HLNet: Modeling High and Low Frequencies for Scene Parsing

Kaiqiang Xu[1,2], Zhulin An[1,*], Hui Zhu[1], Xiaolong Hu[1], Yongjun Xu[1]
[1]*Institute of Computing Technology, Chinese Academy of Sciences, China*
[2]*University of Chinese Academy of Sciences, China*
{xukaiqiang, anzhulin, zhuhui, huxiaolong18g, xyj}@ict.ac.cn

***Abstract*—In this paper we propose to model high and low frequencies of segmentation map, based on the observation that the map can be seen as a mixture of different frequencies. Based on the sparsity of high frequencies and local similarity of low frequencies, we design special building blocks and further a novel High and Low frequency Network (HLNet) with two branches based on FCN to predict high and low frequencies of the segmentation map, respectively. Specifically, we design a high frequency branch with a small kernel size and high-resolution features to predict a sparse high frequency component. Meanwhile, a low frequency branch with similarity computing and low-resolution features is employed to predict a low frequency component. On top of two branches, we combine two different frequency components to generate final result for scene parsing. We empirically demonstrate that the designed model achieves superior performance 44.07% on ADE20K, and 80.14% mIoU on Cityscapes datasets.**

***Index Terms*—Scene Parsing, semantic segmentation, image frequency, deep convolution neural networks**

I. Introduction

Scene parsing, based on semantic segmentation, is a problem of assigning a predefined label to every pixel for a image, and finally, is to get a segmentation map. scene parsing helps human understand the scene according to the segmentation map which reflects the label, location, as well as shape of each element in the image.

With the developments of neural network, it has achieved remarkable results based on Fully Convolutional Networks (FCNs) [1]. However, there are two main limitations in FCN frameworks. First there exists some downsampling operations such as pooling or convolution striding in the frameworks, which results in generating low resolution feature representations and losing an amount of spatial details. Second, due to subject to the limited valid receptive field [2], [3] in neural networks, some pixels lack of enough contexts to discriminate and their classification are ambiguous. Eventually, these limitations result in that segmentation results predicted by FCNs usually exist some problems such as rough elements boundary and misclassification of big objects or stuff.

Various methods have been proposed to overcome above limitations. Some methods [4]–[6] allow several consecutive downsampling operation to enlarge receptive field, and then to remedy the loss of spatial details, middle-layer features are utilized. Some methods [7], [8] reduce the times of downsampling and employ dilated convolution to enlarge receptive field while maintaining the resolution. However, these methods apply unified processing to each pixel and overlook the different processing demand of pixels in different locations. The pixels on objects boundary should strive to retain their spatial information and preserve detailed structure [9] in the processing because of difficulty of recovering, while other pixels having common feature with local adjacent areas need not. Besides, these methods lack of being explainable.

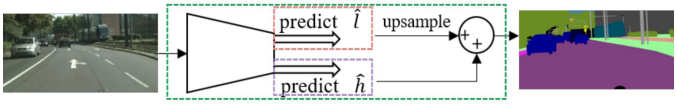

Fig. 1: Reconstruction of the segmentation map by combining predicted high frequencies $\hat{h}$ and low frequencies $\hat{l}$.

In traditional image processing, a natural image can be decomposed into a low frequency component describing its slowly changing structure and a high frequency component describing its rapidly changing structure [10]–[12], and vice versa, a natural image can be reconstructed from its two different frequency components. Similarly, we argue that the segmentation map satisfies the decomposition and reconstruction, so it motivates us to propose novel approach for modeling high frequencies and low frequencies of the segmentation map respectively and combining them as shown in Fig.1. Thus, the pixels located at different frequency components would go through different processing. To accommodate the different frequencies prediction, we design high-frequency branch and low-frequency branch based on the unique characteristics of these frequencies. As the separate prediction of high and low frequencies of the segmentation map, our method enlarges the receptive field when predicting low frequencies and keeps high resolution when processing high frequencies, and thus can alleviate above limitations and improve recognition and location performance.

In principle, we design different branch to solve the segmentation task based on frequency prior, which is different from most CNN based model. We design building blocks for these branches and propose **H**igh and **L**ow frequency Neural **Net**work (HLNet). Our experiments on Cityscapes and ADE20K demonstrate the effectiveness of HLNet. The main contributions of this paper are as following:

*Corresponding author of this work

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

- We propose to view the segmentation map as the composition of two different frequencies and predict them with different branches. As predicting different frequencies, our method can gain larger receptive field while keep high resolution.
- We analyse the characteristics of high and low frequencies, design proper building blocks to extract different frequencies, and further propose a HLNet to handle scene parsing task.
- We extensively study the effect of the propose two branches and achieve superior performance on various scene parsing datasets without bells-and-whistles.

II. Related Works

Driven by deep neural networks [13]–[15], pixel-level prediction tasks like scene parsing have achieved great success. The FCN [1] first convert the fully-connected layer in traditional classification network into the convolution layer to tackle the segmentation task. Following the FCN framework, there are several works trying to improve scene parsing task based on the following two aspects.

Context embedding. Context embedding is a hot direction. U-Net [4] or other variants [16]–[19] use fuse high level feature and low level feature to enhance context information. The atrous spatial pyramid pooling (ASPP) [8] is proposed to capture the nearby context using different dilated rate. The pyramid pooling module (PPM) [20] is proposed to exploit context information from different scale regions. [21]–[23] stress class-dependent context aggregation. [24] use semantic correlation to aggregate shape-variant context. The low-frequency branch in our method can also be viewed as context aggregation.

High resolution designing. Spatial resolution is important for scene parsing task to hold spatial details. [7] propose dilated convolution to avoid reduce the spatial resolution. [25], [26] propose high resolution neural network with parallel convolutions to get high resolution feature representations. [27] focuses lightweight neural architecture and propose two path method to confront with the loss of spatial information and shrinkage of receptive field respectively. Our method also has high resolution branch, however we use it to extract high frequencies explicitly.

Different from above works, we introduce a method based on frequency prior, which learns different frequencies of the segmentation map separately.

III. Methods

To begin with, we briefly review the concepts of image frequency [11] and Laplacian image pyramid [28], which are the basis for understanding this paper.

A. Background

Image frequency. Image frequency is an important concept in traditional image processing. Low frequencies in a image mean pixel values that are changing slowly over spatial dimension, while high frequency content means pixel values that are rapidly changing. That is, high frequencies usually encode fine details and low frequencies usually encode global structure. We usually use lowpass filter, for example Guassian filter, to extract low frequencies. Vice versa, high frequencies is the remaining part after extraction. The extraction procedure can be formulated as:

$$I_{low} = I * G \tag{1}$$

$$\begin{aligned} I_{high} &= I - I_{low} \\ &= I - I * G \\ &= I * (unit - G) \end{aligned} \tag{2}$$

where I, I_{low} and I_{high} indicate the original image, low frequencies and high frequencies of the original image respectively, G denotes a Guassian filter, $*$ represents convolution operator, and $unit - G$ stands for the difference of Gaussian kernel, which approximates the Laplacian of Gaussian (LoG) kernel [28].

Our method is the inverse process of the above decomposition. We try to predict the low frequencies and high frequencies of the segmentation map, and then reconstruct it, i.e.:

$$\hat{L} = up(\hat{L}_{low}) + \hat{L}_{high} \tag{3}$$

Here the superscript $\hat{\cdot}$ means that the variable is predicted by the model, L denotes the final segmentation map, and $up(\cdot)$ is an up-sample function. Note that it is also the first order Laplacian image pyramid reconstruction procedure, in which the low frequencies will be up-sampled and then added to the high frequencies.

B. High-Frequency and Low-Frequency branch

In this subsection we will firstly clarify the characteristics of high frequencies and low frequencies, and then introduce our designed building blocks.

The high frequencies are obtained by applying LoG kernel on the image. The LoG of an image highlights regions of rapid intensity change, so if a local area feature in the image changes, the high frequencies in this area will also change. However, the changes does not effect the high frequencies in other area. So the high-frequency representations are sensitive to local area feature, but insensitive to non-local feature. Besides, high frequencies are sparse and contain critical spatial information. Based on above analysis, we design the basic block of the high-frequency branch to capture high frequencies. The basic block is illustrated in Fig. 2(a), which has small kernel size to avoid non-local effects and keep high resolution to preserve the spatial information. Besides, because of the sparse representation, slim channels is leveraged to relieve memory cost.

The low frequencies are obtained by smoothing the local area, which determines that low frequencies are robust to the changes of the local area. Besides, the low frequencies contain common information between adjacent locations, so it is safe to reduce the spatial resolution by sharing information between

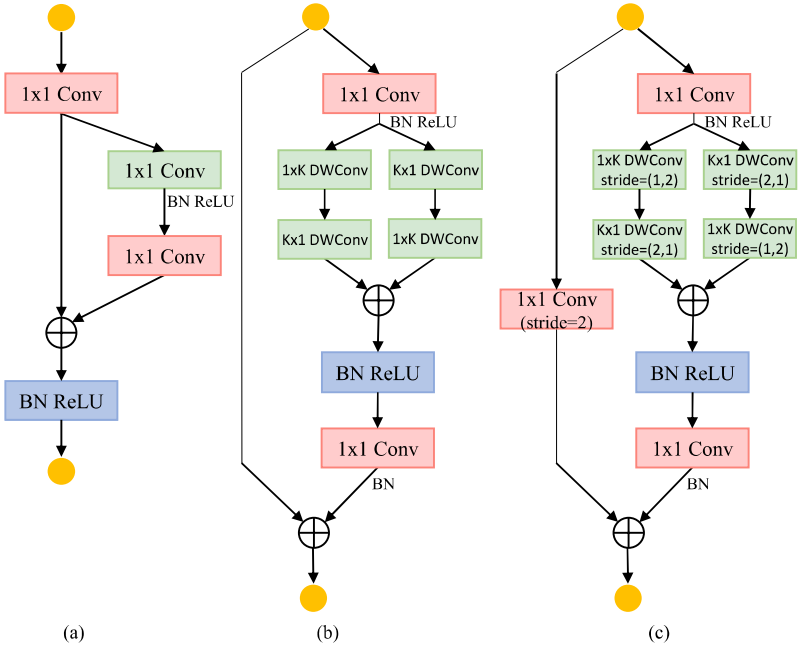

Fig. 2: The building blocks of this work. (a) the basic high-frequency branch unit; (b) the basic low-frequency branch unit; (c) the low-frequency unit with spatial down sampling (2×). **DWConv:** depthwise convolution.

adjacent locations. Therefore, in the low-frequency branch, we adopt ResBlock [15] with large kernel convolution [6] to extract the low frequencies and lower resolution to reduce feature redundancy. The basic blocks of the low-frequency branch is illustrated in Fig. 2(b, c), which shows that we employ the combination of $K \times 1 + 1 \times K$ and $1 \times K + K \times 1$ convolutions to approximate a large kernel convolution to save computation cost.

To enforce these two branches to learn different frequency components of the segmentation map, the high/low frequencies of the ground truth are utilized to supervise these two branches respectively. We will give more detail in next subsection.

C. Overall Framework

Architecture. Based on the designed basic building blocks, we propose the High-Low frequency Network (HLNet) for scene parsing as illustrated in Fig. 3. HLNet consists of three parts, including base network, high-frequency branch and low-frequency branch. We adopt pretrained dilated ResNet [6] as the base network, and high-frequency branch containing two basic blocks (H1 and H2) and low-frequency branch containing three basic blocks (SA [29], L1 and L2) are following in parallel after the base network, which are used to generate different frequency components of the segmentation map respectively. In the base network, we employ dilated convolutions with $rate = 2$ in the last ResNet blocks and remove the subsampling operation, which is a trade-off between computation cost and spatial resolution. The following is low-frequency branch, where we employ spatial attention (SA) module as the first block to compute the feature similarity and enhance feature representations, then two basic low-frequency blocks, L1 and L2 is following. Towards L1 and L2, the former is enabled sub-sampling operation with $stride = 2$ to reduce spatial redundancy, the latter is enabled skip connection to reuse low-frequency feature map. Low-frequency branch also helps model harvest a larger receptive field. In parallel with the low-frequency branch, high-frequency branch consists of two basic high-frequency block, H1 and H2. Considering the sparsity of the high frequencies, which only responses to some area, so dense features in every stage are aggregated after transformation, then subtract the feature map from low-frequency branch which contains sufficient dense information, and finally as the input of the high-frequency branch.

Loss. To enhance the learning ability of branches and learn different things explicitly, we attach loss supervision to these two branch and the final result. The cross entropy loss is employed for these three places. The supervisions are as following,

$$y_{low\_frequencies} = y * G \tag{4}$$

$$y_{high\_frequencies} = y - y_{low\_frequencies} \tag{5}$$

Here y denotes the *mask representation* of the ground truth, $*$ denotes convolution operator, the G denotes the Gussian kernel. $y_{low\_frequencies}$ and $y_{hgih\_frequencies}$ are utilized to supervise low-frequency branch and high-frequency branch respectively. Note that $y_{high\_frequencies} \in (-1, 1]$, we use a trivial transformation to convert its value domain to $[0, 1)$, that is, $(y_{high\_frquencies} + 1)/2$. So the supervisory signal of high-frequency branch is $(y_{high\_frquencies} + 1)/2$. After predicting low frequencies and high frequencies, the final result of the HLNet is the combination of these two branches. Same as Laplacian pyramid reconstruction procedure, where

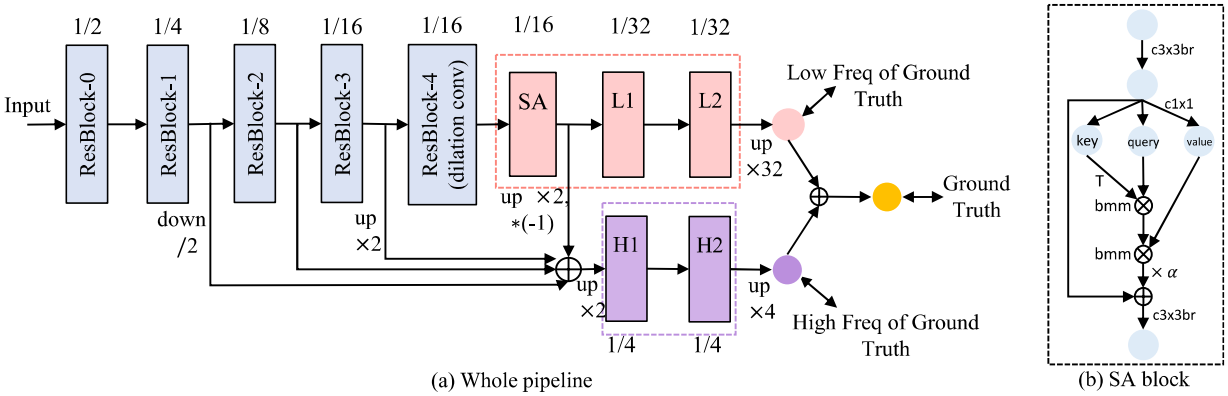

Fig. 3: (a) Overview of High-Low Frequency Network (HLNet). Each box is a computation block in the HLNet. Low frequency network branch is marked by pink dashed box, and high frequency network branch is marked by purple dashed box. The numbers next to boxes indicate the scale of the feature map in that block. *up* or *down* indicates a up-sampling or down-sampling operation. **L1** is a low frequency branch unit with down-sampling shown in Fig.2(c), and **L2** is shown in Fig.2(b). **H1** and **H2** are the high frequency branch unit shown in Fig.2(a). The **pink circle** and the **purple circle** stand for outputs of the low frequency branch and high frequency branch, respectively. The **double arrow** presents that there exists supervision. The detail of Spatial Attention block (**SA block**) is illustrated in (b), where *c3x3br* stands for 3x3conv with BN and ReLU, *T* means matrix transposition, and *bmm* means batch matrix multiplication. Because of low resolution and reduced dimension, the computation of SA module is fast.

high frequencies are added with upsampled low frequencies, we transform the output of high-frequency branch to $\hat{y}_{high\_frequencies} = \hat{y}_{high\_frequencies\_out} * 2 - 1$, and add with $\hat{y_{low_frequencies}}$ directly to get the final result, that is,

$$\hat{y} = \hat{y}_{low-frequencies} + \hat{y}_{high-frequencies} * 2 - 1 \quad (6)$$

supervised by normal ground truth. Following PSPNet [20], to improve the performance and make the deep neural network easier to train, auxiliary loss is enabled. We use four parameters λ_h, λ_l, λ_f and λ_a to reweigh high-frequency loss l_{high}, low-frequency loss l_{low}, final loss l_{final} and auxiliary loss l_{aux}, as shown in Eq.7:

$$Loss = \lambda_h * l_{high} + \lambda_l * l_{low} + \lambda_f * l_{final} + \lambda_a * l_{aux} \quad (7)$$

D. Compare with dual path method

BiSeNet [27] proposes a dual path network, one is Spatial Path, and another is Context Path. The former is to preserve the spatial information, while the latter is to obtain sufficient semantic information with fast downsampling strategy. However, our method is different from it. The way we design the neural network depends on our observation on the high/low frequencies. Just as in image processing, the pioneers designed convolution block based on the fact that each pixel has a relationship to the surrounding pixels, we analyze the characteristics of the high and low frequencies and design specific basic blocks to fit the high and low frequencies, which try to decrease the difficulty of fitting problem. Besides, we give explicit supervision to enhance these two branch.

IV. Experiments

To evaluation the proposed high/low-frequency modeling approach, we conduct extensive experiments on the Citycapes dataset [30] and the ADE20K dataset [31].

- **Cityscapes.** The Cityscapes dataset is urban scene parsing dataset which involve 19 valid classes and 2975, 500, 1525 images in *train*, *val*, and *test* set, respectively. We only use the fine annotated data in our experiments.
- **ADE20K.** ADE20K dataset is used in ImageNet scene parsing challenge 2016 which contains 150 classes and 20K/2K/3K images for training, validation and testing.

In this section, we firstly describe implementation details, and then show ablation study on the Cityscapes dataset to verify the effectiveness of the proposed modeling approach. Comparisons with state-of-the-art on Cityscapes and ADE20K also would be reported at last.

A. Implementation Details

We employ ImageNet pretrained ResNet as our base network with dilated rates in the last ResNet block is set to (2,2,2). During training phrase, following prior works, We employ the polynomial learning rate policy with factor $(1 - \frac{iter}{total\_iters})^{0.9}$, and enable the auxiliary loss if we adopt the backbone ResNet101. Momentum and weight decay coefficients are set as 0.9 and 0.0001, batch size is 16. For the data augmentation, random flipping horizontally, random scaling in the range of [0.5, 2], and random rotating in the range of [-10, 10] are adopted. We use distributed training in Pytorch v1.3 with synchronized batch normalization across multiple GPUs enabled to conduct experiments. We set the coefficients of the losses $\lambda_h = 1.5$, $\lambda_l = 0.8$, $\lambda_f = 1.$ and $\lambda_a = 0.4$ in empirical manner. The mean of class-wise

TABLE I: The comparison with baseline on Cityscapes. The HL Branch is the high-frequency and low-frequency branch, and -SA means that HL Branch excludes the SA block.

Base Network	Block	mIoU(%)
ResNet-50	None	72.35
	+HL Branch (-SA)	76.11
	+HL Branch	78.05
ResNet-101	None	74.32
	+HL Branch (-SA)	77.28
	+HL Branch	79.38

Intersection over Union (mIoU), which can be formulated as $\text{mIoU} = \frac{1}{N}\sum_i^N \frac{t_{ii}}{|C_i|+\sum_j t_{ji}-t_{ii}}$, where N denotes the total of different classes, $|C_i|$ is the number of pixels of class i, and t_{ji} indicates the number of pixels of class j predicted to class i, is employed as the evaluation metric in our experiments. And multi-scale inference scheme with scales 0.5, 0.75, 1.0, 1.25, 1.5, and 1.75 will not be used unless specifically stated. For private setting on these benchmark datasets we will show the following.

Cityscapes: we set crop size as 713×713, initial learning rate as 0.01 and training epoch as 200.

ADE20K: we set crop size as 473×473, initial learning rate as 0.005 and training epoch as 150.

Before feed into SA block shown in Fig.3(b), a 3x3 convolution layer with BN, ReLU is applied on the outputs of ResBlock-4 to reduce the number of channels to 512, then the output of the SA block is also reduced to 512 dimensions. We set the kernel size of low-frequency branch as 7, but we would give ablation study on it later. In addition, we also reduce the dimension of output of every stage in base network to 128 before feeding into H1 block. Here if it needs to down-sample, then $stride = 2$ is enabled during reduction, and if it needs to up-sample, bilinear interpolation operation would be employed after reduction. In the every layer of the low-frequency branch except SA block and high-frequency branch, we set the numbers of channels to 1024 and 128 respectively.

B. Ablation Study

In this subsection, we conduct a series of experiments to verify the effectiveness of the proposed modeling approach and reveal the effect of each branch in our proposed method step by step. First, we give the comparison between HLNet and baseline network which will be introduced later. Then the ablations for which stages high-frequency branch should take as input and for which kernel size low-frequency branch should adopt would be explored. Next, some studies to analyse the function of these losses of two branch in our method would be presented.

Compare with the baseline. To verify the effectiveness of our method, we remove high-frequency branch and low-frequency branch and then get the baseline network. Also, the base network of baseline network is initialized by ImageNet pretrained model. The final result of the baseline are obtained by directly upsampling the output. We also consider the effect of SA block. As shown in Tab. I, our HLNet improves the performance from 72.35% to 76.11% based on ResNet50 and from 74.32% to 77.28% based on ResNet101. And after using SA block to utilize low-frequency similarity, we will get higher performance with 78.05% on ResNet50 and 79.38% on ResNet101, which shows that our improvement is significant.

TABLE II: The effects of the different input of the High-Freq Branch.

Base Network	Input	mIoU(%)
ResNet-50	stage3	76.24
	stage3,2	77.90
	stage3,2,1	78.05
Base Network	Low-freq Interaction	mIoU(%)
ResNet-50	None	74.25
	Add	77.75
	Subtract	78.05

TABLE III: The effect of kernel size in low-frequency branch.

Base Network	Kernel Size	mIoU(%)
ResNet-50	3	77.93
	5	77.95
	7	78.05
	9	77.57
	11	78.01

Which stages should be used to aggregate for high-frequency branch? Tab. II presents the results when aggregating different stages as input of high-frequency branch. As can be seen from Tab. II, the network performance consistently improves with lower stages being used as input. It is intuitive because higher stage has more semantic cues while lower stage contains more local details which is helpful to predict high frequencies. The maximum gap between the default setting with stage3,2,1 as input and other setting with only stage3 as input is up to 1.81% based on ResNet50 backbone.

Which kernel size should be used to extract in low-frequency branch? Large kernel size is benefit to extract low frequencies. Considering different kernel size causing different receptive fields, large kernel size will boost neural network performance. As shown in Tab. III, the network fluctuates slightly and gradually reaches top performance when kernel size reaching 7, and larger kernel size afford no more contribution. We guess the reason is that sufficient receptive field is aggregated, considering output stride of base network and function of SA block.

Empirical analysis of multiple losses. We firstly analyse the effect of multiple losses on the network shown in Tab. IV. Enabling these three losses is better than other settings. Own to design of two branches, other setting also can achieve

TABLE IV: The effect of supervisions.

Base Network	Loss			mIoU(%)
	Final Loss	High-Freq Loss	Low-Freq Loss	
ResNet-101	✓	✓	✓	**79.38**
	✓	✗	✓	78.06
	✓	✓	✗	78.83
	✓	✗	✗	78.57
	✗	✓	✓	71.73

good performance. In addition, these three losses are all cross entropy loss, however, the final loss is using per-class mask (hard label) to supervise, while, the high-freq and low-freq loss using soft label to supervise, shown in Fig. 5. After above analysis, besides advantage of building block in the network for different frequencies, we guess that our model is also benefiting from label smoothing [32], [33].

We also give qualitative result toward the output effected by these multiple losses. We show the visualization of outputs of low frequency branch. Shown as in Fig. 4, the output of low frequency branch more focus on big stuff or objects and the main components of elements, while the final result contains the fine details of the boundaries. This is an expected result since the low frequency branch has low resolution feature and large kernel size, while the final result utilize the help of final supervision and output of high frequency branch. This demonstrates our network has better interpretability.

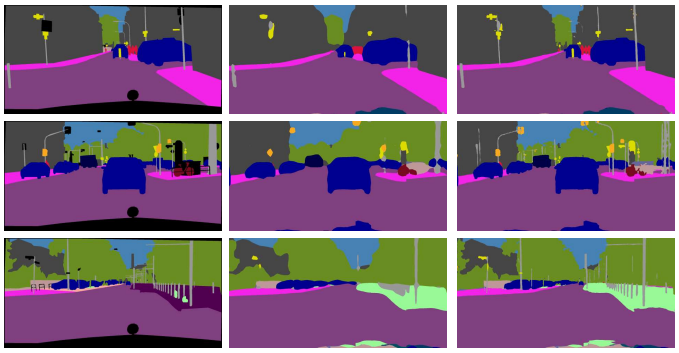

Fig. 4: Visualization of output of low frequency branch and final result. The boundaries of the former is more smooth. The latter contains more fine details.

Discussion. The visual method is found in the mammalian visual system, which decomposes the visual scene into a central high-resolution (foveal) area and a lower resolution surrounded, with attentional shifting to bring the fovea to bear on regions of interest in a visual scene. This biology visual method is highly efficient and motivates many works [29], [34] to introduce *attentional mechanism* into neural network to focus on a some patch in a image. Same as above biology visual method, our method in this paper also introduces low and high resolution, but is different from it in practice. Considering the characteristic of scene parsing task is per-pixel prediction, we introduce parallel high and low resolution performing over the entire image area, instead of separately over different small patches.

0	0	0	0	
0	1	1	0	
0	1	1	0	
0	1	1	0	
0	0	0	0	

(a) Ground truth: mask of this object.

0.25	0.375	0.375	0.25
0.375	0.563	0.563	0.375
0.5	0.75	0.75	0.5
0.375	0.563	0.563	0.375
0.25	0.375	0.375	0.25

(b) Ground truth of low frequencies.

Fig. 5: An example to illustrate the multiple supervision signals. (a) is the normal ground truth of some object with mask representation. (b) is generated by smoothing the (a). Here the Gaussian blur with kernel 3 is adopted. Correspondingly, ground truth of high frequencies is (a) subtracting (b). The depth of the color indicates the degree of supervision.

TABLE V: Comparisions with other state-of-the-arts results on Cityscapes *val* set and ADE20K *val* set. We adopt ResNet101 with output stride 16 as our base network. Our method outperforms most previous methods in mIoU on these benchmarks. Red represents that the result outperforms ours.

Method	Base Network	mIoU(%) on Cityscapes	mIoU(%) on ADE20K
Dilation10 [7]	VGG16	68.7	-
LRR [35]	VGG16	70.0	-
DeepLabV2+CRF [8]	ResNet-101	71.4	-
DUC [36]	ResNet-152	76.7	-
DSSPN [37]	ResNet-101	77.8	43.68
PSPNet [20]	ResNet-101	79.2	44.15
PSANet [38]	ResNet-101	79.4	44.14
CFNet [39]	ResNet-101	79.5	44.89
RefineNet [18]	ResNet-152	-	40.7
UperNet [40]	ResNet-101	-	42.66
HLNet(Ours)	ResNet-101	**80.14**	**44.07**

C. Comparisons with State-of-the-Art

Cityscapes. We evaluate HLNet on the Cityscapes dataset with multi-scale inference scheme. The Tab. V shows that our comparison with other existing leading algorithms on *val* set. Our proposed method, which uses only *train-fine* data, achieves **80.14%** mIoU. Compared with other algorithms, our method reaches superior performance based on ResNet101 without bells-and-whistles.

ADE20K. We also evaluate HLNet on the ADE20K

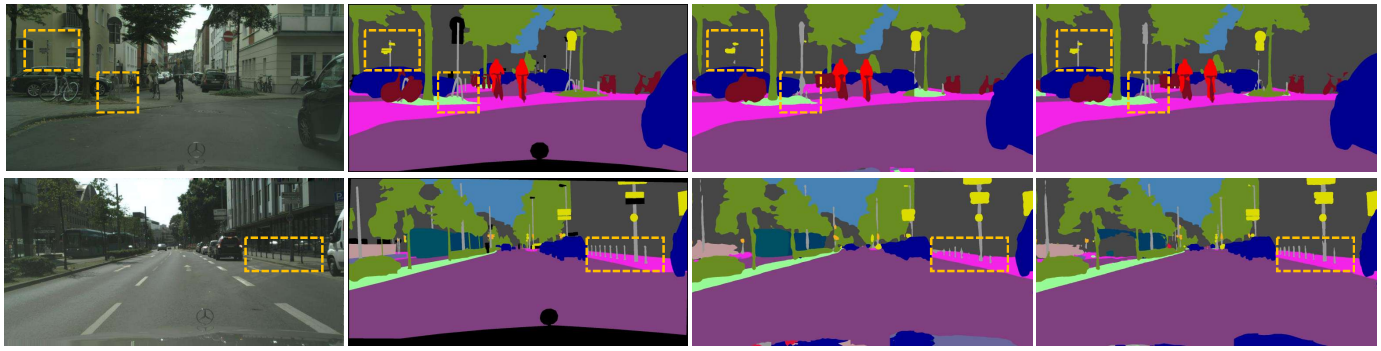

Fig. 6: Visualization of the baseline and HLNet base on ResNet-101 on Cityscapes *val* set. Improved areas are marked with yellow dashed boxes.

dataset with multi-scale strategy being adopted. As shown in last column of Tab. V, our method achieves **44.07%** mIoU on ADE20K *val* set, which is a competitive performance compared with most of the previous approaches. The methods marked as Red in Tab. V are slightly better than ours. However, our method has less computation than them since they all adopt ResNet101 with output stride 8 as base network. For example, given a single scale 473 $\times$ 473 input and same base network ResNet-101, the FLOPs of PSPNet, PSANet and our method are 230.29G, 235.48G, and 89.38G, respectively (We can not calculate FLOPs of CFNet due to the source code is not available). The comparison shows our method has big advantage.

Qualitative Results. We provide the qualitative result in Fig. 6 on Cityscapes benchmarks. We use the yellow dashed boxes to mark those challenging regions that are mis-labeled by the baseline easily but corrected by HLNet. We also note some new errors being introduced in our result, for example, in Fig. 6 the large gray area (its semantic label is building class) superimposed on the green area (train class) under the tree at mid-left in the lower frame, which does not appear in baseline result. We guess that it may be caused by the low weight of low-frequency loss.

V. Conclusion

In this work, we analyse the characteristics of different frequency components and propose two branches to capture high and low frequencies of the segmentation map for scene parsing problem. Different from most existing methods, the proposed approach predicts high frequencies and low frequencies respectively utilizing their characteristics, which results in larger receptive field to extract low-frequency component and higher resolution to keep high-frequency component. The experiments has proved the effectiveness of the proposed approach and analysed the effects of each branch. Experiments on Cityscapes and ADE20K prove the superiority of the proposed approach on scene parsing. More potential application using frequency prior (*e.g.*, lightweight CNN design and neural architecture search) and efficient imitating mammalian visual system remain to be explored in the future.

acknowledgement. We thank the anonymous reviewer for discussion of comparison of our method with that visual method found in the mammalian brain.

References

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*. IEEE Computer Society, 2015, pp. 3431–3440.

[2] W. Luo, Y. Li, R. Urtasun, and R. S. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *NIPS*, 2016, pp. 4898–4906.

[3] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene cnns," in *ICLR*, 2015.

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI (3)*, ser. Lecture Notes in Computer Science, vol. 9351. Springer, 2015, pp. 234–241.

[5] Z. Zhang, X. Zhang, C. Peng, X. Xue, and J. Sun, "Exfuse: Enhancing feature fusion for semantic segmentation," in *ECCV (10)*, ser. Lecture Notes in Computer Science, vol. 11214. Springer, 2018, pp. 273–288.

[6] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters - improve semantic segmentation by global convolutional network," in *CVPR*. IEEE Computer Society, 2017, pp. 1743–1751.

[7] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *ICLR (Poster)*, 2016.

[8] L. 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, 2018.

[9] L. Song, Y. Li, Z. Li, G. Yu, H. Sun, J. Sun, and N. Zheng, "Learnable tree filter for structure-preserving feature transform," in *NeurIPS*, 2019, pp. 1709–1719.

[10] F. Campbell and J. Robson, "Application of fourier analysis to the visibility of gratings," *Journal of Physiology (London)*, vol. 197Channels in humans, pp. 551–556, 08 1968.

[11] R. DeValois and K. DeValois, "Spatial vision," *Oxford psychology series No. 14*, vol. 5, pp. 147–175, 01 2008.

[12] Y. Chen, H. Fan, B. Xu, Z. Yan, Y. Kalantidis, M. Rohrbach, S. Yan, and J. Feng, "Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution," *CoRR*, vol. abs/1904.05049, 2019.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1106–1114.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*. IEEE Computer Society, 2015, pp. 1–9.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*. IEEE Computer Society, 2016, pp. 770–778.

[16] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017.

[17] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *ICCV*. IEEE Computer Society, 2015, pp. 1520–1528.

[18] G. Lin, A. Milan, C. Shen, and I. D. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *CVPR*. IEEE Computer Society, 2017, pp. 5168–5177.

[19] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *DLMIA/ML-CDS@MICCAI*, ser. Lecture Notes in Computer Science, vol. 11045. Springer, 2018, pp. 3–11.

[20] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*. IEEE Computer Society, 2017, pp. 6230–6239.

[21] H. Zhang, K. J. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *CVPR*. IEEE Computer Society, 2018, pp. 7151–7160.

[22] F. Zhang, Y. Chen, Z. Li, Z. Hong, J. Liu, F. Ma, J. Han, and E. Ding, "Acfnet: Attentional class feature network for semantic segmentation," *CoRR*, vol. abs/1909.09408, 2019.

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

[24] H. Ding, X. Jiang, B. Shuai, A. Q. Liu, and G. Wang, "Semantic correlation promoted shape-variant context for segmentation," in *CVPR*. Computer Vision Foundation / IEEE, 2019, pp. 8885–8894.

[25] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," *CoRR*, vol. abs/1908.07919, 2019.

[26] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang, "High-resolution representations for labeling pixels and regions," *CoRR*, vol. abs/1904.04514, 2019.

[27] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *ECCV (13)*, ser. Lecture Notes in Computer Science, vol. 11217. Springer, 2018, pp. 334–349.

[28] P. Burt and E. Adelson, "The laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, April 1983.

[29] X. Wang, R. B. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *CVPR*. IEEE Computer Society, 2018, pp. 7794–7803.

[30] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*. IEEE Computer Society, 2016, pp. 3213–3223.

[31] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *CVPR*. IEEE Computer Society, 2017, pp. 5122–5130.

[32] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *CVPR*. Computer Vision Foundation / IEEE, 2019, pp. 558–567.

[33] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" in *NeurIPS*, 2019, pp. 4696–4705.

[34] C. Yang, Z. An, H. Zhu, X. Hu, K. Zhang, K. Xu, C. Li, and Y. Xu, "Gated convolutional networks with hybrid connectivity for image classification," *CoRR*, vol. abs/1908.09699, 2019.

[35] G. Ghiasi and C. C. Fowlkes, "Laplacian pyramid reconstruction and refinement for semantic segmentation," in *ECCV (3)*, ser. Lecture Notes in Computer Science, vol. 9907. Springer, 2016, pp. 519–534.

[36] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. W. Cottrell, "Understanding convolution for semantic segmentation," in *WACV*. IEEE Computer Society, 2018, pp. 1451–1460.

[37] X. Liang, H. Zhou, and E. P. Xing, "Dynamic-structured semantic propagation network," in *CVPR*. IEEE Computer Society, 2018, pp. 752–761.

[38] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. C. Loy, D. Lin, and J. Jia, "Psanet: Point-wise spatial attention network for scene parsing," in *ECCV (9)*, ser. Lecture Notes in Computer Science, vol. 11213. Springer, 2018, pp. 270–286.

[39] H. Zhang, H. Zhang, C. Wang, and J. Xie, "Co-occurrent features in semantic segmentation," in *CVPR*. Computer Vision Foundation / IEEE, 2019, pp. 548–557.

[40] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *ECCV (5)*, ser. Lecture Notes in Computer Science, vol. 11209. Springer, 2018, pp. 432–448.