Specialize and Fuse: Pyramidal Representation for Semantic Segmentation

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

We present a novel pyramidal representation for semantic segmentation to take advantage of the typical scales of semantic classes (*e.g.*, a road segment is typically larger than a car segment). First, we define a “semantic pyramid" comprising semantic maps at various scales. Each map consists of a grid of cells, and a “unit-cell" contains pixels of a single class. To encourage parsimony, we carefully assign each pixel to the “unit-cell" at the coarsest scale and construct the “unity pyramid" to indicate the assignment. We end-to-end train a joint model to predict both pyramids. At inference, the predicted unity pyramid fuses the semantic pyramid into the final per-pixel semantic map. Our representation reduces the effective number of predictions in favor of parsimony since the number of unit-cells to be fused is significantly less than the number of pixels (*i.e.*, the standard output space). Moreover, our model learns to specialize in the prediction at each scale reflecting the natural distribution of unit-cell for each semantic class (*i.e.*, skies are typically assigned at coarser scales). Finally, we propose a coarse-to-fine contextual module that accords with the essence of our pyramidal representation for further improvements. We validate the effectiveness of each key module in our method through extensive ablation studies. Our approach achieves state-of-the-art performance on ADE20K and COCO-Stuff 10K datasets.

1 Introduction

Given an RGB image, semantic segmentation is aimed at predicting a semantic class for each pixel. Recent approaches widely exploit deep neural networks, and several modules are proposed upon them. Designing new contextual modules in a network is one major line that keeps making progress [4, 5, 6, 11, 23, 24, 25, 30]. However, exploring only the intermediate components for deep neural networks might constrain the improvement. We argue that by incorporating key observation about the task into our solution, we may gain further improvements in an orthogonal direction. We observe that a large portion of pixels in a common image can be predicted in a coarse spatial scale without loss of accuracy (*e.g.*, stuff or central region of an object), and only a small number of pixels need finer treatment. To embrace the *law of parsimony*, we redesign the output format from standard per-pixel to a hierarchical structure. We predict pyramidal semantic segmentation maps with each pixel assigned to the coarsest possible pyramid level that would not sacrifice precision (*i.e.*, a single prediction can be faithfully shared by all pixels it authorizes). A lightweight model head is trained to predict the assignment such that we can fuse the predicted pyramidal semantic maps into one as the final prediction in testing phase. Owing to our parsimony design principle, a very small number of predicted values take charge of constituting the final semantic map, and this design forms a new input and output distribution for the classifier at each pyramid level to specialize.

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

We characterize our key merits as follows: *i*) we introduce a pyramidal "output" representation for semantic segmentation that encourages parsimony and allows the semantic classifier of each level to specialize in different classes or regions; *ii*) we design a contextual module that fits the essence of our pyramidal representation and amplifies its performance gain; *iii*) we demonstrate state-of-the-art results on ADE20K and COCO-Stuff 10K datasets with the proposed model head only having FLOPs roughly the same as a 3×3 convolutional layer.

2 Related work

Contextual modules. Context is important for the task of semantic segmentation, with more and more improvements coming from the newly designed context spreading strategy. PSPNet [27] proposes to pool the deep feature into several small and fixed spatial resolutions to generate global contextual information. Deeplab [2] employs dilated CNN layers with several dilation rates, which help the model capture different ranges of context. Recently, self-attention methods [18, 20] achieve great success in natural language processing and computer vision, with many variants being proposed for semantic segmentation. DANet [6] employs self-attention in spatial dimension and channel dimension. CCNet [11] proposes criss-cross attention in which a pixel attends only to pixels of the same column or row. ANL [30] pools the feature to a fixed spatial size, which acts as the key and value of attention. OCR [23] pools the context according to a coarse prediction and employs attention between the deep feature and the classes centers. The CCNet [11], ANL [30], and OCR [23] greatly reduce the computation via specially designed attending strategy but still retain or even improve the performance. One of this work's proposals, the coarse-to-fine contextual module, is inspired by ANL [30] but cooperates better with another of our proposal—the pyramidal representation.

Hierarchical semantic segmentation prediction. Layer Cascade (LC) [13] predicts three semantic maps of the same resolution at three stages. At each stage, only uncertain pixels are passed to the next stage for further prediction. While all of the LC predictions are of the same scale, our method predicts multi-scale semantic maps trained under the principle of parsimony. Furthermore, LC fuses the semantic maps based on the semantic prediction itself. In contrast, we train a *fuser* with a carefully defined physical meaning to infer with the semantic pyramid.

Two most recent works, PointRend [12] and QGN [3], explore the direction of hierarchical prediction where the final semantic segmentation map is reconstructed in a coarse-to-fine manner instead of a dense prediction from the deep model directly. Both approaches start from the coarsest prediction. PointRend [12] gradually increases the resolution by sampling only uncertain points for the finer prediction, while QGN [3] predicts C+1 classes where the extra 'composite' class indicates whether a point would be propagated to a finer scale in their SparseConv [8] decoder. Both approaches yield high-resolution prediction (the same as input resolution) with their efficient sparseness design. PointRend [12] achieves better mIoU due to the finer results, while QGN [3] focuses on computational efficiency with inferior performance. Unlike their approaches, we predict dense pyramidal output with specially designed training and fusing procedures. Though our prediction, like most prior work, is coarser than [3, 12], our method achieves state-of-the-art performance with a significant improvement over baselines. The 'composite' class in QGN [3] saves the computation but degrades the performance. In contrast, owing to our specially designed training and fusing procedures on the dense output representation, our method can effectively improve with the principle of parsimony.

3 Approach

The core of our approach is fusing a predicted *semantic pyramid* according to a trained *unity pyramid* (Fig. 1). The *semantic pyramid* is a set of semantic maps of different spatial scales, and the *unity pyramid* comprises binary maps indicating whether a cell only covers a single semantic class. For inference, a fusion procedure generates the final prediction by considering both pyramids in a coarse-to-fine and non-repeating manner. Each cell in the fused semantic map comes from only one level of the semantic pyramid based on the unity prediction. For training, the per-pixel ground-truth labels would be dispatched to the coarsest possible level of pyramid, satisfying the condition that all pixels represented by a cell share the same semantic class. Thus, different levels of the semantic pyramid are trained to have their own specialization, while the trained unity pyramid can refer each pixel to the correct semantic pyramid level that is good at predicting the corresponding semantic.

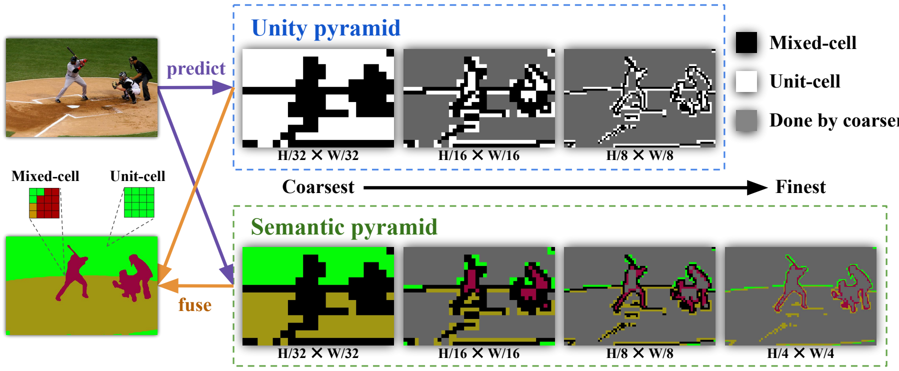

Figure 1: An overview of the proposed pyramidal output representation for semantic segmentation. Classifiers for different pyramid levels could specialize in different classes (*e.g.*, the coarsest scale is seldom responsible for thin and small objects) and areas (*e.g.*, central vs. boundary of an instance). Our model predicts two types of pyramids – *i*) **semantic pyramid** which is semantic prediction of 4 different scales, and *ii*) **unity pyramid** which comprises binary maps indicating whether a cell is 'mixed-cell' or 'unit-cell'. A mixed-cell means that the cell covers more than one semantic class and thus the finer pyramid level should be referred. In contrast, a unit-cell indicates that the semantic prediction of the corresponding cell in the semantic pyramid is ready to be used as a final prediction. To get the final prediction, the pyramid fusing procedure simply refers to the semantic prediction at the first unit-cell from the coarsest to the finest pyramid level for each pixel. Thus, the predictions covered by coarser unit-cells would be ignored (marked as "done by coarser").

3.1 Semantic pyramid and unity pyramid

Pyramid structure. Our pyramidal representation is under the *tree-pyramid* [17] structure, where a finer level has double resolution than its adjacent coarser level, and all cells except those in the finest level have exactly 4 children. Besides, the finest level of our pyramid is set to have the same spatial resolution as the output of CNN backbone, and the width and height of an input image should be divisible by those of the coarsest level (otherwise, we resize the input RGB to the nearest divisible).

Notation. Let D denote the backbone latent dimension, C the number of output classes, and ℓ the index of the pyramid level where $\ell = 1$ is the coarsest level and $\ell = L$ is the finest level (the same resolution as the backbone output in this work). The spatial stride at the pyramid level ℓ is denoted by s_ℓ. With the *tree-pyramid* structure in this work, we have $s_{\ell-1} = 2 \cdot s_\ell$, and s_L is the backbone output stride. Our model takes a feature tensor $X \in \mathbb{R}^{D \times \frac{H}{s_L} \times \frac{W}{s_L}}$ from the backbone, and predicts a **semantic pyramid** $\{\hat{Y}^{(\ell)} \in \mathbb{R}^{C \times \frac{H}{s_\ell} \times \frac{W}{s_\ell}}, \ell = 1, \dots, L\}$ and a **unity pyramid** $\{\hat{U}^{(\ell)} \in \mathbb{R}^{\frac{H}{s_\ell} \times \frac{W}{s_\ell}}, \ell = 1, \dots, L-1\}$. The real-valued predictions $\hat{Y}^{(\ell)}, \hat{U}^{(\ell)}$ can be converted into probabilities using $\mathrm{softmax}$ and $\mathrm{sigmoid}$, and then be converted into class indices and binary maps using argmax and threshold τ respectively, as shown at the top-right and top-left of Fig. 2. The fused semantic from the pyramid is denoted as $\hat{Y} \in \mathbb{R}^{C \times \frac{H}{s_L} \times \frac{W}{s_L}}$, which is the final output by our approach. The pyramidal ground truths $Y^{(\ell)}, U^{(\ell)}$ for training are derived from the conventional per-pixel semantic ground-truth labels Y.

Pyramidal ground truth. At pyramid level ℓ, each cell is responsible for a patch of $s_\ell \times s_\ell$ pixels in the original image. The proposed unity pyramid should tell whether it is ready to predict a shared semantic class for all pixels covered by a cell. Thus, the ground-truth binary values in $U^{(\ell)}$ indicate whether a cell is a 'unit-cell' (positive value, implying that all pixels covered by the cell share the same label) or a 'mixed-cell' (negative value). For the semantic pyramid, if a cell of $Y^{(\ell)}$ is a 'unit-cell', its semantic label is defined as the shared label by all covered pixels in the original per-pixel ground truth Y. Otherwise, the semantic supervision is still ambiguous, which means that an ideal fusion procedure should refer to a finer pyramid level for resolving the semantic during the inference phase. Thus, the semantic label of a 'mixed-cell' in the semantic pyramid is defined as "don't care". A special case is the finest scale ground truth $Y^{(L)}$, which is directly a downsampled version of Y.

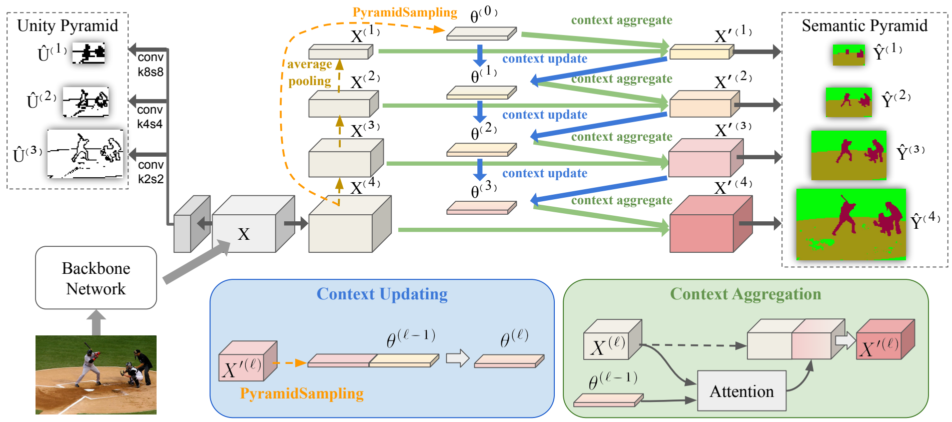

Figure 2: An overview of our network architecture. Both the unity head (top-left) and the semantic head (top-right) take the feature X from the backbone as input. The proposed coarse-to-fine contextual module is employed in the semantic head, and the details of its two operations—context updating and context aggregation—are illustrated in the bottom-central blue and bottom-right green boxes.

Here we skip how we deal with the ground-truth "don't care" annotations which originally exist in the per-pixel ground truth Y. We detail such cases in the supplementary material.

The training phase. Our experiments show that merely training the predicted $\hat{Y}^{(\ell)}, \hat{U}^{(\ell)}$ with their ground-truth counterparts $Y^{(\ell)}, U^{(\ell)}$ can NOT provide any improvement. This setting does not utilize the fact that a large number of pixels belonging to unit-cells are already predicted at a coarser level, and the finer level is still redundantly trained on those predicted regions. Based on the motivation to encourage parsimony and to train specialized pyramid levels, for those cells whose predecessors in the *tree-pyramid* structure are already correctly classified as unit-cells (true positives), our training procedure re-labels them as "don't care" on the fly (we refer to such labels as "done by coarser"). With the re-labeled ground truths in each mini-batch, the training loss is computed as follows:

$$\text{Loss} = \frac{1}{L}\sum_{\ell=1}^{L} \text{CE}(\hat{Y}^{(\ell)},\, Y^{(\ell)}_{\text{re-labeled}}) \;+\; \frac{1}{L-1}\sum_{\ell=1}^{L-1} \text{BCE}(\hat{U}^{(\ell)},\, U^{(\ell)}_{\text{re-labeled}})\,, \tag{1}$$

where CE is cross entropy and BCE is binary cross entropy. We show in the experiments that each level of the semantic pyramid has indeed learned to specialize in characterizing the assigned pixels, otherwise, the results would degrade if the predictions from other levels are also used.

Fuser—fusing semantic pyramid based on unity pyramid. During the training phase, pixel-level ground truths are dispatched to the appropriate pyramid level according to the ground truth labeling. The unity pyramid is trained to imitate the oracle dispatching behavior such that we can do the inverse operation—aggregating the pyramid into one final semantic segmentation map $\hat{Y}$. Given the predicted pyramids $\hat{Y}^{(\ell)}, \hat{U}^{(\ell)}$, the *fuser* follows a fusion procedure that refers each pixel to the semantic prediction at the coarsest unit-cell from its predecessor. For convenience, we assume all cells in the finest level are unit-cells ($\hat{U}^{(L)} = 1$). We now can write the fusing process as

$$\hat{Y} = \sum_{\ell=1}^{L} \left(\prod_{1\le k<\ell}^{\odot} \mathbb{1}\left[\text{Up}(\hat{U}^{(k)}) < \tau\right] \right) \odot \mathbb{1}\left[\text{Up}(\hat{U}^{(\ell)}) \ge \tau\right] \odot \text{Up}(\hat{Y}^{(\ell)})\,, \tag{2}$$

where $\text{Up}(\cdot)$ performs nearest-neighbor upsampling that keeps the *tree-pyramid* structure and resizes the prediction to the spatial resolution of $\hat{Y}$. The indicator function $\mathbb{1}$ thresholds the unity pyramid to binary maps. The underlying operation of the product $\prod$ is element-wise multiplication, which is denoted by $\odot$. For a unit-cell whose unity prediction is greater than the threshold (the second term), it checks whether all of its predecessors are mixed-cells (the product over $1 \le k < \ell$), and hence one and only one level is referred for each spatial location in $\hat{Y}$.

3.2 Predicting the unity pyramid

Fig. 2 illustrates the network architecture for predicting the unity pyramid $\hat{U}^{(\ell)}$ (at the top-left corner). It is a shared channel-reduction layer projecting D to D_u, which is then followed by different strided convolution layers to get the probability prediction at the desired scales. The stride and the kernel size of the CNN layers for the target pyramid level ℓ are set to $2^{(L-\ell)}$ (*e.g.*, if $L=4$, the kernel size and the stride are set to $8, 4, 2$ for $\ell = 1, 2, 3$ respectively). Note that we do not need the unity prediction at the finest level (*i.e.*, $\ell = L$) as there is no subsequent finer-level semantic prediction to refer. In other words, we assume $\hat{U}^{(L)} = 1$, and thus only $L-1$ levels are in the predicted unity pyramid.

3.3 Predicting the semantic pyramid with the coarse-to-fine contextual module

A naive way to predict the semantic pyramid $\hat{Y}^{(\ell)}$ from backbone feature X is pooling X to the L desired spatial sizes with each followed by their CNN layers that project latent dimension D to the number C of classes. With the proposed pyramidal representation, even the just mentioned naive network setting can already achieve promising improvements. Observing the recent success of context spreading strategy for semantic segmentation, we further design a top-down contextual module that fits the nature of our multi-scale semantic output and further boosts the performance.

Coarse-to-fine context updating and aggregation. Our motivation is to enrich the feature at all scales via global context and coarser scale information. Besides, the module should be efficient as we have multiple scales to process. Owing to the reasons above, we extend ANL [30]—a fast and memory-efficient variant of the non-local block [20]—to spread the context and augment the feature in a coarse-to-fine manner.

An overview of the proposed model head for the semantic pyramid is shown at the top-right of Fig. 2. A 1×1 convolution projecting D to D_s is employed to get the initial feature $X^{(L)}$ at the finest scale from backbone feature X, while the initial feature tensors at other desired scales $X^{(1)}, \dots, X^{(L-1)}$ are computed by average pooling from $X^{(L)}$. The operation $\mathrm{PyramidSampling}$ with a set of spatial sizes $N = \{1, 3, 6, 8\}$ is also applied on $X^{(L)}$ to initialize the global context $\theta^{(0)} \in \mathbb{R}^{D \times S \times 1}$ with $S = \sum_{n \in N} n^2$ is the adaptive average pooled feature followed by reshaping and concatenation. Two operations are defined as follows: **context aggregation** to obtain the final augmented multi-scale features $X'^{(1)}, \dots, X'^{(L)}$ and **context updating** to get the refined context $\theta^{(1)}, \dots, \theta^{(L-1)}$ from the coarser-scale feature and the previous context. Note that the resolution of the context $\theta^{(\ell)}$ is designed to be invariant to the pyramid level for saving computation.

Context aggregation. The context aggregation operation is illustrated in the bottom-right green box of Fig. 2. The average-pooled backbone feature $X^{(\ell)}$ aggregates the context and coarser-scale information through the enriched context $\theta^{(\ell-1)}$:

$$X'^{(\ell)} = F_{\mathrm{agg}}^{(\ell)} \left(\mathrm{concat} \left(\mathrm{Attention2D} \left(X_{\mathrm{q}}^{(\ell)}, \theta_{\mathrm{k}}^{(\ell-1)}, \theta_{\mathrm{v}}^{(\ell-1)} \right), X^{(\ell)} \right) \right) \text{ for } \ell = 1, \dots, L. \quad (3)$$

$X_{\mathrm{q}}^{(\ell)}$, which serves as the *query*, is transformed from $X^{(\ell)}$. The *key* $\theta_{\mathrm{k}}^{(\ell-1)}$ and the *value* $\theta_{\mathrm{v}}^{(\ell-1)}$ are transformed from $\theta^{(\ell-1)}$. All transformations are simply a $1{\times}1$ Conv layer. The operation 'concat' performs channel concatenation. For $\ell = 1$, the initial context $\theta^{(0)}$ is used. The layer F_{agg} projects the $2D$ feature concatenation back to the original D feature dimension, where a simple $1{\times}1$ convolutional layer is employed for $\ell = L$ while a residual block is employed for $\ell < L$ due to the much less number of spatial resolution points. The augmented feature $X'^{(\ell)}$ has two follow-up paths: *i*) a $1{\times}1$ convolutional non-linear projection layer to get the segmentation prediction $\hat{Y}^{(\ell)}$ at pyramid level ℓ and *ii*) an update of the global context for the finer-scale context aggregation operation.

Context updating. The context updating operation is illustrated in the bottom-central blue box of Fig. 2. The augmented feature $X'^{(\ell)}$ is semantic meaningful and close to the ℓth-level semantic prediction $\hat{Y}^{(\ell)}$ with only simple non-linear projection involved. We use $X'^{(\ell)}$ to update the context for later finer-scale feature aggregation with the motivation of enabling the finer-scale feature to know what has been predicted at the coarser scale.

$$\theta^{(\ell)} = F_{\mathrm{upd}}^{(\ell)} \left(\mathrm{concat} \left(\mathrm{PyramidSampling} \left(X'^{(\ell)} \right), \theta^{(\ell-1)} \right) \right) \text{ for } \ell = 1, \dots, L-1. \quad (4)$$

The operation of $\mathrm{PyramidSampling}$ works the same as context initialization and produces a tensor of shape $\mathbb{R}^{D\times S\times 1}$. A 1×1 channel reduction layer F_{upd} is applied to the concatenated feature to get the updated context $\theta^{(\ell)}$.

3.4 Computation efficiency and implementation details

We put the detailed derivation in the supplementary material. In summary, if $L = 4$, the number of flops of the overall model head for semantic pyramid is approximately only 1.3 times of a 3×3 Conv directly applied on the finest pyramid level as most of the operations are launched in the coarser pyramid levels where the numbers of spatial points are $\frac{1}{4}$, $\frac{1}{16}$, $\frac{1}{64}$ to the finest level. Please also see the supplementary material for the implementation details about the model and training hyperparameters.

4 Experiments

Datasets and metric. The quantitative segmentation quality is evaluated by the mean intersection-over-union (mIoU). **ADE20K [29]:** The ADE20K dataset is a scene parsing dataset containing 35 stuff classes and 115 objects classes. The data split for training and validation is 20K/2K. To conduct ablation study and model tuning, we randomly split the original 20K training set into 16K/4K. **COCO-Stuff 10K [1]:** The COCO-Stuff 10K dataset is a very challenging dataset. It contains 91 stuff and 80 object classes. The training and the test sets contain 9K and 1K images, respectively.

Comparison with state-of-the-arts. Following the literature, we apply multi-scale and left-right flip testing augmentation to report our best performance. Table 1 summarizes the comparison with recent state-of-the-arts. We choose HRNet [19] as our backbone due to the consideration that it consumes much less computational resources than the dilated ResNet101 [10] with output stride 8 (ResNet101-os8), and it yields high-resolution features for us to generate more pyramid levels without exploring the decoder. For a fairer comparison, we adapt two state-of-the-art methods—CCNet [11] and ANL [30]—by simply replacing their ResNet101-os8 backbone with HRNet48. Online hard pixel mining is also employed for the two baselines. Both CCNet [11] and ANL [30] significantly improve the performance of HRNet backbone and achieve similar or slightly better mIoU to their official ResNet101-os8 based implementation. Finally, compared with the competing state-of-the-art methods, our method still achieves superior performance on the two datasets.

4.1 Ablation study

We conduct extensive ablation studies to verify the effectiveness of our proposals and show the results in Table 2a. We put a detailed description for each experiment setting in the supplementary material and focus on the comparisons and analyses here. For all the experiments, the backbone is the light-weight HRNet32, and a sub-sampled ADE20K training split with 16K/4K is used for training and evaluation in the ablation study. Please note that all the data used here do not overlap with the official 2K validation split, which is used to report the best performance in Table 1. Each experiment in Table 2a is associated with an ID for easier discussion below.

The effectiveness of the pyramidal representation. (Related to *ABCFGH* in Table 2a) We show the effectiveness of the proposed pyramidal representation with various model settings. Under the naive Conv setting, we gain a $+1.65$ mIoU improvement (*A*→*F*). With an ANL module [30] appended to the backbone, the pyramidal representation still contributes to a $+1.05$ improvement (*B*→*G*). Finally, under the proposed coarse-to-fine contextual module, we achieve a significant $+2.31$ improvement (*C*→*H*). The results suggest that the proposed pyramidal representation, leveraging the principle of parsimony, alone can boost the naive setting, and continues improving after we apply the newly designed contextual module.

The effectiveness of the contextual module. (Related to *ABCFGH* in Table 2a) Under the common per-pixel semantic prediction, both the ANL [30] and the coarse-to-fine contextual module can improve the baseline performance by $+1.60$ (*A*→*B*) and $+1.58$ (*A*→*C*), respectively. However, we find that the results of the proposal *C* is similar to the simpler ANL *B*. On the other hand, under the proposed pyramidal representation, the ANL can improve the baseline by $+1.00$ mIoU (*F*→*G*) while the proposed coarse-to-fine version gains a significant $+2.24$ improvement (*F*→*H*). The proposed

Table 1: Quantitative comparison with previous arts on ADE20K [29] validation set and COCO-Stuff 10k [1] test set. The evaluation metric is *mIoU* (%). The 'bos' column indicates the backbone output stride while the 'os' column indicates the final prediction output stride.

Method	Venue	Backbone	bos	os	ADE20K	COCO-Stuff
DSSPN [16]	CVPR2018	ResNet101	8	8	43.68	38.9
EncNet [25]	CVPR2018	ResNet101	8	8	44.65	-
UperNet [21]	ECCV2018	ResNet101	32	4	42.66	-
PSANet [28]	ECCV2018	ResNet101	8	8	43.77	-
SGR [15]	NeurIPS2018	ResNet101	8	8	44.32	39.1
SVCNet [5]	CVPR2019	ResNet101	32	4	-	39.6
DANet [6]	CVPR2019	ResNet101	8	8	-	39.7
CFNet [26]	CVPR2019	ResNet101	8	8	44.89	-
APCNet [9]	CVPR2019	ResNet101	8	8	45.38	-
EMANet [14]	ICCV2019	ResNet101	8	8	-	39.9
CCNet [11]	ICCV2019	ResNet101	8	8	45.22	-
ANL [30]	ICCV2019	ResNet101	8	8	45.24	-
ACNet [7]	ICCV2019	ResNet101	16	2	45.90	40.1
OCR [23]	-	ResNet101	8	8	45.28	39.5
CPNet [22]	CVPR2020	ResNet101	8	8	46.27	-
QGN [3]	WACV2020	ResNet101	32	1	43.91	-
HRNet [19]	TPAMI2019	HRNet48	4	4	44.20	37.9
OCR [23]	-	HRNet48	4	4	45.50	40.6
CCNet [11]†	-	HRNet48	4	4	45.65	39.8
ANL [30]†	-	HRNet48	4	4	45.23	40.6
Ours	-	HRNet48	4	4	**47.37**	**41.4**

†Our reproduction by replacing the backbone with HRNet48

contextual module and ANL achieve similar mIoU under the common per-pixel prediction, but ours is better in the case of cooperating with the pyramidal representation—an extra $+1.24$ mIoU ($G \rightarrow H$) by replacing ANL with ours. Thus, we recommend using the simpler ANL [30] for common single-scale prediction and use the proposed coarse-to-fine contextual module for our pyramidal prediction.

Does the improvement stem from auxiliary supervision? (Related to *CHI* in Table 2a) One may argue that the improvement from the pyramidal representation is due to the rich supervision from the multi-scale outputs. Both *I* and *C* share the same network architecture but *I* has auxiliary supervision given to the multi-scale prediction while *C* is only supervised by the finest scale prediction. The auxiliary loss indeed can improve mIoU by $+0.63$ ($C \rightarrow I$). However, with the designed training and fusing procedure to encourage parsimony, the performance can be improved more significantly by $+2.31$ ($C \rightarrow H$)—an extra $+1.68$ mIoU added to the auxiliary loss setting ($I \rightarrow H$), which suggests that rich supervision is not the root cause to the improvement of the proposed pyramidal representation.

The training procedure. (Related to *ADEF* in Table 2a) In Sec. 3.1 we mention that merely training with the raw pyramid ground-truth $Y^{(\ell)}, U^{(\ell)}$ without parsimony cannot provide any improvement, which is verified by the roughly same performance of *A* and *D*. A simple fix is the setting of *E* that explicitly sets all descendants of a ground-truth unit-cell as "don't care". A clear $+1.12$ improvement is now shown by the simple fix ($A \rightarrow E$). However, the setting *E* still ignores the fact that the unit-cell prediction may produce false negatives where a ground-truth unit-cell is falsely classified as a mixed-cell and thus a finer-scale semantic prediction is referred in inference phase. As a result, the false negatives might make the train-test distributions inconsistent. Finally, the design of our true positive "don't care" re-labeling gains an extra $+0.53$ improvement ($E \rightarrow F$).

Does our pyramidal representation improve more on boundary regions? Finer levels of our pyramid are trained to focus more on semantic boundaries in contrast to standard per-pixel prediction. To examine its effect on the model performance, we show the mIoU evaluated at the boundary pixels and at the whole image in Table 2b. Pyramidal representation (*F*) and contextual module (*C*) individually achieve similar improvements over baseline *A* ($+1.65$ vs. $+1.58$). However, when evaluating

Table 2: Ablation studies and detailed evaluations. Please refer to Sec. 4.1 and Sec. 4.2 for detail.

(a) Ablation study for the proposals.

ID	Contextual module	Pyramid. rep.	Training procedure	mIoU (%)
A	-	-	-	40.42
B	ANL [30]	-	-	42.02
C	ours	-	-	42.00
D	-	✓	raw	40.41
E	-	✓	GT	41.54
F	-	✓	TP	42.07
G	ANL [30]	✓	TP	43.07
H	ours	✓	TP	**44.31**
I	ours	auxiliary	-	42.63

(b) mIoU on boundary pixels vs. all pixels.

	boundary	all
$A \rightarrow C$	+1.91	+1.58
$A \rightarrow F$	**+2.28**	+1.65
$C \rightarrow F$	+0.37	+0.07

(c) mIoU over pyramid levels.

$\ell' \setminus \ell$	4	3	2	1
4	**33.48**	39.52	42.13	38.61
3	31.63	**41.17**	45.98	44.57
2	27.31	38.13	**46.34**	47.52
1	21.94	28.94	40.05	**48.05**

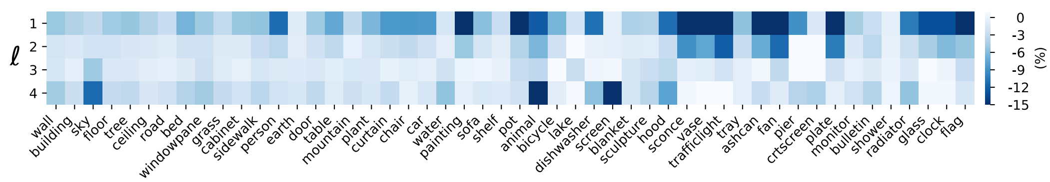

Figure 3: The performance of each single $\hat{Y}^{(\ell)}$ on different classes. For clearer visualization, we show the IoU difference between $\hat{Y}^{(\ell)}$ and the final fused $\hat{Y}$ instead of the original IoU of $\hat{Y}^{(\ell)}$.

on only the pixels close to semantic boundary (within an eight-pixel range), our representation (F) improves more than C (+2.28 vs. +1.91). We show more details in the supplementary material.

4.2 Performance analysis

Table 2c shows the performance of different pyramid levels. We first cluster the pixels into 4 disjoint groups (the columns), where the pyramid level ℓ is the level that a pixel should refer to according to our trained *fuser*. In each cluster, we show the performance of prediction by different semantic levels ℓ' (the rows). The rightmost column has higher mIoU because the pixels are generally at the central area of an instance and thus are easier to classify. We can see that the mIoU is degraded if a pixel refers to a pyramid level that does not agree with the *fuser* ($\ell' \neq \ell$). Therefore, different semantic pyramid levels can learn to specialize in predicting the pixels assigned by our fuser. In contrast, using any semantic pyramid level alone leads to degraded results.

To demonstrate our intuition that different pyramid levels have their specializations in different classes, in each row of Fig. 3, we show the per-class IoUs predicted by each semantic level. The results suggest that each $\hat{Y}^{(\ell)}$ learns better for different categories in practice. For instance, $\hat{Y}^{(4)}$ performs better at trafficlight, while $\hat{Y}^{(2)}$ is good at mountain. We put more visualizations and qualitative results in the supplementary material.

5 Conclusion

We present a novel “output" representation for the task of semantic segmentation. The proposed pyramidal “output" representation and the fusing procedure follow the motivation to assign each pixel to an appropriate pyramid level for better specialization under the parsimony principle. Improvements and motivations are shown through extensive experiments. A newly designed contextual module, which is efficient and fits the essence of the proposed pyramidal structure, improves the performance further. Finally, this work establishes new state-of-the-art results on two public benchmarks. We believe many new explorations on the efficiency and accuracy aspects can be built upon this work.

Broader Impact

Semantic segmentation is a fundamental task in computer vision for building intelligence systems that can perceive and understand the environment and scenes. Its applications like driving assistance, indoor navigation, and event monitoring are emerging in industry. Nevertheless, the robustness and reliability of a learning-based technique must be extensively investigated and validated under various conditions in addition to current benchmarks when it is considered to be implemented in a safety-critical system.

References

[1] H. Caesar, J. R. R. Uijlings, and V. Ferrari. Coco-stuff: Thing and stuff classes in context. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 1209–1218, 2018.

[2] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, pages 833–851, 2018.

[3] K. Chitta, J. M. Álvarez, and M. Hebert. Quadtree generating networks: Efficient hierarchical scene parsing with sparse convolutions. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2020, Snowmass Village, CO, USA, March 1-5, 2020*, pages 2009–2018, 2020.

[4] H. Ding, X. Jiang, A. Q. Liu, N. Magnenat-Thalmann, and G. Wang. Boundary-aware feature propagation for scene segmentation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6818–6828, 2019.

[5] H. Ding, X. Jiang, B. Shuai, A. Q. Liu, and G. Wang. Semantic correlation promoted shape-variant context for segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8885–8894, 2019.

[6] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu. Dual attention network for scene segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 3146–3154, 2019.

[7] J. Fu, J. Liu, Y. Wang, Y. Li, Y. Bao, J. Tang, and H. Lu. Adaptive context network for scene parsing. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6747–6756, 2019.

[8] B. Graham, M. Engelcke, and L. van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 9224–9232, 2018.

[9] J. He, Z. Deng, L. Zhou, Y. Wang, and Y. Qiao. Adaptive pyramid context network for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 7519–7528, 2019.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016.

[11] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu. Ccnet: Criss-cross attention for semantic segmentation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 603–612, 2019.

[12] A. Kirillov, Y. Wu, K. He, and R. B. Girshick. Pointrend: Image segmentation as rendering. *CoRR*, abs/1912.08193, 2019.

[13] X. Li, Z. Liu, P. Luo, C. C. Loy, and X. Tang. Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6459–6468, 2017.

[14] X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu. Expectation-maximization attention networks for semantic segmentation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 9166–9175, 2019.

[15] X. Liang, Z. Hu, H. Zhang, L. Lin, and E. P. Xing. Symbolic graph reasoning meets convolutions. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 1858–1868, 2018.

[16] X. Liang, H. Zhou, and E. P. Xing. Dynamic-structured semantic propagation network. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 752–761, 2018.

[17] M. Sonka, V. Hlavác, and R. Boyle. *Image processing, analysis and and machine vision (3. ed.)*. Thomson, 2008.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference*

on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, pages 5998–6008, 2017.

[19] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2019.

[20] X. Wang, R. B. Girshick, A. Gupta, and K. He. Non-local neural networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7794–7803, 2018.

[21] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. Unified perceptual parsing for scene understanding. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part V*, pages 432–448, 2018.

[22] C. Yu, J. Wang, C. Gao, G. Yu, C. Shen, and N. Sang. Context prior for scene segmentation. *CoRR*, abs/2004.01547, 2020.

[23] Y. Yuan, X. Chen, and J. Wang. Object-contextual representations for semantic segmentation. *CoRR*, abs/1909.11065, 2019.

[24] F. Zhang, Y. Chen, Z. Li, Z. Hong, J. Liu, F. Ma, J. Han, and E. Ding. Acfnet: Attentional class feature network for semantic segmentation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6797–6806, 2019.

[25] H. Zhang, K. J. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal. Context encoding for semantic segmentation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7151–7160, 2018.

[26] H. Zhang, H. Zhang, C. Wang, and J. Xie. Co-occurrent features in semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 548–557, 2019.

[27] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6230–6239, 2017.

[28] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. C. Loy, D. Lin, and J. Jia. Psanet: Point-wise spatial attention network for scene parsing. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IX*, pages 270–286, 2018.

[29] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ADE20K dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5122–5130, 2017.

[30] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai. Asymmetric non-local neural networks for semantic segmentation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 593–602, 2019.