Unified Perceptual Parsing for Scene Understanding

Tete Xiao[1]*, Yingcheng Liu[1]*, Bolei Zhou[2]*, Yuning Jiang[3], Jian Sun[4]

[1]Peking University [2] MIT CSAIL [3] Bytedance Inc. [4] Megvii Inc.
* indicates equal contribution.

Abstract. Humans recognize the visual world at multiple levels: we effortlessly categorize scenes and detect objects inside, while also identifying the textures and surfaces of the objects along with their different compositional parts. In this paper, we study a new task called Unified Perceptual Parsing, which requires the machine vision systems to recognize as many visual concepts as possible from a given image. A multi-task framework called UPerNet and a training strategy are developed to learn from heterogeneous image annotations. We benchmark our framework on Unified Perceptual Parsing and show that it is able to effectively segment a wide range of concepts from images. The trained networks are further applied to discover visual knowledge in natural scenes[1].

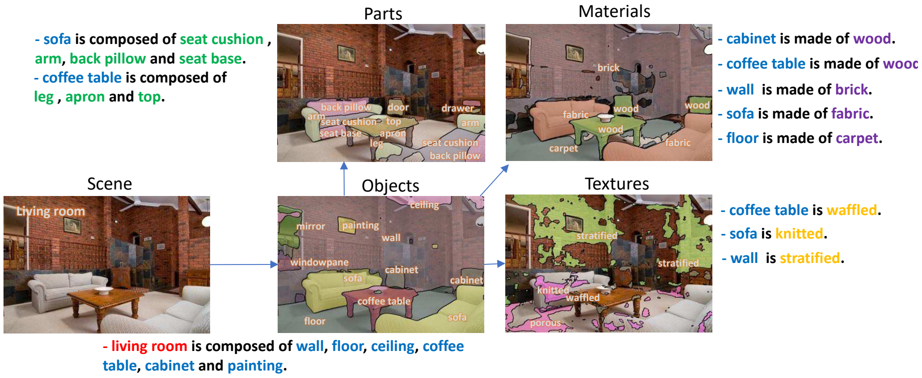
Keywords: Deep neural network, semantic segmentation, scene understanding

1 Introduction

The human visual system is able to extract a remarkable amount of semantic information from a single glance. We not only instantly parse the objects contained within, but also identify the fine-grained attributes of objects, such as their parts, textures and materials. For example in Figure 1, we can recognize that this is a living room with various objects such as a coffee table, a painting, and walls inside. At the same time, we identify that the coffee table has legs, an apron and top, as well as that the coffee table is wooden and the surface of the sofa is knitted. Our interpretation of the visual scene is organized at multiple levels, from the visual perception of the materials and textures to the semantic perception of the objects and parts.

Great progress in computer vision has been made towards human-level visual recognition because of the development of deep neural networks and large-scale image datasets. However, various visual recognition tasks are mostly studied independently. For example, human-level recognition has been reached for object classification [1] and scene recognition [2]; objects and stuff are parsed and segmented precisely at pixel-level [3,2]; Texture and material perception and recognition have been studied in [4] and [5]. Since scene recognition, object detection,

[1] Models are available at https://github.com/CSAILVision/unifiedparsing



- sofa is composed of seat cushion , arm, back pillow and seat base.
- coffee table is composed of leg , apron and top.

- cabinet is made of wood.
- coffee table is made of wooc
- wall is made of brick.
- sofa is made of fabric.
- floor is made of carpet.

- coffee table is waffled.
- sofa is knitted.
- wall is stratified.

- living room is composed of wall, floor, ceiling, coffee table, cabinet and painting.

Fig. 1. Network trained for Unified Perceptual Parsing is able to parse various visual concepts at multiple perceptual levels such as scene, objects, parts, textures, and materials all at once. It also identifies the compositional structures among the detected concepts.

texture and material recognition are intertwined in human visual perception, this raises an important question for the computer vision systems: is it possible for a neural network to solve several visual recognition tasks simultaneously? This motives our work to introduce a new task called Unified Perceptual Parsing (UPP) along with a novel learning method to address it.

There are several challenges in UPP. First, there is no single image dataset annotated with all levels of visual information. Various image datasets are constructed only for specific task, such as ADE20K for scene parsing [2], the Describe Texture Dataset (DTD) for texture recognition [4], and OpenSurfaces for material and surface recognition [6]. Next, annotations from different perceptual levels are heterogeneous. For example, ADE20K has pixel-wise annotations while the annotations for textures in the DTD are image-level.

To address the challenges above we propose a framework that overcomes the heterogeneity of different datasets and learns to detect various visual concepts jointly. On the one hand, at each iteration, we randomly sample a data source, and only update the related layers on the path to infer the concepts from the selected source. Such a design avoids erratic behavior that the gradient with respect to annotations of a certain concept may be noisy. On the other hand, our framework exploits the hierarchical nature of features from a single network, *i.e.*, for concepts with higher-level semantics such as scene classification, the classifier is built on the feature map with the higher semantics only; for lower-level semantics such as object and material segmentation, classifiers are built on feature maps fused across all stages or the feature map with low-level semantics only. We further propose a training method that enables the network to predict pixel-wise texture labels using only image-level annotations.

Our contributions are summarized as follows: 1) We present a new parsing task Unified Perceptual Parsing, which requires systems to parse multiple visual concepts at once. 2) We present a novel network called UPerNet with hierarchical structure to learn from heterogeneous data from multiple image datasets. 3) The model is shown to be able to jointly infer and discover the rich visual knowledge underneath images.

1.1 Related work

Our work is built upon the previous work of semantic segmentation and multi-task learning.

Semantic segmentation. To generate pixel-wise semantic predictions for a given image, image classification networks [7,8,9,1] are extended to generate semantic segmentation masks. Pioneering work by Chen *et al.* [10], based on structure prediction, uses conditional random field (CRF) to refine the activations of the final feature map of CNNs. The most prevalent framework designed for this pixel-level classification task is the Fully Convolutional Network (FCN) [11], which replaces fully-connected layers in classification networks with convolutional layers. Noh *et al.* [12] propose a framework which applies deconvolution [13] to up-sample low resolution feature maps. Yu and Vladlen [14] propose an architecture based on dilated convolution which is able to exponentially expand the receptive field without loss of resolution or coverage. More recently, RefineNet [15] uses a coarse-to-fine architecture which exploits all information available along the down-sampling process. The Pyramid Scene Parsing Network (PSPNet) [16] performs spatial pooling at several grid scales and achieves remarkable performance on several segmentation benchmarks [17,18,2].

Multi-task learning. Multi-task learning, which aims to train models to accomplish multiple tasks at the same time, has attracted attention since long before the era of deep learning. For example, a number of previous research works focus on the combination of recognition and segmentation [19,20,21]. More recently, Elhoseiny *et al.* [22] have proposed a model that performs pose estimation and object classification simultaneously. Eigen and Fergus [23] propose an architecture that jointly addresses depth prediction, surface normal estimation, and semantic labeling. Teichmann *et al.* [24] propose an approach to perform classification, detection, and semantic segmentation via a shared feature extractor. Kokkinos proposes the UberNet [25], a deep architecture that is able to do seven different tasks relying on diverse training sets. Another recent work [3] proposes a partially supervised training paradigm to scale up the segmentation of objects to $3,000$ objects using box annotations only. Comparing our work with previous works on multi-task learning, only a few of them perform multi-task learning on heterogeneous datasets, *i.e.*, a dataset that does not necessarily have all levels of annotations over all tasks. Moreover, although tasks in [25] are formed from low level to high level, such as boundary detection, semantic segmentation and object detection, these tasks do not form the hierarchy of visual concepts. In Section 4.2, we further demonstrate the effectiveness of our proposed tasks and frameworks in discovering the rich visual knowledge from images.

2 Defining Unified Perceptual Parsing

We define the task of Unified Perceptual Parsing as the recognition of many visual concepts as possible from a given image. Possible visual concepts are organized into several levels: from scene labels, objects, and parts of objects, to materials and textures of objects. The task depends on the availability of different kinds of training data. Since there is no single image dataset annotated with all visual concepts at multiple levels, we first construct an image dataset by combining several sources of image annotations.

2.1 Datasets

In order to accomplish segmentation of a wide range of visual concepts from multiple levels, we utilize the Broadly and Densely Labeled Dataset (Broden) [26], a heterogeneous dataset that contains various visual concepts. Broden unifies several densely labeled image datasets, namely ADE20K [2], Pascal-Context [27], Pascal-Part [28], OpenSurfaces [6], and the Describable Textures Dataset (DTD) [4]. These datasets contain samples of a broad range of scenes, objects, object parts, materials and textures in a variety of contexts. Objects, object parts and materials are segmented down to pixel level while textures and scenes are annotated at image level.

The Broden dataset provides a wide range of visual concepts. Nevertheless, since it is originally collected to discover the alignment between visual concepts and hidden units of Convolutional Neural Networks (CNNs) for network interpretability [26,29], we find that samples from different classes are unbalanced. Therefore we standardize the Broden dataset to make it more suitable for training segmentation networks. First, we merge similar concepts across different datasets. For example, objects and parts annotations in ADE20K, Pascal-Context, and Pascal-Part are merged and unified. Second, we only include object classes which appear in at least 50 images *and* contain at least $50,000$ pixels in the whole dataset. Also, object parts which appear in at least 20 images can be considered valid parts. Objects and parts that are conceptually inconsistent are manually removed. Third, we manually merge under-sampled labels in OpenSurfaces. For example, *stone* and *concrete* are merged into *stone*, while *clear plastic* and *opaque plastic* are merged into *plastic*. Labels that appear in less than 50 images are also filtered out. Fourth, we map more than 400 scene labels from the ADE20K dataset to 365 labels from the Places dataset [30].

Table 1 shows some statistics of our standardized Broden, termed as Broden+. It contains $57,095$ images in total, including $22,210$ images from ADE20K, $10,103$ images from Pascal-Context and Pascal-Part, $19,142$ images from OpenSurfaces and $5,640$ images from DTD. Figure 2 shows the distribution of objects as well as parts grouped by the objects to which they belong. We also provide examples from each source of the Broden+ dataset in Figure 3.

Category	Classes	Sources	Eval. Metrics
scene	365	ADE [2]	top-1 acc.
object	335	ADE [2], Pascal-Context[27]	mIoU & pixel acc.
object w/ part	77	ADE [2], Pascal-Context[27]	-
part	152	ADE [2], Pascal-Part [28]	mIoU (bg) & pixel acc.
material	26	OpenSurfaces [6]	mIoU & pixel acc.
texture	47	DTD [4]	top-1 acc.

Table 1. Statistics of each label type in the Broden+ dataset. Evaluation metrics for each type of labels are also listed.

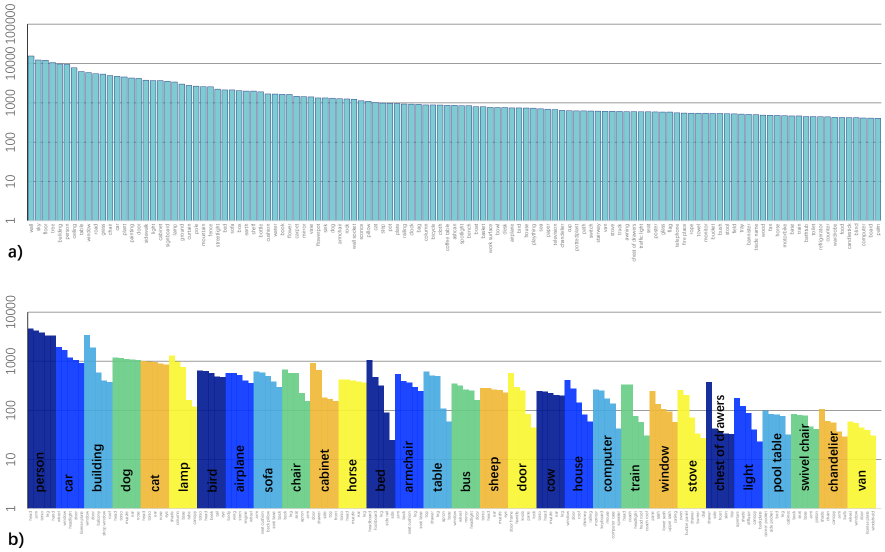


Fig. 2. a) Sorted object classes by frequency: we show top 120 classes selected from the Broden+. Object classes that appear in less than 50 images or contain less than 50, 000 pixels are filtered. b) Frequency of parts grouped by objects. We show only top 30 objects with their top 5 frequent parts. The parts that appear in less than 20 images are filtered.

2.2 Metrics

To quantify the performance of models, we set different metrics based on the annotations of each dataset. Standard metrics to evaluate semantic segmentation tasks include Pixel Accuracy (P.A.), which indicates the proportion of correctly classified pixels, and mean IoU (mIoU), which indicates the intersection-over-union (IoU) between the predicted and ground truth pixels, averaged over all object classes. Note that since there might be unlabeled areas in an image, the mIoU metric will not count the predictions on unlabeled regions. This would encourage people to exclude the background label during training. However, it

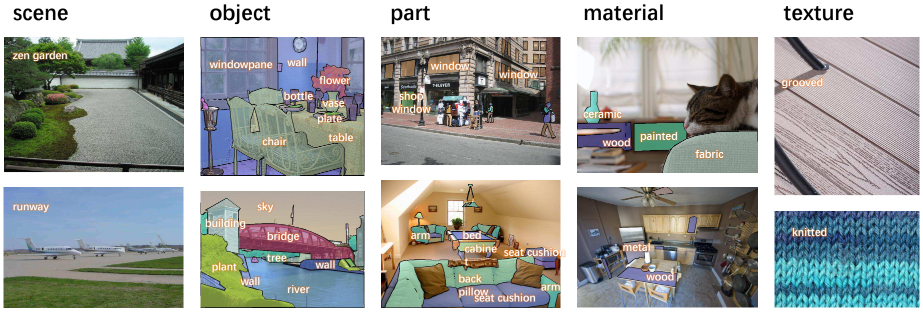


Fig. 3. Samples from the Broden+ dataset. The ground-truth labels for scene and texture are image-level annotations, while for object, part and material are pixel-wise annotations. Object and part are densely annotated, while material is partially annotated. Images with texture labels are mostly such localized object regions.

is not suitable for the evaluation of tasks like part segmentation, because for some objects the regions with part annotations only account for a small number of pixels. Therefore we use mIoU, but count the predictions in the background regions, denoted as mIoU-bg, in certain tasks. In this way, excluding background labels during training will boost P.A. by a small margin. Nonetheless, it will significantly downgrade mIoU-bg performance.

For object and material parsing involving ADE20K, Pascal-Context, and OpenSurfaces, the annotations are at pixel level. Images in ADE20K and Pascal-Context are fully annotated, with the regions that do not belong to any predefined classes categorized into an unlabeled class. Images in OpenSurfaces are partially annotated, *i.e.*, if several regions of material occur in a single image, more than one region may not be annotated. We use P.A. and mIoU metrics for these two tasks.

For object parts we use P.A. and mIoU-bg metrics for the above mentioned reason. The IoU of each part is first averaged within an object category, then averaged over all object classes. For scene and texture classification we report top-1 accuracy. Evaluation metrics are listed in Table 1.

To balance samples across different labels in different categories we first randomly sample 10% of original images as the validation set. We then randomly choose an image both from the training and validation set, and check if the annotations in pixel level are more balanced towards 10% after swapping these two images. The process is performed iteratively. The dataset is split into $51,617$ images for training and $5,478$ images for validation.

3 Designing Networks for Unified Perceptual Parsing

We demonstrate our network design in Figure 4, termed as **UPerNet** (**U**nified **Per**ceptual Parsing **Net**work), based on the Feature Pyramid Network (FPN) [31]. FPN is a generic feature extractor which exploits multi-level feature represen-

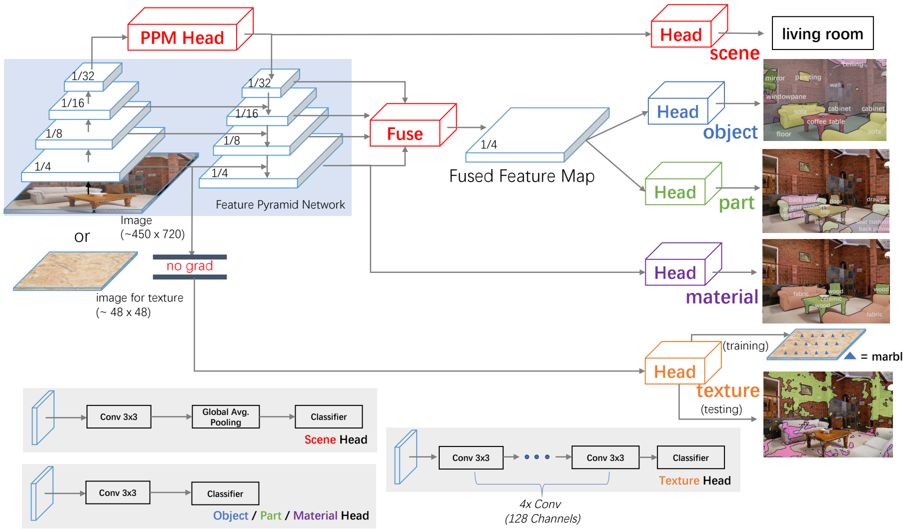


Fig. 4. UPerNet framework for Unified Perceptual Parsing. Top-left: The Feature Pyramid Network (FPN) [31] with a Pyramid Pooling Module (PPM) [16] appended on the last layer of the back-bone network before feeding it into the top-down branch in FPN. Top-right: We use features at various semantic levels. Scene head is attached on the feature map directly after the PPM since image-level information is more suitable for scene classification. Object and part heads are attached on the feature map fused by all the layers put out by FPN. Material head is attached on the feature map in FPN with the highest resolution. Texture head is attached on the Res-2 block in ResNet [1], and fine-tuned after the whole network finishes training on other tasks. Bottom: The illustrations of different heads. Details can be found in Section 3.

tations in an inherent and pyramidal hierarchy. It uses a top-down architecture with lateral connections to fuse high-level semantic information into middle and low levels with marginal extra cost. To overcome the issue raised by Zhou *et al.* [32] that although the theoretical receptive field of deep CNN is large enough, the empirical receptive field of deep CNN is relatively much smaller [33], we apply a Pyramid Pooling Module (PPM) from PSPNet [16] on the last layer of the backbone network before feeding it into the top-down branch in FPN. Empirically we find that the PPM is highly compatible with the FPN architecture by bringing effective global prior representations. For further details on FPN and PPM, we refer the reader to [31] and [16].

With the new framework, we are able to train a single network which is able to unify parsing of visual attributes at multiple levels. Our framework is based on Residual Networks [1]. We denote the set of last feature maps of each stage in ResNet as $\{C_2, C_3, C_4, C_5\}$, and the set of feature maps put out by FPN as $\{P_2, P_3, P_4, P_5\}$, where P_5 is also the feature map directly following PPM. The down-sampling rates are $\{4, 8, 16, 32\}$, respectively. *Scene label*, the highest-level attribute annotated at image-level, is predicted by a global average pooling of P_5

followed by a linear classifier. It is worth noting that, unlike frameworks based on a dilated net, the down-sampling rate of P_5 is relatively large so that the features after global average pooling focus more on high-level semantics. For *object label*, we empirically find that fusing all feature maps of FPN is better than only using the feature map with the highest resolution (P_2). *Object parts* are segmented based on the same feature map as objects. For *materials*, intuitively, if we have prior knowledge that these areas belong to the object "cup", we are able to make a reasonable conjecture that it might be made up of paper or plastics. This context is useful, but we still need local apparent features to decide which one is correct. It should also be noted that an object can be made up of various materials. Based on the above observations, we segment materials on top of P_2 rather than fused features. *Texture label*, given at the image-level, is based on non-natural images. Directly fusing these images with other natural images is harmful to other tasks. Also we hope the network can predict texture labels at pixel level. To achieve such a goal, we append several convolutional layers on top of C_2, and force the network to predict the texture label at every pixel. The gradient of this branch is prevented from back-propagating to layers of backbone networks, and the training images for texture are resized to a smaller size ($\sim 64 \times 64$). The reasons behind these designs are: 1) Texture is the lowest-level perceptual attribute, thus it is purely based on apparent features and does not need any high-level information. 2) Essential features for predicting texture correctly are implicitly learned when trained on other tasks. 3) The receptive field of this branch needs to be small enough, so that the network is able to predict different labels at various regions when an image at normal scale is fed in the network. We only fine-tune the texture branch for a few epochs after the whole network finishes training on other tasks.

When only trained on object supervision, without further enhancements, our framework yields almost identical performance as the state-of-the-art PSPNet, while requiring only 63% of training time for the same number of epochs. It is worth noting that we do not even perform deep supervision or data augmentations used in PSPNet other than scale jitter, according to the experiments in their paper [16]. Ablation experiments are provided in Section 4.1.

3.1 Implementation details

Every classifier is preceded by a separate convolutional head. To fuse the layers with different scales such as $\{P_2, P_3, P_4, P_5\}$, we resize them via bilinear interpolation to the size of P_2 and concatenate these layers. A convolutional layer is then applied to fuse features from different levels as well as to reduce channel dimensions. All extra non-classifier convolutional layers, including those in FPN, have batch normalization [34] with 512-channel output. ReLU [35] is applied after batch normalization. Same as [36], we use the "poly" learning rate policy where the learning rate at current iteration equals the initial learning rate multiplying $\left(1 - \frac{iter}{max\_iter}\right)^{power}$. The initial learning rate and power are set to 0.02 and 0.9, respectively. We use a weight decay of 0.0001 and a momentum of 0.9. During training the input image is resized such that the length of its shorter side is randomly chosen from the set $\{300, 375, 450, 525, 600\}$. For inference we do not apply multi-scale testing for fair comparison, and the length is set to 450. The maximum length of the longer side is set to 1200 in avoidance of GPU memory overflow. The layers in the backbone network are initialized with weights pre-trained on ImageNet [37].

During each iteration, if a mini-batch is composed of images from several sources on various tasks, the gradient with respect to a certain task can be noisy, since the real batch size of each task is in fact decreased. Thus we randomly sample a data source at each iteration based on the scale of each source, and only update the path to infer the concepts related to the selected source. For object and material, we do not calculate loss on unlabeled area. For part, as mentioned in Section 2.2, we add background as a valid label. Also the loss of a part is applied only inside the regions of its super object.

Due to physical memory limitations a mini-batch on each GPU involves only 2 images. We adopt synchronized SGD training across 8 GPUs. It is worth noting that batch size has proven to be important to generate accurate statistics for tasks like classification [38], semantic segmentation [16] and object detection [39]. We implement batch normalization such that it is able to synchronize across multiple GPUs. We do not fix any batch norm layer during training. The number of training iterations of ADE20k (with $\sim 20k$ images) alone is $100k$. If trained on a larger dataset, we linearly increase training iterations based on the number of images in the dataset.

3.2 Design discussion

State-of-the-art segmentation networks are mainly based on fully convolutional networks (FCNs) [11]. Due to a lack of sufficient training samples, segmentation networks are usually initialized from networks pre-trained for image classification [37,7,8]. To enable high-resolution predictions for semantic segmentation, dilated convolution [14], a technique which removes the stride of convolutional layers and adds holes between each location of convolution filters, has been proposed to ease the side effect of down-sampling while maintaining the expansion rate for receptive fields. The dilated network has become the *de facto* paradigm for semantic segmentation.

We argue that such a framework has major drawbacks for the proposed Unified Perceptual Parsing task. First, recently proposed deep CNNs [1,40], which have succeeded on tasks such as image classification and semantic segmentation usually have tens or hundreds of layers. These deep CNNs are intricately designed such that the down-sampling rate grows rapidly in the early stage of the network for the sake of a larger receptive field and lighter computational complexity. For example, in the ResNet with 100 convolutional layers in total, there are 78 convolutional layers in the Res-4 and Res-5 blocks combined, with down-sampling rates of 16 and 32, respectively. In practice, in a dilated segmentation framework, dilated convolution needs to be applied to both blocks to ensure that the maximum down-sampling rate of all feature maps do not exceed 8. Nevertheless, due to the feature maps within the two blocks are increased to

Method	Mean IoU(%)	Pixel Acc.(%)	Overall(%)	Time(hr)
FCN [11]	29.39	71.32	50.36	-
SegNet [42]	21.64	71.00	46.32	-
DilatedNet [14]	32.31	73.55	52.93	-
CascadeNet [2]	34.90	74.52	54.71	-
RefineNet (Res-152) [15]	40.70	-	-	-
DilatedNet*†(Res-50) [16]	34.28	76.35	55.32	53.9
PSPNet†(Res-50) [16]	**41.68**	**80.04**	**60.86**	61.1
FPN (/16)	34.46	76.04	55.25	18.1
FPN (/8)	34.99	76.54	55.77	20.2
FPN (/4)	35.26	76.52	55.89	21.2
FPN+PPM (/4)	40.13	79.61	59.87	27.8
FPN+PPM+Fusion (/4)	**41.22**	**79.98**	**60.60**	38.7

Table 2. Detailed analysis of our framework based on ResNet-50 *v.s.* state-of-the-art methods on ADE20K dataset. Our results are obtained without multi-scale inference or other techniques. FPN baseline is competitive while requiring much less computational resources. Further increasing resolution of feature maps brings consistent gain. PPM is highly compatible with FPN. Empirically we find that fusing features from all levels of FPN yields best performance. *: A stronger reference for DilatedNet reported in [16]. †: Training time is based on our reproduced models. We also use the same codes in FPN baseline.

4 or 16 times of their designated sizes, both the computation complexity and GPU memory footprint are dramatically increased. The second drawback is that such a framework utilizes only the deepest feature map in the network. Prior works [41] have shown the hierarchical nature of the features in the network, *i.e.*, lower layers tend to capture local features such as corners or edge/color conjunctions, while higher layers tend to capture more complex patterns such as parts of some object. Using the features with the highest-level semantics might be reasonable for segmenting high-level concepts such as objects, but it is naturally unfit to segment perceptual attributes at multiple levels, especially the low-level ones such as textures and materials. In what follows, we demonstrate the effectiveness and efficiency of our UPerNet.

4 Experiments

The experiment section is organized as follows: we first introduce the quantitative study of our proposed framework on the original semantic segmentation task and the UPP task in Section 4.1. Then we apply the framework to discover visual common sense knowledge underlying scene understanding in Section 4.2.

4.1 Main results

Overall architecture. To demonstrate the effectiveness of our proposed architecture on semantic segmentation, we report the results trained on ADE20K us-

Training Data					Object		Part		Scene	Material		Texture
+O	+P	+S	+M	+T	mI.	P.A.	mI.(bg)	P.A.	T-1	mI.	P.A.	T-1
✓					24.72	78.03	-	-	-	-	-	-
			✓		-	-	-	-	-	52.78	84.32	-
✓	✓				23.92	77.48	30.21	48.30	-	-	-	-
✓	✓	✓			23.83	77.23	30.10	48.34	71.35	-	-	-
✓	✓	✓	✓		23.36	77.09	28.75	46.92	70.87	54.19	84.45	-
✓	✓	✓	✓	✓	23.36	77.09	28.75	46.92	70.87	54.19	84.45	35.10

Table 3. Results of Unified Perceptual Parsing on the Broden+ dataset. O: Object. P: Part. S: Scene. M: Material. T: Texture. mI.: mean IoU. P.A.: pixel accuracy. mI.(bg): mean IoU including background. T-1: top-1 accuracy.

ing object annotations under various settings in Table 2. In general, FPN demonstrates competitive performance while requiring much less computational resources for semantic segmentation. Using the feature map up-sampled only once with a down-sampling rate of 16 (P_4), it reaches mIoU and P.A. of 34.46/76.04, almost identical to the strong baseline reference reported in [16] while only taking about 1/3 of the training time for the same number of iterations. Performance improves further when the resolution is higher. Adding the Pyramid Pooling Module (PPM) boosts performance by a 4.87/3.09 margin, which demonstrates that FPN also suffers from an insufficient receptive field. Empirically we find that fusing features from all levels of FPN yields best performance, a consistent conclusion also observed in [43].

The performance of FPN is surprising considering its simplicity with feature maps being simply up-sampled by bilinear interpolation instead of time-consuming deconvolution, and the top-down path is fused with bottom-up path by an 1x1 convolutional layer followed by element-wise summation without any complex refinement module. It is the simplicity that accomplishes its efficiency. We therefore adopt this design for Unified Perceptual Parsing.

Multi-task learning with heterogeneous annotations. We report the results trained on separate or fused different sets of annotations. The baseline of object parsing is the model trained on ADE20K and Pascal-Context. It yields mIoU and P.A. of 24.72/78.03. This result, compared with the results for ADE20K, is relatively low because Broden+ has many more object classes. The baseline of material is the model trained on OpenSurfaces. It yields mIoU and P.A. of 52.78/84.32. Joint training of object and part parsing yields 23.92/77.48 on object and 30.21/48.30 on part. The performance on object parsing trained plus part annotations is almost identical to that trained only on object annotations. After adding a scene prediction branch it yields top-1 accuracy of 71.35% on scene classification, with negligible downgrades of object and part performance. When jointly training material with object, part, and scene classification, it yields a performance of 54.19/84.45 on material parsing, 23.36/77.09 on object parsing, and 28.75/46.92 on part parsing. It is worth noting that the object and part both suffer a slight performance degrade due to heterogeneity, while

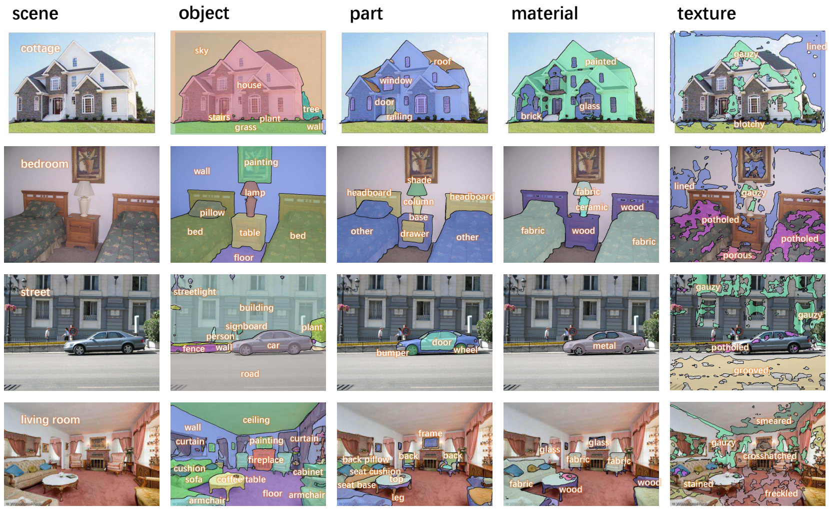


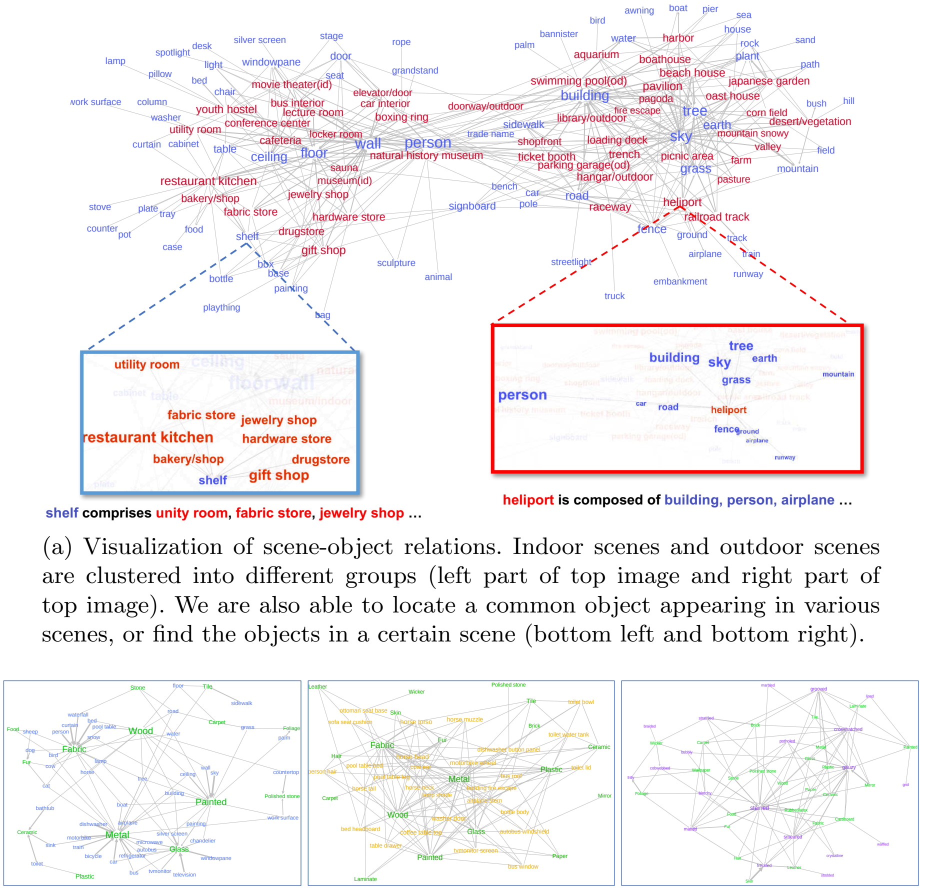
Fig. 5. Predictions on the validation set using UPerNet (ResNet-50). From left to right: scene classification, and object, part, material, and texture parsing.

material enjoys a boost in performance compared with that trained only on OpenSurfaces. We conjecture that it is attributed to the usefulness of information in object as priors for material parsing. As mentioned above, we find that directly fusing texture images with other natural images is harmful to other tasks, since there are nontrivial differences between images in DTD and natural images. After fine-tuning on texture images using the model trained with all other tasks, we can obtain the quantitative texture classification results by picking the most frequent pixel-level predictions as an image-level prediction. It yields classification accuracy of 35.10. The performance on texture indicates that only fine-tuning the network on texture labels is not optimal. However, this is a necessary step to overcome the fusion of natural and synthetic data sources. We hope future research can discover ways to better utilize such image-level annotations for pixel-level predictions.

Qualitative results. We provide qualitative results of UPerNet, as visualized in Figure 5. UPerNet is able to unify compositional visual knowledge and efficiently predicts hierarchical outputs simultaneously.

4.2 Discovering visual knowledge in natural scenes

Unified Perceptual Parsing requires a model that is able to recognize as many visual concepts as possible from a given image. If a model successfully achieves this goal, it could discover rich visual knowledge underlying the real world, such as answering questions like "What are the commonalities between living rooms and bedrooms?" or "What are the materials that make a cup?" The discovery or even the reasoning of visual knowledge in natural scenes will enable future vision systems to understand its surroundings better. In this section, we demonstrate



(a) Visualization of scene-object relations. Indoor scenes and outdoor scenes are clustered into different groups (left part of top image and right part of top image). We are also able to locate a common object appearing in various scenes, or find the objects in a certain scene (bottom left and bottom right).

(b) From left to right: visualizations of object-material relations, part-material relations and material-texture relations. We are able to discover knowledge such as some sinks are ceramic while others are metallic. We can also find out what can be used to describe a material.

Fig. 6. Visualizing discovered compositional relations between various concepts.

that our framework trained on the Broden+ is able to discover compositional visual knowledge at multiple levels. That is also the special application for the network trained on heterogeneous data annotations. We use the validation set of Places-365 [30] containing 36, 500 images from 365 scenes as our testbed, since the Places dataset contains images from a variety of scenes and is closer to real world. We define several relations in a hierarchical way, namely *scene-object* relation, *object-part* relation, *object-material* relation, *part-material* relation and *material-texture* relation. Note that only the object-part relations can be directly read out from the ground-truth annotations, other types of relations can only be extracted from the network predictions.

Scene-object relations. For each scene, we count how many objects show up normalized by the frequency of this scene. According to [44], we formulate the relation as a bipartite graph $G = (V, E)$ comprised of a set $V = V_s \cup V_o$ of scene nodes and object nodes together with a set E of edges. The edge with a weight from v_s to v_o represents the percent likelihood that object v_o shows up in scene v_s. No edge connects two nodes that are both from V_s or both from V_o. We filter the edges whose weight is lower than a threshold and run a clustering algorithm to form a better layout. Due to space limitations, we only sample dozens of nodes and show the visualization of the graph in Figure 6(a). We can clearly see that the indoor scenes mostly share objects such as ceiling, floor, chair, or windowpane while the outdoor scenes mostly share objects such as sky, tree, building, or mountain. What is more interesting is that even in the set of scenes, human-made and natural scenes are clustered into different groups. In the layout, we are also able to locate a common object appearing in various scenes, or find the objects in a certain scene. The bottom-left and bottom-right pictures in Figure 6(a) illustrate an example in which we can reasonably conclude that the shelf often appears in shops, stores, and utility rooms; and that in a heliport there are often trees, fences, runways, persons, and of course, airplanes.

Object(part)-material relations. Apart from scene-object relations, we are able to discover object-material relations as well. Thanks to the ability of our model to predict a label of both object and material at each pixel, it is straightforward to align objects with their associated materials by counting at each pixel what percentage of each material is in every object. Similar to the scene-object relationship, we build a bipartite graph and show its visualization in the left of Figure 6(b). Using this graph we can infer that some sinks are ceramic while others are metallic; different floors have different materials, such as wood, tile, or carpet. Ceiling and wall are painted; the sky is also "painted", more like a metaphor. However, we can also see that most of the bed is fabric instead of wood, a misalignment due to the actual objects on the bed. Intuitively, the material of a part in an object will be more monotonous. We show the part-material visualization in the middle of Figure 6(b).

Material-texture relations. One type of material may have various kinds of textures. But what is the visual description of a material? We show the visualization of material-texture relations in the right of Figure 6(b). It is worth noting that although there is a lack of pixel-level annotations for texture labels, we can still generate a reasonable relation graph. For example, a carpet can be described as matted, blotchy, stained, crosshatched and grooved.

5 Conclusion

This work studies the task of Unified Perceptual Parsing, which aims at parsing visual concepts across scene categories, objects, parts, materials and textures from images. A multi-task network and training strategy of handling heterogeneous annotations are developed and benchmarked. We further utilize the trained network to discover visual knowledge among scenes.

References

1. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
2. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proc. CVPR. (2017)
3. Hu, R., Dollár, P., He, K., Darrell, T., Girshick, R.: Learning to segment every thing. arXiv preprint arXiv:1711.10370 (2017)
4. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE (2014) 3606–3613
5. Liu, C., Sharan, L., Adelson, E.H., Rosenholtz, R.: Exploring features in a bayesian framework for material recognition. In: Proc. CVPR
6. Bell, S., Upchurch, P., Snavely, N., Bala, K.: Opensurfaces: A richly annotated catalog of surface appearance. ACM Transactions on Graphics (TOG) **32**(4) (2013) 111
7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
8. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. (2015)
9. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., et al.: Going deeper with convolutions, Cvpr (2015)
10. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. In: International Conference on Learning Representations (ICLR). (2014)
11. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 3431–3440
12. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1520–1528
13. Zeiler, M.D., Taylor, G.W., Fergus, R.: Adaptive deconvolutional networks for mid and high level feature learning. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 2018–2025
14. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: International Conference on Learning Representations (ICLR). (2016)
15. Lin, G., Milan, A., Shen, C., Reid, I.: Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017)
16. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (2017) 2881–2890
17. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision **88**(2) (2010) 303–338
18. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 3213–3223

19. Keeler, J.D., Rumelhart, D.E., Leow, W.K.: Integrated segmentation and recognition of hand-printed numerals. In: Advances in neural information processing systems. (1991) 557–563
20. Kokkinos, I., Maragos, P.: An expectation maximization approach to the synergy between image segmentation and object categorization. In: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. Volume 1., IEEE (2005) 617–624
21. Maire, M., Stella, X.Y., Perona, P.: Object detection and segmentation from joint embedding of parts and pixels. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 2142–2149
22. Elhoseiny, M., El-Gaaly, T., Bakry, A., Elgammal, A.: Convolutional models for joint object categorization and pose estimation. arXiv preprint arXiv:1511.05175 (2015)
23. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 2650–2658
24. Teichmann, M., Weber, M., Zoellner, M., Cipolla, R., Urtasun, R.: Multinet: Real-time joint semantic reasoning for autonomous driving. arXiv preprint arXiv:1612.07695 (2016)
25. Kokkinos, I.: Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (2017)
26. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: Quantifying interpretability of deep visual representations. In: Proc. CVPR. (2017)
27. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2014)
28. Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., Yuille, A.: Detect what you can: Detecting and representing objects using holistic models and body parts. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2014)
29. Zhou, B., Bau, D., Oliva, A., Torralba, A.: Interpreting deep visual representations via network dissection. In: IEEE Trans. on Pattern Analysis and Machine Intelligence. (2018)
30. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: Advances in neural information processing systems. (2014) 487–495
31. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR. Volume 1. (2017) 4
32. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Object detectors emerge in deep scene cnns. International Conference on Learning Representations (ICLR) (2015)
33. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on, IEEE (2016) 2921–2929
34. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. (2015) 448–456
35. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10). (2010) 807–814

36. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. arXiv preprint arXiv:1606.00915 (2016)
37. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE (2009) 248–255
38. Ioffe, S.: Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. In: Advances in Neural Information Processing Systems. (2017) 1942–1950
39. Peng, C., Xiao, T., Li, Z., Jiang, Y., Zhang, X., Jia, K., Yu, G., Sun, J.: Megdet: A large mini-batch object detector. arXiv preprint arXiv:1711.07240 (2017)
40. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, IEEE (2017) 5987–5995
41. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European conference on computer vision, Springer (2014) 818–833
42. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence **39**(12) (2017) 2481–2495
43. Kirillov, A., He, K., Girshick, R., Dollár, P.: Mscoco challenge 2017: stuff segmentation, team fair. (2017)
44. Brandes, U., Robins, G., McCranie, A., Wasserman, S.: What is network science? Network science **1**(1) (2013) 1–15