A Holistically-Guided Decoder for Deep Representation Learning With Applications to Semantic Segmentation and Object Detection

Jianbo Liu, Junjun He, Yuanjie Zheng, Shuai Yi, Xiaogang Wang, and Hongsheng Li

Abstract—Both high-level and high-resolution feature representations are of great importance in various visual understanding tasks. To acquire high-resolution feature maps with high-level semantic information, one common strategy is to adopt dilated convolutions in the backbone networks to extract high-resolution feature maps, such as the dilatedFCN-based methods for semantic segmentation. However, due to many convolution operations are conducted on the high-resolution feature maps, such methods have large computational complexity and memory consumption. To balance the performance and efficiency, there also exist encoder-decoder structures that gradually recover the spatial information by combining multi-level feature maps from a feature encoder, such as the FPN architecture for object detection and the U-Net for semantic segmentation. Although being more efficient, the performances of existing encoder-decoder methods for semantic segmentation are far from comparable with the dilatedFCN-based methods. In this paper, we propose one novel holistically-guided decoder which is introduced to obtain the high-resolution semantic-rich feature maps via the multi-scale features from the encoder. The decoding is achieved via novel holistic codeword generation and codeword assembly operations, which take advantages of both the high-level and low-level features from the encoder features. With the proposed holistically-guided decoder, we implement the EfficientFCN architecture for semantic segmentation and HGD-FPN for object detection and instance segmentation. The EfficientFCN achieves comparable or even better performance than state-of-the-art methods with only 1/3 of their computational costs for semantic segmentation on PASCAL Context, PASCAL VOC, ADE20K datasets. Meanwhile, the proposed HGD-FPN achieves > 2% higher mean Average Precision (mAP) when integrated into several object detection frameworks with ResNet-50 encoding backbones.

Index Terms—Semantic segmentation, object detection, encoder-decoder, dilated convolution, holistic features, feature pyramids

1 INTRODUCTION

GENERATING high-resolution feature maps with rich semantic information is one of the fundamental components in modern deep neural networks with a wide variety of applications on semantic segmentation, instance segmentation, and object detection. Such tasks all require generating high-resolution and semantic-rich feature maps (e.g., 1/8 or even 1/4 of the input spatial size), which are then fed into the final segmentation or detection heads to generate the final predictions. However, given an input image, the general neural networks first perform pooling and strided convolutions to gradually decrease the spatial sizes of the feature maps and then to increase the receptive fields of the downsampled feature maps.

There are two common strategies for obtaining or recovering high-resolution feature maps. The first type of networks utilize dilated convolution [1], [2], [3] to directly avoid decreasing the spatial sizes of the feature maps too much. The second type of networks adopts the encoder-decoder architectures [4] that gradually upsamples lower-resolution feature maps via deconvolution or nearest/bilinear upsampling to generate feature maps of higher resolutions. However, both strategies have apparent limitations.

The dilated convolution (or atrous convolution) is a powerful technique that can effectively enlarge the receptive field (RF) while maintaining the resolution of higher-level feature maps. By utilizing multiple dilation rates, the dilated convolution can capture multi-scale context information without decreasing the resolution. Combining the dilated convolution with Fully Convolutional Network (FCN), the dilatedFCN based methods (shown in Fig. 1b) dominate state-of-the-art methods for semantic segmentation. As shown in Fig. 1b, by removing the downsampling operations and replacing convolution with the dilated convolution in the later blocks, the dilatedFCN backbone generates the final feature maps of output stride 8 (OS=8). Based on the dilatedFCN backbone, many researcher proposed their context modeling modules to further capture the long-range context information [1], [3], [5], [6], [7], [8]. Despite the superior performance and no extra parameters introduced from dilated convolutions, the high-resolution feature representations require very high computational cost and memory

Digital Object Identifier no. 10.1109/TPAMI.2023.3114342
For instance, for a $512^2$ image and a ResNet-101 backbone encoder, the computational cost of the encoder increases from 44.6 GFlops to 223.6 GFlops by adopting the dilated convolution of strides 2 and 4 into the last two blocks.

The encoder-decoder architecture, as shown in Fig. 1c, is widely adopted in semantic segmentation and object detection, e.g., U-Net [4] and SegNet [9], and Feature Pyramid Network (FPN) [10]. The decoder networks in semantic segmentation or the FPN module in object detection gradually upsample and generate the high-resolution feature maps by aggregating multi-scale feature maps from the encoder or the backbone network. However, on the one hand, the fine-grained structural details are already lost in the topmost high-level encoder feature maps of output stride 32 (OS=32) and the low-level semantic-less feature maps have more precise spatial information. Within U-Net and FPN, low-level and high-level features can indeed be fused to obtain semantic-rich features with the classical skip-connections. However, considering the semantic gap of low-level and high-level features, even with the skip connections, the fusion via simple addition or concatenation operations cannot effectively generate the high-resolution semantic-rich feature maps, which are demonstrated by their inferior segmentation performances. On the other hand, existing feature upsampling operators mainly utilize bilinear upsampling or deconvolution to increase the resolution of the high-level feature maps. These operations can only aggregate local information for feature upsampling. The feature representation at each location of the upsampled feature maps is therefore created from a small receptive filed, which limits the contextual scope of the features.

To tackle the challenges in both types of networks, we propose the Holistically-guided Decoder (HGD) (as illustrated in Fig. 1d) to effectively conduct feature upsampling guided by holistic context information from multi-scale feature maps.

Given the multi-level feature maps from the last three blocks from a common backbone encoder (e.g., ResNet-101), the proposed holistically-guided decoder takes the advantages of both high-level but low-resolution (OS=32) feature maps and also mid-level high-resolution feature maps (OS=8, OS=16). Intuitively, the higher-resolution but mid-level feature maps contain more fine-grained structural information, which is beneficial for spatially guiding feature upsampling; the lower-resolution but high-level feature maps contain more semantic-rich information, which are more suitable to encode the global context. Our HGD first generates a series of holistic codewords to summarize different global context of the input image from the low-resolution feature maps (OS=32). Those codewords are then properly assembled in a high-resolution grid to form the upsampled feature maps with rich semantic information. Following this principle, our HGD generates assembly coefficients from the mid-level high-resolution feature maps (OS=8, OS=16) to guide the linear assembly of the holistic codewords at each high-resolution spatial location to achieve feature upsampling.
To demonstrate the effectiveness and generalization of our proposed holistically-guided decoder, we apply it in semantic segmentation, object detection and instance segmentation. For semantic segmentation, we propose EfficientFCN by directly replacing conventional decoder in encoder-decoder architectures with the proposed HGD to generate semantically strong feature maps of OS=8. For object detection and instance segmentation, the proposed HGD are integrated into the FPN module to implement the HGD-FPN module. Since the original FPN module needs to output multiple feature maps with different scales, our HGD-FPN generates these feature maps by predicting multiple assembly coefficients with different scales and using one shared holistic codewords. In such a way, the feature pyramids generated by our proposed HGD-FPN consider the holistic semantic information from the multi-stage feature maps to boost the performance of object detection and instance segmentation.

Our primary contributions could be summarized into fourfold. (1) We propose a novel holistically-guided decoder, which can efficiently generate the high-resolution feature maps considering holistic contexts of the input image. (2) Because of the light weight and high performance of the proposed holistically-guided decoder, our proposed EfficientFCN for semantic segmentation can adopt the encoder without any dilated convolution but still achieve superior performance. (3) Our EfficientFCN achieves competitive (or better) results compared with the state-of-the-art dilatedFCN based methods on the PASCAL Context, PASCAL VOC, ADE20K datasets, with 1/3 fewer FLOPS. (4) For object detection and instance segmentation, compared to the conventional FPN, our proposed HGD-FPN achieves at least 2% higher mean Average Precision (mAP) for the classical Faster-RCNN [13], RetinaNet [14], FreeAnchor [15] and Mask-RCNN [16] frameworks with the ResNet-50 encoding backbone.

2 RELATED WORK

DilatedFCN based Methods for Semantic Segmentation The Deeplab V2 [1], [2] proposed to exploit dilated convolution in the backbone to generate high-resolution feature maps, which increases the output stride from 32 to 8. However, the dilated convolution in the last two layers of the backbone adds huge additional computation and leaves large memory footprint. Based on the dilated convolution backbone, many works [7], [17], [18], [19] continued to apply different strategies as the segmentation heads to acquire the context-enhanced feature maps. PSPNet [20] utilized the Spatial Pyramid Pooling (SPP) module to increase the receptive field. EncNet [5] proposed an encoding layer to predict a feature re-weighting vector from the global context and selectively high-lights class-dependent feature maps. CFNet [7] exploited an aggregated co-occurrence feature (ACF) module to aggregate the co-occurrence context by the pairwise similarities in the feature space. Gated-SCNN[21] proposed to use a new gating mechanism to connect the intermediate layers and a new loss function that exploits the duality between the tasks of semantic segmentation and semantic boundary prediction. DANet [17] proposed to use two attention modules with the self-attention mechanism to aggregate features from spatial and channel dimensions respectively. ACNet [18] applied a dilated ResNet as the backbone and combined the encoder-decoder strategy for the observation that the global context from high-level features helps the categorization of some large semantically confused regions, while the local context from lower-level visual features helps to generate sharp boundaries of clear details. DMNet [19] generated a set of dynamic filters of different sizes from the multi-scale neighborhoods for handling the scale variations of objects for semantic segmentation. Although these works further improve the performances on different benchmarks, these proposed heads still adds extra computational costs to the already burdensome encoder.

Encoder-Decoder Architecture for Semantic Segmentation Another type of methods focus on efficiently acquire the high-resolution semantic feature maps via the encoder-decoder architectures. Through the upsampling operations and the skip connections, the encoder-decoder architecture [4] can gradually recover the high-resolution feature maps for segmentation. DUsampling [22] designed a data-dependent upsampling module based on fully connected layers for constructing the high-resolution feature maps from the low-resolution feature maps. FastFCN [23] proposed a Joint Pyramid Upsampling (JPU) method via multiple dilated convolution to generate the high-resolution feature maps. One common drawback of these methods is that the feature at each location of the upsampled high-resolution feature maps is constructed via only local feature fusion. Such a property limits their performance in semantic segmentation, where global context is important for the final performance.

Multi-scale Feature Representations for Object Detection Deep learning based object detection methods have shown dramatic improvements in both accuracy and speed recently. Although many deep learning based object detection algorithms, such as Fast R-CNN [24] and Faster RCNN [13] could be conducted on a single layer’s feature maps to detect objects, detecting objects of a large variety of scales and aspect ratios from a single-scale feature map is still challenging. To remedy the issue, an intuitive way is to leverage multi-scale feature representations. There are two types of strategies to generate multi-scale feature representations: multi-scale image pyramids [25], [26] and multi-scale feature pyramids [10], [11], [27], [28]. SNIP [25] is one representative method that exploits multi-scale image pyramids, which proposed a scale normalization method that selectively trains the objects of appropriate sizes in each image scale. To improve the effectiveness of the multi-scale training, SNIPER [26] was proposed to select context regions around the ground-truth instances and sampling background regions for each scale during training. However, the inference time of these methods is too long to make these image pyramid-based methods less favorable for practical applications.

Diferent with the image pyramids, as shown in Fig. 2, feature-based pyramids aim to construct the multi-scale feature representations directly by exploiting the multi-layer features from the encoder network with only one input image. As one of the pioneering works, feature pyramid network (FPN) [10] introduced a top-down pathway and lateral connections between bottom-layer and top-layer features to create multi-scale feature maps. Motivated by

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
METHODS

(a) FPN [10] (b) PA-FPN [11] (c) NAS-FPN [12] (d) our proposed HGD-FPN

Fig. 2. Feature pyramid network (FPN) variants for object detection. (a) FPN [10] generates the feature pyramids for object detection and instance segmentation through the top-down pathway and lateral connections. (b) PA-FPN [11] extends FPN by introducing an extra bottom-up pathway to enhance information fusion. (c) NAS-FPN [12] applies neural architecture search to automatically discover how to merge features at different scales. (d) Our HGD-FPN adopts the proposed holistically-guided decoder to modify the original FPN architecture for generating more discriminative feature pyramids.

FPN, PANet [11] further enhanced the feature hierarchies by additional bottom-up path augmentation and proposed adaptive feature pooling to aggregate features from all levels. Libra RCNN [27] proposed the blanced FPN architecture which applies one non-local module to produce one balanced semantic features. CARAFE [28] incorporated a cascade structure and dynamic filtering to improve the quality of object proposals and detection results. More recently, NAS-FPN [12] leveraged neural architecture search to automatically design feature network topology. Although it achieves better performance, NAS-FPN [12] requires thousands of GPU hours during search, and the resulting feature network is irregular and thus difficult to interpret. Feature pyramids grids (FPG) [29] proposed a deep multi-pathway feature pyramid network by representing the feature scale-space as a regular grid of parallel pyramid pathways. However, similar to encoder-decoder methods for semantic segmentation, most of these FPN-variants reconstruct the multi-scale feature representations in a local feature fusion manner and cannot efficiently fuse the global semantic information into the final multi-stage feature maps. In this paper, our proposed HGD can be integrated with the FPN module to implement our HGD-FPN, which is able to generate discriminative multi-scale feature representations that contain the holistic semantic information from the multi-level feature maps.

3 METHODS

3.1 Overview

In state-of-the-art Deep Convolutional Neural Network (DCNN) backbones, the high-resolution mid-level feature maps at earlier stages can better encode fine-grained structures, while the low-resolution high-level features at later stages are generally more discriminative for category prediction. They are essential and complementary information sources for generating high-resolution semantic information from multi-level feature maps. Therefore, it is incapable of generating high-resolution semantic-rich feature maps to identify fine-grained image structures. To solve this drawback, we propose a novel Holistically-guided Decoder (HGD), as shown in Fig. 3, which decomposes the feature upsampling process into the generation of a series of holistic codewords from multi-level feature maps to capture global contexts, and linearly assembling codewords at each spatial location for achieving semantic-rich feature upsampling. Such a decoder can exploit the global contextual information to effectively guide the feature upsampling process and is able to recover fine-grained details. Thus, our proposed HGD can be integrated into the deep learning frameworks, where the high-resolution feature representations are needed, such as the semantic...
segmentation, instance segmentation, and object detection. In the following subsections, we first introduce our proposed HGD generating the high-resolution (OS=8) semantic-rich feature maps from multi-scale feature maps of OS=8, OS=16 and OS=32, which can be directly used as the decoder module into the encoder-decoder architectures for semantic segmentation. As object detection and instance segmentation are conducted on feature pyramids, we show how to generalize our HGD module with some minor modifications to implement one novel feature pyramid generation method HGD-FPN, which can encode discriminative multi-scale feature representations to significantly improve the performance of popular detection and instance segmentation frameworks, including Faster R-CNN [13], Mask-RCNN [16], RetinaNet [14] and FreeAnchor [15]. To help read this paper, we summarize the mathematical notations in Table I.

### Table I

| Notations | Descriptions |
|-----------|--------------|
| $e_k$, $f_k$, $m_k$ | The feature maps used in the semantic segmentation, where the number $k$ denotes its spatial resolution is $1/k$ times of the original resolution. |
| $C$ | The generated holistic codewords. |
| $G$ | The assembly guidance feature maps. |
| $W$ | The assembly coefficients. |
| $(P2 - P6)$ | The feature maps used in the object detection, where $(P2 - P6)$ correspond to the feature maps of scales OS=4 to OS=64. |
| $a_i, r_i, s_i, t_i$ | The linear non-negative parameters for feature fusion in object detection. |

3.2 Holistically-guided Decoder

**Multi-Scale Features Fusion.** Given the multi-scale feature maps from the encoder, we aim at generating a series of holistic codewords to encode different aspects of the input image’s global context, and linear assembly coefficients for combining the holistic codewords for creating the high-resolution feature maps. Without loss of generality, we assume that the encoder generates three feature maps with different scales (OS=8, 16 and 32) and the decoder generates semantic-rich feature maps of scale OS=8.

Although we can directly generate the holistic codewords from the topmost high-level OS=32 feature maps and also directly produce the codeword assembly coefficients from the mid-level OS=16 and OS=8 feature maps, we observe the simple fusion of multi-scale encoder feature maps can lead to better performance. Specifically, for the OS=8, OS=16, OS=32 encoder feature maps, we first adopt separate $1 \times 1$ convolutions to compress each of their channels to 512 for reducing the follow-up computational complexity, obtaining $e_8 \in \mathbb{R}^{512 \times (H/8) \times (W/8)}$, $e_{16} \in \mathbb{R}^{512 \times (H/16) \times (W/16)}$, $e_{32} \in \mathbb{R}^{512 \times (H/32) \times (W/32)}$, respectively, where $H$ and $W$ are the input image’s height and width. The multi-scale fused OS=32 feature maps $m_{32}$ are then obtained by downsampling $e_8$ and $e_{16}$ to the size of OS=32 and concatenating them along the channel dimension with $e_{32}$ as $m_{32} = [e_8; e_{16}; e_{32}] \in \mathbb{R}^{1536 \times (H/32) \times (W/32)}$, where $;$ represents bilinear downsampling, and $[;]$ denotes concatenation along the channel dimension. We can also obtain the multi-scale fused OS=8 feature maps, $m_{32} = [e_8; e_{16}; e_{32}] \in \mathbb{R}^{1536 \times (H/8) \times (W/8)}$, in a similar manner.

**Holistic Codeword Generation.** Although the multi-scale fused feature maps $m_{32}$ are created to integrate both high-level and mid-level features, their small resolutions make them lose many structural details of the scene. On the other hand, because $e_{32}$ is encoded from the deepest layer, $m_{32}$ is
able to encode rich categorical representations of the image. We therefore propose to generate a series of unordered holistic codewords from \( m_{32} \) to implicitly model different aspects of the global context.

To generate \( n \) holistic codewords, a codeword bases map \( B \in \mathbb{R}^{1024 \times (H/32) \times (W/32)} \) and \( n \) spatial weighting maps \( A \in \mathbb{R}^{n \times (H/32) \times (W/32)} \) are first computed from the fused multi-scale feature maps \( m_{32} \) by two separate \( 1 \times 1 \) convolutions. For the bases map \( B \), we denote \( B(x, y) \in \mathbb{R}^{1024} \) as the 1024-d feature of \( x, y \). With the guidance feature weighting maps eventu-

\[
\hat{A}_i(x, y) = \frac{\exp(A_i(x, y))}{\sum_{p,q} \exp(A_i(p,q))}.
\]

The \( i \)-th codeword \( c_i \in \mathbb{R}^{1024} \) can be obtained as the weighted average of all codeword bases \( B(x, y) \), i.e.,

\[
c_i = \sum_{p,q} \hat{A}_i(p,q) B(p,q).
\]

In other words, each spatial weighting map \( \hat{A}_i \) learns to linearly combine all codeword bases \( B(x, y) \) from all spatial locations to form a single codeword, which captures certain aspect of the global context. The \( n \) weighting maps eventually lead to \( n \) holistic codewords \( C = [c_1, \ldots, c_n] \in \mathbb{R}^{1024 \times n} \) to encode high-level global features.

**Codeword Assembly for High-Resolution Feature Upsampling.** The holistic codewords can capture various aspects of the global context of the input image. They are perfect ingredients for generating the high-resolution semantic-rich feature maps as they are encoded from the high-level features \( m_{32} \). However, since their structural information have been mostly removed during the downsampling process, we turn to use the OS=8 multi-scale fused features \( m_8 \) to predict the linear assembly coefficients of the \( n \) codewords at each spatial location for creating a high-resolution feature map.

More specifically, we first create a raw codeword assembly guidance feature map \( G \in \mathbb{R}^{1024 \times (H/8) \times (W/8)} \) to predict the assembly coefficients at each spatial location, which are obtained by applying a \( 1 \times 1 \) convolution on the multi-scale fused features \( m_8 \). However, the OS=8 fused features \( m_8 \) have no information on the holistic codewords as they are all generated from \( m_{32} \). We therefore consider the general codeword information as the global average vector of the codeword based map \( B \in \mathbb{R}^{1024} \) and location-wise add it to the raw assembly guidance feature map to obtain the novel guidance feature map \( G = G \odot B \), where \( \odot \) represents location-wise addition. Another \( 1 \times 1 \) convolution applied on the guidance feature map \( G \) generates the linear assembly coefficients or weighting map of the \( n \) codewords \( W \in \mathbb{R}^{n \times (H/8) \times (W/8)} \) for all \( (H/8) \times (W/8) \) spatial locations. By reshaping the weighting map \( W \) as an \( n \times (HW/8^2) \) matrix, the holistically-guided upsampled feature \( \tilde{f}_8 \) of OS=8 can be easily obtained as

\[
\tilde{f}_8 = CW.
\]

Given the holistically-guided upsampled feature map \( \tilde{f}_8 \), we obtain the final upsampled feature map \( \hat{f}_8 \) by concatenating the feature map \( \tilde{f}_8 \) with the guidance feature map \( G \), i.e., \( \hat{f}_8 = [\tilde{f}_8; G] \). Such an upsampled feature map \( \hat{f}_8 \) takes advantages of both \( m_8 \) and \( m_{32} \), and contains semantic-rich and also structure-preserved features for achieving one powerful high-resolution representation. If other output scales are needed (e.g., \( f_{16} \) of OS=16), they can be generated in the similar manner as \( \hat{f}_8 \).

### 3.3 EfficientFCN with HGD for Semantic Segmentation

Semantic segmentation aims to assign one categorical label to each pixel on the given image. For predicting the semantic segmentation result, one \( 1 \times 1 \) convolution layer is adopted as the final classifier appended to the final feature maps. Most of the state-of-the-art methods [1], [3], [5], [6], [7] use the feature maps with OS=8 to make the categorical prediction, whose results are then upsampled to the original input resolution with bilinear interpolation. How to obtain the high-resolution OS=8 feature maps with rich semantic information is therefore essential for achieving high segmentation accuracy. To compared with state-of-the-art methods, in this work, we focus on utilizing the proposed HGD to generate OS=8 feature maps, although we can use it to generate feature maps of even higher resolutions.

As illustrated in Fig. 1, there exist two types of architectures to acquire the high-resolution OS=8 feature maps for semantic segmentation. The DilatedFCN based methods [1], [3], [5], [6], [7] current dominate the benchmark leaderboards and replace the traditional convolution with the dilated convolution at the last two stages of the encoder network. However, these methods introduce very heavy computational overhead. For the traditional encoder-decoder methods [4], [22], the upsample operations, bilinear interpolation or deconvolution, generate the upsampled feature maps by fusing information of only local neighborhoods, which are sub-optimal for the semantic segmentation that needs contextual understanding.

To remedy these shortcomings, we propose our EfficientFCN architecture that integrates our HGD into the encoder-decoder architecture with commonly used encoder backbones (e.g., ResNet [31], etc.) but adopts HGD to replace existing upsampling operations in the conventional decoder.

Specifically, the encoder takes an image of interest as input for feature encoding and outputs the initial feature maps, \( f_{32}, f_{16}, f_{8} \), of three scales, which are then fed into our proposed HGD to generate the 

### 3.4 HGD-FPN for Object Detection and Instance Segmentation

The Feature Pyramid Network (FPN) [10] and its variants [11], [12], [28], [29] are commonly adopted to generate multi-scale feature representations for capturing objects of different scales in object detection and instance segmentation. The FPN has been shown as one of the most important
components in modern deep learning based object detectors. Similar to the problems of encoder-decoder architectures for semantic segmentation, most existing FPN methods for object detection mainly generate the multi-scale feature representations from information of local neighborhoods via nearest interpolation and the (de-)convolution operator with a small and limited receptive field. Obviously, it cannot efficiently fuse global semantic information into the final multi-scale feature maps.

To better incorporate global context into the generation of multi-scale feature maps, we integrate our proposed HGD module into the classical FPN architecture and implement one novel feature pyramids method, which is named HGD-FPN, for object detection and instance segmentation. Unlike HGD in EfficientFCN, which only needs to generate high-resolution feature maps of a single scale $OS=8$, HGD-FPN needs to generate feature maps of multiple scales. Therefore, as shown in Fig. 4, our designed HGD-FPN architecture makes modifications to the three above mentioned operations: multi-scale encoder feature fusion, holistic codeword generation and codeword assembly for feature upsampling.

Specifically, for codeword generation we first fuse all multi-scale encoder feature maps ($P2 – P6$) so that the learned holistic codewords contain semantic information of different scales. We upsample the feature map $P6$ and downsample the feature maps ($P2 – P4$) to make them have the same resolution as $P5$ via nearest interpolation or max-pooling operation. The multiple feature maps are linearly combined to generate the multi-scale fused feature maps $m_{\text{code}}$,

$$m_{\text{code}} = a_0 \cdot \uparrow P6 + a_1 \cdot P5 + a_2 \cdot \downarrow P4 + a_3 \cdot \downarrow P3 + a_4 \cdot \downarrow P2,$$

where the symbols $\uparrow$ and $\downarrow$ denote the nearest upsampling interpolation and max-pooling operations, respectively. For one-stage detectors, feature maps $P2 – P6$ are replaced with $P3 – P7$.

Fig. 4. Illustration of the proposed HGD-FPN architecture for object detection and instance segmentation. HGD-FPN integrates our proposed holistically-guided decoder into the FPN architecture to generate the feature pyramids. It also consists of three main components. Given the multi-scale feature maps $P2 – P6$, the multi-scale feature fusion component fuses multi-scale features for the codeword generation and codeword assembly components respectively. The HGD-FPN applies the codeword generation component to produce one shared set of holistic codewords from the multi-scale feature maps $P2 – P6$. Then, it exploits three codeword assembly components to generate three feature maps with different scales. Finally, the residual connections are used to generate the output feature pyramids. In our HGD-FPN for object detection, the symbols $\uparrow$ and $\downarrow$ denote the nearest upsampling interpolation and max-pooling operations, respectively. For one-stage detectors, feature maps $P2 – P6$ are replaced with $P3 – P7$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
where the fusion coefficients \( r_i, s_i, t_i \) are learned non-negative parameters and satisfy \( \sum r_i = 3, \sum s_i = 3 \) and \( \sum t_i = 3 \). \( m_3, m_4, m_5 \) can be used to generate the codeword assembly coefficients or weighting maps of these three scales with different and independent operations (i.e., convolution kernels are not shared across scales) following the descriptions in Section 3.2. The reconstructed feature maps \( P_3, P_4 \) and \( P_5 \) (corresponding to the scales of \( P_3, P_4, P_5 \)) can then be obtained by linearly combining the codewords \( C \) with the linear assembly coefficients following Eq. (3). On the other hand, for feature maps \( P_2 \), since they have too large resolution, to avoid too much computation, we directly upsample \( P_3 \) to generate \( P_2 \), i.e., \( P_2 = \uparrow P_3 \). Similarly, as \( P_6 \) has too small resolution, we directly downsample \( P_5 \) to generate \( P_6 \), i.e., \( P_6 = \downarrow P_5 \). Residual connections are further added between \( P_2-P_6 \) and \( P_2-P_6 \) to generate the final output feature maps \( P_2^*-P_6^* \) for the proposed HGD-FPN,

\[
\begin{align*}
P_2^* &= P_2 + \hat{P}_2, \\
P_3^* &= P_3 + \hat{P}_3, \\
\hat{P}_4^* &= P_4 + \hat{P}_4, \\
\hat{P}_5^* &= P_5 + \hat{P}_5, \\
P_6^* &= P_6 + \hat{P}_6.
\end{align*}
\]

The resulting multi-scale feature maps \( P_2^*-P_6^* \) can be used similarly to those from conventional FPN for ROI pooling in existing object detection frameworks. Such multi-scale features incorporate global contexts and can help improve the final detection performance. Following the latest work on recurrent FPN architectures [12], [29], [32], in our HGD-FPN, we find our HGD-FPN module can also be stacked for multiple \( k \) times to further enhance the representation ability of the resulting multi-scale feature maps. We set \( k = 4 \) and other different \( k \)'s are also explored in experiments.

We integrate the proposed HGD-FPN into three popular object detection frameworks (Faster RCNN[13], RetinaNet [14] and FreeAnchor [15]) and one instance segmentation network (Mask RCNN [16]). ResNet is adopted as the encoder network and the classical FPN is exploited to produce the multi-scale semantic features \( (P_2-P_6) \) of channel dimension 256, which are input into our HGD-FPN to generate the output multi-scale feature maps. For Faster RCNN and Mask RCNN, the feature maps \( (P_2-P_6) \) correspond to the feature maps of scales \( OS=4 \) to \( OS=64 \). For RetinaNet and FreeAnchor, the feature maps \( (P_2-P_6) \) are replaced with feature maps \( (P_3-P_7) \) which correspond to the feature maps of scales \( OS=8 \) to \( OS=128 \).

4 Experiments

In this section, we introduce the implementation details, training strategies and evaluation metrics of our experiments for three applications: semantic segmentation, object detection and instance segmentation. For semantic segmentation, to evaluate our proposed EfficientFCN model, we conduct comprehensive experiments on three public datasets: PASCAL Context [33], PASCAL VOC 2012 [34] and ADE20K [35]. To further evaluate the contributions of individual components in our model, we conduct detailed ablation studies on the PASCAL Context dataset. Then, we perform experiments on the MS COCO 2017 dataset [36] to demonstrate the capability of our proposed HGD-FPN module. In principle, our proposed HGD-FPN module can be flexibly integrated into any methods for object detection and instance segmentation by replacing the FPN module with our HGD-FPN. In this paper, we choose four representative frameworks, Faster RCNN [13], RetinaNet [14], FreeAnchor [15] and Mask RCNN [16], to verify its effectiveness.

4.1 Semantic Segmentation

4.1.1 Implementation Details

Network Structure. Different with the dilatedFCN based methods, which remove the stride of the last two blocks of the backbone networks and adopt the dilated convolution with the dilation rates 2 and 4, we use the original ResNet [31] as our encoder backbone network. Thus the size of the output feature maps from the last ResBlock is \( 32 \times \) smaller than that of the input image. After feeding the encoder feature maps into our proposed holistic-guided decoder, the classification is performed on the output upsampled feature map \( f_k \). The ImageNet [37] pre-trained weights are utilized to initialize the encoder network.

Training Settng. A poly learning rate policy [1] is used in our experiments. We set the initial learning rates as 0.001 for PASCAL Context [33], 0.002 for PASCAL VOC 2012 [34] and ADE20K [35]. The power of poly learning rate policy is set as 0.9. The optimizer is stochastic gradient descent (SGD) [38] with momentum 0.9 and weight decay 0.0001. We train our EfficientFCN for 120 epochs on PASCAL Context, 80 epochs on PASCAL 2012, 120 epochs on ADE20K and 400 epochs on Cityscapes, respectively. We set the crop size to \( 512 \times 512 \) on PASCAL Context and PASCAL 2012. Since the average image size is larger than other two datasets, we use \( 576 \times 576 \) as the crop size on ADE20K and use \( 864 \times 864 \) as the training size on Cityscapes. For data augmentation, we only randomly flip the input image and scale it randomly in the range \([0.5, 2.0]\).

Evaluation Metrics. We choose the standard evaluation metrics of pixel accuracy (pixAcc) and mean Intersection of Union (mIoU) as the evaluation metrics in our experiments. Following the best practice [5], [6], [17], we apply the strategy of averaging the network predictions in multiple scales for evaluation. For each input image, we first randomly resize the input image with a scaling factor sampled uniformly from \([0.5, 2.0]\) and also randomly horizontally flip the image. These predictions are then averaged to generate the final prediction.

4.1.2 Results on PASCAL Context

The PASCAL Context dataset consists of 4,998 training images and 5,105 testing images for scene parsing. It is a complex and challenging dataset based on PASCAL VOC 2010 with more annotations and fine-grained scene classes, which includes 59 foreground classes and one background class. We take the same experimental settings and evaluation strategies following previous works [5], [7], [17], [18], [19]. We first conduct ablation studies on this dataset to demonstrate the effectiveness of each individual module design of our proposed EfficientFCN and then compare our
model with state-of-the-art methods. The ablation studies are conducted with a ResNet101 encoder backbone.

**Comparison With the Classical Encoder-Decoders.** For the classical encoder-decoder based methods, the feature upsampling is achieved via either bilinear interpolation or deconvolution. We implement two classical encoder-decoder based methods, which include the feature upsampling operation (bilinear upsampling or deconvolution) and the skip-connections. To verify the effectiveness of our proposed HGD, these two methods are trained and tested on the PASCAL Context dataset with the same training setting as our model. The results are shown in Table 2. Although the classical encoder-decoder methods have similar computational complexities, their performances are generally far inferior than our EfficientFCN. The key reason is that their upsampled feature maps are created in a local manner. The simple bilinear interpolation or deconvolution cannot effectively upsample the OS=32 feature maps even with the skip-connected OS=8 and OS=16 feature maps. In contrast, our proposed HGD can effectively upsample the high-resolution semantic-rich feature maps not only based on the fine-grained structural information in the OS=8 and OS=16 feature maps but also from the holistic semantic information from the OS=32 feature maps.

We conduct comparison with the light-weight decoder of DeepLabV3+ [39] but with the same ResNet101 pretrained model as an encoder on PASCAL-Context dataset. With the same experimental settings, the performance and FLOPS are added in Table 1. As shown in Table 1, our proposed HGD significantly increases the accuracy with the less computational load and the similar parameters.

**Comparison With the Classical HRNet [40].** We also compare our method with the recent HRNet[40]. HRNet[40] is one important work that can produce high-resolution representations by connecting high-to-low resolution convolutions in parallel and repeatedly conducting multi-scale fusions across parallel streams. HRNet can be regarded as one novel encoder network while our HGD is one decoder module. They share the same motivation but adopt different complementary strategies. We setup two comparisons with the HRNetV2-W48. As shown in Table 1, as one powerful feature encoder to produce high resolution feature maps, the performance of the HRNetV2-W48 encoder is much better than that of the ResNet101 encoder. However, the combination of ResNet-101 and our HGD is able to obtain better accuracy than HRNetV2-W48 with fewer parameters and less computational cost (69.6GFlops vs. 94.2GFlops on the same test size). In addition, we implement one model by combining our proposed decoder HGD with the HRNet encoder. Since the output of HRNet has four feature maps from OS=4 to OS=32, we replace the input of our HGD with four feature maps and make it upsample to obtain feature maps of OS=4. The combination of HRNet encoder and our proposed HGD could further improve the accuracy (mIoU: 53.2% → 55.6%).

**Multi-Scale Feature Fusion.** We conduct two experiments on multi-scale features fusion to verify their effects on holistic codeword generation and codeword assembly for feature upsampling. In our holistically-guided decoder, the holistic codewords are generated based on the OS=32 multi-scale fused feature maps \( m_{32} \) and the codeword assembly coefficients are predicted from the OS=8 multi-scale fused feature maps \( m_s \). For the codeword generation, we conduct three experiments to generate the holistic codewords from multi-scale fused features with different numbers of scales. As shown in Table 3, when reducing the number of fusion scales from 3 to 2 and from 2 to 1, the performances of our EfficientFCN slightly decrease. The phenomenon is reasonable as the deepest feature maps contain more categorical information than the OS=8 and OS=16 feature maps. For the codeword assembly coefficient estimation, the similar experiments are conducted, where results are shown in Table 4. However, different from the above results, when fewer scales of feature maps are used to form the multi-scale fused OS=8 feature map \( m_s \), the performances of our EfficientFCN show significant drops. These results demonstrate that although the OS=8 feature maps contain more fine-grained structural information, the semantic features from higher-level feature maps are essential for guiding the recovery of the semantic-rich high-resolution feature maps.

**Number of Holistic Codewords.** We also conduct experiments to survey the effectiveness of the number of codewords in our predicted holistic codewords for feature upsampling. As shown in Table 5, when \( n = 0 \), without the proposed HGD, the prediction is made only from the feature maps \( m_s \) and its mIoU is 49.0%, which shows that the guided upsampling of HGD could dramatically improve

| Method         | Backbone      | OS  | mIoU% | Parameters (MB) | GFlops (G) |
|----------------|---------------|-----|-------|-----------------|------------|
| FCN-32s        | ResNet101     | 32  | 43.3  | 54.0            | 44.6       |
| dilatedFCN-8s | dilated-ResNet101 | 8   | 47.2  | 54.0            | 223.6      |
| UNet-Bilinear  | ResNet101     | 8   | 49.3  | 60.7            | 87.9       |
| UNet-Deconv    | ResNet101     | 8   | 49.1  | 62.8            | 93.2       |
| DeepLabV3+     | ResNet101     | 4   | 53.5  | 62.3            | 112.7      |
| HRNetV2-W48    | -             | 4   | 53.2  | 65.8            | 94.2       |
| EfficientFCN   | ResNet101     | 8   | 55.3  | 55.8            | 69.6       |
| EfficientFCN   | HRNetV2-W48   | 4   | 55.6  | 68.2            | 103.5      |

| Number of Holistic Codewords | pixAcc | mIoU |
|------------------------------|--------|------|
| \( (32) \)                  | 80.0   | 54.8 |
| \( (16,32) \)               | 80.1   | 55.1 |
| \( (8,16,32) \)             | 80.3   | 55.3 |
with the holistically-guided decoder, the comparisons with state-of-the-art methods are shown in Table 6. The dilatedFCN based methods dominate the top performances for semantic segmentation. However, our work is still able to achieve the best results compared to the dilatedFCN based methods on the PASCAL Context validation set without using any dilated convolution and has significantly less computational cost. Because of the efficient design of our HGD, our EfficientFCN only has 1/3 of the computational cost of state-of-the-arts methods but can still achieve the best performance.

**Importance of the Codeword Information Transfer for Accurate Assembly Coefficient Estimation.** The key of our proposed HGD is how to linearly assemble holistic codewords at each spatial location to form high-resolution upsampled feature maps based on the feature maps $m_8$. In our HGD, although the OS=8 features have well maintained structural image information, we argue that directly using OS=8 features to predict codeword assembly coefficients are less effective since they have no information about the codewords. We propose to transfer the codeword information as the average codeword basis, which is location-wisely added to the OS=8 feature maps. To verify this argument, we design an experiment that removes the additive information transfer, and only utilizes two $1 \times 1$ convolutions with the same output channels on the OS=8 feature maps $m_8$ for directly predicting assembly coefficients. The mIoU of this implementation is 54.2%, which has a clear performance drop if there is no codeword information transfer from the codeword generation branch to the codeword coefficient prediction branch.

**Visualization of the Weighting Maps and Example Results.** To better interpret the obtained holistic codewords, we visualize the weighting maps $\tilde{A}$ for creating the holistic codewords in Fig. 5, where each column shows one weighting map $\tilde{A}_i$ for generating one holistic codeword. Some weighting maps focus on summarizing foreground objects or regions to create holistic codewords, while some other weighting maps pay attention to summarizing background contextual regions or objects as the holistic codewords. The visualization shows that the learned codewords implicitly capture different global contexts from the scenes. In Fig. 6, we also visualize some predictions by the baseline DilatedFCN-8s and by
our EfficientFCN, where our model significantly improves the visualized results with the proposed HGD.

4.1.3 Results on PASCAL VOC
The original PASCAL VOC 2012 dataset consists of 1,464 images for training, 1,449 for validation, and 1,456 for testing, which is a major benchmark dataset for semantic object segmentation. It includes 20 foreground objects classed and one background class. The augmented training set of 10,582 images, namely train-aug, is adopted as the training set following the previous experimental set in [7]. To further demonstrate the effectiveness of our proposed HGD. We adopt all the best strategies of HGD design and compare it with state-of-the-art methods on the test set of PASCAL-VOC 2012, which is evaluated on the official online server. As shown in Table 7, the dilatedFCN based methods dominate the top performances on the PASCAL VOC benchmark. However, our EfficientFCN with a backbone having no dilated convolution can still achieve the best results among all the ResNet101-based methods.

4.1.4 Results on ADE20K
The ADE20K dataset consists of 20K images for training, 2K images for validation, and 3K images for testing, which were used for ImageNet Scene Parsing Challenge 2016. This dataset is more complex and challenging with 150 labeled classes and more diverse scenes. As shown in Table 6, our EfficientFCN achieves the competitive performance than the dilatedFCN based methods but has only 1/3 of their computational cost.

4.1.5 Results on Cityscapes
Cityscapes [58] densely annotates 19 object categories in images with urban scenes. It contains 5000 finely annotated images including 2975, 500 and 1525 for training, validation and testing respectively. The images are all captured at a high resolution of $2048 \times 1024$. Due to the image resolution in the Cityscapes dataset is very large, we use four input multi-scale features (OS = 4, 8, 16 and 32) from the encoder ResNet-101 and make the prediction on the generated feature maps (OS = 4). As shown in the Table 8, the

Fig. 5. (a) Input images from the PASCAL Context and ADE20K dataset. (b-e) Different weighting maps $\tilde{A}_i$ for creating the holistic codewords.

Fig. 6. Visualization results from the PASCAL Context dataset.
performance of our method EfficientFCN achieves 82.1%, which are both better than that of the recently dilated-ResNet based methods and the recently proposed HRNet.

4.2 Object Detection and Instance Segmentation

We verify the effectiveness of our proposed HGD-FPN to generate multi-scale feature pyramids for object detection and instance segmentation on COCO 2017 detection dataset [36]. This dataset contains objects of 80 categories. The training dataset consists of ~118k samples, and the validation dataset contains 5k images (minival). The performances on the minival set are reported. The evaluation metrics include Average Precision (AP), AP$_{50}$ (AP for IoU threshold 50%), AP$_{75}$ (AP for IoU threshold 75%), AP$_M$ (results on small scale), AP$_L$ (results on medium scale) and AP$_L$ (results on large scale).

We integrate and test our HGD-FPN in three object detection frameworks (Faster RCNN [13], RetinaNet [14] and FreeAnchor [15]), and one framework for instance segmentation (Mask RCNN [16]) with ResNet-50 [31] backbones. The 1x training schedule settings are adopted following Detectron [59] and MMDetection [60]. For fair comparison, all experiments are conducted based on MMDetection [60] implementations. The default parameters for our HGD-FPN are set as: the number of codewords $n = 128$, the dimension of codewords $c = 512$, and the FPN recurrence number $k = 4$. The HGD-FPN module shares its parameters at different recurrence stages.

4.2.1 Ablation Studies

To investigate the effects of different components of our proposed HGD-FPN, we design a series of experiments on HGD-FPN integrated in the Faster RCNN [13] framework.

Dimensions of the Shared Holistic Codewords. For our HGD-FPN, the core contribution is to generate the multi-scale feature maps via a linear combination of the predicted holistic codewords. Then, the number of codewords $n$ and the dimension of the codewords $c$ have major impacts to the performance of the proposed HGD-FPN. As the channel of the input feature pyramids $f_P$ have been reduced to 256 [13], [14], we design a series of experiments with varying $n$ and $c$ to investigate their influences in Table 9.

The implementation with $n = 0$ and $c = 0$ means keeping only the multi-feature fusion operation while removing the holistic codeword generation and codeword assembly operations in our HGD module. The HGD-FPN with $n = 0$ and $c = 0$ would still lead to a better result than the original FPN method. Second, compared with the performance of our HGD-FPN with $n = 0$ and $c = 0$, the results of our HGD-FPN with $n = 1$ and $c = 1$ show significant improvements, which demonstrate that our proposed HGD module can generate discriminative multi-scale feature maps for object detection. Finally, as $n$ and $c$ increase, the better performance can be achieved with higher computational cost. We choose $n = 128$ and $c = 512$ as our default parameter settings in the following experiments as trade-offs between the performance and the computational efficiency.

The Recurrence Number $k$ of HGD-FPN. With the fixed dimensions for the holistic codewords, we further investigate the effect of the recurrence number $k$ of the proposed HGD-FPN module. We conduct five experiments with varying $k$’s from 1 to 5. As shown in the Table 11, increasing the

### TABLE 7

| Method         | Backbone | mIoU (%) |
|---------------|----------|----------|
| PSNet [51]    | dilated-ResNet101 | 80.1     |
| ANN [44]      |          | 81.3     |
| CNPNet [52]   |          | 81.3     |
| CCNet [53]    |          | 81.4     |
| DANet [17]    |          | 81.5     |
| SpysGr [54]   |          | 81.6     |
| ACNet [55]    |          | 81.8     |
| OCR [56]      |          | 81.8     |
| SNeT [57]     |          | 81.8     |
| HRNNet [57]   | HRNetV2-W48 | 81.9     |
| Ours           | ResNet101 | 82.1     |

Our EfficientFCN obtains 85.4% without MS COCO dataset pre-training and 87.6% with MS COCO dataset pre-training. (For each column, the best two entries are masked in shaded regions.)
The Effects of Different Dimensions of Holistic Codewords in Our HGD-FPN When Integrated in Faster RCNN With a ResNet-50 Backbone on MS COCO 2017 minival

| Feature Fusion     | $n$  | $c$ | FLOPs (G) | Params (M) | Task   | AP  | AP$_{50}$ | AP$_{75}$ | AP$_{S}$ | AP$_{M}$ | AP$_{L}$ |
|-------------------|------|----|----------|-----------|--------|-----|-----------|-----------|---------|---------|---------|
| FPN [10]          | -    | -  | 214.8    | 41.5      | BBox   | 36.3| 58.2      | 39.1      | 21.6    | 40.1    | 46.4    |
| HGD-FPN           | 0    | 0  | 266.4    | 43.3      | BBox   | 37.5| 59.2      | 40.3      | 21.6    | 41.0    | 48.6    |
|                   | 64   | 256| 384.9    | 47.9      | BBox   | 39.4| 61.3      | 42.4      | 24.6    | 42.5    | 51.0    |
|                   | 512  | 554.8| 54.2     | 42.7      | BBox   | 39.7| 61.6      | 42.7      | 24.4    | 43.3    | 50.9    |
|                   | 1024 | 894.8| 66.9     | 43.7      | BBox   | 40.1| 61.8      | 43.7      | 24.5    | 43.9    | 51.0    |
|                   | 128  | 256| 397.8    | 48.4      | BBox   | 39.4| 61.1      | 42.6      | 24.0    | 42.9    | 50.8    |
|                   | 512  | 580.7| 55.1     | 43.6      | BBox   | 40.0| 61.9      | 43.6      | 24.5    | 43.6    | 51.3    |
|                   | 1024 | 946.4| 65.7     | 42.9      | BBox   | 39.7| 61.5      | 42.9      | 24.1    | 43.1    | 51.3    |
|                   | 256  | 423.8| 49.7     | 42.7      | BBox   | 39.5| 61.2      | 42.7      | 23.9    | 43.0    | 50.7    |
|                   | 512  | 632.4| 56.9     | 43.0      | BBox   | 39.7| 61.6      | 43.0      | 24.3    | 43.2    | 51.4    |
|                   | 1024 | 1049.6| 72.3    | 43.3      | BBox   | 40.0| 61.9      | 43.7      | 24.4    | 43.6    | 51.9    |

The number of recurrence of HGD $k = 4$.

The Effects of Removing $P^2$ and $P^6$ feature maps in our HGD-FPN when integrated in Faster RCNN with a ResNet-50 backbone on MS COCO 2017 minival

| Feature Fusion     | $P^2$ | $P^6$ | FLOPs (G) | Params (M) | Task   | AP  | AP$_{50}$ | AP$_{75}$ | AP$_{S}$ | AP$_{M}$ | AP$_{L}$ |
|-------------------|-------|-------|----------|-----------|--------|-----|-----------|-----------|---------|---------|---------|
| FPN [10]          | -     | -     | 214.8    | 41.5      | BBox   | 36.3| 58.2      | 39.1      | 21.6    | 40.1    | 46.4    |
| HGD-FPN           | $\times$ | $\times$ | 580.3    | 55.1      | BBox   | 38.5| 59.6      | 41.5      | 21.4    | 42.1    | 50.7    |
|                   | $\times$ | $\checkmark$ | 580.3    | 55.1      | BBox   | 38.5| 60.0      | 42.1      | 21.6    | 42.3    | 51.3    |
|                   | $\checkmark$ | $\times$ | 580.7    | 55.1      | BBox   | 39.4| 61.4      | 42.7      | 24.3    | 43.3    | 50.6    |
|                   | $\checkmark$ | $\checkmark$ | 580.7    | 55.1      | BBox   | 40.0| 61.9      | 43.6      | 24.5    | 43.6    | 51.4    |

The number of recurrence of HGD $k = 4$. 

recurrence number $k$ of HGD-FPN results in better performance, which demonstrates that the recurrence of the HGD-FPN module could further strengthen the multi-scale feature fusion and increase the discriminativeness of the feature pyramids. Even when $k = 1$, the performance of our HGD-FPN is still better than that of conventional FPN [10] and similar to that of CARAFE [28]. Although the method CARAFE [28] has a very smaller computational cost than our HGD-FPN with $k = 1$, the method CARAFE is only designed for context-aware feature upsampling operation which can be integrated into the FPN module in the top-down path and cannot be used for the feature fusion from the bottom-up path. Then, similar to the weakness of the classical FPN method [12], we found that stacking the CARAFE architecture does not improve the performance. The FPN variants, BiFPN [32], NAS-FPN [12] and FPG [29], both investigate recurrent or multiple repeated FPN architectures. BiFPN [32] proposes to stack the weighted bi-directional feature pyramid network. As shown in Table 11, although BiFPN enables light-weight multi-scale feature fusion, our proposed HGD-FPN significantly increases the accuracy with the same feature encoding backbone. NAS-FPN [12] stacks the searched FPN architecture and FPG [29] stacks the fully-connected FPN architecture. In Table 11, we observe that our proposed HGD-FPN ($k = 4$) has higher accuracy than NAS-FPN ($k = 7$) [12] and FPG ($k = 9$) [29] with less computational cost. In the remaining parts of the paper, we set the default $k = 4$.

Parameter Sharing of the Recurrent HGD-FPN Module. We also investigate the effect of whether to share parameters of the recurrent HGD-FPN at different stages. For the shared-parameter setting, we only initialize one set of parameters for our HGD-FPN and stack it for $k$ times. For the non-sharing parameter setting, we use $k$ different sets of parameters for HGD-FPN. In such cases, for $k = 4$, the mAP of the Faster RCNN with a ResNet 50 backbone is 39.6, which is smaller than that of the method with shared parameters. Then, we choose the parameter sharing strategy for our proposed HGD-FPN to reduce the memory cost.

Removing the $P^2$ and/or $P^6$ feature maps in our HGD-FPN method. In our proposed HGD-FPN architecture, despite the input and output have five feature maps, the holistically-guided decoder mainly generates three feature maps ($P^3$, $P^4$ and $P^5$). To investigate the contribution of the feature maps $P^2$ and $P^6$ in our proposed HGD-FPN, we conduct one ablation study in Table 10, removing the feature maps $P^2$ and $P^6$ has almost the same computational load and parameters, as the feature fusions use the addition operation in Eqs. (4), (5), (6), (7), and (8). In Table 10, we find that both feature maps $P^2$ and $P^6$ could contribute to increasing the final performance without adding obvious computational load.

4.2.2 Comparison with state-of-the-art FPN variants

To further demonstrate the effectiveness of our proposed HGD-FPN for object detection, we integrate it into two other classical object detection frameworks, RetinaNet [14] and FreeAnchor [15]. Different from Faster RCNN [13], which is a two-stage object detector, RetinaNet [14] and FreeAnchor [15] are one-stage object detection frameworks. Our proposed HGD-FPN is compared with state-of-the-art multi-scale FPN variants within such object detection frameworks.
As shown in Table 12, our proposed HGD-FPN achieves the best performance when integrated into the three object detection frameworks. Both the conventional FPN [10] and PA-FPN [11] obtain the upsampled feature maps from local neighborhoods, which lack the ability of capturing long-range context information. Although CARAFE [28] uses an adaptive feature upsampling scheme, the receptive fields of its predicted kernels are still limited. For Balanced FPN [27], it adopts the non-local module to generate balanced multi-scale features for enhancing the output feature pyramids. However, such features might not be able to capture the global context for each feature scale. Since the holistic code-words in HGD-FPN are generated by considering the holistic information and the multi-scale feature maps are generated via linearly combining them with predicted assembly coefficients, our HGD-FPN could adaptively enhance each feature scale with the global contexts and improve the final performance for object detection.

Besides object detection, we integrate our proposed HGD-FPN into MaskRCNN for instance segmentation. As shown in Table 13, our HGD-FPN also leads to the most AP gain compared to other FPN variants when integrated with Mask RCNN [16]. The good performance validates that the proposed HGD could yield the discriminative semantic feature maps and help boost the performance of instance segmentation.

HRNet[40] is one novel multi-scale feature representation learning network that can produce high-resolution representations by connecting high-to-low resolution convolutions in parallel and repeatedly conducting multi-scale fusions across parallel streams. HRNet has been proven its superiority in object detection compared with the traditional feature encoding backbone. For the same method, Faster-RCNN, the combination of ResNet-50 and our proposed HGD-FPN (40.0%) is able to achieve better accuracy than the HRNetV2p-W32 (39.5%) with comparable parameter size.

Meanwhile, we also provide the object detection results on the MS COCO 2017 test-dev dataset in Table 14. These results are reported on a single image of size 1280 × 832. As shown in Table 14, on the MS COCO 2017 test-dev dataset, compared with the classical FPN method, our proposed HGD-FPN could introduce about 4% improvement on the test-dev dataset.

5 CONCLUSION

In this paper, we propose the holistically-guided decoder for achieving discriminative deep feature representations. In fact, this novel decoder is flexible to reconstruct semantic-rich feature maps of any resolution from the holistic information in the feature pyramids of the encoder. To

### TABLE 11

The effects of integrating different FPN variants in Faster RCNN with a ResNet-50 backbone on MS COCO 2017 minival

| Feature Fusion | Task | AP | AP\_50 | AP\_75 | AP\_S | AP\_M | AP\_L |
|----------------|------|----|--------|--------|-------|-------|-------|
| FPN [10]       | BBox | 36.3| 58.2   | 39.1   | 21.6  | 40.1  | 46.4  |
| CARAFE [28]    | BBox | 37.2| 58.8   | 40.0   | 22.1  | 40.8  | 47.5  |
| BiFPN [32]     | BBox | 37.2| 58.8   | 40.0   | 22.1  | 40.8  | 47.5  |
| NAS-FPN [12]   | BBox | 39.0| 59.5   | 42.4   | 22.4  | 42.6  | 47.8  |
| FPG [29]       | BBox | 39.2| 60.8   | 42.7   | 22.7  | 41.9  | 48.4  |
| HGD-FPN 1      | BBox | 37.9| 60.7   | 40.6   | 23.8  | 42.0  | 48.1  |
| HGD-FPN 2      | BBox | 38.6| 60.9   | 41.9   | 24.0  | 42.4  | 49.4  |
| HGD-FPN 3      | BBox | 39.3| 61.5   | 42.6   | 24.2  | 42.8  | 50.6  |
| HGD-FPN 4      | BBox | 40.0| 61.9   | 43.6   | 25.0  | 43.6  | 51.4  |
| HGD-FPN 5      | BBox | 39.8| 61.4   | 42.9   | 23.4  | 43.4  | 51.9  |

### TABLE 12

Object Detection performances of different FPN variants and frameworks on COCO 2017 minival set

| Method         | Feature Fusion | Backbone | Task | AP | AP\_50 | AP\_75 | AP\_S | AP\_M | AP\_L |
|----------------|----------------|----------|------|----|--------|--------|-------|-------|-------|
| Faster R-CNN [13] | FPN [10]      | ResNet-50 | BBox | 36.3| 58.2   | 39.1   | 21.6  | 40.1  | 46.4  |
| PA-FPN [11]    | ResNet-50     | BBox     | 36.7| 58.5 | 39.7   | 21.6  | 40.4  | 47.7  |
| Balanced FPN [27] | ResNet-50     | BBox     | 37.2| 59.8 | 40.1   | 23.2  | 41.2  | 47.1  |
| CARAFE [28]    | ResNet-50     | BBox     | 38.1| 60.7 | 41.0   | 22.8  | 41.2  | 46.9  |
| HGD-FPN        | ResNet-50     | BBox     | 40.0| 61.9 | 43.6   | 24.5  | 43.6  | 51.4  |
| RetinaNet [14] | FPN [10]      | ResNet-50 | BBox | 35.3| 55.1   | 37.7   | 19.6  | 39.2  | 46.9  |
| PA-FPN [11]    | ResNet-50     | BBox     | 35.8| 55.6 | 38.3   | 20.1  | 39.9  | 47.1  |
| Balanced FPN [27] | ResNet-50     | BBox     | 36.4| 57.2 | 38.9   | 21.6  | 40.5  | 47.0  |
| CARAFE [28]    | ResNet-50     | BBox     | 36.1| 56.5 | 38.3   | 20.9  | 40.3  | 46.8  |
| HGD-FPN        | ResNet-50     | BBox     | 37.8| 58.2 | 40.7   | 22.0  | 42.0  | 49.6  |
| FreeAnchor [15] | FPN [10]      | ResNet-50 | BBox | 38.5| 57.3   | 41.2   | 21.1  | 41.8  | 51.5  |
| PA-FPN [11]    | ResNet-50     | BBox     | 38.9| 57.6 | 41.6   | 21.8  | 41.8  | 50.9  |
| Balanced FPN [27] | ResNet-50     | BBox     | 39.1| 58.5 | 41.6   | 22.0  | 42.9  | 51.4  |
| CARAFE [28]    | ResNet-50     | BBox     | 39.2| 58.4 | 42.0   | 22.5  | 42.9  | 51.5  |
| HGD-FPN        | ResNet-50     | BBox     | 40.8| 59.7 | 43.7   | 23.2  | 44.0  | 54.5  |
validate the effectiveness of its feature upsampling performance, we test it in two computer vision tasks and design two instantiations for semantic segmentation and the object detection/instance segmentation. With the proposed HGD, our EfficientFCN, with much fewer parameters and less computational cost, achieves competitive or even better performance compared with state-of-the-art dilatedFCN based methods. For object detection/instance segmentation, we propose a novel HGD-FPN to produce multi-scale feature pyramids. Both the proposed EfficientFCN and the HGD-FPN architectures demonstrate that our proposed HGD has powerful capability of encoding discriminative multi-scale feature representations.

REFERENCES

[1] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 4, pp. 834–848, Apr. 2018.
[2] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” 2017, arXiv:1706.05587.
[3] F. Yu, V. Koltun, and T. Funkhouser, “Dilated residual networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 472–480.
[4] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in Proc. Int. Conf. Med. Image Comput. Comput. Assist. Intervention, 2015, pp. 234–241.
[5] H. Zhang et al., “Context encoding for semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 7151–7160.
[6] J. He, Z. Deng, L. Zhou, Y. Wang, and Y. Qiao, “Adaptive pyramid context network for semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 548–557.
[7] H. Zhang, H. Zhang, C. Wang, and J. Xie, “Co-occurrent features in semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 7519–7528.
[8] J. Liu, J. He, Y. Qiao, J. S. Ren, and H. Li, “Learning to predict context-adaptive convolution for semantic segmentation,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 769–786.
[9] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” 2015, arXiv:1511.05617.
[10] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Biringen, “Feature pyramid networks for object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2117–2125.
[11] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 8759–8768.
[12] G. Ghiaci, T.-Y. Lin, and Q. V. Le, “NAS-FPN: Learning scalable feature pyramid architecture for object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 7036–7045.
[13] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 91–99.
[14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 2980–2988.
[15] X. Zhang, F. Wan, C. Liu, R. Ji, and Q. Ye, “Freeanchor: Learning to match anchors for visual object detection,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 147–155.
[16] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 2961–2969.
[17] J. Fu et al., “Dual attention network for scene segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 3146–3154.
[18] J. Fu et al., “Adaptive context network for scene parsing,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 6747–6756.
[19] J. He, Z. Deng, and Y. Qiao, “Dynamic multi-scale filters for semantic segmentation,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 3562–3572.
[20] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2891–2899.
[21] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, “Gated-SCNN: Gated shape CNNs for semantic segmentation,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 5229–5238.
[22] Z. Tian, T. He, C. Shen, and Y. Yan, “Decoders matter for semantic segmentation: Data-dependent decoding enables flexible feature aggregation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 3126–3135.
[23] H. Wu, J. Zhang, K. Huang, K. Liang, and Y. Yu, “Fastfcn: Rethinking dilated convolution in the backbone for semantic segmentation,” 2019, arXiv:1903.11816.
[24] R. Girshick, “Fast R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 1440–1448.

[25] B. Singh and L. S. Davis, “An analysis of scale invariance in object detection snip,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 3578–3587.

[26] B. Singh, M. Najibi, and L. S. Davis, “Sniper: Efficient multi-scale training,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 9310–9320.

[27] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, “Libra R-CNN: Towards balanced learning for object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 821–830.

[28] J. Wang, K. Chen, R. Xu, Z. Liu, C. C. Loy, and D. Lin, “Carafe: Content-aware reassembly of features,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 3007–3016.

[29] K. Chen, Y. Cao, C. C. Loy, D. Lin, and C. Feichtenhofer, “Feature pyramid grids,” 2020, arXiv:2004.03580.

[30] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit., 2018, pp. 7132–7141.

[31] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 777–787.

[32] M. Tan, R. Pang, and Q. V. Le, “EfficientDet: Scalable and efficient object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 10781–10790.

[33] R. Mottaghi et al., “The role of context for object detection and semantic segmentation in the wild,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2014, pp. 879–888.

[34] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (VOC) challenge,” Int. J. Comput. Vis., vol. 88, no. 2, pp. 303–338, 2010.

[35] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, “Scene parsing through ade20k dataset,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 633–641.

[36] Y.-Y. Lin et al., “Microsoft coco: Common objects in context,” in Proc. Comput. Vis. Pattern Recognit., 2014, pp. 740–755.

[37] O. Russakovsky et al., “Imagenet large scale visual recognition challenge,” Int. J. Comput. Vis., vol. 115, no. 3, pp. 211–252, 2015.

[38] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in Proc. COMPSTAT, 2010, pp. 177–186.

[39] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 833–851.

[40] J. Wang et al., “Deep high-resolution representation learning for visual recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 10, pp. 3349–3364, Oct. 2021.

[41] G. Lin, A. Milan, C. Shen, and I. Reid, “RefineNet: Multi-path refinement networks for high-resolution semantic segmentation,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit., 2017, pp. 1925–1934.

[42] D. Lin, Y. Ji, D. Lischinski, D. Cohen-Or, and H. Huang, “Multi-scale context intertwining for semantic segmentation,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 603–619.

[43] R. Zhang, S. Tang, Y. Zhang, J. Li, and S. Yan, “Scale-adaptive convolutions for scene parsing,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 2031–2039.

[44] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai, “Asymmetric non-local neural networks for semantic segmentation,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 593–602.

[45] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 3431–3440.

[46] S. Zheng et al., “Conditional random fields as recurrent neural networks,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 1529–1537.

[47] H. Noh, S. Hong, and B. Han, “Learning deconvolution network for semantic segmentation,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 1520–1528.

[48] Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang, “Semantic image segmnetation via deep parsing network,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 1377–1385.

[49] G. Lin, C. Shen, A. Van Den Hengel, and I. Reid, “Efficient piece-wise training of deep structured models for semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 3194–3203.

[50] Z. Wu, C. Shen, and A. v. d. Hengel, “Wider or deeper: Revisiting the resnet model for visual recognition,” 2016, arXiv:1611.10080.

[51] H. Zhao et al., “PSANet: Point-wise spatial attention network for scene parsing,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 267–283.

[52] S. Yu, J. Wang, C. Gao, G. Yu, C. Shen, and N. Sang, “Context prior for scene segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 12 416–12 425.

[53] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, “CCNet: Criss-cross attention for semantic segmentation,” in Proc. IEEE Int. Conf. Comput. Vis., early access, Jul. 3, 2020, doi: 10.1109/ TPAMI.2020.3007032.

[54] X. Li, Y. Yang, Q. Zhao, T. Shen, Z. Lin, and H. Liu, “Spatial pyramid based graph reasoning for semantic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 8950–8959.

[55] F. Zhang et al., “ACFNet: Attentional class feature network for semantic segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 6798–6807.

[56] Y. Yuan, L. Huang, J. Guo, C. Zhang, X. Chen, and J. Wang, “OCNet: Object context network for scene parsing,” 2018, arXiv:1809.00916.

[57] X. Li et al., “Semantic flow for fast and accurate scene parsing,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 775–793.

[58] Y. Yuan et al., “The cityscapes dataset for semantic urban scene understanding,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 3213–3223.

[59] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollar, and K. He, “Detectron,” 2018. [Online]. Available: https://github.com/facebookresearch/detectron

[60] K. Chen et al., “MMDetector: Open mmlab detection toolbox and benchmark,” 2019, arXiv:1906.07155.

Jianbo Liu received the BS degree from Lanzhou University in 2012 and the MS degree from the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, in 2015. He is currently working toward the PhD degree with the Department of Electronic Engineering, Chinese University of Hong Kong. His research interests include computer vision, deep learning, and scene understanding.

Junjun He received the bachelor’s degree from Sichuan University, Chengdu, China, in 2015. He is currently working toward the PhD degree with the School of Biomedical Engineering, Shanghai Jiao Tong University. His research interests include semantic segmentation, object detection, video classification, and medical image computing.

Yuanjie Zheng received the PhD degree from Shanghai Jiao Tong University, China, in 2006. He is currently a full professor with the School of Information Sciences and Engineering, Shandong Normal University of China and also the dean. He was a senior research investigator with the Perelman School of Medicine, University of Pennsylvania, USA. His research interests include computer vision, artificial intelligence, medical image analysis, and translational medicine.
Shuai Yi received the BEng degree in electronic engineering from Tsinghua University, in 2012 and the PhD degree in electronic engineering from the Chinese University of Hong Kong, in 2016. His research interests include computer vision and machine learning.

Xiaogang Wang received the BS degree from the University of Science and Technology of China in 2001, the MS degree from The Chinese University of Hong Kong in 2003, and the PhD degree from the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, in 2009. He is currently an associate professor with the Department of Electronic Engineering, Chinese University of Hong Kong. His research interests include computer vision and machine learning.

Hongsheng Li received the bachelor’s degree in automation from the East China University of Science and Technology in 2006, and the master’s and doctorate degrees in computer science from Lehigh University, Pennsylvania, in 2010 and 2012, respectively. He is currently an assistant professor with the Department of Electronic Engineering, Chinese University of Hong Kong. His research interests include computer vision, medical image analysis, and machine learning.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.