Deep Hierarchical Semantic Segmentation

Liulei Li¹,⁵*, Tianfei Zhou², Wenguan Wang³†, Jianwu Li¹, Yi Yang⁴

¹ Beijing Institute of Technology ² ETH Zurich ³ ReLER, AAII, University of Technology Sydney ⁴ CCAI, Zhejiang University ⁵ Baidu Research

https://github.com/0liliulei/HieraSeg

Abstract

Humans are able to recognize structured relations in observation, allowing us to decompose complex scenes into simpler parts and abstract the visual world in multiple levels. However, such hierarchical reasoning ability of human perception remains largely unexplored in current literature of semantic segmentation. Existing work is often aware of flatten labels and predicts target classes exclusively for each pixel. In this paper, we instead address hierarchical semantic segmentation (HSS), which aims at structured, pixel-wise description of visual observation in terms of a class hierarchy. We devise HSSN, a general HSS framework that tackles two critical issues in this task: i) how to efficiently adapt existing hierarchy-agnostic segmentation networks to the HSS setting, and ii) how to leverage the hierarchy information to regularize HSS network learning. To address i), HSSN directly casts HSS as a pixel-wise multi-label classification task, only bringing minimal architecture change to current segmentation models. To solve ii), HSSN first explores inherent properties of the hierarchy as a training objective, which enforces segmentation predictions to obey the hierarchy structure. Further, with hierarchy-induced margin constraints, HSSN reshapes the pixel embedding space, so as to generate well-structured pixel representations and improve segmentation eventually. We conduct experiments on four semantic segmentation datasets (i.e., Mapillary Vistas 2.0, Cityscapes, LIP, and PASCAL-Person-Part), with different class hierarchies, segmentation network architectures and backbones, showing the generalization and superiority of HSSN.

1. Introduction

Semantic segmentation, which aims to identify semantic categories for pixel observations, is viewed as a vital step towards intelligent scene understanding [82]. The vast majority of modern segmentation models simply assume that all the target classes are disjoint and should be distinguished exclusively during pixel-wise prediction. This fails to capture the structured nature of the visual world [53]; complex scenes arise from the composition of simpler entities. Walking city, vehicles and pedestrian fill our view (Fig. 1). After focusing on the vehicles, we identify cars, buses, and trucks, which consist of more fine-grained parts like wheel and window. On the other hand, structured understanding of our world in terms of relations and hierarchies is a central ability in human cognition [68, 95]. We group chair and bed as furniture, while cat and dog as pet. We understand this world over multiple levels of abstraction, in order to maintain stable, coherent percepts in the face of complex visual inputs [37].

The ubiquity of hierarchical decomposition serves as a core motivation behind many structured machine learning models [20, 85], which have shown wide success in document classification [39, 55] and protein function prediction [8, 75].

In semantic segmentation literature, surprisingly little is understood about how to accommodate pixel recognition into semantic hierarchies. [43, 45, 56, 80, 81, 83, 89] are rare exceptions that exploit class hierarchies in segmentation networks. Nevertheless, they either focus specifically on the structured organization of human body parts [80, 81, 83], or introduce hierarchy-induced architectural changes to the
segmentation network [43, 45, 56, 89], both hindering generality. More essentially, these methods are more aware of making efficient information propagation over the hierarchies (e.g., graph message passing [43, 83, 109], multi-task learning [89]), without imposing tree-structured label dependencies/constraints into prediction and learning.

To mimic human hierarchical visual perception, we propose a novel approach for hierarchical semantic segmentation (HSS). In HSS, classes are not arranged in a “flat” structure, but organized as a tree-shaped hierarchy. Thus each pixel observation is associated to a root-to-leaf path of the class hierarchy (e.g., human→rider→bicyclist), capturing general-to-specific relations between classes. Our algorithm, called HSSN, addresses two core issues in HSS, yet untouched before. First, instead of previous structured segmentation models focusing on sophisticated network design, HSSN directly formulates HSS as a pixel-wise multi-label classification task. This allows to easily adapt existing segmentation models to the HSS setting, densely linking the fields of classic hierarchy-agnostic segmentation and HSS together. Second, HSSN makes full use of the class hierarchy in HSS network learning. To make pixel predictions coherent with the class hierarchy, HSSN explores two hierarchy constraints, i.e., i) a pixel sample belonging to a given class must also belong to all its ancestors in the hierarchy, ii) a pixel sample not belonging to a given class must also not belong to all its descendants, as optimization criterion. This leads to a pixel-wise hierarchical segmentation learning strategy, which enforces segmentation predictions to obey the hierarchy structure during training. HSSN further encodes the structured knowledge introduced by the class hierarchy into the pixel embedding space. This leads to a pixel-wise hierarchical representation learning strategy, which inspires tree-induced margin separation for embedding space reshaping. As the hierarchy characterizes the underlying relationships between classes, HSSN is able to enrich pixel embeddings by pulling semantically similar pixels (e.g., bicycle and motorcycle) closer, while pushing semantically dissimilar pixels (e.g., pedestrian and lamppost) farther away. This leads to more efficient learning by discovering and reusing common patterns [27], facilitating hierarchical segmentation eventually. This also allows our model to take different levels of mistakes into consideration. This is essential for some critical systems [7]. Take autonomous driving as an example: mistaking a bicycle for a motorcycle is less of a problem than confusing a pedestrian with a lamppost.

This work represents a solid step towards HSS. Our approach is elegant and principle; it is readily incorporated to arbitrary previous hierarchy-agnostic segmentation networks, with only marginal modification on the segmentation head. We train and test HSSN over four public benchmarks (i.e., Mapillary Vistas 2.0 [58], Cityscapes [18], LIP [44], PASCAL-Person-Part [87]), with different class hierarchies for urban street scene parsing and human semantic parsing. Extensive experimental results with different segmentation network architectures (i.e., DeepLabV3+ [13], OCRNet [98], MaskFormer [16] and backbones (i.e., ResNet-101 [34], HRNetV2-W48 [79], Swin-Small [49]) verify the generalization and effectiveness of HSSN.

2. Related Work

(Hierarchy-Agnostic) Semantic Segmentation. Semantic segmentation is to partition an image into regions with different semantic categories, which can be viewed as a pixel-wise classification task. Typical solutions for semantic segmentation follow a hierarchy-agnostic setting, where each pixel is assigned to a single label from a set of disjoint semantic categories. In 2015, Long et al. proposed fully convolutional networks (FCNs) [50], which are advantageous in end-to-end dense representation modeling, laying the foundation for modern semantic segmentation algorithms. As FCNs suffer from limited visual context with local receptive fields, how to effectively capture cross-pixel relations became the main focus of follow-up studies. Scholars devised many promising solutions, by enlarging receptive fields [10, 13, 19, 93, 97, 105], building image pyramids [33, 46], exploring encoder-decoder architectures [3, 13, 62], utilizing boundary clues [23, 41, 99], or incorporating neural attention [25, 32, 35, 40, 42, 73, 84, 106, 108, 113]. Recently, a new family of semantic segmentation models [16, 69, 90, 107], built upon the full attention (Transformer [76]) architecture, yielded impressive performance, as it overcomes the issues in long-range cross-pixel dependency modeling.

Though impressive, existing semantic segmentation solutions rarely explore the structures between semantic concepts. We take a further step towards class relation aware semantic segmentation, which better reflects the structured nature of our visual world, and echoes the hierarchical reasoning mode of human visual perception. An appealing advantage of our hierarchical solution is that, it can adapt existing class hierarchy-agnostic segmentation architectures, no matter FCN-based or Transformer-like, to the structured setting, in a simple and cheap manner.

Scene Parsing/Hierarchical Semantic Segmentation. Our work is, at a high level, relevant to classical image parsing algorithms [31, 70, 71, 74, 96]. Image parsing has been extensively studied in the pre-deep learning era, dating back to [74]. Image parsing seeks a parse graph that explains visual observation following a “divide-and-conquer”strategy: a football game image is first parsed into person, sports field, and spectator, which are further decomposed, e.g., person consists of face and body patterns. In the deep learning era, human parsing, as a sub-field of scene parsing, became active. Some recent human parsers explored human part relations, based on the human hierarchy [36, 56, 80, 83,
Only very few efforts [43, 45, 89, 104] are concerned with utilizing structured knowledge to aid the training of general-purpose semantic segmentation networks.

To accommodate the semantic structures imposed by the hierarchy, previous methods tend to greatly change the segmentation network, through the use of different graph neural networks. They hence put all emphasis on how to aggregate information over the structured network. Beyond their specific solutions, we propose a general framework for both HSS network design and training. This leads to an elegant view of how to adapt typical segmentation networks to the class hierarchy with only minimal architecture change, and how to involve the hierarchy for regularizing network training, which are core problems yet ignored by prior methods.

Hierarchical Classification. Considering class hierarchies when designing classifiers is a common issue across various machine learning application domains [67], such as text categorization [63], functional genomics [4], and image classification [6, 21]. Depending on whether each datapoint can be assigned a single path or multiple paths in the hierarchy, hierarchy-aware classification can be categorized into hierarchical classification [20, 39, 55, 72] and, a more general setting, hierarchical multi-label classification [8, 29, 85]. In the field of computer vision, exiting efforts for class taxonomy aware image classification can be broadly divided into three groups [7]: i) Label-embedding methods [2, 6, 24, 88] that embed class labels to vectors whose relative locations represent semantic relationships; ii) Hierarchical losses [7, 9, 21, 78, 103] which are designed to inspire the coherence between the prediction and class hierarchy; and iii) Hierarchical architectures [1, 91, 112, 114] that adapt the classifier architecture to the class hierarchy.

Drawing inspiration from these past efforts, we advocate for holistic visual scene understanding through pixel-level hierarchical reasoning. We leverage tree-structured class dependencies as supervision signal to guide hierarchy-coherent pixel prediction and structured pixel embedding.

Hierarchical Embedding. The objective of an embedding algorithm is to organize data samples (e.g., words, images) into a high-dimensional space where their distance reflects their semantic similarity [59]. As semantics are inherently structured, it is necessary to integrate different levels of concept abstraction into representation embedding. Some algorithms directly parameterize the hierarchical embedding space into hierarchical models [14, 57, 60, 78, 86, 92]. While straightforward, they are computationally intensive and have to adjust the network architecture when handling different hierarchies. Some alternatives [5, 28, 38, 94] design hierarchy-aware metric learning objectives [26, 59, 65] to directly shape the embedding space.

With a similar spirit, in this work, we adopt semantic hierarchy-induced margin separation to reinforce pixel representation learning and make prediction less ambiguous.

3. Our Approach

Our goal is to accommodate standard semantic segmentation networks to the HSS problem and then exploit structured class relations in order to generate hierarchy-coherent representations and predictions, and improve performance. Given this goal, we develop HIERARCHICAL SEMANTIC SEGMENTATION NETWORKS (HSSN), a general framework for HSS network design (§3.1) and training (§3.2).

3.1. Hierarchical Semantic Segmentation Networks

Rather than typical segmentation methods treating semantic classes as disjoint labels, in the HSS setting, the underlying dependencies between classes are considered and formalized in a form of a tree-structured hierarchy, $T = (V, E)$. Each node $v \in V$ denotes a semantic class/concept, while each edge $(u, v) \in E$ encodes the decomposition relationship between two classes, $u, v \in V$, i.e., parent node $v$ is a more general, superclass of child node $u$, such as $(u, v) = \{\text{bicycle}, \text{vehicle}\}$. We assume $(v, v) \in E$, thus every class is both a subclass and superclass of itself. The root node of $T$, i.e., $v^*$, denotes the most general class. The leaf nodes, i.e., $V_\chi$, refer to the most fine-grained classes, such as $V_\chi = \{\text{tree}, \text{bicyclist}, \cdots\}$ in urban street scene parsing, and $V_\chi = \{\text{head}, \text{leg}, \cdots\}$ in human parsing.

For a typical hierarchy-agnostic segmentation network, an encoder $f_{\text{ENC}}$ is first adopted to map an image $I$ into a dense feature tensor $f = f_{\text{ENC}}(I) \in \mathbb{R}^{H \times W \times |V|}$, where $i \in I$ is the embedding of pixel $i \in I$. Then a segmentation head $f_{\text{SEG}}$ is used to get a score map $Y = \text{softmax}(f_{\text{SEG}}(I)) \in [0, 1]^{H \times W \times |V|}$ w.r.t. the leaf node set $V_\chi$. Given the score vector $y = [y_v]_{v \in V_\chi} \in [0, 1]^{|V_\chi|}$ and groundtruth leaf label $v_\chi \in V_\chi$ for pixel $i$, the categorical cross-entropy loss is optimized:

$$L_{\text{CCE}}(y) = -\log(y_{v_\chi}).$$

During inference, pixel $i$ is associated to a single leaf node: $v^*_\chi = \arg \max_{v_\chi} y_{v_\chi}$.

To accommodate classic segmentation networks to the HSS setting with minimum change, our HSSN first formulates HSS as a pixel-wise multi-label classification task, i.e., map pixels with their corresponding classes in the hierarchy as a whole. Specifically, only the segmentation head $f_{\text{SEG}}$ is modified to predict an augmented score map $S = \text{sigmoid}(f_{\text{SEG}}(I)) \in [0, 1]^{H \times W \times |V|}$ w.r.t. the entire class hierarchy $V$. Given the score vector $s = [s_v]_{v \in V} \in [0, 1]^{|V|}$ and groundtruth binary label set $l = [l_v]_{v \in V} \in \{0, 1\}^{|V|}$ for pixel $i$, the binary cross-entropy loss is optimized:

$$L_{\text{BCE}}(s) = \sum_{v \in V} \hat{l}_v \log(s_v) - (1-\hat{l}_v)\log(1-s_v).$$

During inference, each pixel $i$ is associated with the top-scoring root-to-leaf path in the class hierarchy $T$:

$$\{v^*_1, \cdots, v^*_{|P|}\} = \arg \max_{P \subseteq T} \sum_{v_p \in P} s_{v_p},$$
Figure 2. Hierarchy constraints used in our pixel-wise hierarchical segmentation learning (§3.2.1). (a) In the class hierarchy, the filled circles represent the positive classes, while empty circles indicate the negative classes. The positive and negative T-properties are highlighted in the red and blue regions, respectively. (b) The original score vector \( s \) predicted for the class hierarchy. The predictions which violate the positive and negative T-constraints are highlighted in the red and blue rectangles, respectively. (c) The updated score vector \( p \), which satisfies the T-constraints. With \( \mathcal{L}^{\text{TM}} \), the penalties for the wrong predictions, i.e., ‘0.6’ and ‘0.3’, are increased twice, compared with applying \( \mathcal{L}^{\text{BCE}} \) on \( b \).

where \( \mathcal{P} = \{ v_1, \ldots, v_{|\mathcal{P}|} \} \subseteq \mathcal{T} \) denotes a feasible root-to-leaf path of \( \mathcal{T} \), i.e., \( v_1 \in \mathcal{V}_r, v_{|\mathcal{P}|} = v^r \), and \( \forall v_p, v_{p+1} \in \mathcal{P} \Rightarrow (v_p, v_{p+1}) \in \mathcal{E} \). Although Eq. 3 ensures the coherence between pixel-wise prediction and the class hierarchy during the inference stage, there is no class relation information used for segmentation network training, as the binary cross-entropy loss in Eq. 2 is computed over each class independently. To alleviate this issue, we propose a hierarchy-aware segmentation learning scheme (§3.2.2), which incorporates the semantic structures into the training of HSSN.

3.2. Hierarchy-Aware Segmentation Learning

Our hierarchy-aware segmentation learning scheme includes two major components: i) a pixel-wise hierarchical segmentation learning strategy (§3.2.1) which supervises the segmentation prediction \( \mathcal{S} \) in a hierarchy-coherent manner, and ii) a pixel-wise hierarchical representation learning strategy (§3.2.2) that makes hierarchy-induced margin separation for reshaping the pixel embedding space \( f_{\text{ENC}} \).

3.2.1 Pixel-Wise Hierarchical Segmentation Learning

For each pixel, the assigned labels are hierarchically consistent if they satisfy the following two properties (Fig. 2):

Definition 3.2.1 (Positive T-Property). For each pixel, if a class is labeled positive, all its ancestor nodes (i.e., superclasses) in \( \mathcal{T} \) should be labeled positive.

Definition 3.2.2 (Negative T-Property). For each pixel, if a class is labeled negative, all its child nodes (i.e., subclasses) in \( \mathcal{T} \) should be labeled negative.

The first property, also known as T-property [8], was explored in some hierarchical classification work [29, 77, 85], while the second property is ignored. Actually, these two properties are complementary and crucial for consistent hierarchical prediction. Specifically, to incorporate these two label consistency properties into the supervision of HSSN, we further derive the following two hierarchy constraints w.r.t. per-pixel prediction, i.e., \( s = [s_v]_{v \in \mathcal{V}} \in [0, 1]^{|\mathcal{V}|} \):

\[
\mathcal{L}^{\text{TM}}(p) = \sum_{v \in \mathcal{V}} -\hat{l}_v \log(p_v) - (1 - \hat{l}_v) \log(1 - p_v),
\]

\[
\sum_{v \in \mathcal{V}} -\hat{l}_v \log(\min_{u \in \mathcal{A}_v} (s_u)) - (1 - \hat{l}_v) \log(1 - \max_{u \in \mathcal{C}_v} (s_u)).
\]

Compared with \( \mathcal{L}^{\text{BCE}}(s) \), \( \mathcal{L}^{\text{TM}}(p) \) is more favored as the structured score distribution \( p \) is constructed by strictly following the hierarchy constraints (cf. Eq. 4), and hence the violation of the hierarchy properties (i.e., any undesired prediction of \( p \)) can be explicitly penalized (see Fig. 2(c)).

Focal Tree-Min Loss. Inspired by the focal loss [47], we add a modulating factor to the tree-min loss (cf. Eq. 5), so as to reduce the relative loss for well-classified pixel samples and focus on those difficult ones:

\[
\mathcal{L}^{\text{TM}}(p) = \sum_{v \in \mathcal{V}} -\hat{l}_v (1 - p_v) \gamma \log(p_v) - (1 - \hat{l}_v) p_v \gamma \log(1 - p_v),
\]

\[
\sum_{v \in \mathcal{V}} -\hat{l}_v (1 - \min_{u \in \mathcal{A}_v} (s_u)) \gamma \log(\min_{u \in \mathcal{A}_v} (s_u)) - (1 - \hat{l}_v) (\max_{u \in \mathcal{C}_v} (s_u)) \gamma \log(1 - \max_{u \in \mathcal{C}_v} (s_u)).
\]
where $\gamma \geq 0$ is a tunable focusing parameter controlling the rate at which easy classes are down-weighted. When $\gamma = 0$, $L^{\text{FTM}}(p)$ is equivalent to $L^{\text{TM}}(p)$. Fig. 3 shows representative visual effects of $L^{\text{FTM}}$ against $L^{\text{BCE}}$. We see that $L^{\text{FTM}}$ yields more precise and coherent results. In §4.4, we provide quantitative comparison results for $L^{\text{BCE}}(s)$ (cf. Eq. 2), $L^{\text{TM}}(p)$ (cf. Eq. 5), and $L^{\text{FTM}}(p)$ (cf. Eq. 6).

### 3.2.2 Pixel-Wise Hierarchical Representation Learning

Through mapping pixels with their corresponding semantic classes in the hierarchy $T$ as a whole (cf. §3.1), we exploit intrinsic properties of $T$ (cf. Defs. 3.2.1-3.2.2) as constraints (cf. Defs. 3.2.3-3.2.4) to encourage hierarchy-coherent segmentation prediction $S$ (cf. Eqs. 5-6). As the class hierarchy provides rich semantic relations among categories over different levels of concept abstraction, next we will exploit such structured knowledge to reshape the pixel embedding space $f_{\text{ENC}}$, so as to generate more efficient pixel representations and improve final segmentation performance.

With this purpose, we put forward a margin based pixelwise hierarchical representation learning strategy, where the learned pixel embeddings are well separated with structured margins imposed by the class hierarchy $T$. Specifically, for any pair of labels $u, v \in V$, let $\psi(u, v)$ denote their distance in the tree $T$. That is, $\psi(u, v)$ is defined as the length (in edges) of the shortest path between $u$ and $v$ in $T$. The distance function $\psi(\cdot, \cdot)$ is in fact a semantic similarity metric defined over $T$ [20]; it is a non-negative and symmetric function, $\psi(v, v) = 0$, $\psi(u, v) = \psi(v, u)$, and the triangle inequality always holds with equality.

In HSSN, the structured margin constraints are defined by the tree distance $\psi(\cdot, \cdot)$, leading to a tree-triplet loss. This loss is optimized on a set of pixel triplets $\{i, i^+, i^-\}$, where $i, i^+, i^-$ are anchor, positive and negative pixel samples, respectively. $\{i, i^+, i^-=\}$ are sampled from the whole training batch, such that $\psi(\hat{v}_i, \hat{v}_{i^+}) < \psi(\hat{v}_i, \hat{v}_{i^-})$, where $\hat{v}_i, \hat{v}_{i^+}, \hat{v}_{i^-} \in V_N$ are the groundtruth leaf labels of $i$, $i^+$, and $i^-$, respectively. As such, in our tree-triplet loss, the positive samples are more semantically similar to the anchor pixels (i.e., closer in $T$), compared with the negative pixels. Note that this is different from the classic, hierarchy-agnostic triplet loss [66], where the anchor and positive samples are from the same class, while the anchor and negative samples are from different classes, i.e., $\hat{v}_i = \hat{v}_{i^+}$, and $\hat{v}_i \neq \hat{v}_{i^-}$. With a valid training triplet $\{i, i^+, i^-=\}$, our loss is given as:

$$L^{\text{TT}}(i, i^+, i^-) = \max\{\langle i, i^+ \rangle - \langle i, i^- \rangle + m, 0\}, \quad (7)$$

where $i, i^+, i^- \in \mathbb{R}^C$ are the embeddings of $i$, $i^+$, and $i^-$, respectively, obtained from the encoder $f_{\text{ENC}}$, $\langle \cdot, \cdot \rangle$ is a distance function to measure the similarity of two inputs; we use the cosine distance, i.e., $(x, y) = \frac{1}{2}(1 - \frac{x^T y}{\|x\|\|y\|}) \in [0, 1]$. The margin $m$ forces the gap of $\langle i, i^- \rangle$ and $\langle i, i^+ \rangle$ larger than $m$. When the gap is larger than $m$, the loss value would be zero. The separation margin $m$ is determined as:

$$m = m_\varepsilon + 0.5m_\tau$$

$$m_\tau = (\psi(\hat{v}_i, \hat{v}_{i^+}) - \psi(\hat{v}_i, \hat{v}_{i^-}))/2D,$$  \quad (8)$$

where $m_\varepsilon = 0.1$ is set as a constant for the tolerance of the intra-class variance, i.e., maximum intra-class distance, $m_\tau \in [0, 1]$ is a dynamic violate margin, which is computed according to the semantic relationships among $i$, $i^+$, and $i^-$ over the class hierarchy $T$, and $D$ refers to the height of $T$.

Eq. 7 encourages $f_{\text{ENC}}$ as a hierarchically-structured embedding space (Fig. 4): pixels with similar semantics (i.e., nearby in $T$) are pushed closer than those with dissimilar semantics (i.e., faraway in $T$), guided by the hierarchy-induced margin $m$. Related experiments are given in §4.4.

### 3.3. Implementation Detail

#### Network Architecture.

HSSN is a general HSS framework; it is readily applied to any hierarchy-agnostic segmentation models. **i)** The segmentation encoder $f_{\text{ENC}}$ (§3.1) maps each input image $I$ into a dense feature $f \in \mathbb{R}^{H \times W \times C}$, and can be implemented as any backbone networks. In §4, we experiment with two CNN-based (i.e., ResNet-101 [34] and HRNetV2-W48 [79]) and a Transformer-based (i.e., Swin-Transformer [49]) backbones. **ii)** The segmentation head $f_{\text{SEG}}$ (§3.1) projects $I$ into a structured score map $S \in \mathbb{R}^{H \times W \times |V|}$ for all the classes in $V$. Segmentation heads used in recent segmentation models (i.e., DeepLabV3+ [13], OCRNet [98], MaskFormer [16]) are used and modified.

#### Training Objective.

HSSN is end-to-end trained by minimizing the combinatorial loss of our focal tree-min loss ($L^{\text{FTM}}$ in Eq. 6) and tree-triplet loss ($L^{\text{TT}}$ in Eq. 7): $L^{\text{TT}} + \beta L^{\text{FTM}}$, where the coefficient $\beta \in [0, 0.5]$ is scheduled following a cosine annealing policy [51]. The focusing parameter $\gamma$ in $L^{\text{FTM}}$ is set as 2. Furthermore, following the common practice in metric learning, a projection function $f_{\text{ROI}}$ is used in $L^{\text{TT}}$. It maps each pixel embedding $i$ into a 256-d vector. $f_{\text{ROI}}$ consists of two $1 \times 1$ convolutional layers and one ReLU between them, and is discarded after training, causing no extra computational cost in deployment.

**Inference.** For each pixel, the label assignment follows Eq. 3.
4. Experiment

4.1. Experimental Setup

Datasets. We conduct experiments on two popular urban street scene parsing datasets [18, 58] and two human body parsing datasets [44, 87]. The corresponding class hierarchies are either the officially provided ones [18, 58] or generated by following the conventions [44, 87].

- **Mapillary Vistas 2.0** [58] is an urban egocentric street-view dataset with high-resolution images. It contains 18,000, 2,000 and 5,000 images for train, val and test, respectively. It provides annotations for 144 semantic concepts, which are organized in a three-level hierarchy, covering 4/16/124 concepts, respectively.

- **Cityscapes** [18] contains 5,000 elaborately annotated urban scene images, which are split into 2,975/500/1,524 for train/val/test. It is associated with 19 fine-grained concepts, which are grouped into 6 super-classes.

- **PASCAL-Person-Part** [87] has 1,716 and 1,817 images for train and test, with precise annotations for 6 human parts. Following [80, 83], we group 20 fine-grained parts (e.g., head, left-arm) into two superclasses upper-body and lower-body, which are further combined into full-body.

- **LIP** [44] includes 50,462 single-person images gathered from real-world scenarios, with 30,462/10,000/10,000 for train/val/test splits. The hierarchy is similar to the one in PASCAL-Person-Part, but the leaf layer has 19 fine-grained semantic parts.

Training. For fair comparison, we follow [13, 80, 102, 105] to set the training hyper-parameters. Specifically, for CNN-based models, we use SGD as the optimizer with base learning rate 1e-2, momentum 0.9 and weight decay 1e-4. For Transformer-based models, we use AdamW [52] with base learning rate 6e-5 and weight decay 0.01. The learning rate is scheduled by the polynomial annealing policy [11]. All backbones are initialized using the weights pre-trained on ImageNet-1K [22], while the remaining layers are randomly initialized. During training, we use standard data augmentation techniques, i.e., horizontal flipping and random scaling with a ratio between 0.5 and 2.0. We train 240K and 80K iterations for Mapillary Vistas 2.0 and Cityscapes, with batch size 8 and crop size 512 × 1024. For PASCAL-Person-Part and LIP, we use batch size 16 and crop size 480 × 480, and train models for 80K and 160K iterations, respectively.

Testing. The inference follows Eq. 3. As in [16, 35, 36, 80, 83, 98], we report the segmentation scores at multiple scales \(\{0.5, 0.75, 1.0, 1.25, 1.5, 1.75\}\) with horizontal flipping.

Evaluation Metric. The mean intersection-over-union (mIoU) is adopted for evaluation. Particularly, we report the average score, i.e., \(\text{mIoU}^i\), for classes in each hierarchy level \(i\) independently. For reference, we also report the scores of each level for hierarchy-agnostic methods. The results of each non-leaf layer are obtained by merging the segmentation predictions of its subclasses together.

4.2. Quantitative Results

Mapillary Vistas 2.0 [58]. Table 1 presents comparisons of our HSSN against several top-leading semantic segmen-
Table 3. Hierarchical human parsing results (§4.2) on PASCAL-Person-Part [87] test. All models use ResNet-101 as the backbone.

<table>
<thead>
<tr>
<th>Method</th>
<th>Head</th>
<th>Torso</th>
<th>U-Arm</th>
<th>L-Arm</th>
<th>U-Leg</th>
<th>L-Leg</th>
<th>U-Body</th>
<th>L-Body</th>
<th>F-Body</th>
<th>B.G.</th>
<th>mIoU↑</th>
<th>mIoU↓</th>
<th>mIoU↑↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepLabV3+ [13]</td>
<td>87.02</td>
<td>72.02</td>
<td>60.37</td>
<td>57.36</td>
<td>53.54</td>
<td>48.52</td>
<td>90.07</td>
<td>65.88</td>
<td>93.02</td>
<td>96.07</td>
<td>94.55</td>
<td>84.01</td>
<td>67.84</td>
</tr>
<tr>
<td>SPNNet [15]</td>
<td>87.67</td>
<td>71.41</td>
<td>61.69</td>
<td>60.35</td>
<td>52.62</td>
<td>48.80</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td>PCN [30]</td>
<td>90.89</td>
<td>75.12</td>
<td>55.83</td>
<td>64.61</td>
<td>55.42</td>
<td>41.57</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td>CINF [80]</td>
<td>88.02</td>
<td>72.91</td>
<td>64.31</td>
<td>63.52</td>
<td>55.61</td>
<td>54.96</td>
<td>91.82</td>
<td>66.56</td>
<td>94.33</td>
<td>96.02</td>
<td>95.18</td>
<td>84.80</td>
<td>70.76</td>
</tr>
<tr>
<td>SemaTree [36]</td>
<td>89.15</td>
<td>74.76</td>
<td>63.90</td>
<td>63.95</td>
<td>57.53</td>
<td>54.62</td>
<td>92.36</td>
<td>67.13</td>
<td>95.11</td>
<td>96.84</td>
<td>95.98</td>
<td>85.44</td>
<td>71.59</td>
</tr>
<tr>
<td>HHP [83]</td>
<td>89.73</td>
<td>75.22</td>
<td>66.87</td>
<td>66.21</td>
<td>58.69</td>
<td>58.17</td>
<td>93.44</td>
<td>68.02</td>
<td>96.77</td>
<td>96.94</td>
<td>96.86</td>
<td>86.13</td>
<td>73.12</td>
</tr>
<tr>
<td>BGIN [102]</td>
<td>90.18</td>
<td>77.44</td>
<td>64.93</td>
<td>67.15</td>
<td>60.79</td>
<td>59.27</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>74.42</td>
</tr>
<tr>
<td>PCNet [101]</td>
<td>90.04</td>
<td>76.89</td>
<td>69.11</td>
<td>68.40</td>
<td>60.78</td>
<td>60.14</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
<td>74.59</td>
</tr>
<tr>
<td>HSSN DeepLabV3+</td>
<td>90.19</td>
<td>78.72</td>
<td>70.67</td>
<td>67.71</td>
<td>61.15</td>
<td>60.44</td>
<td>95.86</td>
<td>71.56</td>
<td>98.20</td>
<td>97.18</td>
<td>97.69</td>
<td>88.20</td>
<td>75.44</td>
</tr>
</tbody>
</table>

Table 4. Hierarchical human parsing results (§4.2) on LIP val.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>mIoU↑</th>
<th>mIoU↓</th>
<th>mIoU↑↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepLabV2 [10]</td>
<td>ResNet-101</td>
<td>-</td>
<td>-</td>
<td>41.64</td>
</tr>
<tr>
<td>Attention [12]</td>
<td>ResNet-101</td>
<td>-</td>
<td>-</td>
<td>42.92</td>
</tr>
<tr>
<td>MMAP [54]</td>
<td>ResNet-101</td>
<td>-</td>
<td>-</td>
<td>46.93</td>
</tr>
<tr>
<td>CEP2 [64]</td>
<td>ResNet-101</td>
<td>-</td>
<td>-</td>
<td>53.10</td>
</tr>
<tr>
<td>BraidNet [48]</td>
<td>ResNet-101</td>
<td>-</td>
<td>-</td>
<td>54.42</td>
</tr>
<tr>
<td>SemaTree [36]</td>
<td>ResNet-101</td>
<td>90.78</td>
<td>87.12</td>
<td>54.73</td>
</tr>
<tr>
<td>BGIN [102]</td>
<td>ResNet-101</td>
<td>-</td>
<td>-</td>
<td>56.82</td>
</tr>
<tr>
<td>PCNet [101]</td>
<td>ResNet-101</td>
<td>-</td>
<td>-</td>
<td>57.03</td>
</tr>
<tr>
<td>CINF [80]</td>
<td>ResNet-101</td>
<td>95.92</td>
<td>91.83</td>
<td>57.74</td>
</tr>
<tr>
<td>HRNet [79]</td>
<td>HRNet-W48</td>
<td>95.53</td>
<td>91.21</td>
<td>57.23</td>
</tr>
<tr>
<td>OCRNet [98]</td>
<td>HRNet-W48</td>
<td>96.78</td>
<td>92.56</td>
<td>58.47</td>
</tr>
<tr>
<td>HHP [83]</td>
<td>HRNet-W48</td>
<td>97.41</td>
<td>93.43</td>
<td>59.25</td>
</tr>
<tr>
<td>HSSN DeepLabV3+</td>
<td>ResNet-101</td>
<td>98.86</td>
<td>94.75</td>
<td>60.37</td>
</tr>
</tbody>
</table>

Fig. 5 and Fig. 6 depict representative visual results on four datasets. As seen, HSSN yields more precise segmentation results in comparison with some top-performing methods (i.e., MaskFormer in Fig. 5 and DeepLabV3+ in Fig. 6), and shows strong robustness to various challenging scenarios with occlusions, small objects and densely arranged targets, etc. Moreover, as shown in the last column of Fig. 5, MaskFormer makes a severe mistake that misclassifies a part of background structure as truck. In contrast, benefiting from hierarchy-aware segmentation learning, HSSN naturally address the issue of mistake severity, i.e., distinguishes significantly different concepts with larger margins.

4.3. Qualitative Results

To gain more insights into HSSN, we conduct a set of ablative studies on Mapillary Vistas 2.0 [58] and PASCAL-Person-Part [87], with ResNet-101 as the backbone.

**Key Component Analysis.** First, we investigate the essential designs in HSSN, i.e., hierarchical segmentation learning (§3.2.1) with $L_{TT}^{PTM}$ (cf. Eq. 6) and hierarchical representation learning (§3.2.2) with $L_{TTC}^{EE}$ (cf. Eq. 7). The results are summarized in Table 5. The first row refers to a hierarchy-agnostic baseline that only concerns the leaf nodes and is trained using the categorical cross-entropy loss $L_{CCE}^{E}$ (cf. Eq. 1). Three crucial conclusions can be drawn. First, our $L_{TT}^{PTM}$ leads to significant performance improvements against the baseline across all the metrics on both datasets. This evidences that our hierarchical segmentation
learning strategy is able to produce hierarchy-coherent predictions. Second, we also observe compelling gains by incorporating $L^{TT}$ into the baseline. This proves the importance of hierarchical representation learning. Third, our full model achieves the best performance by combining our $L^{FTM}$ and $L^{TT}$ together, confirming the necessity of joint hierarchical segmentation and embedding learning.

**Focal Tree-Min Loss.** We next examine the design of our focal tree-min loss $L^{FTM}$ (cf. Eq. 6). As shown in Table 6, we compare $L^{FTM}$ with four different losses, i.e., categorical cross-entropy loss $L^{CCE}$ (cf. Eq. 1), binary cross-entropy loss $L^{BCE}$ (cf. Eq. 2), focal loss [47], and our tree-min loss $L^{TM}$ (cf. Eq. 5). We can see that our $L^{TM}$ generates impressive results, and $L^{FTM}$ is even better than $L^{TM}$. Then, in Table 7, we analyze the impact of the focusing parameter $\gamma$ in $L^{FTM}$.

The loss of $\gamma$ is increased, and the gain becomes marginal when $\gamma = 2$. Hence, we choose $\gamma = 2$ by default.

**Tree-Triplet Loss.** We further investigate the design of our tree-triplet loss $L^{TT}$ (cf. Eq. 7). In Table 8, “Vanilla” refers to the vanilla triplet loss with a constant margin [66]. By constructing hierarchy-aware triplet samples, our tree-triplet loss $L^{TT}$ (also with a constant margin) outperforms “Vanilla”. The gains become larger when further applying the hierarchy-induced margin constraint. These results confirm the designs of our tree-triplet loss. Finally, we assess the impact of the distance measurement $\langle \cdot , \cdot \rangle$ used in $L^{TT}$. We study Cosine and Euclidean distances. Table 9 shows that Cosine distance performs much better than Euclidean distance, corroborating relevant observations in [26, 59, 65].

5. Conclusion

In this paper, we presented HSSN, a structured solution for semantic segmentation. HSSN is capable of exploiting taxonomic semantic relations for structured scene parsing, by only slightly changing existing hierarchy-agnostic segmentation networks. By exploiting hierarchy properties as optimization criteria, hierarchical violation in the segmentation predictions can be explicitly penalized. Through hierarchy-induced margin separation, more effective pixel representations can be generated. We experimentally show that HSSN outperforms many existing segmentation models on four famous datasets. We wish this work to pave the way for future research on hierarchical semantic segmentation.

**Acknowledgements**

This work was supported in part by the Beijing Natural Science Foundation under Grant L191004, CCF-Baidu Open Fund, and ARC DECRA DE220101390.
References


[10] 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. TPAMI, 2017. 2, 6, 7


[16] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In NeurIPS, 2021. 2, 5, 6, 7


[18] 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 CVPR, 2016. 2, 6, 7


[21] Jia Deng, Alexander C Berg, Kai Li, and Li Fei-Fei. What does classifying more than 10,000 image categories tell us? In ECCV, 2010. 3

[22] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In CVPR, 2009. 6


[41] Xiaotai Li, Xia Li, Li Zhang, Guangliang Cheng, Jianping Shi, Zhouchen Lin, Shaohua Tan, and Yuhai Tong. Improving semantic segmentation via decoupled body and edge supervision. In ECCV, 2020. 2


[44] Xiaodan Liang, Ke Gong, Xiaohui Shen, and Liang Lin. Look into person: Joint body parsing & pose estimation network and a new benchmark. TPAMI, 2018. 2, 6, 7, 8


