ExFuse: Enhancing Feature Fusion for Semantic Segmentation

Zhenli Zhang[1](✉), Xiangyu Zhang[2], Chao Peng[2], Xiangyang Xue[1], and Jian Sun[2]

[1] Fudan University, Shanghai, China
{zhenlizhang14,xyxue}@fudan.edu.cn
[2] Megvii Inc., Beijing, China
{zhangxiangyu,pengchao,sunjian}@megvii.com

Abstract. Modern semantic segmentation frameworks usually combine low-level and high-level features from pre-trained backbone convolutional models to boost performance. In this paper, we first point out that a simple fusion of low-level and high-level features could be less effective because of the gap in semantic levels and spatial resolution. We find that introducing semantic information into low-level features and high-resolution details into high-level features is more effective for the later fusion. Based on this observation, we propose a new framework, named ExFuse, to bridge the gap between low-level and high-level features thus significantly improve the segmentation quality by 4.0% in total. Furthermore, we evaluate our approach on the challenging PASCAL VOC 2012 segmentation benchmark and achieve 87.9% mean IoU, which outperforms the previous state-of-the-art results.

Keywords: Semantic segmentation · Convolutional neural networks

1 Introduction

Most state-of-the-art semantic segmentation frameworks [2–6,12,22,26,28,35,38,40] follow the design of Fully Convolutional Network (FCN) [25]. FCN has a typical encoder-decoder structure – semantic information is firstly embedded into the feature maps via encoder then the decoder takes responsibility for generating segmentation results. Usually the encoder is the pre-trained convolutional model to extract image features and the decoder contains multiple upsampling components to recover resolution. Although the top-most feature maps of the encoder could be highly semantic, its ability to reconstruct precise details in segmentation maps is limited due to insufficient resolution, which is very common in modern backbone models such as [15,16,20,31,33,37]. To address this, an "U-Net" architecture is proposed [28] and adopted in many recent work [2,12,22,25,26,28]. The core idea of *U-Net* is to gradually fuse high-level low-resolution features from top layers with low-level but high-resolution features from bottom layers, which

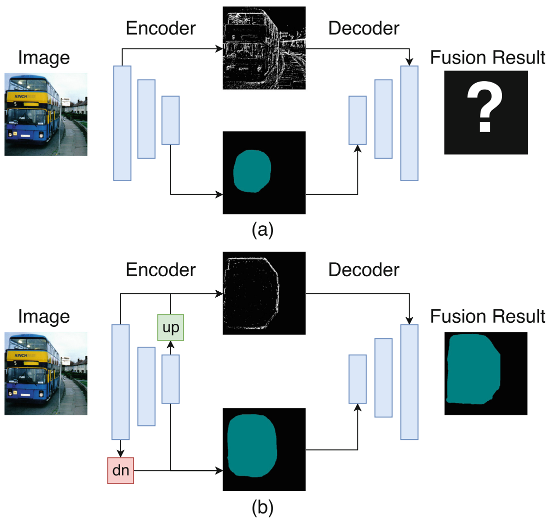

Fig. 1. Fusion of low-level and high-level features. (a) “Pure” low-level high-resolution and “pure” high-level low-resolution features are difficult to be fused because of the significant semantic and resolution gaps. (b) Introducing semantic information into low-level features or spatial information into high-level features benefits the feature fusion. “dn” and“up” blocks represent *abstract* up/down-sampling feature embedding.

is expected to be helpful for the decoder to generate high-resolution semantic results.

Though the great success of U-Net, the working mechanism is still unknown and worth further investigating. Low-level and high-level features are complementary by nature, where low-level features are rich in spatial details but lack semantic information and vice versa. Consider the extreme case that “pure” low-level features only encode low-level concepts such as points, lines or edges. Intuitively, the fusion of high-level features with such “pure” low-level features helps little, because low-level features are too noisy to provide sufficient high-resolution semantic guidance. In contrast, if low-level features include more semantic information, for example, encode relatively clearer semantic boundaries, then the fusion becomes easy – fine segmentation results could be obtained by aligning high-level feature maps to the boundary. Similarly, “pure” high-level features with little spatial information cannot take full advantage of low-level features; however, with additional high-resolution features embedded, high-level features may have chance to refine itself by aligning to the nearest low-level boundary. Figure 1 illustrates the above concepts. Empirically, the semantic and resolution overlap between low-level and high-level features plays an important role in the effectiveness of feature fusion. In other words, feature fusion could be enhanced by introducing more semantic concepts into low-level features or by embedding more spatial information into high-level features.

Motivated by the above observation, we propose to boost the feature fusion by bridging the semantic and resolution gap between low-level and high-level feature maps. We propose a framework named *ExFuse*, which addresses the gap from the following two aspects: (1) to introduce more semantic information into low-level features, we suggest three solutions – *layer rearrangement*, *semantic supervision* and *semantic embedding branch*; (2) to embed more spatial information into high-level features, we propose two novel methods: *explicit channel resolution embedding* and *densely adjacent prediction*. Significant improvements are obtained by either approach and a total increase of 4% is obtained by the combination. Furthermore, we evaluate our method on the challenging PASCAL VOC 2012 [10] semantic segmentation task. In the test dataset, we achieve the score of 87.9% mean IoU, surpassing the previous state-of-the-art methods.

Our contributions can be summerized as follows:

– We suggest a new perspective to boost semantic segmentation performance, i.e. bridging the semantic and resolution gap between low-level and high-level features by more effective feature fusion.
– We propose a novel framework named ExFuse, which introduces more semantic information into low-level features and more spatial high-resolution information into high-level features. Significant improvements are obtained from the enhanced feature fusion.
– Our fully-equipped model achieves the new state-of-the-art result on the test set of PASCAL VOC 2012 segmentation benchmark.

2 Related Work

Feature Fusion in Semantic Segmentation. Feature fusion is frequently employed in semantic segmentation for different purposes and concepts. A lot of methods fuse low-level but high-resolution features and high-level low-resolution features together [2,12,22,25,26,28]. Besides, *ASPP* module is proposed in DeepLab [4–6] to fuse multi-scale features to tackle objects of different size. Pyramid pooling module in PSPNet [40] serves the same purpose through different implementation. BoxSup [8] empirically fuses feature maps of bounding boxes and segmentation maps to further enhance segmentation.

Deeply Supervised Learning. To the best of our knowledge, deeply supervised training is initially proposed in [21], which aims to ease the training process of very deep neural networks since depth is the key limitation for training modern neural networks until batch normalization [18] and residual networks [15] are proposed. Extra losses are utilized in GoogleNet [33] for the same purpose. Recently, PSPNet [40] also employs this method to ease the optimization when training deeper networks.

Upsampling. There are mainly three approaches to upsample a feature map. The first one is bilinear interpolation, which is widely used in [4–6,40]. The second

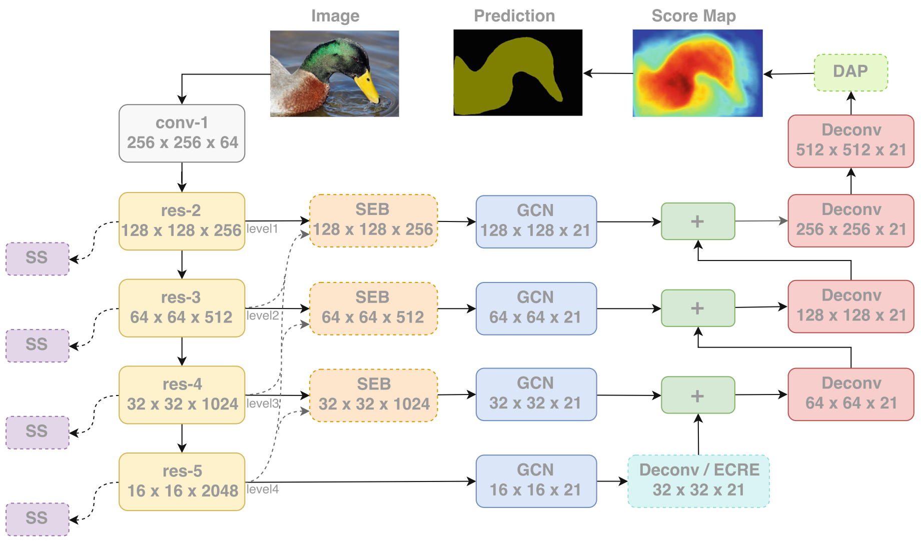

Fig. 2. Overall architecture of our approach. Components with solid boxes belong to the backbone *GCN* framework [26], while others with dashed lines are proposed in this work. Similar to [26], *Boundary Refinement* blocks are actually used but omitted in the figure. Numbers ($H \times W \times C$) in blocks specify the output dimension of each component. **SS** – semantic supervision. **ECRE** – explicit channel resolution embedding. **SEB** – semantic embedding branch. **DAP** – densely adjacent prediction. (Color figure online)

method is deconvolution, which is initially proposed in FCN [25] and utilized in later work such as [2,3,22,26,28]. The third one is called "sub-pixel convolution", which derives from [1,30] in super resolution task and is widely broadcast to other tasks such as semantic segmentation. For instance, [35] employs it to replace the traditional deconvolution operation.

3 Approach

In this work we mainly focus on the feature fusion problem in "U-Net" segmentation frameworks [2,12,22,25,26,28]. In general, *U-Net* have an encoder-decoder structure as shown in Fig. 1. Usually the encoder part is based on a convolutional model pretrained on large-scale classification dataset (e.g. ImageNet [9]), which generates low-level but high-resolution features from the bottom layers and high-level low-resolution features from the top layers. Then the decoder part mixes up the features to predict segmentation results. A common way of feature fusion [2,12,14,22,26–28] is to formulate as a residual form:

$$\mathbf{y}_l = Upsample(\mathbf{y}_{l+1}) + \mathcal{F}(\mathbf{x}_l) \tag{1}$$

where $\mathbf{y}_l$ is the fused feature at l-th level; $\mathbf{x}_l$ stands for the l-th feature generated by the encoder. Features with larger l have higher semantic level but lower spatial resolution and vice versa (see Fig. 2).

In Sect. 1 we argue that feature fusion could become less effective if there is a large semantic or resolution gap between low-level and high-level features. To study and verify the impact, we choose one of the start-of-the-art “U-Net” frameworks – *Global Convolutional Network (GCN)* [26] – as our backbone segmentation architecture (see Fig. 2 for details). In GCN, 4 different semantic levels of feature maps are extracted from the encoder network, whose spatial resolutions, given the 512×512 input, are $\{128, 64, 32, 16\}$ respectively. To examine the effectiveness of feature fusion, we select several subsets of feature levels and use them to retrain the whole system. Results are shown in Table 1. It is clear that even though the segmentation quality increases with the fusion of more feature levels, the performance tends to saturate quickly. Especially, the lowest two feature levels (1 and 2) only contribute marginal improvements (0.24% for ResNet 50 and 0.05% for ResNeXt 101), which implies the fusion of low-level and high-level features is rather ineffective in this framework.

In the following subsections we will introduce our solutions to bridge the gap between low-level and high-level features – embedding more semantic information into low-level features and more spatial resolution clues into high-level features. First of all, we introduce our baseline settings:

Table 1. *GCN* [26] segmentation results using given feature levels. Performances are evaluated by standard mean IoU(%) on PASCAL VOC 2012 validation set. Lower feature level involves less semantic but higher-resolution features and vice versa (see Fig. 2). The feature extractor is based on pretrained ResNet50 [15] and ResNeXt101 [37] model. Performance is evaluated in mIoU.

Feature levels	ResNet 50 (%)	ResNeXt 101 (%)
$\{4\}$	70.04	73.79
$\{3, 4\}$	72.17	75.97
$\{2, 3, 4\}$	72.28	75.98
$\{1, 2, 3, 4\}$	72.41	76.02

Baseline Settings. The overall semantic segmentation framework follows the fully-equipped *GCN* [26] architecture, as shown in Fig. 2. For the backbone encoder network, we use ResNeXt 101 [37] model pretrained on ImageNet by default[1] unless otherwise mentioned. We use two public-available semantic segmentation benchmarks – *PASCAL VOC 2012* [10] and *Semantic Boundaries Dataset* [13] – for training and evaluate performances on PASCAL VOC 2012 validation set, which is consistent with many previous work [2–6,12,22,25–27,35,38,40]. The performance is measured by standard mean intersection-over-union (mean IoU). Other training and test details or hyper-parameters are

[1] Though ResNeXt 101 performs much better than ResNet 101 [15] on ImageNet classification task (21.2% vs. 23.6% in top-1 error), we find there are no significant differences on the semantic segmentation results (both are 76.0% mIoU).

exactly the same as [26]. Our reproduced GCN baseline score is 76.0%, shown in Table 3 (#1).

3.1 Introducing More Semantic Information into Low-Level Features

Our solutions are inspired by the fact: for convolutional neural networks, feature maps close to semantic supervisions (e.g. classification loss) tend to encode more semantic information, which has been confirmed by some visualization work [39]. We propose three methods as follows:

Layer Rearrangement. In our framework, features are extracted from the tail of each stage in the encoder part (res-2 to res-5 in Fig. 2). To make low-level features (res-2 or res-3) 'closer' to the supervisions, one straight-forward approach is to arrange more layers in the early stages rather than the latter. For example, ResNeXt 101 [37] model has $\{3, 4, 23, 3\}$ building blocks for Stage 2–5 respectively; we rearrange the assignment into $\{8, 8, 9, 8\}$ and adjust the number of channels to ensure the same overall computational complexity. Experiment shows that even though the ImageNet classification score of the newly designed model is almost unchanged, its segmentation performance increases by 0.8% (Table 3, compare #2 with #3), which implies the quality of low-level feature might be improved.

Semantic Supervision. We come up with another way to improve low-level features, named *Semantic Supervision (SS)*, by assigning auxiliary supervisions directly to the early stages of the encoder network (see Fig. 2). To generate semantic outputs in the auxiliary branches, low-level features are forced to encode more semantic concepts, which is expected to be helpful for later feature fusion. Such methodology is inspired by *Deeply Supervised Learning* used in some old classification networks [21,33] to ease the training of deep networks. However, more sophisticated classification models [15–17,32,34,37] suggest end-to-end training without auxiliary losses, which is proved to have no convergence issue even for models over 100 layers. Our experiment also shows that for ResNet or ResNeXt models deeply supervised training is useless or even harms the classification accuracy (see Table 2). Therefore, our *Semantic Supervision* approach mainly focuses on improving the quality of low-level features, rather than boosting the backbone model itself.

Figure 3 shows the detailed structure of our Semantic Supervision block. When pretraining the backbone encoder network, the components are attached to the tail of each stage as auxiliary supervisions (see Fig. 2). The overall classification loss equals to a weighted summation of all auxiliary branches. Then after pretraining, we remove these branches and use the remaining part for fine tuning. Experiment shows the method boosts the segmentation result by 1.1%. Moreover, we find that if features are extracted from the second convolutional layer in the auxiliary module for fine tuning (Fig. 3), more improvement (1.5%)

Table 2. Effects of *Semantic Supervision (SS)*. Classification scores are evaluated on ImageNet 2012 validation set.

Model	Cls err (top-1, %)	Seg mIoU (%)
Res50	24.15	72.4
SS Res50	24.77	73.5

is obtained (see Table 3, compare #1 with #2), which supports our intuition that feature maps closer to the supervision tend to encode more semantic information.

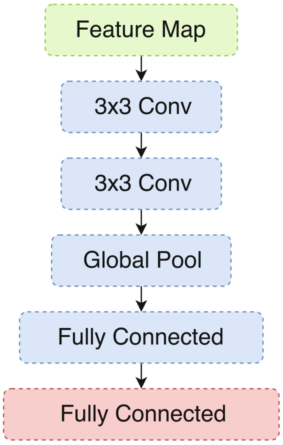

Fig. 3. Details of *Semantic Supervision (SS)* component in our pipeline.

It is worth noting that the recent semantic segmentation work *PSPNet* [40] also employs deeply supervised learning and reports the improvements. Different from ours, the architecture of [40] do not extract feature maps supervised by the auxiliary explicitly; and their main purpose is to ease the optimization during training. However, in our framework we find the improvements may result from different reasons. For instance, we choose a relatively shallower network ResNet 50 [15] and pretrain with or without semantic supervision. From Table 2, we find the auxiliary losses do not improve the classification score, which implies ResNet 50 is unlikely to suffer from optimization difficulty. However, it still boosts the segmentation result by 1.1%, which is comparable to the deeper case of ResNeXt 101 (1.0%). We believe the enhancement in our framework mainly results from more "semantic" low-level features.

Semantic Embedding Branch. As mentioned above, many "U-Net" structures involve low-level feature as the residue to the upsampled high-level feature. In Eq. 1 the residual term $\mathcal{F}(\mathbf{x}_l)$ is a function of low-level but high-resolution feature, which is used to fill the spatial details. However, if the low-level feature contains little semantic information, it is insufficient to recover the semantic resolution. To address the drawback, we generalize the fusion as follows:

$$\mathbf{y}_l = Upsample\left(\mathbf{y}_{l+1}\right) + \mathcal{F}(\mathbf{x}_l, \mathbf{x}_{l+1}, \ldots, \mathbf{x}_L) \tag{2}$$

where L is the number of feature levels. Our insight is to involve more semantic information from high-level features to guide the resolution fusion.

The detailed design of function $\mathcal{F}(\cdot)$ is illustrated in Fig. 4, named *Semantic Embedding Branch, (SEB)*. We use the component for features of Level 1-3 (see Fig. 2). In our experiment SEB improves the performance by 0.7% (Table 3, compare #3 with #5).

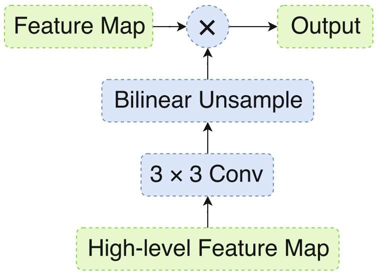

Fig. 4. Design of the *Semantic Embedding Branch* in Fig. 2. The "×" sign means element-wise multiplication. If there are more than one groups of high-level features, the component outputs the production of each feature map after upsampling.

3.2 Embedding More Spatial Resolution into High-Level Features

For most backbone feature extractor networks, high-level features have very limited spatial resolution. For example, the spatial size of top-most feature map in ResNet or ResNeXt is 7 × 7 for 224 × 224 input size. To encode more spatial details, a widely used approach is *dilated strategy* [4–6,35,38,40], which is able to enlarge feature resolution without retraining the backbone network. However, since high-level feature maps involve a lot of channels, larger spatial size significantly increases the computational cost. So in this work we mainly consider another direction – we do not try to increase the "physical" resolution of the feature maps; instead, **we expect more resolution information encoded within channels**. We propose the following two methods:

Explicit Channel Resolution Embedding. In our overall framework, segmentation loss is only connected to the output of decoder network (see Fig. 2), which is considered to have less impact on the spatial information of high-level features by intuition. One straight-forward solution is to borrow the idea of *Semantic Supervision* (Sect. 3.1) – we could add an auxiliary supervision branch

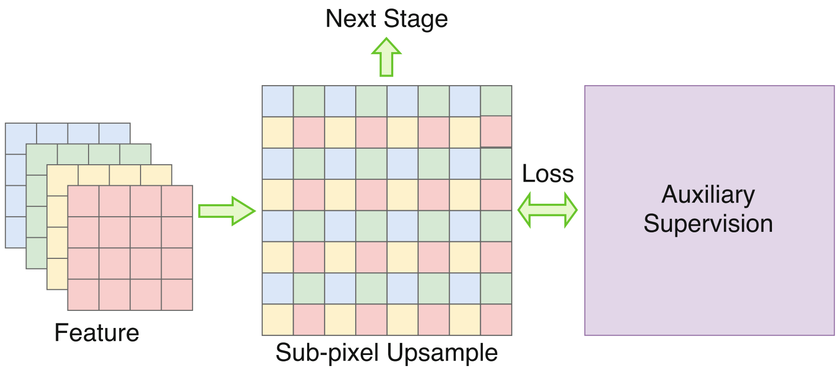

Fig. 5. Illustration of the design of *Explicit Channel Resolution Embedding (ECRE)* module in Fig. 2.

to the high-level feature map, upsample and force it to learn fine segmentation map. Following the insight, firstly we try adding an extra segmentation loss to the first deconvolution module (the light-blue component in Fig. 2), however, no improvements are obtained (Table 4, #2).

Table 3. Ablation experiments of the methods in Sect. 3. Performances are evaluated by standard mean IoU(%) on PASCAL VOC 2012 validation set. The baseline model is [26] (our impl.) **SS** – semantic supervision. **LR** – layer rearrangement. **ECRE** – explicit channel resolution embedding. **SEB** – semantic embedding branch. **DAP** – densely adjacent prediction.

Index	Baseline	SS	LR	ECRE	SEB	DAP	mIoU (%)
1	✓						76.0
2	✓	✓					77.5
3	✓	✓	✓				78.3
4	✓	✓	✓	✓			78.8
5	✓	✓	✓		✓		79.0
6	✓	✓	✓		✓	✓	79.6
7	✓	✓	✓	✓	✓	✓	**80.0**

Why does the auxiliary loss fail to work? Note that the purpose of the supervision is to embed high resolution information "explicitly" into feature map channels. However, since deconvolution layer includes weights, the embedding becomes implicit. To overcome this issue, we adopt a parameter-free upsampling method – *Sub-pixel Upsample* [1,30] – to replace the original deconvolution. Since sub-pixel upsample enlarge the feature map just by reshaping the spatial and channel dimensions, the auxiliary supervision is able to *explicitly* impact the features. Details of the component are shown in Fig. 5. Experiment shows that it enhances the performance by 0.5% (see Tables 4 and 3).

Table 4. Ablation study on the design of *Explicit Channel Resolution Embedding, (ECRE)*. The baseline model is in Table 3 (#3)

Index	Method	mIoU (%)
1	Baseline	78.3
2	Deconv + Supervised	78.2
3	Sub-pixel upsample only	77.6
4	ECRE (Fig. 5)	**78.8**

Moreover, to demonstrate that the improvement is brought by explicit resolution embedding rather than sub-pixel upsampling itself, we also try to replace the deconvolution layer only without auxiliary supervision. Table 4 (#3) shows the result, which is even worse than the baseline.

Densely Adjacent Prediction. In the decoder upstream of the original architecture (Fig. 2), feature point at the spatial location (i, j) mainly takes responsibility for the semantic information at the same place. To encode as much spatial information into channels, we propose a novel mechanism named *Densely Adjacent Prediction (DAP)*, which allows to predict results at the adjacent position, e.g. $(i-1, j+1)$. Then to get the final segmentation map, result at the position (i, j) can be generated by averaging the associated scores. Formally, given the window size $k \times k$, we divide the feature channels into $k \times k$ groups, then DAP works as follows:

$$\mathbf{r}_{i,j} = \frac{1}{k \times k} \sum_{0 \le l,m < k} \mathbf{x}_{i+l-\lfloor k/2 \rfloor, j+m-\lfloor k/2 \rfloor}^{(l \times k+m)} \tag{3}$$

where $\mathbf{r}_{i,j}$ denotes the result at the position (i, j) and $\mathbf{x}_{i,j}^{(c)}$ stands for the features at the position (i, j) belonging to channel group c. In Fig. 6 we illustrate the concept of DAP.

We use DAP on the output of our decoder (see Fig. 2). In our experiment we set $k = 3$. Note that DAP requires the number of feature channels increased by $k \times k$ times, so we increase the output channels of each deconvolution block to 189 $(21 \times 3 \times 3)$. For fair comparison, we also evaluate the baseline model with the same number of channels. Results are shown in Table 5. It is clear that DAP improves the performance by 0.6% while the counterpart model without DAP only obtains marginal gain, which implies DAP may be helpful for feature maps to embed more spatial information.

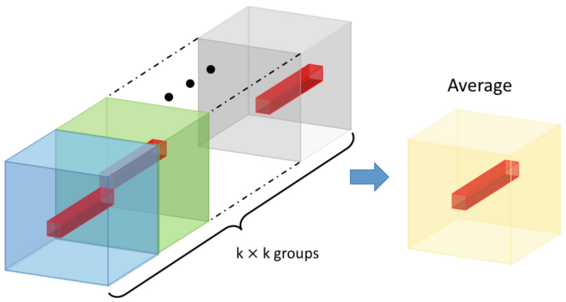

Fig. 6. Illustration of *Densely Adjacent Prediction (DAP)* component in Fig. 2.

Table 5. Ablation study on the effect of *Densely Adjacent Prediction (DAP)*. The baseline model is in Table 3 (#5)

Index	Method	mIoU (%)
1	Baseline	79.0
2	Baseline (more channels)	79.1
3	DAP (Fig. 6)	**79.6**

3.3 Discussions

Is Feature Fusion Enhanced? At the beginning of Sect. 3 we demonstrate that feature fusion in our baseline architecture (*GCN* [26]) is ineffective. Only marginal improvements are obtained by fusing low-level features (Level 1 and 2), as shown in Table 1. We attribute the issue to the semantic and resolution gap between low-level and high-level features. In Sects. 3.1 and 3.2, we propose a series of solutions to introduce more semantic information into low-level features and more spatial details into high-level features.

Despite the improved performance, a question raises: is feature fusion in the framework *really* improved? To justify this, similar to Table 1 we compare several subsets of different feature levels and use them to train original baseline (GCN) and our proposed model (*ExFuse*) respectively. For the ExFuse model, all the 5 approaches in Sects. 3.1 and 3.2 are used. Table 6 shows the results. We find that combined with low-level feature maps (Level 1 and 2) the proposed ExFuse still achieves considerable performance gain (~1.3%), while the baseline model cannot benefit from them. The comparison implies our insights and methodology enhance the feature fusion indeed.

Table 6 also shows that the proposed model is much better than the baseline in the case that only top-most feature maps (Level 4) are used, which implies the superior high-level feature quality to the original model. Our further study shows that methods in Sect. 3.2 contribute most of the improvement. Empirically we conclude that boosting high-level features not only benefits feature fusion, but also contributes directly to the segmentation performance.

Table 6. Comparison of original GCN [26] and ExFuse on segmentation results using given feature levels. The backbone feature extractor networks are both ResNeXt 101.

Feature levels	Original GCN [26] (%)	ExFuse (%)
$\{4\}$	73.79	77.29
$\{3, 4\}$	75.97	78.69
$\{2, 3, 4\}$	75.98	79.11
$\{1, 2, 3, 4\}$	76.02	80.04

Could the Perspective and Techniques Generalize to Other Computer Vision Tasks? Since U-Net structure is widely applied to other vision tasks such as low-level vision [29] and detection [23], a question raises naturally: could the proposed perspective and techniques generalize to other tasks? We carefully conducted ablation experiments and observe positive results. We leave detailed discussion for future work.

4 PASCAL VOC 2012 Experiment

In the last section we introduce our methodology and evaluate their effectiveness via ablation experiments. In this section we investigate the fully-equipped system and report benchmark results on PASCAL VOC 2012 test set.

To further improve the feature quality, we use deeper ResNeXt 131 as our backbone feature extractor, in which *Squeeze-and-excitation* modules [17] are also involved. The number of building blocks for Stage 2-5 is $\{8, 8, 19, 8\}$ respectively, which follows the idea of Sect. 3.1. With ResNeXt 131, we get 0.8% performance gain and achieve 80.8% mIoU when training with 10582 images from *PASCAL VOC 2012* [10] and *Semantic Boundaries Dataset (SBD)* [13], which is 2.3% better than *DeepLabv3* [6] at the same settings.

Table 7. Strategies and results on PASCAL VOC 2012 validation set

Index	ResNeXt 131	COCO	Flip	mIoU (%)
1	(ResNeXt 101)			80.0
2	✓			80.8
3	✓	✓		85.4
4	✓	✓	✓	**85.8**

Following the same procedure as [2, 4–6, 12, 22, 26, 35, 40], we employ Microsoft COCO dataset [24] to pretrain our model. COCO has 80 classes and we only retain images including the same 20 classes in PASCAL VOC 2012 and all other classes are regarded as background. Training process has 3 stages. In stage-1, we mix up all images in COCO, SBD and standard PASCAL VOC 2012. In stage-2, we utilize SBD and PASCAL VOC 2012 training images. Finally for stage-3, we only employ standard PASCAL VOC 2012 training set. We keep image crop size unchanged during the whole training procedure and all other settings are exactly the same as [26]. COCO pretraining brings about another 4.6% increase in performance, as shown in Table 7 (#2 and #3).

We further average the score map of an image with its horizontal flipped version and eventually get a 85.8% mIoU on PASCAL VOC 2012 validation set, which is 2.3% better than DeepLabv3+ [7] (Table 7 #4).

Resembling [6], we then freeze the batch normalization parameters and fine tune our model on official PASCAL VOC 2012 *trainval* set. In particular, we duplicate the images that contain hard classes (namely bicycle, chair, dining table, potted plant and sofa). Finally, our ExFuse framework achieves **87.9%** mIoU on PASCAL VOC 2012 test set without any DenseCRF [19] post-processing, which surpasses previous state-of-the-art results, as shown in Table 8. For fair comparison, we also evaluate our model using a standard ResNet101 and it achieves 86.2% mIoU, which is better than DeepLabv3 at the same setting.

Table 8. Performance on PASCAL VOC 2012 test set

Method	mIOU
Tusimple [35]	83.1
Large_Kernel_Matters [26]	83.6
Multipath_RefineNet [22]	84.2
ResNet_38_MS_COCO [36]	84.9
PSPNet [40]	85.4
DeepLabv3 [6]	85.7
SDN [11]	86.6
DeepLabv3+ (Xception) [7]	87.8
ExFuse_ResNet101 (ours)	**86.2**
ExFuse_ResNeXt131 (ours)	**87.9**

Figure 7 visualizes some representative results of the GCN [26] baseline and our proposed ExFuse framework. It is clear that the visualization quality of our method is much better than the baseline. For example, the boundary in ExFuse is more precise than GCN.

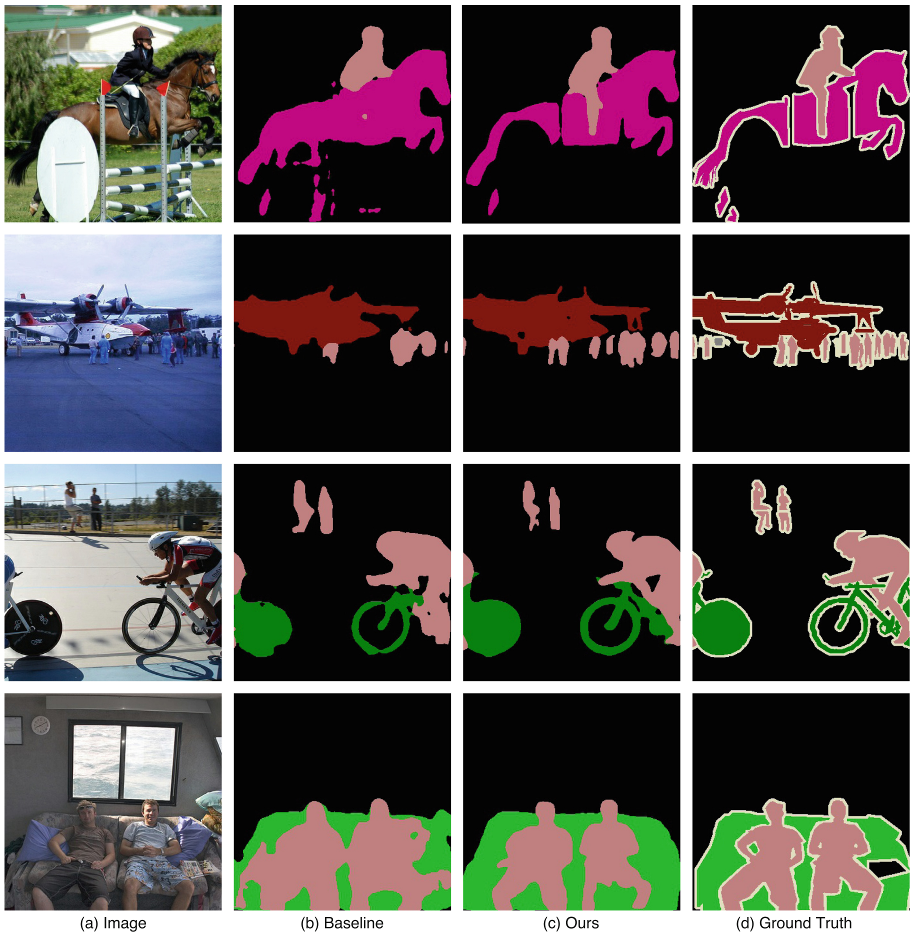

Fig. 7. Examples of semantic segmentation results on PASCAL VOC 2012 validation set. (b) is our GCN [26] baseline which achieves 81.0% mIoU on val set. (c) is our method which achieves 85.4% on val set, as shown in Table 7 #3.

5 Conclusions

In this work, we first point out the ineffective feature fusion problem in current *U-Net* structure. Then, we propose our *ExFuse* framework to tackle this problem via bridging the gap between high-level low-resolution and low-level high-resolution features. Eventually, better feature fusion is demonstrated by the performance boost when fusing with original low-level features and the overall segmentation performance is improved by a large margin. Our *ExFuse* framework also achieves new state-of-the-art performance on PASCAL VOC 2012 benchmark.

References

1. Aitken, A., Ledig, C., Theis, L., Caballero, J., Wang, Z., Shi, W.: Checkerboard artifact free sub-pixel convolution: a note on sub-pixel convolution, resize convolution and convolution resize (2017)
2. Amirul Islam, M., Rochan, M., Bruce, N.D.B., Wang, Y.: Gated feedback refinement network for dense image labeling. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017
3. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: a deep convolutional encoder-decoder architecture for scene segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **PP**(99), 1 (2017)
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. Comput. Sci. (4), 357–361 (2014)
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. **PP**(99), 1 (2016)
6. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation (2017)
7. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation (2018)
8. Dai, J., He, K., Sun, J.: BoxSup: exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In: IEEE International Conference on Computer Vision, pp. 1635–1643 (2015)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Li, F.F.: ImageNet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 248–255 (2009)
10. Everingham, M., Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
11. Fu, J., Liu, J., Wang, Y., Lu, H.: Stacked deconvolutional network for semantic segmentation (2017)
12. Ghiasi, G., Fowlkes, C.C.: Laplacian pyramid reconstruction and refinement for semantic segmentation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 519–534. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_32
13. Hariharan, B., Arbelaez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: International Conference on Computer Vision, pp. 991–998 (2011)
14. Hariharan, B., Arbelaez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization, pp. 447–456 (2014)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Computer Vision and Pattern Recognition, pp. 770–778 (2016)
16. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 630–645. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_38
17. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. arXiv preprint arXiv:1709.01507 (2017)
18. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift, pp. 448–456 (2015)

19. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected CRFs with Gaussian edge potentials. In: Advances in Neural Information Processing Systems, pp. 109–117 (2011)
20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems, pp. 1097–1105 (2012)
21. Lee, C.Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z.: Deeply-supervised nets. Eprint Arxiv, pp. 562–570 (2014)
22. Lin, G., Milan, A., Shen, C., Reid, I.: RefineNet: multi-path refinement networks for high-resolution semantic segmentation (2016)
23. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection (2016)
24. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
25. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
26. Peng, C., Zhang, X., Yu, G., Luo, G., Sun, J.: Large kernel matters - improve semantic segmentation by global convolutional network (2017)
27. Pohlen, T., Hermans, A., Mathias, M., Leibe, B.: Full-resolution residual networks for semantic segmentation in street scenes (2016)
28. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
29. Shen, X., Chen, Y.C., Tao, X., Jia, J.: Convolutional neural pyramid for image processing (2017)
30. Shi, W., et al.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, pp. 1874–1883 (2016)
31. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Computer Science (2014)
32. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-ResNet and the impact of residual connections on learning. In: AAAI, pp. 4278–4284 (2017)
33. Szegedy, C., et al.: Going deeper with convolutions. In: Computer Vision and Pattern Recognition, pp. 1–9 (2015)
34. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
35. Wang, P., et al.: Understanding convolution for semantic segmentation (2017)
36. Wu, Z., Shen, C., Hengel, A.V.D.: Wider or deeper: revisiting the ResNet model for visual recognition (2016)
37. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks (2016)
38. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions (2015)
39. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_53
40. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network (2016)