Received 7 May 2024, accepted 18 June 2024, date of publication 24 June 2024, date of current version 1 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3418515

RESEARCH ARTICLE

Multi-Level Segmentation Data Generation Based on a Scene-Specific Word Tree

SOOMIN KIM AND JUYOUN PARK
Korea Institute of Science and Technology, Seoul 02792, Republic of Korea

Corresponding author: Juyoun Park (juyounpark@kist.re.kr; juyoun726@gmail.com)

This work was supported by Korea Institute of Science and Technology (KIST) Institutional Programs [Project No. 2E33003 and 2E32992].

ABSTRACT We, humans, perceive the scene utilizing pre-learned language categories. Our vocabulary system inherently possesses a hierarchy, aiding humans in understanding scenes at multiple levels. For example, when a person passes by chairs and desks from a distance rather than interacting with them up close, the objects are perceived from a broader perspective and recognized as furniture at a higher category level. In this work, we propose a multi-level semantic segmentation data generation method based on a scene-specific word tree to mimic human multi-level scene recognition. Multi-level semantic segmentation data encompasses diverse levels of grouped segmented areas with different degrees of detail, from the finest level of conventional semantic segmentation to coarser levels. Our scene-specific word trees leverage linguistic hierarchies to group scene components by considering relationships between words present in the scene. Furthermore, in the proposed data generation method, each word tree is constructed within a single image, allowing us to group the objects into user-selected levels, taking into account the relative relationship between objects in that scene. We demonstrate the effectiveness of our data generation method by building a multi-level scene segmentation network and training the model with the generated dataset, which reflects the scene-specific word tree.

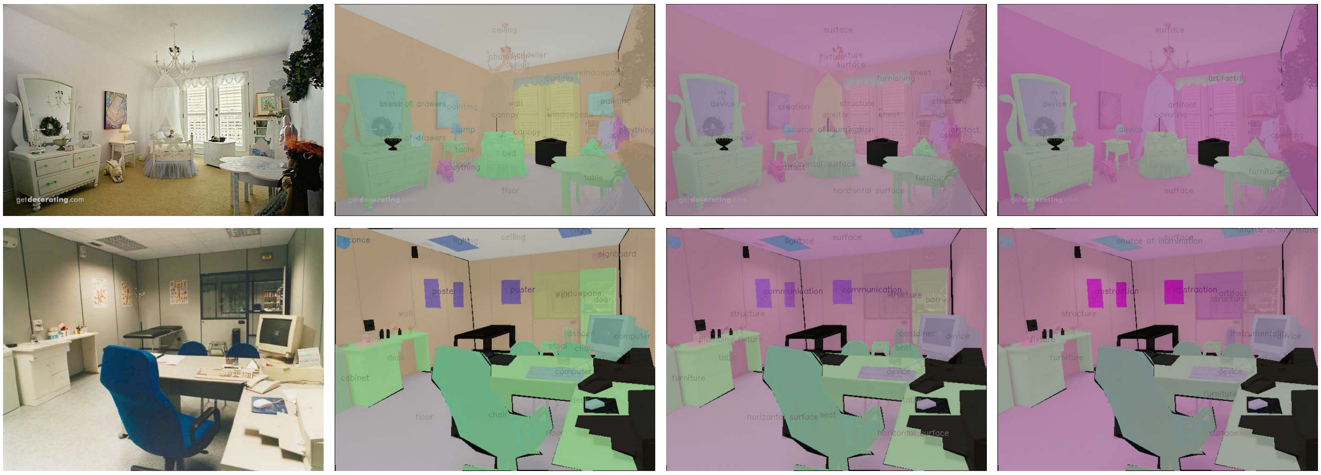
INDEX TERMS Segmentation, semantic grouping, language hierarchy, dataset generation, multi-level analysis.

I. INTRODUCTION

Humans typically perceive the scene with a single glance, even when there are numerous objects or living beings present. Normally, we do not pay attention to many details of the scene unless we need to focus on a specific group. For instance, when observing a scene like the upper image of Figure 1, at first glance, we recognize that there are pieces of furniture in the room like Level 2. However, as we focus more, we begin to recognize the table and chair as Level 1. Subsequently, in situations such as when using a particular object, humans recognize each object separately, such as the desk, chair, and seat as in the Level 0 scene. In other words, humans perceive and group objects existing in the same scene into different conceptual levels depending on the context. This unconscious grouping is related to our pre-learned language categories. The ability to perceive objects at multiple levels and group them, as we humans do, can be utilized to implement agents such as mobile robots with advanced intelligence capable of behaving like humans according to the situation. The human-like visual intelligence can be particularly useful when mobile robots are navigating or moving. When a mobile robot intends to enter a building, the robot needs to perceive the building at a coarse level from a distance, and as it approaches, it needs to understand the scene at a finer level to locate the door. Once the robot finds the door, it needs to identify the door knob to open it and enter. Therefore, we aim to implement visual intelligence for computers that recognizes and groups objects into different levels depending on the context, similar to how humans do.

For a computer to understand a scene or image, image segmentation is a crucial task. With the rise of Convolutional

The associate editor coordinating the review of this manuscript and approving it for publication was Byung-Gyu Kim.



(a)RGB image (b) Level 0 segmentation (c) Level 1 segmentation (d) Level 2 segmentation

FIGURE 1. Our proposed scene-specific word tree incorporated dataset examples. The objects in the finest level (b) are merged as the level increases (c, d). For example, 'chair' and 'stool' from (b) are merged into 'seat' in (c) and further merged into 'furniture' in (d). Moreover, 'drawers', 'table', and 'bed' on level 0 are merged into 'furniture' at level 2 in the end. Also, since we build a scene-specific word tree for generating multi-level, depending on the scene configuration, grouping is done differently. For example, upper image objects are grouped more compared to the objects in the lower image, which includes various types of objects.

Neural Network (CNN)-based image classification [1], [2], [3], [4], large-scale ground-truth labeled datasets become available which lead image segmentation methods to advance as well. Accordingly, the necessity of large faithful semantic segmentation datasets is also increasing. Even though there have been various large-scale datasets with different levels of objectiveness, range of details, and scene properties [5], [6], [7], [8], [9], most of them are labeled with a single object per pixel, making them unsuitable for training human cognitive models that recognize objects at multiple levels. Therefore, we propose a method for generating multi-level segmentation datasets capable of recognizing and grouping objects within the same scene into different levels depending on the context by integrating pre-learned linguistic knowledge similar to humans. Even though UPerNet [10] and All-Inclusive Multi-Level Segmentation (AIMS) [11] proposed to segment areas into multiple levels such as objects and parts, they do not utilize the hierarchy of objects themselves. Our proposed method utilizes object hierarchies and generates semantic multi-level dataset, which can help teach the computer to perceive the scene as humans do.

Recently, with the advent of Large Language Models (LLMs) [12], [13], [14], the use of word semantic knowledge has increased, and it has even become possible to solve various problems on an open-vocabulary basis using learned models. In addition, it has been integrated with data from other domains such as images (Contrastive Language-Image Pre-Training; CLIP [14]), making multi-modal applications possible. However, LLMs, which learn given words as feature points in the same space, cannot contain information about superordinate words (hypernyms) and cannot handle homonyms. Therefore, when building a scene-specific word tree in our method, we not only utilize extensive open-vocabulary word semantic information by applying LLM but also it is based on an open word database such as WordNet [15] when dealing with homonyms or hypernyms. In other words, we propose a method that combines a state-of-the-art learning-based approach and a conventional word tree method that includes structured language information to compensate for the shortcomings of both methods and take advantage of each of them.

For data generation, we utilize our proposed scene-specific word tree-building method to reconstruct the existing scene segmentation dataset with multiple levels of coarseness, as shown in Figure 1. Given RGB scene images (column (a) in Figure 1) and the corresponding scene segmentation data (Level 0 segmentation data; column (b) in Figure 1), the proposed method generates Level 1 and Level 2 segmentation data (columns (c) and (d) in Figure 1, respectively). For example, 'chair' and 'stool' from (b) are merged into 'seat' in (c), and in (d), they are merged into 'furniture'. To respond to the distinct scene configuration, we build a scene-specific word tree, which is an individual word tree representing the relationships of objects in each scene. Since the word tree is composed of object labels existing in one image rather than all the words in the dataset, the proposed method varies the degree of grouping objects differently for each scene. In other words, it enables a relative visual understanding of each scene as perceived by human beings. For example, when an armchair and a desk appear in one scene, they are recognized as furniture in the next level, but when an armchair and a stool appear in one scene, they are recognized as a chair in the next level. Furthermore, we build a multi-level scene segmentation grouping network and train it with our scene-specific word tree incorporated datasets to demonstrate the effectiveness of our proposed dataset generation method.

In summary, our contributions are as follows:

- Present a novel scene-specific word tree-building method for generating the multi-level segmentation dataset, which contains the hierarchy of each scene driven from linguistic information.
- Propose a homonym handling method using pretrained CLIP features.
- Propose a multi-level scene segmentation network that outputs the coarser levels of semantic segmentation labels containing the scene-specific word tree.

II. RELATED WORK

A. SEMANTIC SEGMENTATION DATASETS

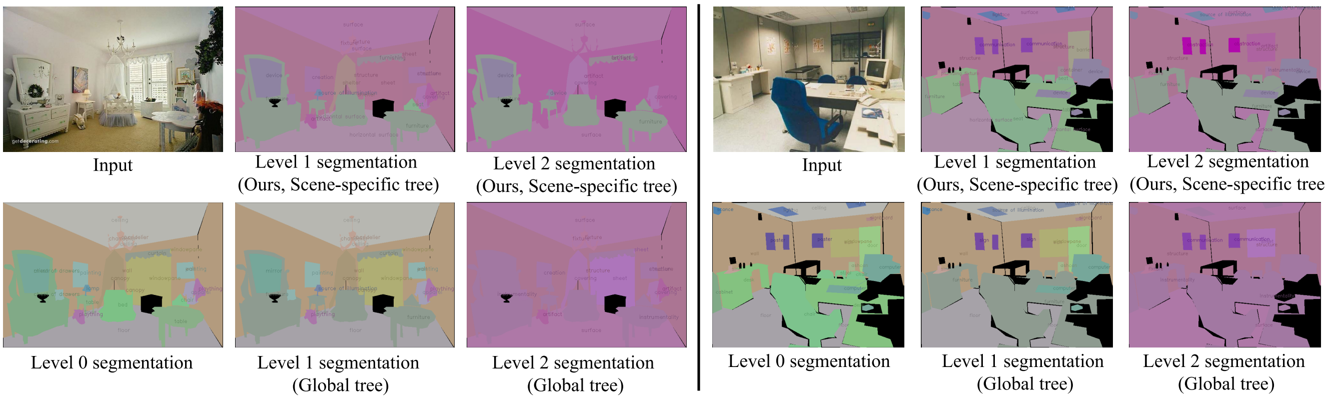
Numerous datasets have been introduced for specific purposes, such as object recognition [5], [16], [17], semantic understanding of scenes [6], [7], [9], [18], and object parts recognition [19], [20], [21], [22]. PASCAL VOC [5] and MS COCO [6] are widely used large-scale datasets with pixel-level labels. However, their annotations are limited to subsets of existing foreground objects. There are several datasets with denser annotation as well, for example, PASCAL-context [23] includes pixel-wise labels for all training images with more than 400 classes. Furthermore, SUN database [24] for scene categorization, Cityscapes dataset [7] for semantic understanding of urban street scenes, and NYU Depth V2 [8] for various indoor scenes recorded by both RGB and depth cameras from the Microsoft Kinect, were proposed. Also, ADE20K [9], a densely annotated dataset with parts information for scene parsing, was published. Although there are many different datasets for semantic segmentation, most of the datasets consist of single object labels per pixel, making them unsuitable for training human visual intelligence to recognize objects at different conceptual levels across scenes.

There are a few datasets that provide a tree structure to represent the hierarchical relationship of objects. The Mapillary Vistas 2.0 [25] and Cityscapes [7] datasets contain images from an egocentric perspective in urban environments such as roads. The PASCAL-Person-Part [26] and LIP [27] datasets provide segmentation data for each part of the human body, and the hierarchical relationship between them is represented in a tree structure. However, since these datasets represent the hierarchical relationship between objects in a fixed tree structure, they are not segmented reflecting scene characteristics such as the diversity of objects in the scene. Therefore, we propose a method to generate multi-level segmentation data based on creating different object hierarchy trees considering the composition and diversity of different objects across scenes (scene-specific trees). Unlike existing datasets, where objects are grouped into single hierarchical concepts irrespective of the scene, our approach allows for context-aware cognition by grouping objects differently into higher-level concepts depending on the situation. This enables the generation of datasets that facilitate human visual intelligence learning.

B. SCENE/SEMANTIC SEGMENTATION

CNN-based image classification such as AlexNet [1], VGGNet [2], ResNet [3], and GoogLeNet [4] shows remarkable results and those successes are extended for semantic segmentation as well since they can be used for generating pixel-wise labels. Noh et al. [28] introduced semantic segmentation with encoder-decoder models. RefineNet [29] applies a coarse-to-fine structure by multi-path refinement network for utilizing all pixel information available. The Pyramid Scene Parsing Network (PSPNet) [30] utilizes spatial pooling at multiple grid scales and shows impressive results. Zhang et al. [31] performed instance segmentation based on semantic attention and scale complementary network. SegFormer [32] utilizes a hierarchical transformer encoder to output multi-scale features and lightweight multilayer perception (MLP) decoders for combining those features for representing both local and global attention. However, the mentioned methods segment each pixel into a single object concept; thus, they cannot implement human visual intelligence to differentiate and recognize one scene at different levels depending on the context. Recently, methods for performing multi-level scene segmentation have also been proposed. Xiao et al. [10] proposed UPerNet for finer-level segmentation, such as textures, parts, and material as well as conventional object segmentation. AIMS [11] segments areas into three levels: part, entity, and relation. Although these two studies are called multi-level segmentation, they are multi-task models that classify regions according to multiple criteria rather than recognizing objects at different category levels. SceneScript [33] generates scene models incorporating structured language commands using an autoregressive, token-based approach. They employ LLM to understand scenes; However, their method requires multiple images of one scene with different angles for generating full 3D scenes, and they do not group detected objects hierarchically.

A few studies considering the hierarchical relationship between objects in scene segmentation have been conducted. In [34], hierarchical cross-entropy (HXE) loss, which reflects high-level object information, is used for training based on a word tree that represents the relationship between objects. Relationship-enhanced semantic graph (ReSG) model [35] learns locally discriminative semantic concepts. Mask2Former [36] improves the performance of scene segmentation by considering the surrounding area of an object from fine to coarse. The mentioned methods reflect the hierarchical relationships of objects when performing conventional scene segmentation, meaning they only leverage the hierarchical nature of objects in a single-level segmentation task. In contrast, Li et al. [37] proposed a model that performs multi-level scene segmentation. However, in that paper, a dataset containing multi-level object information is not proposed, and experiments are only performed on a dataset with object classes with an existing single hierarchy. In other words, because the model is based on a single object hierarchy rather than configuring a hierarchy that



Input

Level 1 segmentation
(Ours, Scene-specific tree)

Level 2 segmentation
(Ours, Scene-specific tree)

Input

Level 1 segmentation
(Ours, Scene-specific tree)

Level 2 segmentation
(Ours, Scene-specific tree)

Level 0 segmentation

Level 1 segmentation
(Global tree)

Level 2 segmentation
(Global tree)

Level 0 segmentation

Level 1 segmentation
(Global tree)

Level 2 segmentation
(Global tree)

FIGURE 2. Segmentation comparison between our scene-specific tree and global tree. The results for two scenes selected from the ADE20K dataset are shown in the left and right parts, respectively. For each image, the level 1 and 2 segmentation labels generated from the level 0 segmentation for the scene in the left column are listed as rows. (Top: based on our scene-specific tree, Bottom: based on a global tree).

takes into account the situation of each scene, scene-specific characteristics such as relationships with surrounding objects are not reflected.

C. OBJECT RECOGNITION WITH SEMANTIC HIERARCHY

A husky dog is represented as a species of dog, which is a mammal and, more broadly, a living creature. As such, there are hierarchical associations between objects, and a graph representing these relationships is often used for object recognition. The Dual Accuracy Reward Trade-off Search (DARTS) algorithm classifies object images into object categories based on semantic hierarchy [38]. The Hierarchy and Exclusion (HEX) graph captures the subsumption, exclusion, and overlap relationships between objects, and based on these, an object classification model was proposed [39]. In [40], based on the hypernym relationships between words from an open word database such as WordNet [15], semantic hierarchy is constructed and utilized for scene parsing. Also, Cao et al. [41] proposed a framework to measure the strength of interactions among objects within an image. In these studies, object segmentation/detection is performed on the leaf nodes of the semantic hierarchy, so objects which belong to the same superclass are not grouped together. Also, since the hierarchy including the entire objects is created in advance, so the correlations of objects in a specific scene are hard to be known.

III. METHODOLOGY

A. MOTIVATION

As an initial step toward realizing human visual intelligence, we propose a method to generate multi-level data. Our motivation is to aggregate existing scene segmentation data into multi-level segmentation data to actively make use of a massive existing dataset. In particular, we consider the fact that a person perceives a scene at multiple levels using different criteria based on the type and composition of objects present in the scene. This effect cannot be achieved using existing global single-object word trees. In this context, our contribution is that we propose a *scene-specific word tree* generation technique that creates a word tree for each scene rather than using a *global tree* applied to all scenes. Figure 2 shows the difference between the segmentation according to the level applied with the global tree (lower) and those with our scene-specific tree (upper). Using a global word tree for multi-level scene segmentation applies the same level-setting standard to all scenes. Specifically, a global word tree was constructed using the hypernyms of objects at level 0(finest level). Based on the global tree depth, and the labels existing in each part were designated as levels 1 and 2 equally across all the data. In this case, the unique object compositions of individual scenes, such as the number and variety of existing objects, cannot be reflected fully. For example, the left side image of Figure 2 contains relatively similar objects, which results in the entire scene being split into two level concepts with a global tree. On the other hand, our scene-specific word tree can address this issues by performing segmentation to different degrees depending on the characteristics of each scene.

B. SYNSET SELECTION WITH CLIP

Given an image, we aim to derive word-semantic relationships between objects in the scene and group objects that belong to similar categories. In order to find the relationship between the objects, a scene-specific word tree is created from input images and segmentation labels corresponding to each of them, which are generated based on a public word database such as WordNet [15]. The name of the object given as the segmentation label is searched in the word database. Since the WordNet database provides word information and its hypernyms, we utilize this hypernym information to build a scene-specific word tree.

WordNet handles homonyms of the word by assigning different synsets to each homonym. For example, *plant* in the WordNet has multiple synsets, one for *vascular plants* and another for *power plants*. Selecting correct synset is essential for generating plausible scene-specific word tree. Figure 4 shows how wrong synset selection affects the grouping of a scene-specific word tree. Suppose we have

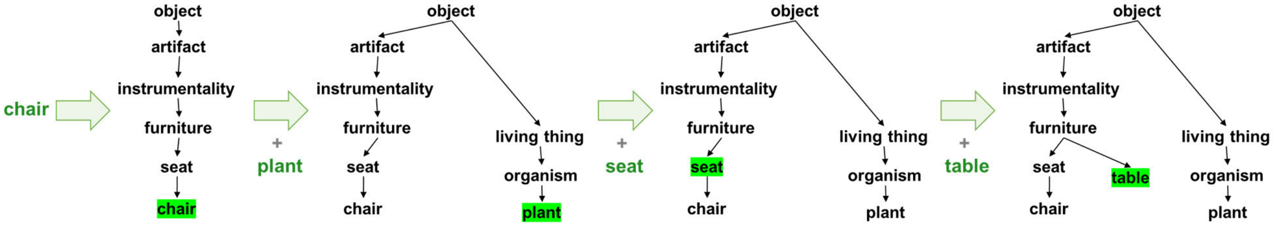


FIGURE 3. Illustration of the process for generating a scene-specific word tree. When the initial word 'chair' is given, the path formed by its inherited hypernyms creates a subtree and serves as the starting point for tree generation. Subsequent words are given, and the paths formed by their hypernyms merge with the existing subtrees to form the scene-specific word tree.

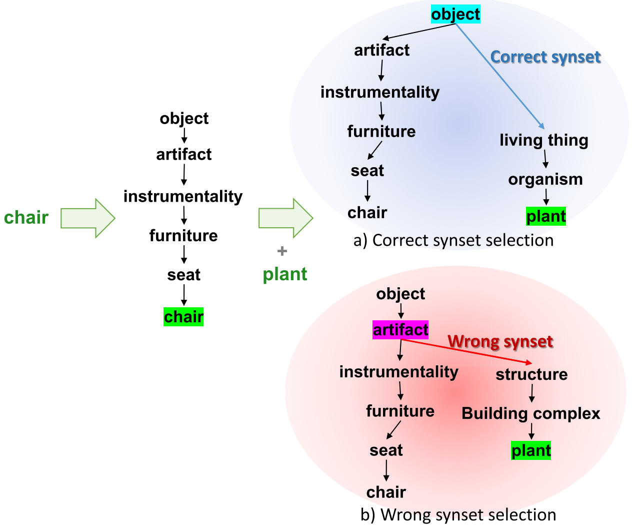


FIGURE 4. Illustration of how wrong synset selection affects grouping of the scene-specific word tree. If the synset of 'plant' is selected incorrectly (b), referring to a power plant, its inherited hypernyms are totally different from the correct ones, resulting in semantically incorrect groupings.

a plant pot in the room for an indoor scene image. If the synset of 'plant' is selected incorrectly (b), meaning a power plant, its inherited hypernyms are totally different from the correct ones, grouping semantically incorrect. As a result, the common ancestor of 'chair' and 'plant' become 'artifacts' rather than 'objects.' Since most dataset label do not specify which homonym they refer to in their label words, we propose to utilize pretrained CLIP text and image features [14] to handle this issue.

CLIP is a type of vision-language model (VLM) that learns the mutual relationship between images and text by aligning the information extracted from each data type in the same space through Contrastive Learning [14]. The CLIP model is trained so that the inferred image and text features have a high cosine similarity in each image-text pair. Therefore, to fully utilize the pretrained CLIP features, we crop the RGB images according to the given per-pixel ground truth object labels. Then, for each label word, there are cropped RGB images corresponding to multiple objects in the scene.

To select the correct synset from multiple candidate synsets (homonyms) belonging to a given label word, we utilize the CLIP image features described earlier and the hypernym sets provided by WordNet [15] for each synset word. Specifically, we use a function provided by the Python library NLTK, which allows obtaining WordNet corpus information, to output inherited hypernyms for given a synset. We calculate the confidence values between the actual meaning of the object in the image and each candidate synset to select the synset with the highest confidence value. For obtaining confidence value, the following three assumptions are made beforehand. First, we assume that multiple synsets with different meanings have different sister words and hypernyms. For example, the first synset of *plant* has *industrial plant* as a sister word and *building complex* as a hyernym, while the second synset has *flora* as a sister word and *organism* as a hypernym. Therefore, the cosine similarity between the CLIP text features of these sister words or hypernyms and the CLIP image features of the cropped object image can contribute to a confidence value that indicates whether each candidate synset refers to the object in that image. Second, we assume that the order of the WordNet synset list reflects the frequency of word use in everyday life, meaning that front-listed synsets are likely to appear more often than the latter ones. For example, the word *grass* usually means *lawn* rather than *police informer* in daily life. Third, we assume that the synset selected by different cropped object images for the same label word in succession is likely to be the correct synset.

With the above first and second assumptions, we define the cost function that calculates the confidence of each synset. For word $w_i \in labels$, if there are j_{max} number of candidate synsets, one of candiate synsets is $Syn_j^{w_i}$, and each cropped image for w_i is $I_{w_i}^k$ where $k \in [1, k_{max}]$. Here j_{max} is the number of synonyms of w_i, and k_{max} is the number of cropped images that contain w_i. $Sis[Syn_j^{w_i}]$ represents the sister word of given synset $Syn_j^{w_i}$ and $Hyp[Syn_j^{w_i}]$ is the hypernym of given synset $Syn_j^{w_i}$. Also for considering the synset order from WordNet, we define the weight $\gamma_j \in [1, 0.1]$ that weighs more for front synsets and weighs less for latter synsets with the gap of $1/j_{max}$. Therefore, confidence $Conf$ of synset $Syn_j^{w_i}$ with the image $I_{w_i}^k$ is as follows.

$$\begin{aligned} Conf\,(Syn_j^{w_i}, I_{w_i}^k) &= \gamma_j[\lambda_1 CLIP_{text}(Sis[Syn_j^{w_i}])) \cdot CLIP_{img}(I_{w_i}^k) \\ &\quad + \lambda_2 CLIP_{text}(Hyp[Syn_j^{w_i}]) \cdot CLIP_{img}(I_{w_i}^k)] \end{aligned} \tag{1}$$

Since we have k_{max} number of cropped object images for each w_i, we select the synset that has the highest confidence value consecutively for $I^k_{w_i}$. Here, we assign λ_1 to be 0.25 and λ_2 to be 0.75. When selecting a synset, synsets whose inherited hypernyms contain words such as *psychologicalfeature* or *yangity* are skipped. Because these words cannot correspond to the meaning of objects present within the scene. Also, the word *glass*'s synset is pre-assigned since its transparency affects badly to CLIP features.

C. SCENE-SPECIFIC WORD TREE GENERATION

Since we select the appropriate synset for each word in the previous step, we can acquire inherited hypernyms of the word as well. These hypernym words become nodes forming a path from a leaf node representing the corresponding object to the root of the word tree. The connection of words constituting this path is considered as a sub-word tree, and after obtaining these sub-trees for all words, they are combined to form a scene-specific word tree, as illustrated in Figure 3. The figure briefly depicts the process of creating a scene-specific word tree when the words 'chair', 'plant', 'seat', and 'table' are given in sequence. Notably, if a word like 'seat' that corresponds to a node in the middle of the existing subtree is given, the tree remains unchanged; and if a word like 'table', where some of its hypernyms already exist in the subtree, is given, the remainder of the path merges below the lowest common hypernym.

The scene-specific tree samples generated through the described process are shown in Figure 5. The figure visualizes a scene-specific word tree generated for each image on the left. Unlike a single global word tree, as scenes become more complex, meaning there are diverse and numerous objects within the image, the number of scene-specific word tree nodes increases, making the structure more complex.

D. MULTI-LEVEL SEGMENTATION LABEL EXTRACTION

Generated word tree is utilized for extracting user-defined D multi-level segmentation label. Let a scene-specific word tree $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of node size $|\mathcal{V}| = N$ and each node $v \in \mathcal{V}$. A directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a set of vertices and $\mathcal{E}$ is a set of directed edges. Each node v in the tree is associated with the corresponding object name and information about the area occupied by the object such as pixel coordinates. D denotes the maximum level of the segmentation label we want to extract, and $\mathcal{G}$ denotes a scene-specific word tree. D is a value that determines how many levels of labels will be extracted when applying the proposed data label generation method. There is no maximum limit on how many times the label extraction process can be performed within the algorithm. However, as words are grouped into higher-level concepts, they eventually fall under the top-level concept of 'entity'. Empirically, when exceeding 3 levels, most of the images are annotated as 'entity'. Therefore, in this paper, we set D to 3 $\mathbf{W}_0$ represents a set of segmentation labels provided by the original dataset for a specific scene. In the proposed data generation method, we generate a base word tree $\mathcal{G}_b$ and define the set of words included in it as $\mathbf{W}_b$. $\mathcal{G}_b$ is a sub-tree that is built by merging paths composed of hypernym words of each word belonging to $\mathbf{W}_0$ and performing refinement by deleting all the nodes having one child node. $\mathcal{G}_b$ can be seen as an essential word tree for all the scenes in the original dataset. The indegree of v is denoted as $deg^-(v)$ and its outdegree is denoted as $deg^+(v)$.

Algorithm 1 Multi-Level Segmentation Label Extraction

```
Input $D > 0$, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathbf{W}_0$, $\mathbf{W}_b$
Output $\mathbf{L}_d$ for $d \in [0, D-1]$
$Leaf(\mathcal{G})$ denotes leaf nodes of graph $\mathcal{G}$
$Root(\mathcal{G})$ denotes the root node of graph $\mathcal{G}$
$d \leftarrow 0$
$\mathbf{T} \leftarrow \{v | v \in \mathbf{W}_0, deg^+(v) \neq 0\}$
$\mathbf{R} \leftarrow \varnothing$
$\mathbf{L}_d \leftarrow \varnothing$ for $d \in [0, D-1]$
for $v \in \mathcal{V}$ do
    if $deg^-(v) > 0, deg^+(v) = 1, v \notin \mathbf{W}_b$ then
        Remove $v$ from $\mathcal{G}$ // If the node has a child node, the child node is connected to the parent node of the node as a child node.
    end if
end for
while $d < D$ do
    $\mathbf{T}' = \{v | v \in \mathbf{T}, deg^+(v) \neq 0\}$
    $\mathbf{T}' = \mathbf{T}' \cup \{v | v \in \mathbf{T}, v = Root(\mathcal{G})\}$
    for $v \in Leaf(\mathcal{G})$ where $v \neq Root(\mathcal{G})$ do
        $\mathbf{R} = \mathbf{R} \cup \{v\}$
        Deliver node attributes to the parent node $v_p$ of $v$ // Combining the node's object area information into the parent node's object area information
        $\mathbf{T}' = \mathbf{T}' \cup \{v_p\}$
    end for
    $\mathbf{L}_d = \mathbf{T} \cup Leaf(\mathcal{G})$
    $\mathbf{T} = \mathbf{T}'$
    $\mathbf{T}' = \varnothing$
    for $v \in \mathbf{R}$ do
        Remove $v$ from $\mathcal{G}$ // If the node has a child node, the child node is connected to the parent node of the node as a child node.
    end for
end while
```

When a scene-specific word tree corresponding to an image is given, redundant nodes are deleted from the word tree and the segmentation labels composing each level are extracted, following the procedure described in Algorithm 1. For the words belonging to $\mathcal{G}_b$, we do not delete them when refining the scene-specific word tree so that the objects in the scene can be grouped into meaningful upper categories even in scenes with few types of objects. The words of the base tree are considered crucial because they appear

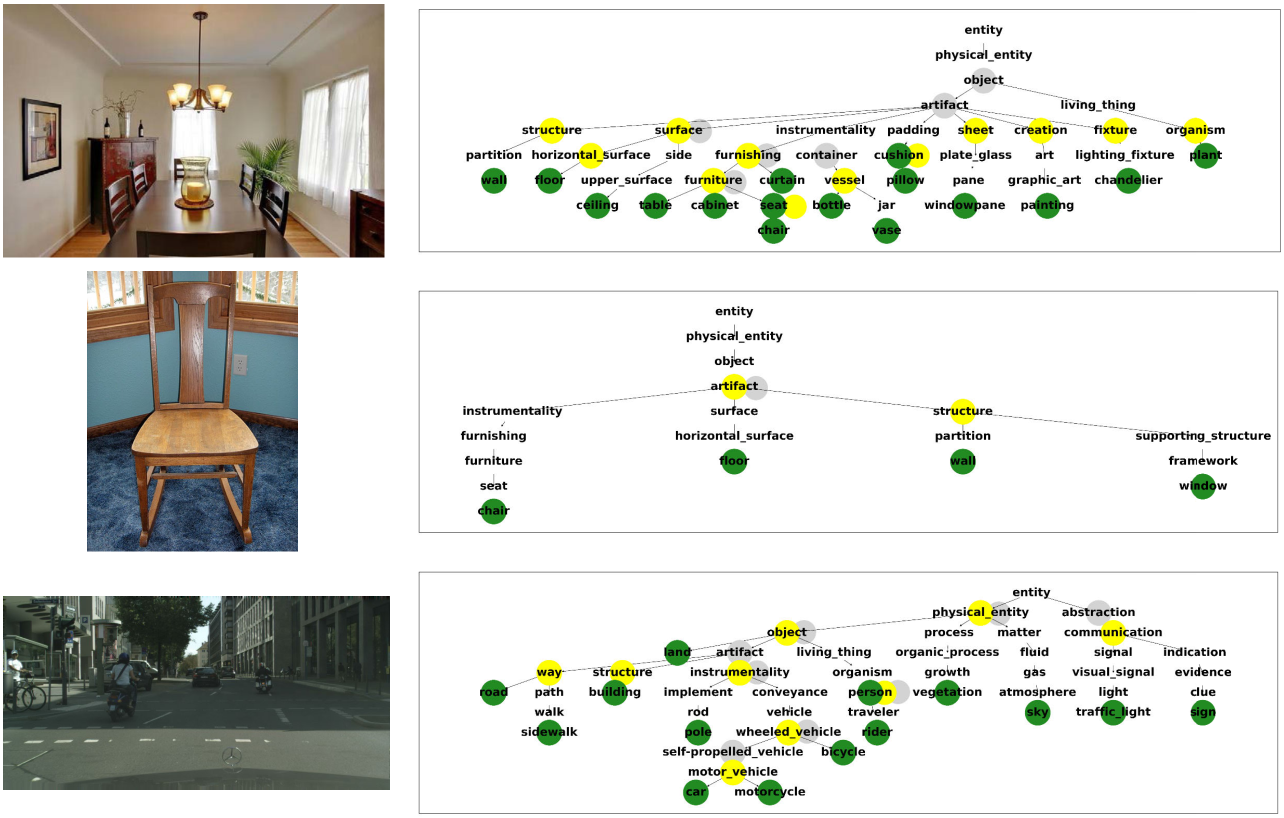


FIGURE 5. Samples of generated scene-specific word trees. A scene-specific word tree in the right column is generated from each input image in the left column. Segmentation labels for each level are extracted from the scene-specific word tree. Green, yellow, and gray nodes represent level 0, level 1, and level 2 segmentation labels, respectively.

relatively more frequently in the entire dataset. Therefore, during tree trimming, the words of the base tree are prevented from being removed. Therefore, by utilizing the base word tree $\mathcal{G}_b$, crucial words remain in the scene-specific word tree, which helps global consistency across the entire dataset when generating coarse-level scene segmentation results according to the scene-specific word tree.

$\mathbf{L}_d$ a set of segmentation levels is extracted for each level d. When a node is removed from $\mathcal{G}$, if the node has a child node, the child node is connected to the parent node of the node as a child node (Algorithm 1 Lines 9 and 24). Delivering a node's attributes to a parent node stands for the process of combining the node's object area information into the parent node's object area information (Algorithm 1 Line 17). A higher-level concept is created that includes lower-level objects, thereby grouping objects within the same category. Samples of segmentation labels for each level extracted from each scene-specific word tree are shown in Figure 5.

E. MULTI-LEVEL SEGMENTATION DATA GENERATION AND VISUALIZATION

Figure 6 shows examples of multi-level segmentation data generated by our method. Each row in the figure shows the scene image, level 0 segmentation label, and generated level 1 and 2 segmentation labels for the ADE20K, PASCAL-Context, and Cityscapes datasets, respectively. As the level increases, grouping of objects occurs, resulting in coarser scene segmentation. Moreover, the proposed multi-level extraction method is based on scene-specific word trees, and the algorithm can handle complex multi-object images as in the first row as well as single-object images as in the second row by reflecting the scene-specific characteristics. Also, for colorization, we want our dataset to have 1) each object have a unique color, and 2) objects with similar semantic meanings have similar colors. In order to handle the above two requirements, we first build an entire word tree $\mathbb{G}$, that contains all the words across the dataset $\mathbb{W}$, which are total unions of all words for each scene image $\mathbf{W}$. Second, we utilize the graph layout method called Kamada-Kawai algorithm [42] on 256 × 256 × 256 RGB space for assigning colors. Kamada-Kawai cost function simulates spring forces according to graph theoretical distances, and each graph node is located to minimize this cost function. In this way, we are able to color semantically similar objects in the dataset with similar colors.

In the proposed data generation method, the part added to the existing segmentation dataset is the word tree information generated from no more than 20 words existing in each scene image. Moreover, this information is generated and deleted for each scene image when creating multi-level

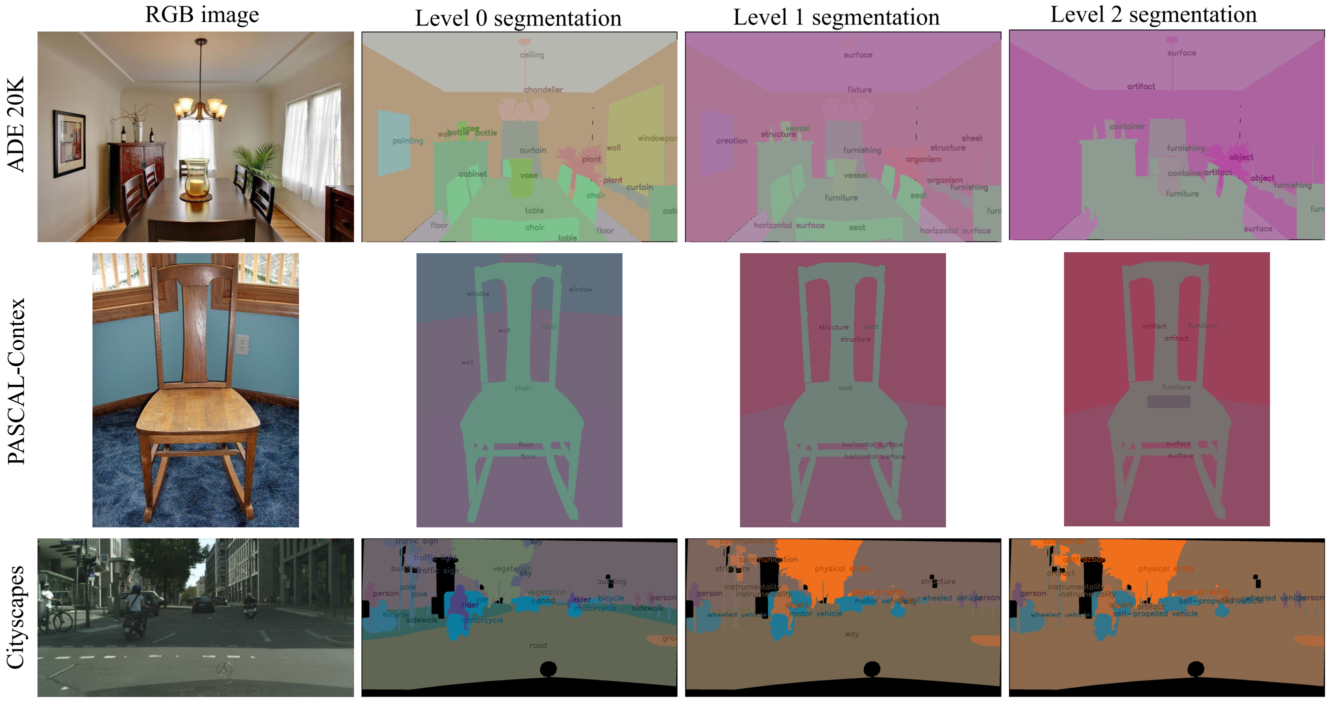


FIGURE 6. Examples of segmentation labels at each level with various datasets. First-row image is from ADE20K dataset [9], the second image is from PASCAL-context dataset [23], and 3rd image is from Cityscapes dataset [7]. For all various types of images, our scene-specific word tree algorithm can extract the appropriate word for each level. Since our level extraction is based on the scene-specific word tree, our algorithm can handle not only the single object image (second row) but also complex multiple object image (first row).

segmentation label images, so there is no significant increase in computational load or data volume. In the subsequent sections, we will introduce a multi-level segmentation model capable of learning from the generated multi-level scene segmentation dataset to validate its effectiveness. We will then proceed to discuss the findings through experimental results.

IV. MULTI-LEVEL SEGMENTATION MODEL

Our proposed method builds multi-level segmentation data based on generating scene-specific word tree. Coarse high-level segmentation through appropriate grouping varies depending on the characteristics of each scene, so network learning for those grouping is needed for this purpose. Therefore, we develop a model that can perform multi-level scene segmentation based on the input RGB image along with the conventional semantic segmentation result.

A. NETWORK ARCHITECTURE

The proposed model receives input of an RGB scene image concatenated with the result of the typical scene segmentation. We concatenate the RGB channels of the scene image (HxWx3 in size) with the object class channels of the segmented image (HxWxC in size, where C denotes the number of entire object classes), resulting in an input size of HxWx(3+C). Figure 7 shows our network structure, here we colorize the segmentation label for better visualization. The segmentation label images actually inputted to or outputted from the segmentation model use a binary format where the value of the corresponding object class channel for each pixel is 1, while the values of other channels are 0. This format, with a size of HxWxC when C denotes the total number of object classes, cannot be directly displayed as an image. Therefore, we present images in the figure using the colorization method described in Section III-E for visualization. The result of any existing scene segmentation model can be used as the input level 0 segmentation along with the scene image, and multiple coarser-level segmentation results are derived. Assuming that the user wants a total of D levels of segmentation, segmentation results for D-1 levels excluding level 0, which is entered

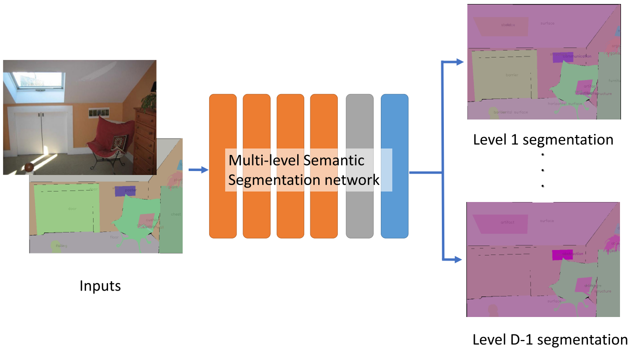


FIGURE 7. Diagram of network structure. Given a scene image and a level 0 segmentation label, the proposed multi-level segmentation network derives high-level segmentation results.

as an input, are obtained. Once the unified score map for all objects that make up D-1 levels is obtained, the object with the highest score for each pixel of the partial score map corresponding to each level is segmented. The proposed multi-level segmentation model can be built by utilizing any existing scene segmentation model. Network modification is performed with different dimensions of input and output, and parameters are initialized with a pre-trained model for the same parts as the base segmentation model. Here, our base model is Segformer [32], since the Segformer model shows the most state-of-the-art performance, especially for the ADE20K dataset.

B. LOSS FUNCTION

The multi-level segmentation model is trained using a unified cross-entropy loss obtained by calculating and adding up the cross-entropy loss for each level. Let $\mathbf{y}$ be the multi-level segmentation results from the model, and $\hat{\mathbf{y}}$ be the ground-truth. Denote the segmentation result for level d as $\mathbf{y}^d$ and the corresponding ground-truth as $\hat{\mathbf{y}}^d$. The unified cross-entropy loss $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})$ is the sum of the cross-entropy losses of each level, $\mathcal{L}_d(\mathbf{y}^d, \hat{\mathbf{y}}^d)$, as follows:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = \sum_d \mathcal{L}_d(\mathbf{y}^d, \hat{\mathbf{y}}^d) = -\sum_d \sum_i \hat{y}_i^d \log y_i^d \tag{2}$$

where $\mathcal{L}_d(\mathbf{y}^d, \hat{\mathbf{y}}^d) = -\sum_{i\in[1,N_d]} \hat{y}_i^d \log y_i^d$ denotes the cross-entropy loss for level $d \in [1, D-1]$ and input data index $i \in [1, N_d]$. D and N_d represent the target number of scene segmentation levels and the number of training data, respectively. To uniformly enhance segmentation performance across each level, we apply the unified cross-entropy loss. Therefore, at coarser levels(e.g. level 2) where diverse objects may belong to the same class and potentially confuse the model, the segmentation performance improves with multi-level output using a single network. This is attributed to the network's understanding of finer-level divisions, enabling it to leverage this information when generating grouped higher-level segmentation results. In other words, by training the multi-level segmentation model using the unified loss function, scene segmentation for each higher level is guided to achieve better performance.

V. EXPERIMENTS

A. SEMANTIC SEGMENTATION

For the base scene segmentation model comprising our multi-level segmentation model, we utilized a state-of-the-art semantic segmentation method called SegFormer [32], that has its encoder pre-trained on the ImageNet-1K dataset [43], while the decoder is randomly initialized. SegFormer performs segmentation through a Mix Transformer (MiT) encoder that analyzes coarse-to-fine features with a hierarchical structure and an ALL-MLP decoder. MiT models include MiT-B0 to MiT-B5, which have the same architecture but different sizes, of which MiT-B5 is the largest model. SegFormer-B5, based on the MiT-B5 encoder, has the best performance among all SegFormer models. We used the SegFormer-B5 model, which has the best performance among SegFormer models that can reflect the hierarchical relationship of objects through a hierarchical encoder, as the backbone model in the following experiments. The implementation details described in their paper were followed. According to Algorithm 1, segmentation labels for each of the D levels (here we define D as 3, therefore level 0, level 1, and level 2) were extracted, and the extracted labels were used for training our model with the unified loss function (2).

For performance evaluation, the following metrics were used: mean intersection over union (mIoU), mean accuracy of each class (mAcc), and all pixel accuracy (aAcc). aAcc evaluates performance in terms of the proportion of correctly classified pixels, and this metric can be biased depending on the results of a few object classes that occupy a large area within the entire dataset. mAcc is the average of the proportion of correctly classified pixels for each class, allowing performance in reducing the effects of the biases to be evaluated. mIoU is the average of the proportion of correctly classified areas for each object instance, allowing it to be spatially verified whether each object is actually well classified. Therefore, we utilize all three of them to evaluate performance in a balanced manner.

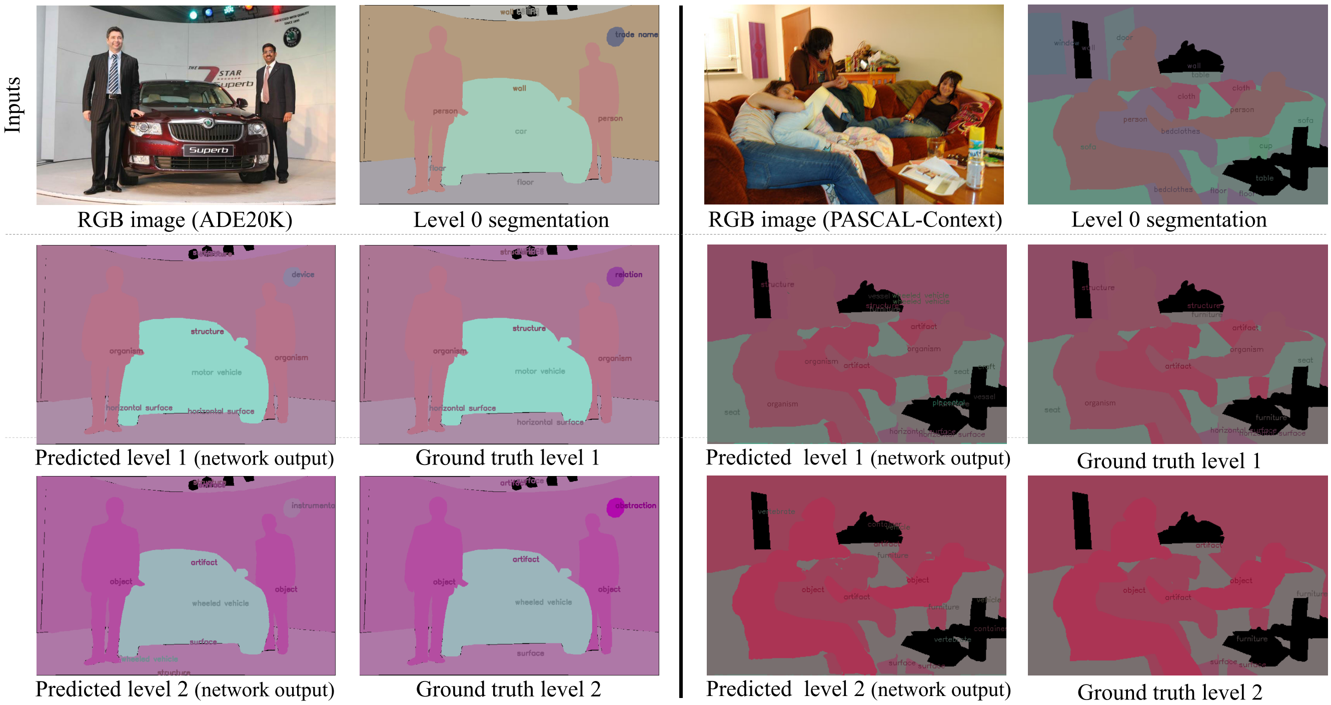
Data augmentation of random resizing, and random cropping, and random horizontal flipping was performed, and input images of 640 × 640 size were generated and used for training. AdamW optimizer for 160K iterations was used. The initial learning rate was established at 0.00006, and the default 'poly' learning rate schedule was subsequently employed with a factor of 1.0.

B. GENERATED DATASET ANALYSIS

Multi-level segmentation experiments were performed on the ADE Challenge 2016, PASCAL-context, and Cityscapes datasets. To demonstrate our approach's flexibility, we select three datasets with different characteristics. The ADE Challenge 2016 dataset is mainly used as a benchmark for comparing semantic segmentation performance for the ADE20K dataset. ADE20K dataset is made for addressing challenges of scene parsing, and it mainly consists of indoor scenes with diverse object categories with pixel-level annotations. The scene segmentation of 150 objects in the ADE Challenge 2016 dataset was set as level 0 labels, and segmentation data for the additional two levels were generated through the proposed multi-level segmentation data generation method based on a scene-specific word tree. 61 and 36 segmentation object labels were generated for levels 1 and 2, respectively. Also, we utilize PASCAL-context dataset, which is derived from a subset of PASCAL VOC images, primarily focuses on object detection; therefore, a large portion of the images contains primary salient main objects and the context of background information around them. The dataset contains both indoor and outdoor, but the number of annotated object classes is relatively small. For the PASCAL-context dataset, level 1 segmentation data

TABLE 1. Segmentation evaluation on the datasets ADE Challenge 2016, PASCAL-context, and Cityscapes in terms of mean intersection over union (mIoU), mean accuracy of each class (mAcc), and all pixel accuracy (aAcc).

Model	Level	ADE Challenge 2016			PASCAL-context			Cityscapes		
		mIoU	mAcc	aAcc	mIoU	mAcc	aAcc	mIoU	mAcc	aAcc
Single-level	Level 1	85.69	91.84	97.87	76.27	81.53	94.22	83.85	87.87	98.39
	Level 2	81.71	90.04	94.45	69.58	75.27	91.56	70.7	77.93	93.13
Multi-level	Level 1	83.73	90.52	97.58	75.74	81.08	94.01	83.59	89.13	98.25
	Level 2	82.29	90.91	95.2	70.22	75.65	92.36	72.55	79.2	93.97



RGB image (ADE20K) | Level 0 segmentation | RGB image (PASCAL-Context) | Level 0 segmentation

Predicted level 1 (network output) | Ground truth level 1 | Predicted level 1 (network output) | Ground truth level 1

Predicted level 2 (network output) | Ground truth level 2 | Predicted level 2 (network output) | Ground truth level 2

FIGURE 8. Multi-level segmentation results. The left and right parts are the results for each sample selected from the ADE20K and PASCAL-context datasets, respectively. The multi-level segmentation results when the scene image and level 0 segmentation labels in the first row are given, and the ground-truth for comparison are shown in the left and right columns of the two lower rows, respectively.

consisting of 33 objects and level 2 segmentation data consisting of 24 objects, and they were generated based on the original semantic segmentation label consisting of 59 objects (excluding the 'background'). Lastly, we utilize Cityscapes dataset, which focuses on semantic understanding of urban street scenes; therefore, its diversity is mainly applied to 50 different cities, seasons, and weather conditions, not to the various object categories. For the Cityscapes dataset, the segmentation results for 27 object labels were used as level 0 data, excluding the following ones that overlap with other labels or do not indicate the type of physically existing object: 'unlabeled', 'ego vehicle', 'rectification border', 'out of roi', 'static', 'dynamic', 'license plate', and 'polegroup'. The level 1 and level 2 scene segmentation data generated from the level 0 data consist of 15 and 11 object labels, respectively. The list of objects for each level generated for the datasets is listed in Tables 3, 4, and 5. Using the generated multi-level segmentation data, we trained our model to derive coarser scene segmentation results of levels 1 and 2, given the combined input of an RGB image and level 0 scene segmentation results.

C. EXPERIMENTAL RESULTS

1) SEGMENTATION PERFORMANCE

The numerical performance of the proposed multi-level segmentation model is listed in Table 1. The test accuracy for each level was calculated separately for each level segmentation output. To analyze the performance of the multi-level segmentation model introduced in Section IV (*Multi-level* in the table), we borrowed the SegFormer [32] structure and conducted experiments to perform single-level segmentation for level 1 and level 2, respectively, for performance comparison (*Single-level* in the table). We refrain from comparing existing multi-level segmentation algorithms because previous approaches primarily segment objects and their parts at a finer level, whereas our method segments objects and groups them into semantically coarser (higher) level chunks, serving a different goal in comparison.

According to the experimental results, the single-level model outperformed the multi-level model in level 1 segmentation; while the multi-level model exhibited superior performance in level 2 segmentation. In the case of level 1, the multi-level model maintains the backbone model of SegFormer [32] and modifies the last layers for multi-task learning, so it is considered that performance was lowered by performing two tasks with a model of the same complexity. However, in the case of level 2, performance improved in the multi-level model compared to the single-level model. This is believed to be because, in the multi-level segmentation, level 2 segmentation is performed with level 1 segmentation simultaneously, which provides some level of guidance to level 2 segmentation, especially when areas with different visual characteristics are grouped into the same label at level 2. For each dataset, level 2 segmentation performance was higher than level 1 segmentation performance. We hypothesize that this is due to the fact that as the scene is divided into larger sections with higher-level concepts, different objects with various visual characteristics are grouped under a single label, making it challenging for the network to learn. In particular, for all datasets used in the experiments, our model showed mIoU performance of more than 70% for each level (especially over 80% for the ADE Challenge 2016 dataset), and it was confirmed that the dataset built by our data generation method reflecting the characteristics of the scene, can be effectively learned.

2) QUALITATIVE ANALYSIS

We also visualized the results of multi-level segmentation based on the trained model, as shown in Figure 8. Each sample was selected from ADE Challenge 2016 and PASCAL-context datasets, and the level 1 and level 2 segmentation results were derived from our multi-level segmentation model. Each data sample is visualized by listing the segmentation results along with the inputs (RGB image and the level 0 segmentation). As shown in the figure, it was verified that our model performed well in level 1 and level 2 segmentation, which grouped objects belonging to the same category based on both given scene and level 0 segmentation results.

Most parts of the ADE Challenge 2016 sample results listed on the left of the figure were segmented correctly, but the 'trade name' object in level 0 was recognized as 'device' rather than 'relation' in level 1. The 'trade name' should be an 'abstraction' at level 2, but it was recognized as 'instrumentality' in our model. However, the result proved that our model performed qualitatively better than the numerical results of the quantitative test, as the results were not entirely incorrect by common sense standards. The visualized results listed on the right side of the figure show that our multi-level segmentation model performed well compared to the ground truth, also on the PASCAL-context dataset.

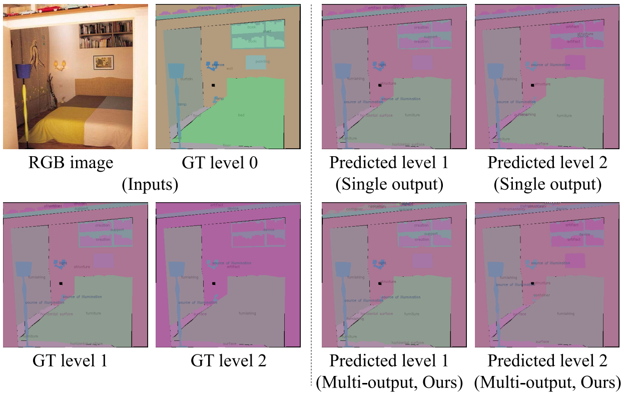

RGB image
GT level 0
(Inputs)
Predicted level 1
(Single output)
Predicted level 2
(Single output)
GT level 1
GT level 2
Predicted level 1
(Multi-output, Ours)
Predicted level 2
(Multi-output, Ours)

FIGURE 9. Multi-level output vs. single-level output model. Given the RGB image and ground-truth level 0, our multi-level model shows better grouping compared to the ablated single-level model.

TABLE 2. Segmentation evaluation using predicted level 0 segmentation results as inputs on the dataset ADE Challenge 2016.

Level 0 Model	Level	ADE CHALLENGE 2016 mIoU	mAcc	aAcc
SegFormer [32]	Level 1	53.37	66.11	84.63
	Level 2	54.07	67.4	78.63
Mask2Former [36]	Level 1	56.28	70.56	85.89
	Level 2	54.65	69.65	79.33

3) ABLATION STUDY

a: MULTI-LEVEL OUTPUT VS. SINGLE-LEVEL OUTPUT

We compared the results for each level segmentation obtained from the multi-level model with the results from a single-level model trained only on the corresponding level segmentation. As mentioned in Table 1 and Section V-C1, our multi-level model showed lower performance than the single-level model at level 1 but higher performance at level 2. Here, we visualized the scene segmentation results and qualitatively compared the two models for each level. Figure 9 shows the segmentation results of our multi-level output model and the ablated single-level output model for comparison. Note that the listed input RGB image and ground-truth level 0 image in Figure 9 were used as input for both models. Our model yielded cleaner segmentation results than the single-level model by reflecting the tendency to group similar objects. We consider that learning grouping in multi-output segmentation provides additional guidance for segmenting objects with clearer boundaries. In particular, as shown in the figure, the 'bed' at level 0 should be grouped from 'furniture' to 'furnishing' as it progresses to levels 1 and 2. In our multi-level segmentation model, it can be confirmed that the result of level 2 is recognized correctly with the guidance of level 1, while in the single-level model, it is recognized as 'furniture' even at level 2 and is not grouped.

b: REAL-WORLD INPUT ANALYSIS

Additional performance evaluation was conducted for the case where real-world level 0 segmentation is given to our multi-level segmentation model. Assuming we have the

TABLE 3. **ADE dataset object list.**

Segmentation level	ADE dataset object list
0	wall, building, sky, floor, tree, ceiling, road, bed, windowpane, grass, cabinet, sidewalk, person, earth, door, table, mountain, plant, curtain, chair, car, water, painting, sofa, shelf, house, sea, mirror, rug, field, armchair, seat, fence, desk, rock, wardrobe, lamp, bathtub, railing, cushion, base, box, column, signboard, chest of drawers, counter, sand, sink, skyscraper, fireplace, refrigerator, grandstand, path, stairs, runway, case, pool table, pillow, screen door, stairway, river, bridge, bookcase, blind, coffee table, toilet, flower, book, hill, bench, countertop, stove, palm, kitchen island, computer, swivel chair, boat, bar, arcade machine, hovel, bus, towel, light, truck, tower, chandelier, awning, streetlight, booth, television receiver, airplane, dirt track, apparel, pole, land, bannister, escalator, ottoman, bottle, buffet, poster, stage, van, ship, fountain, conveyer belt, canopy, washer, plaything, swimming pool, stool, barrel, basket, waterfall, tent, bag, minibike, cradle, oven, ball, food, step, tank, trade name, microwave, pot, animal, bicycle, lake, dishwasher, screen, blanket, sculpture, hood, sconce, vase, traffic light, tray, ashcan, fan, pier, crt screen, plate, monitor, bulletin board, shower, radiator, glass, clock, flag
1	horizontal surface, vascular plant, furniture, barrier, line, lamp, stairway, container, plant, structure, surface, organism, artifact, sheet, seat, creation, source of illumination, electronic equipment, device, fluid, substance, natural elevation, object, plaything, furnishing, vessel, plumbing fixture, door, screen, light, support, matter, fixture, covering, communication, body of water, earth, cushion, chair, table, shelter, shape, path, wheeled vehicle, way, relation, car, canopy, conveyance, motor vehicle, instrumentality, region, kitchen appliance, white goods, location, counter, tree, platform, craft, group, building
2	structure, surface, plant, furnishing, group, source of illumination, way, organism, artifact, instrumentality, furniture, device, object, matter, relation, container, fixture, barrier, covering, physical entity, abstraction, substance, seat, craft, vehicle, motor vehicle, shelter, communication, conveyance, wheeled vehicle, location, home appliance, table, vascular plant, horizontal surface, shape

TABLE 4. **PASCAL-context dataset object list.**

Segmentation level	PASCAL-CONTEXT dataset object list
0	aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, table, dog, horse, motorbike, person, pottedplant, sheep, sofa, train, tvmonitor, bag, bed, bench, book, building, cabinet, ceiling, cloth, computer, cup, door, fence, floor, flower, food, grass, ground, keyboard, light, mountain, mouse, curtain, platform, sign, plate, road, rock, shelves, sidewalk, sky, snow, bedclothes, track, tree, truck, wall, water, window, wood
1	placental, object, vertebrate, vessel, fluid, vascular plant, location, public transport, entity, horizontal surface, organism, seat, structure, furniture, motor vehicle, way, device, artifact, instrumentality, animal, bovid, wheeled vehicle, path, ungulate, craft, food, matter, substance, furnishing, surface, container, plant, abstraction
2	vertebrate, object, animal, craft, matter, plant, conveyance, entity, surface, furniture, artifact, wheeled vehicle, instrumentality, container, organism, ungulate, vehicle, way, placental, physical entity, furnishing, substance, location, abstraction

TABLE 5. **Cityscapes dataset object list.**

Segmentation level	CITYSCAPES dataset object list
0	ground, road, sidewalk, parking, rail track, building, wall, fence, guard rail, bridge, tunnel, pole, traffic light, traffic sign, vegetation, terrain, sky, person, rider, car, truck, bus, caravan, trailer, train, motorcycle, bicycle
1	way, barrier, communication, motor vehicle, abstraction, structure, instrumentality, physical entity, object, person, wheeled vehicle, public transport, conveyance, self-propelled vehicle, artifact
2	abstraction, structure, self-propelled vehicle, instrumentality, physical entity, object, artifact, person, wheeled vehicle, entity, conveyance

existing semantic segmentation model output as level 0 input, which may not be perfect compared to the ground truth. We call such level 0 segmentation labels as *predicted level 0* for our model. Here, we performed scene segmentation using SegFormer [32] and Mask2Former [36], respectively, and used the resulting images as the predicted level 0 input. Figure 10 shows the qualitative comparison of ground-truth level 0 input results and predicted level 0 input results. For a given RGB scene image on the left in Figure 10, the first row includes the ground-truth level 0 input and the level 1 and 2 outputs generated from it. The second and third rows contain the SegFormer and Mask2Former prediction level 0 inputs and the level 1 and 2 outputs generated from them, respectively. As shown in the SegFormer predicted level 0 input, it can be observed that the ‘skyscrapper’ and ‘building’ labels were recognized as a mix for the area labeled ‘building’ in the ground-truth. For this flawed SegFormer level 0 input, our multi-level segmentation model produced clean output. However, they were recognized as lower-level concepts ‘building’ and ‘structure’ rather than ‘structure’ and ‘artifact’ at ground-truth levels 1 and 2, respectively. Nevertheless, it can be concluded that the grouping performed well, as this phenomenon occurred by compensating for the imperfect segmentation performance of input level 0.

In the case of Mask2Former level 0 input(third row of Figure 10), the performance of predicted level 0 can be seen as better than the ground-truth. Two different buildings were recognized separately as ‘skyscrapper’ and ‘building’ according to their visual characteristics. It can be seen that our multi-level segmentation model performs relative grouping

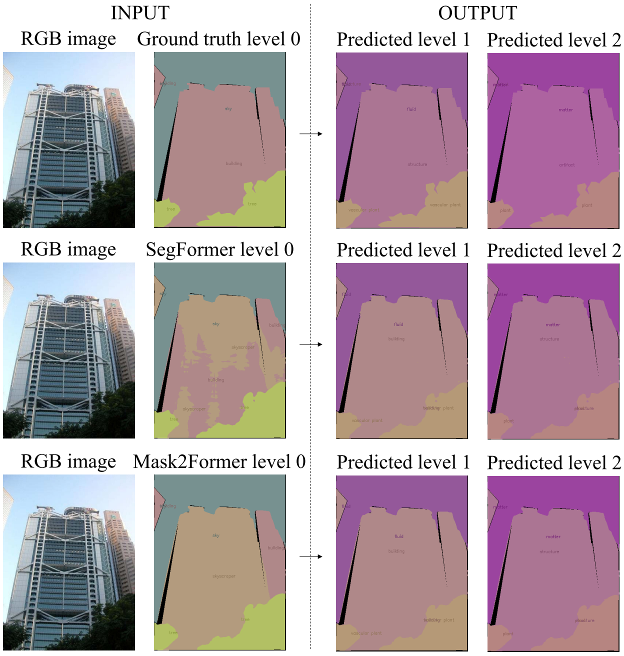


FIGURE 10. Real-word input analysis for the scene on the left. Level 1 and 2 segmentation results according to different level 0 inputs along with the RGB image are listed. (First row: ground-truth level 0, Second row: author-pretrained SegFormer [32] result (level 0), Third row: author-pretrained Mask2Former [36] result (level 0)).

according to the scene well based on the scene-specific word tree even for level 0 inputs that are different from the ground-truth, generating clean level 1 and 2 segmentation results. Quantitative results that include this case can also be confirmed in Table 2, which lists the numerically measured segmentation results of levels 1 and 2 according to each predicted level 0 input (achieving the performance of more than 50% mIoU).

VI. CONCLUSION

We conducted a study to simulate human vision intelligence that recognizes the scene with language hierarchy. Specifically, we proposed a data generation method that derives multi-level segmentation results by scene-specific word tree generation. We also verified the effectiveness of our proposed data by building a model that performs multi-level segmentation for a given scene and evaluating the model using the data generated by the proposed method. The proposed data generation method is versatile, so it can be applied to any other scene segmentation dataset, which may expand the potential capabilities of our scene-specific word tree. The method we proposed will serve as a starting point for implementing the visual intelligence of a person who 'understands' a given scene according to the situation rather than 'looking' at a given scene fragmentarily by combining structured linguistic knowledge with scene segmentation. In particular, the proposed multi-level segmentation that incorporates the hierarchy of object words is a source technology that can be used for situation-based scene understanding in various fields. Specifically, the proposed method will enable scene recognition for mobile robots according to the distance from the target point or object recognition for robot manipulators performing at various categorical levels depending on the task type.

However, in the process of generating label data, it was difficult to select a dictionary definition for the object in the image in the case of an object with a homonymous name. To solve this, we introduced a method to select the most similar definition by comparing the visual information of the hypernym text and the corresponding object using CLIP features. Nevertheless, the effectiveness of the proposed method with a broader vocabulary than currently utilized needs to be verified, necessitating further research.

VII. FUTURE WORK

In future work, we plan to apply our multi-level scene segmentation model to help operate mobile robots. When the camera moves, as in the case of a mobile robot, different widths of the same scene are captured depending on the Depth of Field in photography. By employing multi-level scene segmentation using the proposed model, objects with different degrees of detail can be recognized depending on the distance even when looking at the same scene. Specifically, we consider to implement for the scenario when a mobile robot enters a house. The proposed model will be applied to differentiate between recognizing the entire house as a separate entity from the background when it is viewed from a distance, and identifying finer details such as doors and windows as the robot approaches for entry.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2012.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[6] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* USA: Springer, 2014, pp. 740–755.

[7] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.

[8] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 7576. Springer, 2012, pp. 746–760.

[9] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ADE20K dataset," *Int. J. Comput. Vis.*, vol. 127, no. 3, pp. 302–321, Mar. 2019.

[10] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 418–434.

[11] L. Qi, J. Kuen, W. Guo, J. Gu, Z. Lin, B. Du, Y. Xu, and M.-H. Yang, "AIMS: All-inclusive multi-level segmentation for anything," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–20.

[12] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI, USA, Tech. Rep., 2018. [Online]. Available: https://openai.com/index/language-unsupervised/

[13] J. D. M.-W. C. Kenton and L. K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, vol. 1, 2019, p. 2.

[14] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, vol. 139, 2021, pp. 8748–8763.

[15] G. A. Miller, *WordNet: An Electronic Lexical Database*. Cambridge, MA, USA: MIT Press, 1998.

[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[17] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[18] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 1–8.

[19] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille, "Detect what you can: Detecting and representing objects using holistic models and body parts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1979–1986.

[20] S. Bell, P. Upchurch, N. Snavely, and K. Bala, "OpenSurfaces: A richly annotated catalog of surface appearance," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 1–17, Jul. 2013.

[21] S. Bell, P. Upchurch, N. Snavely, and K. Bala, "Material recognition in the wild with the materials in context database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3479–3487.

[22] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3606–3613.

[23] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 891–898.

[24] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3485–3492.

[25] G. Neuhold, T. Ollmann, S. R. Bulò, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4990–4999.

[26] F. Xia, P. Wang, X. Chen, and A. L. Yuille, "Joint multi-person pose estimation and semantic part segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6769–6778.

[27] X. Liang, K. Gong, X. Shen, and L. Lin, "Look into person: Joint body parsing & pose estimation network and a new benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 871–885, Apr. 2019.

[28] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1520–1528.

[29] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1925–1934.

[30] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.

[31] T. Zhang, X. Zhang, P. Zhu, X. Tang, C. Li, L. Jiao, and H. Zhou, "Semantic attention and scale complementary network for instance segmentation in remote sensing images," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10999–11013, Oct. 2022.

[32] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. Neural Inf. Process. Syst. (NeurIPS)*, 2021, pp. 1–14.

[33] A. Avetisyan, C. Xie, H. Howard-Jenkins, T.-Y. Yang, S. Aroudj, S. Patra, F. Zhang, D. Frost, L. Holland, C. Orme, J. Engel, E. Miller, R. Newcombe, and V. Balntas, "SceneScript: Reconstructing scenes with an autoregressive structured language model," 2024, *arXiv:2403.13064*.

[34] L. Bertinetto, R. Mueller, K. Tertikas, S. Samangooei, and N. A. Lord, "Making better mistakes: Leveraging class hierarchies with deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12506–12515.

[35] X. Liu, Y. He, Y.-M. Cheung, X. Xu, and N. Wang, "Learning relationship-enhanced semantic graph for fine-grained image–text matching," *IEEE Trans. Cybern.*, vol. 54, no. 2, pp. 948–961, Feb. 2024.

[36] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1290–1299.

[37] L. Li, T. Zhou, W. Wang, J. Li, and Y. Yang, "Deep hierarchical semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1246–1257.

[38] J. Deng, J. Krause, A. C. Berg, and L. Fei-Fei, "Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3450–3457.

[39] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam, "Large-scale object classification using label relation graphs," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2014, pp. 48–64.

[40] H. Zhao, X. Puig, B. Zhou, S. Fidler, and A. Torralba, "Open vocabulary scene parsing," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2002–2010.

[41] X. Cao, X. Wei, Y. Han, and X. Chen, "An object-level high-order contextual descriptor based on semantic, spatial, and scale cues," *IEEE Trans. Cybern.*, vol. 45, no. 7, pp. 1327–1339, Jul. 2015.

[42] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Inf. Process. Lett.*, vol. 31, no. 1, pp. 7–15, Apr. 1989.

[43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

SOOMIN KIM received the B.S., M.S., and Ph.D. degrees in computer science from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea, in 2013, 2015, and 2021, respectively. She is currently a Senior Researcher with the Center for Artificial Intelligence, Korea Institute of Science and Technology (KIST), Seoul, Republic of Korea. Her research interests include computer vision and machine learning.

JUYOUN PARK received the B.S. and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea, in 2015 and 2019, respectively.

In 2019, she was a Postdoctoral Researcher with the Information and Electronics Research Institute, KAIST. In 2020, she was a Postdoctoral Scientist with the Department of Biomedical Engineering, The George Washington University, Washington, DC, USA. Since 2021, she has been a Senior Researcher with the Center for Intelligent and Interactive Robotics, Korea Institute of Science and Technology (KIST), Seoul, Republic of Korea. Her research interests include but are not limited to the areas of artificial intelligence (AI) for autonomous agents, including robots. In particular, she is interested in developing novel machine learning or deep learning methods to allow robots to autonomously recognize and understand the environments. She is also interested in developing a framework for practical application of the developed intelligence to various robots, such as social robots and surgical assistive robots and in industrial fields.