
The Multicut objective can be extended to solve a joint graph partitioning and labeling problem [17]. One such extension is the Asymmetric Multiway Cut [24] that is used for simultaneous instance and semantic segmentation. This formulation has been applied to natural images by [20] and to biological images by [22].

The Node Labeling Multicut Problem (NLMP) [29] further generalizes this problem to larger feasible sets, extending the range of applications to human pose estimation and multiple object tracking. In practice, the computational complexity of the NLMP only allows for approximate solutions, possibly combined with reducing the problem size by over-segmentation into superpixels.
Similar to the semantic segmentation use case, CNNs can be used to predict pixel and superpixel affinities which serve as edge weights in the graph partitioning problem [28,33,31].

3 The Semantic Mutex Watershed

In this section, we introduce an extension to the Mutex Watershed algorithm for semantic instance segmentation. To this end, we build a graph of image pixels (voxels) or superpixels and formulate the semantic instance segmentation problem as the joint graph partitioning and labeling.

Semantic Mutex Watershed

SMWS$(\mathcal{G}(V, E'),\ w : E' \to \mathbb{R},$ boolean connect_all$)$**:**

$A^+ \leftarrow \emptyset; \quad A^- \leftarrow \emptyset$

for $(i, j) = e \in E'$ *in descending order of* $|w_e|$ **do**

if $e \in E^+$ **then**

if not mutex$(i, j; A^+, A^-)$

and not differentclass(i, j, A^+, A^S) **then**

if not connected$(i, j; A^+)$ **or** connect_all **then**

merge(i, j): $A^+ \leftarrow A^+ \cup e$

// merge i and j and inherit the mutual

// exclusions from the parent clusters

else if $e \in E^-$ **then**

if not connected$(i, j; A^+)$ **then**

addmutex(i, j): $A^- \leftarrow A^- \cup e$

// add mutual exclusion between i and j

else if $e \in E^S$ **then**

if *class*$(i, A^+, A^S) = \emptyset$ **or** *class*$(i, A^+, A^S) = l_j$ **then**

assignLabel(i, j): $A \leftarrow A \cup e$

return A

Algorithm 1: The Semantic Mutex Watershed algorithm. The differences to the Mutex Watershed are marked in blue.

Weighted graph with terminal nodes. To partition an undirected weighted graph $G = G(V, E, w)$ into instances Wolf *et al.* [44] separate the set of edges into two sets: *attractive edges* $E^+ = \{e \in E \,|\, w_e >= 0\}$ and *repulsive edges* $E^- = \{e \in E \,|\, w_e < 0\}$, based on their weight. These are used in the Mutex Watershed to find a graph partitioning. To model label assignments, we will augment this graph with additional nodes and edges and refer to V as *internal nodes* and edges $E = E^+ \cup E^-$ as *internal edges*.

Semantic instance segmentation is achieved by clustering the internal nodes and assigning a semantic label $l \in \{l_0, ..., l_k\}$ to each cluster. We extend G by k *terminal* nodes $\{t_0, ..., t_k\} \in T$ where each t_i is associated with a label l_i. Every internal node $v \in V$ is connected to every t by a weighted *semantic edge* $e \in E^S$. Here, a large *semantic weight* $w_{ut} \subseteq \mathbb{R}^+$ implies a strong association of internal node u with the label of the terminal node $t \in T$. The extended graph thus becomes $\mathcal{G}(V', E', w')$ with $V' = V \cup T,\ E' = E \cup E^S$ and $w' = w \cup \{w_{ut} | \forall t \in T,\ \forall u \in V\}$. Figure 1 shows an example of such an extended graph.

3.1 The Semantic Mutex Watershed Algorithm.

We will now extend the Mutex Watershed Algorithm to the extended graph $\mathcal{G}$ for joint graph partitioning and labeling. The algorithm finds a clustering and label assignment described by a set of *active* edges: $A \subseteq E'$ where $A^+ := A \cap E^+$, $A^- := A \cap E^-$ and $A^S := A \cap E^S$ encode clusters, mutual exclusions and label

assignments, respectively. For example, $(u, v) \in A^+$ assigns nodes u and v to the same cluster. Similarly, $(u, t_k) \in A^S$ assigns node u to class k. However, not all possible A represent a consistent partitioning and labeling. To ensure consistency, we will make the following definitions:

We define two internal nodes $i, j \in V$ as connected if they are connected by active attractive edges, i.e.

$$\forall i, j \in V : \tag{1}$$
$$\Pi_{i \to j} = \{\text{paths } \pi \text{ from } i \text{ to } j \text{ with } \pi \subseteq E'\} \tag{2}$$
$$\text{connected}(i, j; A^+) \Leftrightarrow \exists \text{ path } \pi \in \Pi_{i \to j} \text{ with } \pi \subseteq A^+ \tag{3}$$
$$\text{cluster}(i; A^+) = \{i\} \cup \{j \mid \text{connected}(i, j; A^+)\} \tag{4}$$

and the mutual exclusion between two nodes as

$$\text{mutex}(i, j; A^+, A^-) \Leftrightarrow \exists\, e = (k, l) \in A^- \text{ with} \tag{5}$$
$$k \in \text{cluster}(i; A^+) \text{ and} \tag{6}$$
$$l \in \text{cluster}(j; A^+) \text{ and} \tag{7}$$
$$\text{cluster}(i; A^+) \neq \text{cluster}(j; A^+) \tag{8}$$

Two nodes are thus mutual exclusive if they are connected by a path from i to j with exactly one repulsive edge.

Furthermore, a label l_j is assigned to a node i if this node is connected to the corresponding terminal node t_j by attractive and semantic edges:

$$\text{class}(i, A^+, A^S) = l_j \Leftrightarrow \exists\, \pi \in \Pi_{i \to j} \text{ with } \pi \subseteq A^+ \cup A^S. \tag{9}$$

For unlabeled nodes i, where $\text{class}(i, A^+, A^S) \neq c \quad \forall c \in \{l_0, ..., l_k\}$, we use the notation $\text{class}(i, A^+, A^S) = \emptyset$ and use it to define the following predicate

$$\text{differentclass}(i, j, A^+, A^S) \quad \Leftrightarrow \quad \text{class}(i, A^+, A^S) \neq \text{class}(j, A^+, A^S) \text{ and} \tag{10}$$
$$\text{class}(i, A^+, A^S) \neq \emptyset \text{ and} \tag{11}$$
$$\text{class}(j, A^+, A^S) \neq \emptyset \tag{12}$$

The graph partitioning assignments $A^+ \cup A^-$ must be chosen such that the clustering and labeling is consistent. This means:

1. Nodes engaged in a mutual exclusion constraint cannot be in the same cluster [44]:
$$\text{mutex}(i, j; A^+, A^-) \Rightarrow \text{ not connected}(i, j; A^+) \tag{13}$$
2. Nodes in the same cluster must have the same label, or equivalently:
$$\text{connected}(i, j; A^+) \Rightarrow \text{not differentclass}(i, j, A^+, A^S) \tag{14}$$

Algorithm. The Semantic Mutex Watershed algorithm is an extension of the Mutex Watershed algorithm introduced by Wolf *et al.* [44]. It augments the partitioning of the latter with a consistent labeling. The algorithm is shown in algorithm 1 with the additions to [44] highlighted. In the following we explain the syntax and procedure of the shown pseudocode.

For each edge $e \in E'$ it will be decided if it should be added to the active set A. The decisions are made in descending order of the absolute edge-weights and follow rules depending on the type of each edge:
Attractive edges: The edge is added if the incident nodes are not mutual exclusive and not labeled differently. We call this a merge because the two incident nodes will be connected afterwards.
Repulsive edges: The edge is added if the incident nodes are not connected.
Semantic edges: The edge is added if the node is either unlabeled or already has the same label as the edge's terminal node.

Note, that the set A never violates eqs. (13) and (14) during the procedure. Therefore, after following these rules, the set of attractive edges in the final set $A \cap E^+$ form clusters in the graph G, which are each connected to a single terminal node indicating the labeling. Figure 1(b) shows a simple example of such an active set. Note, that the Mutex Watershed algorithm is embedded in the Semantic Mutex Watershed for the special case when there are zero or one label ($|T| \in \{0, 1\}$).

Efficient Implementation with Maximum-Spanning-Trees. The SMWS is similar to the efficient Kruskal's maximum spanning tree algorithm [25] and can feasibly be applied to pixel-graphs of large images and even image volumes. Our implementation utilizes an efficient union-find data structure; mutual exclusions are realized through a hash table.

3.2 The Semantic Mutex Watershed Objective

The Semantic Mutex Watershed, introduced in the previous section, operates on a graph with terminal nodes identical to the graph for the Asymmetric Multiway Cut (AMWC) [24]. In this section we prove that the Semantic Mutex Watershed optimizes a precise objective and show how it relates to the Asymmetric Multiway Cut objective. To this end, we will extend the proof by [44] to the Semantic Mutex Watershed. Let us first recall their definitions of *dominant powers* and *conflicted cycles*.

Dominant power. Let $\mathcal{G} = (V', E', w)$ be an edge-weighted graph, with unique weights $w : E' \to \mathbb{R}$. We call $p \in \mathbb{N}^+$ a dominant power if:

$$|w_e|^p > \sum_{t \in E',\, w_t < w_e} |w_t|^p \qquad \forall e \in E', \tag{15}$$

Note that there exists a dominant power for any finite set of edges, since for any $e \in E$ we can divide (15) by w_e^p and observe that the normalized weights w_t^p / w_e^p (and any finite sum of these weights) converges to 0 when p tends to infinity.

Conflicted cycles. We call a cycle of $\mathcal{G}$ conflicted w.r.t. $(\mathcal{G}, w)$ if it contains precisely one repulsive edge $e \in E^-$, s.t. $w_e < 0$. We denote by $\mathcal{C}^-(\mathcal{G}, w) \subseteq \mathcal{C}(\mathcal{G}, w)$ the set of all conflicted cycles. Furthermore, given a set of edges $A \subseteq E$, we denote by $\mathcal{C}^-(A, \mathcal{G}, w) \subseteq \mathcal{C}^-(\mathcal{G}, w)$ the set of conflicted cycles involving only edges in A. If there are no conflicted cycles $\mathcal{C}^-(G, A, w) = \emptyset$ then A implies a consistent graph partitioning [26]. In other words, ensuring that there are no conflicted cycles ensures that two nodes that are mutual exclusive can not be connected.

Furthermore, we define the set $\mathcal{P}(A)$ of all paths π that connect two distinct terminal nodes through attractive and semantic edges:

$$\mathcal{P}(A) := \{\, \pi \,|\, \pi \in \Pi_{t \to t'},\, \pi \in A \cap (E^+ \cup E^S),\, t, t' \in T\,, t \neq t' \,\} \tag{16}$$

The algorithm must never connect two terminal nodes through such a path, thus we define the **label constraint** $\mathcal{P}(A) = \emptyset$. This ensures the consistency between the partitioning and labeling.

Lemma 1 (Optimality of the Semantic Mutex Watershed).
Let $\mathcal{G} = (V', E', w) = (V \cup T, E \cup E^S, w)$ be an edge-weighted graph extended by terminal nodes T, with unique weights $w' : E' \to \mathbb{R}$, $w_t > 0 \,\forall\, t \in T$ and $p \in \mathbb{R}^+$ a dominant power. The edge indicator given by the Semantic Mutex Watershed

$$x^{\mathbf{SMWS}} := \mathbb{1}$$

is the optimal solution to the integer linear program

$$\begin{aligned}
&\arg\min_{x \in \{0,1\}^{|E'|}} \sum_{e \in E'} |w_e|^p x_e && (17)\\
\text{s.t.}\quad &\mathcal{C}^-(G, A, w) = \emptyset, && (18)\\
&\mathcal{P}(A) = \emptyset, && (19)\\
\text{with}\quad &A := \{\, e \in E \mid x_e = 0 \,\}. && (20)
\end{aligned}$$

Proof. This proof is completely analogous to the optimality proof of the Mutex Watershed (see Theorems 4.1 in [44]) and even identical for $T = \emptyset$. The SMWS finds the optimal solution because it enjoys the properties *optimal substructure* and *greedy choice*. Showing the optimal substructure of the Mutex Watershed does not rely on the specific constraints in the ILP. Thus it can also be applied with the additional constraint in eq. (19), giving the ILP eqs. (17) to (20) optimal substructure.

In every iteration the SMWS adds the feasible edge e with the largest weight to the active set. Due to the dominant power, its energy contribution is larger than for any combination of edges e' with $w'_e < w_e$. Thus, SMWS has the greedy choice property [10]. It follows by induction that the SMWS algorithm finds the globally optimal solution to the SMWS objective.

Relation to the Asymmetric Multiway Cut. To understand the relation of the Semantic Mutex Watershed to the Asymmetric Multiway Cut we will transform the SMWS problem (eqs. (17) to (20)) into an ILP with the same minimal energy solution as the AMWC.

First, let us review the AMWC [24] as an ILP:

$$\underset{y\in\{0,1\}^{|E'|}}{\arg\min} \sum_{e\in E'} \operatorname{sign}(w_e)|w_e|^p y(x,e) \tag{21}$$

$$\text{subject to} \quad y_e \leq \sum_{e'\in C\setminus\{e\}} y_{e'} \qquad \forall C \in \text{ cycles } (G) \forall e \in C \tag{22}$$

$$\sum_{t\in T} y_{tv} = |T| - 1, \qquad \text{if } T \neq \emptyset, \forall v \in V\setminus T \tag{23}$$

$$y_{tt'} = 1, \qquad \forall t, t' \in T, t \neq t' c, f \tag{24}$$

$$y_{tu} + y_{uv} \geq y_{tv}, \qquad \forall (u,v) \in E, t \in T \tag{25}$$

$$y_{tv} + y_{uv} \geq y_{tu}, \qquad \forall (u,v) \in E, t \in T. \tag{26}$$

We have reformulated the objective by [24] slightly, to highlight the relations of the following cases: For $p = 1$ and $T \neq \emptyset$, this ILP corresponds to the Asymmetric Multiway Cut. Without semantic classes (*i.e.* $T = \emptyset$) eqs. (23) to (26) are superfluous and the problem reduces to the Multi Cut for $p = 1$ and the Mutex Watershed objective when p is large enough to be dominant [44].

We will now show, for $T \neq \emptyset$ and dominant p, that eqs. (21) to (26) can be solved to optimality with the SMWS. To this end, we identify the indicator variables x in eq. (17) with the AMWC indicators y.

For attractive and semantic edges both indicators represent the same graph partitions and class assignments. In particular, given the associated indicators x and y of any graph partitioning and labeling, $x_e = y(x,e) \quad \forall\, e \in E^+ \cup E^S$ holds. For repulsive edges $e^- \in E^-$ however, x_{e^-} indicates a mutex edge and therefore a necessary cut, hence $y_{e^-} = 1 - x_{e^-}$. Additionally, the Asymmetric Multiway Cut introduces repulsive edges between terminal nodes and constrains them to be always cut. In conclusion we can translate between both indicators with

$$y(x,e) = \begin{cases} x_e & \text{if } e \in E^+ \cup E^S \\ 1 - x_e & \text{if } e \in E^- \\ 1 & \text{if } e \in (T \times T) \end{cases} \tag{27}$$

Using eq. (27) we translate the SWMS objective eq. (17)

$$\sum_{e\in E'} |w_e|^p x_e = \sum_{e\in E^+} |w_e|^p y(x,e) + \underbrace{\left(\sum_{e\in E^-} 1\right)}_{\mathcal{L}_{\text{triv}}} - \sum_{e\in E^-} |w_e|^p y(x,e) + \sum_{e\in E^S} |w_e|^p y(x,e) \tag{28}$$

$$= \sum_{e\in E'} \operatorname{sign}(w_e)|w_e|^p y(x,e) + \mathcal{L}_{\text{triv}} \tag{29}$$

Note that the constant $\mathcal{L}_{\text{triv}}$ does not affect the minimum energy solution.

Second, we will add the constraints

$$\sum_{t \in T} y_{tv} = |T| - 1 \qquad \forall v \in V \tag{30}$$

$$y_e \leq \sum_{e' \in C \setminus \{e\}} y_{e'} \qquad \forall C \in \text{ cycles } (G) \forall e \in C \tag{31}$$

to the Semantic Mutex Watershed ILP eqs. (17) to (19) and observe, since $y(x^{\mathbf{SMWS}})$ always fulfills eqs. (30) and (31). Therefore, $y(x^{\mathbf{SMWS}})$ also minimizes eq. (17) subject to the tighter constraints eqs. (19), (20), (30) and (31). Using Equation (30) and Lemma 2 (see Appendix A) we can replace the path constraints eq. (19) by

$$\begin{aligned}
\mathcal{P}(A) = \emptyset \quad &\Leftrightarrow \quad \sum_{e \in P} y(x, e) \geq 1 && \forall P \in \pi_{t \rightsquigarrow t'} \; \forall t, t' \in T, \, t \neq t' && (32)\\
&\Leftrightarrow \quad y_{ut} + y_{uv} + y_{vt'} \geq 1 && \forall (u, v) \in E \; \forall t, t' \in T, \, t \neq t' && (33)\\
&\Leftrightarrow \quad y_{tu} + y_{uv} \geq y_{tv}, && \forall uv \in E, t \in T && (34)\\
&\qquad\quad y_{tv} + y_{uv} \geq y_{tu}, && \forall uv \in E, t \in T. && (35)
\end{aligned}$$

We conclude that $y(x^{\mathbf{SMWS}})$ minimizes the objective eqs. (21) to (26) highlighting the close connection to the Asymmetric Mutiway Cut objective. In fact, although unlikely in practical applications, for graphs $\mathcal{G}$ where $d = 1$ is a dominant power, the Semantic Mutex Watershed solves the Asymmetric Mutiway Cut to optimality.

4 Experiments

We will now demonstrate how to apply the SMWS algorithm to semantic instance segmentation of 2D and 3D images. We show how existing CNNs can be used as graph weight estimators and compare different sources of edge weights on the Cityscapes dataset. Additionally, we apply the SMWS to a 3D electron microscopy volume and demonstrate its efficiency and scalability. Our SMWS implementation is available at `www.github.com/constantinpape/affogato`

4.1 Affinity Generation with Neural Networks

The only input to the SMWS are the graph weights; it does not require any hyperparamters such as thresholds. Consequently, its segmentation quality relies on good estimates of the graph weights w'. In this section we present how state-of-the-art CNNs can be used as sources for these weights.

Affinity Learning. Affinities are commonly used in instance segmentation; for many modern algorithms CNNs are trained to directly predict pixel affinities. A common approach is to employ a stencil pattern that describes for each pixel which neighbours to consider for the affinity computation. Regularly spaced, multi-scale stencil patterns are widely used for natural images [33,31] and biomedical data [45,28]. These affinities are usually in the interval $[0, 1]$ and can be interpreted as pseudo-probabilities. We use these affinities directly as weights for the attractive edges and invert them to get the repulsive edge weights. Therefore the set of affinities from a single source (*e.g.* a single CNN) forms a weighted graph an which the SMWS can be applied. When multiple sources of affinities are used, each one adds a new set of weighted edges to the graph. If two sources yield different weights for the same edge, only the maximum absolute weight for this edge will be considered by the SMWS algorithm.

Mask-RCNN produces overlapping masks that have to be resolved for a consistent panoptic segmentation. We achieve this with the SMWS by deriving affinities from the foreground probabilities of each mask. A straightforward approach is to compute the (attractive) affinity $a(i, j)$ of two pixels as their joint foreground probability, weighted by the classification score s: $a(i, j) = s\, p(i)\, p(j)$.

We find that sparse repulsive edges work well in practice, as they lead to faster inference and reduced over-segmentation on the instance boundaries. For this reason, we sample random points from all pairs of masks and add (repulsive) edges with weight proportional to a soft intersection over union of two masks m and n: $w_{nm} = 1 - \frac{\sum_{q \in V} p_m(q) p_n(q)}{\sum_{q \in V} \max\left(p_m(q),\ p_n(q)\right)}$.

Semantic Segmentation CNNs. State of the art CNNs [7,50] achieve high quality results on semantic segmentation tasks. The output of the last softmax layer usually used in these networks can be interpreted as the normalized probability of each pixel belonging to each class. Thus, we can use these predictions directly as semantic weights. Additionally, we derive affinities of two pixels i and j from the stuff class probabilities, using their joint probability of being in each stuff class c, i.e.: $a_c(i, j) = p_{\mathrm{c}}(i)\, p_{\mathrm{c}}(j)$.

4.2 Panoptic Segmentation on Cityscapes

We apply the SMWS on the challenging task of panoptic segmentation on the Cityscapes dataset [9]. We illustrate how the different sources of affinities can be used and combined and show their different strengths and weaknesses.

Dataset. The Cityscapes dataset consists of urban street scene images taken from a driver's perspective. It has 5k densely annotated images separated into train (2975), val (500) and test (1525) set. We report all results on the validation set. There are 19 classes with 11 stuff classes and 8 thing classes.

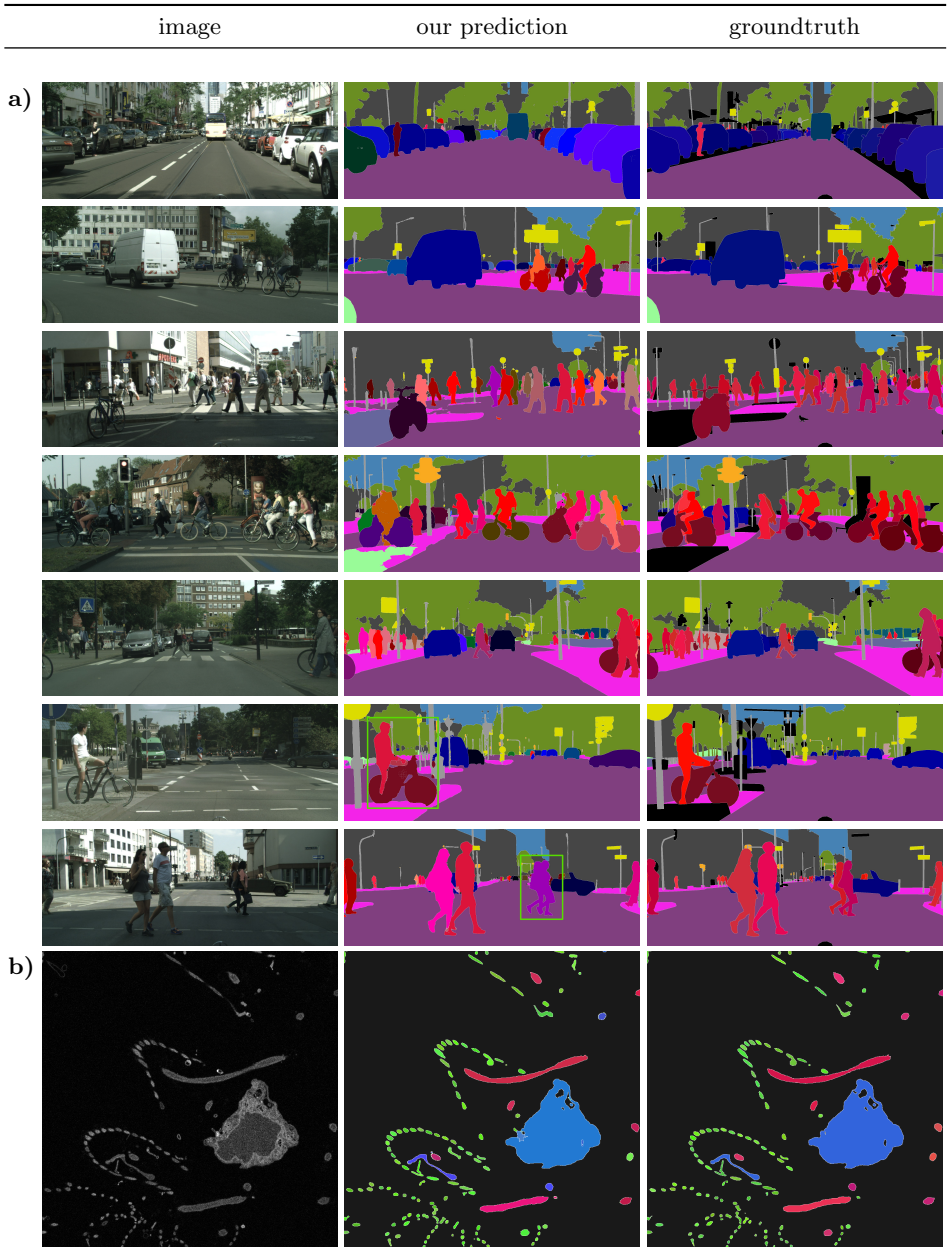

Figure 2: **a)** Semantic instance segmentation. Results on Cityscapes using semantic unaries (Deeplab 3+ network) and affinities derived from Mask-RCNN foreground probability. Colors indicate predicted semantic classes with variations for separate instances. The last two rows show failure cases highlighted in green. Cyclists and their bicycle often form separate components with few to no graph connections between them resulting in a common failure for graph-based segmentation in general, and the SMWS in particular. **b)** Results for the 3D sponge dataset. Cell-bodies are colored in blue, microvilli in green and flagella in red.

MRCNN[15]		GMIS[31]		DEEPLAB[7]			**Cityscapes**		
att	rep	att	rep	att	rep	sem	PQ	PQ^{Th}	PQ^{St}
✓	✓					✓	59.3	50.6	65.7
	✓	✓				✓	58.6	48.8	65.7
		✓	✓			✓	56.1	42.8	65.7
✓	✓	✓	✓	✓	✓	✓	48.7	38.7	55.9
		✓	✓	✓	✓	✓	47.3	35.5	55.9
				✓	✓	✓	46.3	33.1	56.0

Table 1: Panoptic segmentation quality PQ of the SMWS on top of diverse sources of graph weights.

Cityscapes	PQ	PQ^{Th}	PQ^{St}
AdaptIS[41]	62.0	64.4	64.4
SSAP[14]	61.1	55.0	-
SMWS	59.3	50.6	65.7
UPSNet[46]	59.3	54.6	62.7
AUNet[30]	59.0	54.8	62.1
PFPN[18]	58.1	52.0	62.5

Sponge	PQ	PQ^{Th}	PQ^{St}
SMWS	**51.6**	**62.1**	20.0
MWS-MAX	48.1	56.2	**23.8**
CC_{sem}	43.4	55.6	06.7
CC_{aff}	24.3	27.7	13.9

Table 2: Comparison of panoptic segmentation quality on Cityscapes and Sponge dataset. For Cityscapes, the SMWS uses attractive and repulsive graph weights derived from a Masked-RCNN and semantic class probabilities predicted by a Deeplab 3+ network. For Sponge the weights are estimated by two 3D-U-Nets.

Implementation Details. To derive graph weights, we use multiple neural networks trained for affinity, semantic class probability and bounding box prediction (see subsection 4.1). First, we train two Deeplab 3+ [7] networks to predict semantic class probabilities and affinities on the full image resolution. We adopt the training procedure of [7], for both networks. For the affinities we employ the stencil pattern by [31] and train with the Sorensen Dice Loss [45]. The training is done with a batch size of 12, 70k training iterations and without test time augmentations. We will refer to these networks as DEEPLAB in Table 1.

Additionally, we, use a more sophisticated method for affinity prediction and a second Deeplab 3+ network trained on re-scaled crops (GMIS in Table 1). This method was proposed by [31], who kindly provided their trained models allowing us to use their affinities. Their clustering utilizes a threshold, which we use as the splitting point between attractive and repulsive edge weights, i.e. affinities below the threshold are inverted and all affinities are scaled to [0, 1].

Finally, we train a Mask-RCNN with the training procedure described in [15] using the implementation from [34]. We derive graph weights, as described in subsection 4.1, for attractive edges in a regular 8-neighborhood with distances of {1, 2, 4} pixels, and for repulsive edges between pairs of masks. To avoid the large combinatorial number of all pixel pairs between masks, we restrict the

repulsive edges to 5 random pixel per mask. The affinities from this procedure are referred to as MRCNN in Table 1.

Study of Affinity Sources. We evaluate the semantic instance segmentation performance of the SMWS in terms of the "panoptic" metric using different combinations of the graph weight sources discussed above. In table 1 we compare the PQ metric on the Cityscapes dataset. The best performance can be achieved with a combination of Mask-RCNN affinities and Deeplab 3+ for semantic predictions outperforming the strong baseline of [18] listed in table 2 and shown in fig. 2. We find that Mask-RCNN affinities are more reliable in detecting small objects and connecting fragmented instances. Note that PQ measures detection quality, weighted by the segmentation quality of the found instances, hence the detection strength of the Mask-RCNN shines through. Using all sources together leads to a performance drop of 10 percentage points below the best result. We believe this is due to the greedy nature of the SMWS which selects the strongest of all provided edges. This example demonstrates how important it is to carefully select and train the algorithm input.

4.3 Semantic Instance Segmentation of 3D EM Volumes

Semantic instance segmentation is an important task in bio-medical image analysis where classes naturally arise through cellular ultra-structure. We use a 3D EM image dataset to compare the SMWS to algorithms that separately optimize instance segmentation and semantic class assignment.

Dataset. The dataset consists of two FIBSEM volumes of a sponge choanocyte chamber. The data was acquired in [35] to study proto-neural cells in sponges using the segmentation approach introduced in [37]. These cells filter nutrients from water by creating a flow with the beating of a flagellum and absorbing the nutrients through microvilli that surround the flagellum in a collar [27] (see fig. 2). To investigate this process in detail, a precise semantic instance segmentation of the cell-bodies, flagella and microvilli is needed. The dataset consists of three EM image volumes of size $96 \times 896 \times 896$ pixel ($2 \times 18 \times 18\,\mu$m).

Implementation Details. We predict affinities with two separate 3D U-Nets [8] to derive graph edge weights and semantic class probabilities respectively. We adopt the training procedure by [45], which uses the Dice Coefficient as the loss function. Two volumes are used for training and one for testing.

Results. We implement baseline approaches which start from the same network predictions, but do not perform joint labeling and partitioning. We compare to instance segmentation with the Mutex Watershed, followed by assigning instances the semantic label of the strongest semantic edge (MWS-MAX). As a further baseline, we compute connected components of the semantic predictions (CC_{sem}) and short-range affinities (CC_{aff}). The PQ values in table 2 show that the SMWS outperforms the baselines approaches that separately optimize

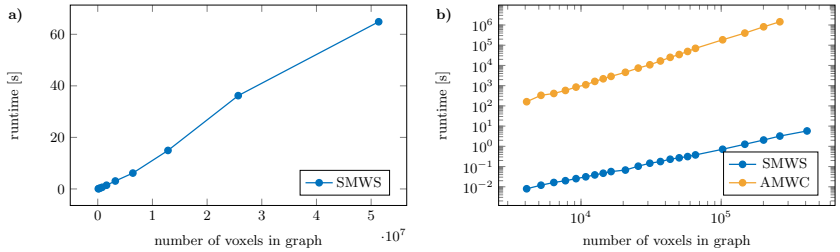

Figure 3: Runtime scaling of the SMWS. **a)** The runtime of the SMWS is evaluated on different volume sizes of the 3D Sponge dataset. We find an almost linear relation between runtime and number of voxels. **b)** Runtime comparison of **(blue)** the SMWS (minimizing (17) with $p \to \infty$) with **(orange)** a KLj*r solver [29] (minimizing the AMWC objective [24], (17) with $p = 1$). The runtime is evaluated on 2D slices of the 3D Sponge dataset with varying size. On the largest feasible slice the SMWS is marginally less accurate with PQ=49.2 (compared to AMWC PQ=52.0), but 5 orders of magnitude faster. We use the implementation of [29] for the AMWC optimization.

instance segmentation and semantic class assignment. Additionally, we measure the runtime of the SMWS on crops of the EM-volume with varying number of voxels, shown in Figure 3). The inference on the full volume (with $\sim 5 \cdot 10^7$ voxels) takes 65 seconds. In the analyzed volume domain the runtime appears to scale linearly with the number of volxels, suggesting that even larger volumes can be processed in reasonable time. We also compare the runtime of the SMWS with an NLMP solver introduced in [29] and find that it is about 5 orders of magnitude faster with only marginally decreased segmentation quality.

5 Conclusion

We introduced a new method for joint partitioning and labeling of weighted graphs as a generalization of the Mutex Watershed algorithm. This algorithm optimally solves an objective function closely related to the objective of the Asymmetric Multiway Cut problem. Our experiments demonstrate that the SMWS with graph edge weights predicted by convolutional neural networks outperform strong baselines on natural and biological images. Any improvement in the CNN performance will translate directly to an improvement of the SMWS results. However, we also observe that the extreme value selection used by the SMWS to assign edges to the active set can lead to sub-optimal performance when diverse edge weights sources are combined. Empirically, the algorithm scales almost linearly with the number of graph edges N making it applicable to large images and volumes without prior over-segmentation into superpixels.

Acknowledgements Funded in part by the Deutsche Forschungsgemeinschft (DFG, German Research Foundation) – Projektnummer 240245660 - SFB 1129

References

1. Andres, B., Briggman, K.L., Korogod, N., Knott, G., Koethe, U., Hamprecht, F.A.: Globally Optimal Closed-Surface Segmentation for Connectomics. In: Computer Vision – ECCV 2012, vol. 7574, pp. 778–791. Springer Berlin Heidelberg (2012)
2. Andres, B., Kappes, J.H., Beier, T., Köthe, U., Hamprecht, F.A.: Probabilistic image segmentation with closedness constraints. In: 2011 International Conference on Computer Vision. pp. 2611–2618. IEEE (2011)
3. Bai, M., Urtasun, R.: Deep watershed transform for instance segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2858–2866. IEEE (2017)
4. Beier, T., Pape, C., Rahaman, N., Prange, T., Berg, S., Bock, D.D., Cardona, A., Knott, G.W., Plaza, S.M., Scheffer, L.K., et al.: Multicut brings automated neurite segmentation closer to human performance. Nature Methods **14**(2), 101 (2017)
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. In: ICLR (2016)
6. Chen, L., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. CoRR **abs/1706.05587** (2017)
7. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 801–818 (2018)
8. Çiçek, O., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 424–432. Springer (2016)
9. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes Dataset for Semantic Urban Scene Understanding. arXiv:1604.01685 [cs] (2016)
10. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, Third Edition. The MIT Press, 3rd edn. (2009)
11. Dai, J., He, K., Sun, J.: Instance-Aware Semantic Segmentation via Multi-task Network Cascades. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3150–3158. IEEE (2016)
12. Fathi, A., Wojna, Z., Rathod, V., Wang, P., Song, H.O., Guadarrama, S., Murphy, K.P.: Semantic Instance Segmentation via Deep Metric Learning. arXiv:1703.10277 [cs] (2017)
13. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient Graph-Based Image Segmentation. International Journal of Computer Vision **59**(2), 167–181 (2004)
14. Gao, N., Shan, Y., Wang, Y., Zhao, X., Yu, Y., Yang, M., Huang, K.: Ssap: Single-shot instance segmentation with affinity pyramid. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 642–651 (2019)
15. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 2980–2988 (2017)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
17. Kappes, J.H., Speth, M., Andres, B., Reinelt, G., Schn, C.: Globally optimal image partitioning by multicuts. In: Boykov, Y., Kahl, F., Lempitsky, V., Schmidt, F.R. (eds.) Energy Minimization Methods in Computer Vision and Pattern Recognition, vol. 6819, pp. 31–44. Springer Berlin Heidelberg (2011)

18. Kirillov, A., Girshick, R., He, K., Dollár, P.: Panoptic feature pyramid networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6399–6408 (2019)
19. Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P.: Panoptic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 9404–9413 (2019)
20. Kirillov, A., Levinkov, E., Andres, B., Savchynskyy, B., Rother, C.: Instancecut: From edges to instances with multicut. In: CVPR. vol. 3, p. 9 (2017)
21. Kong, S., Fowlkes, C.C.: Recurrent pixel embedding for instance grouping. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 9018–9028. IEEE Computer Society (2018)
22. Krasowski, N., Beier, T., Knott, G., Kothe, U., Hamprecht, F.A., Kreshuk, A.: Neuron Segmentation With High-Level Biological Priors. IEEE Transactions on Medical Imaging **37**(4), 829–839 (2018)
23. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Communications of the ACM **60**(6), 84–90 (2017)
24. Kroeger, T., Kappes, J.H., Beier, T., Koethe, U., Hamprecht, F.A.: Asymmetric Cuts: Joint Image Labeling and Partitioning. In: Pattern Recognition, vol. 8753, pp. 199–211. Springer International Publishing (2014)
25. Kruskal, J.B.: On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. Proceedings of the American Mathematical Society p. 3 (1956)
26. Lange, J.H., Karrenbauer, A., Andres, B.: Partial optimality and fast lower bounds for weighted correlation clustering. In: International Conference on Machine Learning. pp. 2898–2907 (2018)
27. Langenbruch, P.F., Weissenfels, N.: Canal systems and choanocyte chambers in freshwater sponges (porifera, spongillidae). Zoomorphology **107**(1), 11–16 (1987)
28. Lee, K., Zung, J., Li, P., Jain, V., Seung, H.S.: Superhuman accuracy on the SNEMI3D connectomics challenge. CoRR **abs/1706.00120** (2017), `http://arxiv.org/abs/1706.00120`
29. Levinkov, E., Uhrig, J., Tang, S., Omran, M., Insafutdinov, E., Kirillov, A., Rother, C., Brox, T., Schiele, B., Andres, B.: Joint Graph Decomposition & Node Labeling: Problem, Algorithms, Applications. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1904–1912. IEEE (2017)
30. Li, Y., Chen, X., Zhu, Z., Xie, L., Huang, G., Du, D., Wang, X.: Attention-guided unified network for panoptic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7026–7035 (2019)
31. Liu, Y., Yang, S., Li, B., Zhou, W., Xu, J.Z., Li, H., Lu, Y.: Affinity Derivation and Graph Merge for Instance Segmentation. In: The European Conference on Computer Vision (ECCV). p. 18 (2018)
32. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3431–3440 (2015)
33. Maire, M., Narihira, T., Yu, S.X.: Affinity CNN: Learning pixel-centric pairwise relations for figure/ground embedding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 174–182 (2016)
34. Massa, F., Girshick, R.: Maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch (2018)

35. Musser, J.M., Schippers, K.J., Nickel, M., Mizzon, G., Kohn, A.B., Pape, C., Hammel, J.U., Wolf, F., Liang, C., Hernández-Plaza, A., et al.: Profiling cellular diversity in sponges informs animal cell type and nervous system evolution. BioRxiv p. 758276 (2019)
36. Pape, C., Beier, T., Li, P., Jain, V., Bock, D.D., Kreshuk, A.: Solving Large Multicut Problems for Connectomics via Domain Decomposition. In: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW). pp. 1–10. IEEE (2017)
37. Pape, C., Matskevych, A., Wolny, A., Hennies, J., Mizzon, G., Louveaux, M., Musser, J., Maizel, A., Arendt, D., Kreshuk, A.: Leveraging domain knowledge to improve microscopy image segmentation with lifted multicuts. Frontiers in Computer Science **1**, 6 (2019)
38. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems 28, pp. 91–99. Curran Associates, Inc. (2015)
39. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 234–241. Springer (2015)
40. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for large-scale image recognition. In: 3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings (2015)
41. Sofiiuk, K., Barinova, O., Konushin, A.: Adaptis: Adaptive instance selection network. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 7355–7363 (2019)
42. Tighe, J., Niethammer, M., Lazebnik, S.: Scene Parsing with Object Instance Inference Using Regions and Per-exemplar Detectors. International Journal of Computer Vision **112**(2), 150–171 (2015)
43. Tu, Z., Chen, X., Yuille, A.L., Zhu, S.C.: Image Parsing: Unifying Segmentation, Detection, and Recognition. International Journal of Computer Vision **63**(2), 113–140 (2005)
44. Wolf, S., Bailoni, A., Pape, C., Rahaman, N., Kreshuk, A., Köthe, U., Hamprecht, F.A.: The Mutex Watershed and its Objective: Efficient, Parameter-Free Image Partitioning. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
45. Wolf, S., Pape, C., Rahaman, N., Kreshuk, A., Kothe, U., Hamprecht, F.A.: The Mutex Watershed: Efficient, Parameter-Free Image Partitioning. In: The European Conference on Computer Vision (ECCV). p. 17 (2018)
46. Xiong, Y., Liao, R., Zhao, H., Hu, R., Bai, M., Yumer, E., Urtasun, R.: Upsnet: A unified panoptic segmentation network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8818–8826 (2019)
47. Yao, J., Fidler, S., Urtasun, R.: Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 702–709. IEEE (2012)
48. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Nong, S.: Learning a discriminative feature network for semantic segmentation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1857–1866 (2018)
49. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: 4th International Conference on Learning Representations, ICLR 2016, Conference Track Proceedings (2016)

50. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 2881–2890 (2017)