This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.2983686, IEEE Transactions on Pattern Analysis and Machine Intelligence

Deep High-Resolution Representation Learning for Visual Recognition

Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao

Abstract—High-resolution representations are essential for position-sensitive vision problems, such as human pose estimation, semantic segmentation, and object detection. Existing state-of-the-art frameworks first encode the input image as a low-resolution representation through a subnetwork that is formed by connecting high-to-low resolution convolutions *in series* (e.g., ResNet, VGGNet), and then recover the high-resolution representation from the encoded low-resolution representation. Instead, our proposed network, named as High-Resolution Network (HRNet), maintains high-resolution representations through the whole process. There are two key characteristics: (i) Connect the high-to-low resolution convolution streams *in parallel*; (ii) Repeatedly exchange the information across resolutions. The benefit is that the resulting representation is semantically richer and spatially more precise. We show the superiority of the proposed HRNet in a wide range of applications, including human pose estimation, semantic segmentation, and object detection, suggesting that the HRNet is a stronger backbone for computer vision problems. All the codes are available at https://github.com/HRNet.

Index Terms—HRNet, high-resolution representations, low-resolution representations, human pose estimation, semantic segmentation, object detection.

1 Introduction

Deep convolutional neural networks (DCNNs) have achieved state-of-the-art results in many computer vision tasks, such as image classification, object detection, semantic segmentation, human pose estimation, and so on. The strength is that DCNNs are able to learn richer representations than conventional hand-crafted representations.

Most recently-developed classification networks, including AlexNet [61], VGGNet [102], GoogleNet [109], ResNet [40], etc., follow the design rule of LeNet-5 [63]. The rule is depicted in Figure 1 (a): gradually reduce the spatial size of the feature maps, connect the convolutions from high resolution to low resolution in series, and lead to a *low-resolution representation*, which is further processed for classification.

High-resolution representations are needed for position-sensitive tasks, e.g., semantic segmentation, human pose estimation, and object detection. The previous state-of-the-art methods adopt the high-resolution recovery process to raise the representation resolution from the low-resolution representation outputted by a classification or classification-like network as depicted in Figure 1 (b), e.g., Hourglass [85], SegNet [3], DeconvNet [87], U-Net [97], SimpleBaseline [126], and encoder-decoder [92]. In addition, dilated convolutions are used to remove some down-sample layers and thus yield medium-resolution representations [15], [148].

We present a novel architecture, namely High-Resolution Net (HRNet), which is able to *maintain high-resolution representations* through the whole process. We start from a high-resolution convolution stream, gradually add high-to-low resolution convolution streams one by one, and connect the multi-resolution streams in parallel. The resulting network consists of several (4 in this paper) stages as depicted in Figure 2, and the nth stage contains n streams corresponding to n resolutions. We conduct repeated multi-resolution fusions by exchanging the information across the parallel streams over and over.

The high-resolution representations learned from HRNet are not only semantically strong but also spatially precise. This comes from two aspects. (i) Our approach connects high-to-low resolution convolution streams in parallel rather than in series. Thus, our approach is able to maintain the high resolution instead of recovering high resolution from low resolution, and accordingly the learned representation is potentially spatially more precise. (ii) Most existing fusion schemes aggregate high-resolution low-level and high-level representations obtained by upsampling low-resolution representations. Instead, we repeat multi-resolution fusions to boost the high-resolution representations with the help of the low-resolution representations, and vice versa. As a result, all the high-to-low resolution representations are semantically strong.

We present two versions of HRNet. The first one, named as HRNetV1, only outputs the high-resolution representation computed from the high-resolution convolution stream. We apply it to human pose estimation by following the heatmap estimation framework. We empirically demonstrate the superior pose estimation performance on the COCO keypoint detection dataset [76].

The other one, named as HRNetV2, combines the representations from all the high-to-low resolution parallel streams. We apply it to semantic segmentation through estimating segmentation maps from the combined high-resolution representation. The proposed approach achieves state-of-the-art results on PASCAL-Context, Cityscapes, and LIP with similar model sizes and lower computation com-

- *J. Wang is with Microsoft Research, Beijing, P.R. China. E-mail: jingdw@microsoft.com*

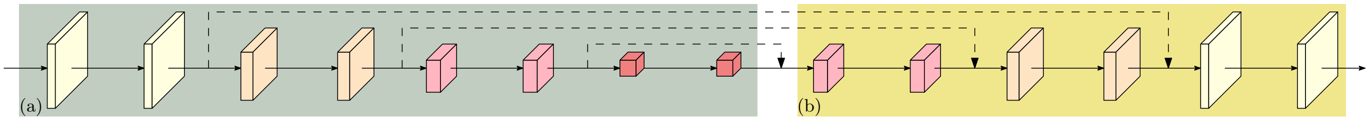


Fig. 1. The structure of recovering high resolution from low resolution. (a) A low-resolution representation learning subnetwork (such as VGGNet [102], ResNet [40]), which is formed by connecting high-to-low convolutions in series. (b) A high-resolution representation recovering subnetwork, which is formed by connecting low-to-high convolutions in series. Representative examples include SegNet [3], DeconvNet [87], U-Net [97] and Hourglass [85], encoder-decoder [92], and SimpleBaseline [126].

plexity. We observe similar performance for HRNetV1 and HRNetV2 over COCO pose estimation, and the superiority of HRNetV2 to HRNet1 in semantic segmentation.

In addition, we construct a multi-level representation, named as HRNetV2p, from the high-resolution representation output from HRNetV2, and apply it to state-of-the-art detection frameworks, including Faster R-CNN, Cascade R-CNN [9], FCOS [112], and CenterNet [28], and state-of-the-art joint detection and instance segmentation frameworks, including Mask R-CNN [39], Cascade Mask R-CNN, and Hybrid Task Cascade [12]. The results show that our method gets detection performance improvement and in particular dramatic improvement for small objects.

2 Related Work

We review closely-related representation learning techniques developed mainly for human pose estimation [43], semantic segmentation and object detection, from three aspects: low-resolution representation learning, high-resolution representation recovering, and high-resolution representation maintaining. Besides, we mention about some works related to multi-scale fusion.

Learning low-resolution representations. The fully-convolutional network approaches [81], [100] compute low-resolution representations by removing the fully-connected layers in a classification network, and estimate their coarse segmentation maps. The estimated segmentation maps are improved by combining the fine segmentation score maps estimated from intermediate low-level medium-resolution representations [81], or iterating the processes [60]. Similar techniques have also been applied to edge detection, e.g., holistic edge detection [130].

The fully convolutional network is extended, by replacing a few (typically two) strided convolutions and the associated convolutions with dilated convolutions, to the dilation version, leading to medium-resolution representations [14], [15], [68], [138], [148]. The representations are further augmented to multi-scale contextual representations [15], [17], [148] through feature pyramids for segmenting objects at multiple scales.

Recovering high-resolution representations. An upsample process can be used to gradually recover the high-resolution representations from the low-resolution representations. The upsample subnetwork could be a symmetric version of the downsample process (e.g., VGGNet), with skipping connection over some mirrored layers to transform the pooling indices, e.g., SegNet [3] and DeconvNet [87], or copying the feature maps, e.g., U-Net [97] and Hourglass [6], [7], [22], [25], [53], [85], [110], [134], [135], encoder-decoder [92], and so on. An extension of U-Net, full-resolution residual network [94], introduces an extra full-resolution stream that carries information at the full image resolution, to replace the skip connections, and each unit in the downsample and upsample subnetworks receives information from and sends information to the full-resolution stream.

The asymmetric upsample process is also widely studied. RefineNet [72] improves the combination of upsampled representations and the representations of the same resolution copied from the downsample process. Other works include: light upsample process [5], [19], [74], [126], possibly with dilated convolutions used in the backbone [49], [71], [93]; light downsample and heavy upsample processes [116], recombinator networks [41]; improving skip connections with more or complicated convolutional units [50], [91], [147], as well as sending information from low-resolution skip connections to high-resolution skip connections [155] or exchanging information between them [35]; studying the details of the upsample process [122]; combining multi-scale pyramid representations [18], [127]; stacking multiple DeconvNets/U-Nets/Hourglass [32], [124] with dense connections [111].

Maintaining high-resolution representations. Our work is closely related to several works that can also generate high-resolution representations, e.g., convolutional neural fabrics [99], interlinked CNNs [154], GridNet [30], and multi-scale DenseNet [44].

The two early works, convolutional neural fabrics [99] and interlinked CNNs [154], lack careful design on when to start low-resolution parallel streams, and how and where to exchange information across parallel streams, and do not use batch normalization and residual connections, thus not showing satisfactory performance. GridNet [30] is like a combination of multiple U-Nets and includes two symmetric information exchange stages: the first stage passes information only from high resolution to low resolution, and the second stage passes information only from low resolution to high resolution. This limits its segmentation quality. Multi-scale DenseNet [44] is not able to learn strong high-resolution representations as there is no information received from low-resolution representations.

Multi-scale fusion. Multi-scale fusion[1] is widely studied [8], [15], [19], [30], [44], [52], [98], [99], [130], [133], [148], [154]. The straightforward way is to feed multi-resolution images separately into multiple networks and aggregate the output response maps [113]. Hourglass [85], U-Net [97], and SegNet [3] combine low-level features in the high-to-low downsample process into the same-resolution high-level features in the low-to-high upsample process progressively through skip connections. PSPNet [148] and DeepLabV2/3 [15] fuse the pyramid features obtained by pyramid pooling module

1. In this paper, Multi-scale fusion and multi-resolution fusion are interchangeable, but in other contexts, they may not be interchangeable.

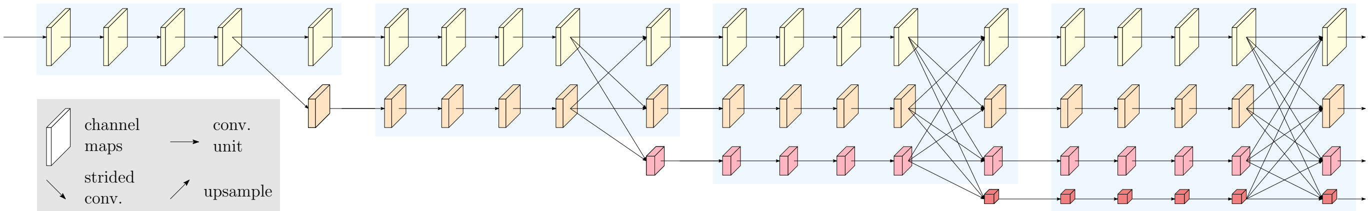


Fig. 2. An example of a high-resolution network. Only the main body is illustrated, and the stem (two stride-2 3×3 convolutions) is not included. There are four stages. The 1st stage consists of high-resolution convolutions. The 2nd (3rd, 4th) stage repeats two-resolution (three-resolution, four-resolution) blocks. The detail is given in Section 3.

and atrous spatial pyramid pooling. Our multi-scale (resolution) fusion module resembles the two pooling modules. The differences include: (1) Our fusion outputs four-resolution representations other than only one, and (2) our fusion modules are repeated several times which is inspired by deep fusion [105], [118], [128], [145], [151].

Our approach. Our network connects high-to-low convolution streams in parallel. It maintains high-resolution representations through the whole process, and generates reliable high-resolution representations with strong position sensitivity through repeatedly fusing the representations from multi-resolution streams.

This paper represents a very substantial extension of our previous conference paper [106] with an additional material added from our unpublished technical report [107] as well as more object detection results under recently-developed start-of-the-art object detection and instance segmentation frameworks. The main technical novelties compared with [106] lie in threefold. (1) We extend the network (named as HRNetV1) proposed in [106], to two versions: HRNetV2 and HRNetV2p, which explore all the four-resolution representations. (2) We build the connection between multi-resolution fusion and regular convolution, which provides an evidence for the necessity of exploring all the four-resolution representations in HRNetV2 and HRNetV2p. (3) We show the superiority of HRNetV2 and HRNetV2p over HRNetV1 and present the applications of HRNetV2 and HRNetV2p in a broad range of vision problems, including semantic segmentation and object detection.

3 High-Resolution Networks

We input the image into a stem, which consists of two stride-2 3×3 convolutions decreasing the resolution to $\frac{1}{4}$, and subsequently the main body that outputs the representation with the same resolution ($\frac{1}{4}$). The main body, illustrated in Figure 2 and detailed below, consists of several components: parallel multi-resolution convolutions, repeated multi-resolution fusions, and representation head that is shown in Figure 4.

3.1 Parallel Multi-Resolution Convolutions

We start from a high-resolution convolution stream as the first stage, gradually add high-to-low resolution streams one by one, forming new stages, and connect the multi-resolution streams in parallel. As a result, the resolutions for the parallel streams of a later stage consists of the resolutions from the previous stage, and an extra lower one.

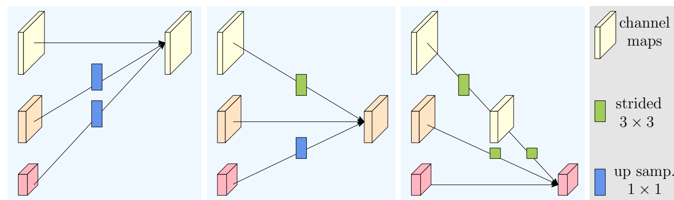


Fig. 3. Illustrating how the fusion module aggregates the information for high, medium and low resolutions from left to right, respectively. Right legend: strided 3×3 = stride-2 3×3 convolution, up samp. 1×1 = bilinear upsampling followed by a 1×1 convolution.

An example network structure illustrated in Figure 2, containing 4 parallel streams, is logically as follows,

$$\begin{array}{ccccccc} \mathcal{N}_{11} & \rightarrow & \mathcal{N}_{21} & \rightarrow & \mathcal{N}_{31} & \rightarrow & \mathcal{N}_{41} \\ & \searrow & \mathcal{N}_{22} & \rightarrow & \mathcal{N}_{32} & \rightarrow & \mathcal{N}_{42} \\ & & & \searrow & \mathcal{N}_{33} & \rightarrow & \mathcal{N}_{43} \\ & & & & & \searrow & \mathcal{N}_{44}, \end{array} \tag{1}$$

where $\mathcal{N}_{sr}$ is a sub-stream in the sth stage and r is the resolution index. The resolution index of the first stream is $r = 1$. The resolution of index r is $\frac{1}{2^{r-1}}$ of the resolution of the first stream.

3.2 Repeated Multi-Resolution Fusions

The goal of the fusion module is to exchange the information across multi-resolution representations. It is repeated several times (e.g., every 4 residual units).

Let us look at an example of fusing 3-resolution representations, which is illustrated in Figure 3. Fusing 2 representations and 4 representations can be easily derived. The input consists of three representations: $\{\mathbf{R}^i_r, r = 1, 2, 3\}$, with r is the resolution index, and the associated output representations are $\{\mathbf{R}^o_r, r = 1, 2, 3\}$. Each output representation is the sum of the transformed representations of the three inputs: $\mathbf{R}^o_r = f_{1r}(\mathbf{R}^i_1) + f_{2r}(\mathbf{R}^i_2) + f_{3r}(\mathbf{R}^i_3)$. The fusion across stages (from stage 3 to stage 4) has an extra output: $\mathbf{R}^o_4 = f_{14}(\mathbf{R}^i_1) + f_{24}(\mathbf{R}^i_2) + f_{34}(\mathbf{R}^i_3)$.

The choice of the transform function $f_{xr}(\cdot)$ is dependent on the input resolution index x and the output resolution index r. If $x = r$, $f_{xr}(\mathbf{R}) = \mathbf{R}$. If $x < r$, $f_{xr}(\mathbf{R})$ downsamples the input representation $\mathbf{R}$ through $(r - s)$ stride-2 3×3 convolutions. For instance, one stride-2 3×3 convolution for $2\times$ downsampling, and two consecutive stride-2 3×3 convolutions for $4\times$ downsampling. If $x > r$, $f_{xr}(\mathbf{R})$ upsamples the input representation $\mathbf{R}$ through the bilinear upsampling followed by a 1×1 convolution for aligning the number of channels. The functions are depicted in Figure 3.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.2983686, IEEE Transactions on Pattern Analysis and Machine Intelligence

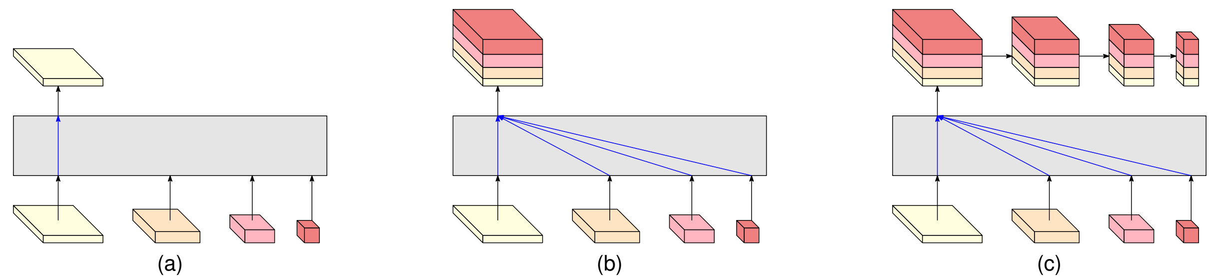


Fig. 4. (a) HRNetV1: only output the representation from the high-resolution convolution stream. (b) HRNetV2: Concatenate the (upsampled) representations that are from all the resolutions (the subsequent 1×1 convolution is not shown for clarity). (c) HRNetV2p: form a feature pyramid from the representation by HRNetV2. The four-resolution representations at the bottom in each sub-figure are outputted from the network in Figure 2, and the gray box indicates how the output representation is obtained from the input four-resolution representations.

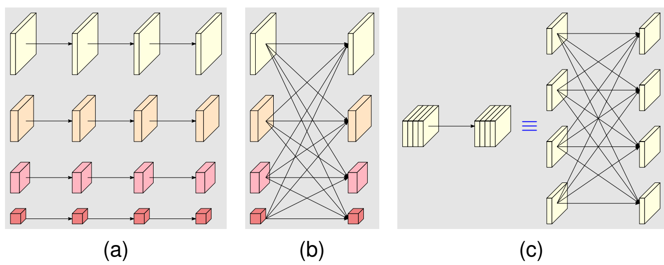


Fig. 5. (a) Multi-resolution parallel convolution, (b) multi-resolution fusion. (c) A normal convolution (left) is equivalent to fully-connected multi-branch convolutions (right).

3.3 Representation Head

We have three kinds of representation heads that are illustrated in Figure 4, and call them as HRNetV1, HRNetV2, and HRNetV1p, respectively.

HRNetV1**.** The output is the representation only from the high-resolution stream. Other three representations are ignored. This is illustrated in Figure 4 (a).

HRNetV2**.** We rescale the low-resolution representations through bilinear upsampling without changing the number of channels to the high resolution, and concatenate the four representations, followed by a 1×1 convolution to mix the four representations. This is illustrated in Figure 4 (b).

HRNetV2**p.** We construct multi-level representations by downsampling the high-resolution representation output from HRNetV2 to multiple levels. This is depicted in Figure 4 (c).

In this paper, we will show the results of applying HRNetV1 to human pose estimation, HRNetV2 to semantic segmentation, and HRNetV2p to object detection.

3.4 Instantiation

The main body contains four stages with four parallel convolution streams. The resolutions are $1/4$, $1/8$, $1/16$, and $1/32$. The first stage contains 4 residual units where each unit is formed by a bottleneck with the width 64, and is followed by one 3×3 convolution changing the width of feature maps to C. The 2nd, 3rd, 4th stages contain 1, 4, 3 modularized blocks, respectively. Each branch in multi-resolution parallel convolution of the modularized block contains 4 residual units. Each unit contains two 3×3 convolutions for each resolution, where each convolution is followed by batch normalization and the nonlinear activation ReLU. The widths (numbers of channels) of the convolutions of the four resolutions are C, $2C$, $4C$, and $8C$, respectively. An example is depicted in Figure 2.

3.5 Analysis

We analyze the modularized block that is divided into two components: multi-resolution parallel convolutions (Figure 5 (a)), and multi-resolution fusion (Figure 5 (b)). The multi-resolution parallel convolution resembles the group convolution. It divides the input channels into several subsets of channels and performs a regular convolution over each subset over different spatial resolutions separately, while in the group convolution, the resolutions are the same. This connection implies that the multi-resolution parallel convolution enjoys some benefit of the group convolution.

The multi-resolution fusion unit resembles the multi-branch full-connection form of the regular convolution, illustrated in Figure 5 (c). A regular convolution can be divided as multiple small convolutions as explained in [145]. The input channels are divided into several subsets, and the output channels are also divided into several subsets. The input and output subsets are connected in a fully-connected fashion, and each connection is a regular convolution. Each subset of output channels is a summation of the outputs of the convolutions over each subset of input channels. The differences lie in that our multi-resolution fusion needs to handle the resolution change. The connection between multi-resolution fusion and regular convolution provides an evidence for exploring all the four-resolution representations done in HRNetV2 and HRNetV2p.

4 Human Pose Estimation

Human pose estimation, a.k.a. keypoint detection, aims to detect the locations of K keypoints or parts (e.g., elbow, wrist, etc) from an image $\mathbf{I}$ of size $W \times H \times 3$. We follow the state-of-the-art framework and transform this problem to estimating K heatmaps of size $\frac{W}{4} \times \frac{H}{4}$, $\{\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_K\}$, where each heatmap $\mathbf{H}_k$ indicates the location confidence of the kth keypoint.

We regress the heatmaps over the high-resolution representations output by HRNetV1. We empirically observe that the performance is almost the same for HRNetV1 and HRNetV2, and thus we choose HRNetV1 as its computation complexity is a little lower. The loss function, defined as the mean squared error, is applied for comparing the predicted heatmaps and the groundtruth heatmaps. The groundtruth heatmaps are generated by applying 2D Gaussian with

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.2983686, IEEE Transactions on Pattern Analysis and Machine Intelligence

Fig. 6. Qualitative COCO human pose estimation results over representative images with various human size, different poses, or clutter background.

TABLE 1
Comparisons on COCO `val`. Under the input size 256×192, our approach with a small model HRNetV1-W32, trained from scratch, performs better than previous state-of-the-art methods. Under the input size 384×288, our approach with a small model HRNetV1-W32 achieves a higher AP score than SimpleBaseline with a large model. In particular, the improvement of our approach for AP^{75}, a strict evaluation scheme, is more significant than AP^{50}, a loose evaluation scheme. Pretrain = pretrain the backbone on ImageNet. OHKM = online hard keypoints mining [19]. #Params and FLOPs are calculated for the pose estimation network, and those for human detection and keypoint grouping are not included.

Method	Backbone	Pretrain	Input size	#Params	GFLOPs	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
8-stage Hourglass [85]	8-stage Hourglass	N	256×192	25.1M	14.3	66.9	–	–	–	–	–
CPN [19]	ResNet-50	Y	256×192	27.0M	6.20	68.6	–	–	–	–	–
CPN + OHKM [19]	ResNet-50	Y	256×192	27.0M	6.20	69.4	–	–	–	–	–
SimpleBaseline [126]	ResNet-50	Y	256×192	34.0M	8.90	70.4	88.6	78.3	67.1	77.2	76.3
SimpleBaseline [126]	ResNet-101	Y	256×192	53.0M	12.4	71.4	89.3	79.3	68.1	78.1	77.1
SimpleBaseline [126]	ResNet-152	Y	256×192	68.6M	15.7	72.0	89.3	79.8	68.7	78.9	77.8
HRNetV1	HRNetV1-W32	N	256×192	28.5M	7.10	73.4	89.5	80.7	70.2	80.1	78.9
HRNetV1	HRNetV1-W32	Y	256×192	28.5M	7.10	74.4	90.5	81.9	70.8	81.0	79.8
HRNetV1	HRNetV1-W48	Y	256×192	63.6M	14.6	75.1	90.6	82.2	71.5	81.8	80.4
SimpleBaseline [126]	ResNet-152	Y	384×288	68.6M	35.6	74.3	89.6	81.1	70.5	79.7	79.7
HRNetV1	HRNetV1-W32	Y	384×288	28.5M	16.0	75.8	90.6	82.7	71.9	82.8	81.0
HRNetV1	HRNetV1-W48	Y	384×288	63.6M	32.9	**76.3**	**90.8**	**82.9**	**72.3**	**83.4**	**81.2**

TABLE 2
Comparisons on COCO `test-dev`. The observations are similar to the results on COCO `val`.

Method	Backbone	Input size	#Params	GFLOPs	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
Bottom-up: keypoint detection and grouping										
OpenPose [11]	–	–	–	–	61.8	84.9	67.5	57.1	68.2	66.5
Associative Embedding [84]	–	–	–	–	65.5	86.8	72.3	60.6	72.6	70.2
PersonLab [88]	–	–	–	–	68.7	89.0	75.4	64.1	75.5	75.4
MultiPoseNet [57]	–	–	–	–	69.6	86.3	76.6	65.0	76.3	73.5
Top-down: human detection and single-person keypoint detection										
Mask-RCNN [39]	ResNet-50-FPN	–	–	–	63.1	87.3	68.7	57.8	71.4	–
G-RMI [89]	ResNet-101	353×257	42.6M	57.0	64.9	85.5	71.3	62.3	70.0	69.7
Integral Pose Regression [108]	ResNet-101	256×256	45.0M	11.0	67.8	88.2	74.8	63.9	74.0	–
G-RMI + extra data [89]	ResNet-101	353×257	42.6M	57.0	68.5	87.1	75.5	65.8	73.3	73.3
CPN [19]	ResNet-Inception	384×288	–	–	72.1	91.4	80.0	68.7	77.2	78.5
RMPE [29]	PyraNet [135]	320×256	28.1M	26.7	72.3	89.2	79.1	68.0	78.6	–
CFN [46]	–	–	–	–	72.6	86.1	69.7	78.3	64.1	–
CPN (ensemble) [19]	ResNet-Inception	384×288	–	–	73.0	91.7	80.9	69.5	78.1	79.0
SimpleBaseline [126]	ResNet-152	384×288	68.6M	35.6	73.7	91.9	81.1	70.3	80.0	79.0
HRNetV1	HRNetV1-W32	384×288	28.5M	16.0	74.9	92.5	82.8	71.3	80.9	80.1
HRNetV1	HRNetV1-W48	384×288	63.6M	32.9	75.5	92.5	83.3	71.9	81.5	80.5
HRNetV1 + extra data	HRNetV1-W48	384×288	63.6M	32.9	**77.0**	**92.7**	**84.5**	**73.4**	**83.1**	**82.0**

standard deviation of 2 pixel centered on the groundtruth location of each keypoint. Some example results are given in Figure 6.

Dataset. The COCO dataset [76] contains over $200,000$ images and $250,000$ person instances labeled with 17 keypoints. We train our model on the COCO `train2017` set, including $57K$ images and $150K$ person instances. We evaluate our approach on the `val2017` and `test-dev2017` sets, containing 5000 images and $20K$ images, respectively.

Evaluation metric. The standard evaluation metric is based on Object Keypoint Similarity (OKS): $\mathrm{OKS} = \frac{\sum_i \exp(-d_i^2/2s^2k_i^2)\delta(v_i>0)}{\sum_i \delta(v_i>0)}$. Here d_i is the Euclidean distance between the detected keypoint and the corresponding ground truth, v_i is the visibility flag of the ground truth, s is the object scale, and k_i is a per-keypoint constant that controls falloff. We report standard average precision and recall scores[2]: AP^{50} (AP at OKS $=0.50$), AP^{75}, AP (the mean of AP scores at 10 OKS positions, $0.50, 0.55, \ldots, 0.90, 0.95$); AP^{M} for medium objects, AP^{L} for large objects, and AR (the mean of AR scores at 10 OKS positions, $0.50, 0.55, \ldots, 0.90, 0.95$).

Training. We extend the human detection box in height or width to a fixed aspect ratio: $\mathrm{height} : \mathrm{width} = 4 : 3$, and then crop the box from the image, which is resized to a fixed size, 256×192 or 384×288. The data augmenta-

2. http://cocodataset.org/#keypoints-eval

tion scheme includes random rotation ($[-45°, 45°]$), random scale ($[0.65, 1.35]$), and flipping. Following [121], half body data augmentation is also involved.

We use the Adam optimizer [56]. The learning schedule follows the setting [126]. The base learning rate is set as $1e{-}3$, and is dropped to $1e{-}4$ and $1e{-}5$ at the 170th and 200th epochs, respectively. The training process is terminated within 210 epochs. The models are trained on 4 V100 GPUs and it takes around 60 (80) hours for HRNet-W32 (HRNet-W48).

Testing. The two-stage top-down paradigm similar as [19], [89], [126] is used: detect the person instance using a person detector, and then predict detection keypoints.

We use the same person detectors provided by SimpleBaseline[3] for both the `val` and `test-dev` sets. Following [19], [85], [126], we compute the heatmap by averaging the heatmaps of the original and flipped images. Each keypoint location is predicted by adjusting the highest heatvalue location with a quarter offset in the direction from the highest response to the second highest response.

Results on the `val` set. We report the results of our method and other state-of–the-art methods in Table 1. The network - HRNetV1-W32, trained from scratch with the input size 256×192, achieves an AP score 73.4, outperforming other methods with the same input size. (i) Compared to Hourglass [85], our network improves AP by 6.5 points, and the GFLOP of our network is much lower and less than half, while the numbers of parameters are similar and ours is slightly larger. (ii) Compared to CPN [19] w/o and w/ OHKM, our network, with slightly larger model size and slightly higher complexity, achieves 4.8 and 4.0 points gain, respectively. (iii) Compared to the previous best-performed method SimpleBaseline [126], our HRNetV1-W32 obtains significant improvements: 3.0 points gain for the backbone ResNet-50 with a similar model size and GFLOPs, and 1.4 points gain for the backbone ResNet-152 whose model size (#Params) and GFLOPs are twice as many as ours.

Our network can benefit from (i) training from the model pretrained on the ImageNet: The gain is 1.0 points for HRNetV1-W32; (ii) increasing the capacity by increasing the width: HRNetV1-W48 gets 0.7 and 0.5 points gain for the input sizes 256×192 and 384×288, respectively.

Considering the input size 384×288, our HRNetV1-W32 and HRNetV1-W48, get the 75.8 and 76.3 AP, which have 1.4 and 1.2 improvements compared to the input size 256×192. In comparison to SimpleBaseline [126] that uses ResNet-152 as the backbone, our HRNetV1-W32 and HRNetV1-W48 attain 1.5 and 2.0 points gain in terms of AP at 45% and 92.4% computational cost, respectively.

Results on the `test-dev` set. Table 2 reports the pose estimation performances of our approach and the existing state-of-the-art approaches. Our approach is significantly better than bottom-up approaches. On the other hand, our small network, HRNetV1-W32, achieves an AP of 74.9. It outperforms all the other top-down approaches, and is more efficient in terms of model size (#Params) and computation complexity (GFLOPs). Our big model, HRNetV1-W48, achieves the highest AP score 75.5. Compared to SimpleBaseline [126] with the same input size, our small and big networks receive 1.2 and 1.8 improvements, respectively. With the additional data from AI Challenger [123] for training, our single big network can obtain an AP of 77.0.

3. https://github.com/Microsoft/human-pose-estimation.pytorch

TABLE 3
Semantic segmentation results on Cityscapes `val` (single scale and no flipping). The GFLOPs is calculated on the input size 1024×2048. The small model HRNetV2-W40 with the smallest GFLOPs performs better than two representative contextual methods (Deeplab and PSPNet). Our approach combined with the recently-developed object contextual (OCR) representation scheme [139] gets further improvement. D-ResNet-101 = Dilated-ResNet-101.

	backbone	#param.	GFLOPs	mIoU
UNet++ [155]	ResNet-101	59.5M	748.5	75.5
Dilated-ResNet [40]	D-ResNet-101	52.1M	1661.6	75.7
DeepLabv3 [16]	D-ResNet-101	58.0M	1778.7	78.5
DeepLabv3+ [18]	D-Xception-71	43.5M	1444.6	79.6
PSPNet [148]	D-ResNet-101	65.9M	2017.6	79.7
HRNetV2	HRNetV2-W40	45.2M	493.2	80.2
HRNetV2	HRNetV2-W48	65.9M	696.2	81.1
HRNetV2 + OCR [139]	HRNetV2-W48	70.3M	1206.3	**81.6**

TABLE 4
Semantic segmentation results on Cityscapes `test`. We use HRNetV2-W48, whose parameter complexity and computation complexity are comparable to dilated-ResNet-101 based networks, for comparison. Our results are superior in terms of the four evaluation metrics. The result from the combination with OCR [139] is further improved. D-ResNet-101 = Dilated-ResNet-101.

	backbone	mIoU	iIoU cla.	IoU cat.	iIoU cat.
Model learned on the `train` set					
PSPNet [148]	D-ResNet-101	78.4	56.7	90.6	78.6
PSANet [149]	D-ResNet-101	78.6	-	-	-
PAN [64]	D-ResNet-101	78.6	-	-	-
AAF [54]	D-ResNet-101	79.1	-	-	-
HRNetV2	HRNetV2-W48	**80.4**	**59.2**	**91.5**	**80.8**
Model learned on the `train+val` set					
GridNet [30]	-	69.5	44.1	87.9	71.1
LRR-4x [33]	-	69.7	48.0	88.2	74.7
DeepLab [15]	D-ResNet-101	70.4	42.6	86.4	67.7
LC [66]	-	71.1	-	-	-
Piecewise [73]	VGG-16	71.6	51.7	87.3	74.1
FRRN [94]	-	71.8	45.5	88.9	75.1
RefineNet [72]	ResNet-101	73.6	47.2	87.9	70.6
PEARL [51]	D-ResNet-101	75.4	51.6	89.2	75.1
DSSPN [70]	D-ResNet-101	76.6	56.2	89.6	77.8
LKM [91]	ResNet-152	76.9	-	-	-
DUC-HDC [119]	-	77.6	53.6	90.1	75.2
SAC [143]	D-ResNet-101	78.1	-	-	-
DepthSeg [58]	D-ResNet-101	78.2	-	-	-
ResNet38 [125]	WResNet-38	78.4	59.1	90.9	78.1
BiSeNet [136]	ResNet-101	78.9	-	-	-
DFN [137]	ResNet-101	79.3	-	-	-
PSANet [149]	D-ResNet-101	80.1	-	-	-
PADNet [131]	D-ResNet-101	80.3	58.8	90.8	78.5
CFNet [142]	D-ResNet-101	79.6	-	-	-
Auto-DeepLab [77]	-	80.4	-	-	-
DenseASPP [148]	WDenseNet-161	80.6	59.1	90.9	78.1
SVCNet [27]	ResNet-101	81.0	-	-	-
ANN [158]	D-ResNet-101	81.3	-	-	-
CCNet [47]	D-ResNet-101	81.4	-	-	-
DANet [31]	D-ResNet-101	81.5	-	-	-
HRNetV2	HRNetV2-W48	81.6	**61.8**	**92.1**	**82.2**
HRNetV2 + OCR [139]	HRNetV2-W48	**82.5**	61.7	**92.1**	81.6

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.2983686, IEEE Transactions on Pattern Analysis and Machine Intelligence

Fig. 7. Qualitative segmentation examples from Cityscapes (left two), PASCAL-Context (middle two), and LIP (right two).

TABLE 5
Semantic segmentation results on PASCAL-Context. The methods are evaluated on 59 classes and 60 classes. Our approach performs the best for 60 classes, and performs worse for 59 classes than APCN [37] that developed a strong contextual method. Our approach, combined with OCR [139], achieves significant gain, and performs the best. D-ResNet-101 = Dilated-ResNet-101.

	backbone	mIoU (59)	mIoU (60)
FCN-8s [101]	VGG-16	-	35.1
BoxSup [24]	-	-	40.5
HO_CRF [2]	-	-	41.3
Piecewise [73]	VGG-16	-	43.3
DeepLab-v2 [15]	D-ResNet-101	-	45.7
RefineNet [72]	ResNet-152	-	47.3
UNet++ [155]	ResNet-101	47.7	-
PSPNet [148]	D-ResNet-101	47.8	-
Ding et al. [26]	ResNet-101	51.6	-
EncNet [141]	D-ResNet-101	52.6	-
DANet [31]	D-ResNet-101	52.6	-
ANN [158]	D-ResNet-101	52.8	-
SVCNet [27]	ResNet-101	53.2	-
CFNet [142]	D-ResNet-101	54.0	-
APCN [37]	D-ResNet-101	55.6	-
HRNetV2	HRNetV2-W48	54.0	48.3
HRNetV2 + OCR [139]	HRNetV2-W48	**56.2**	**50.1**

TABLE 6
Semantic segmentation results on LIP. Our method doesn't exploit any extra information, e.g., pose or edge. The overall performance of our approach is the best, and the OCR scheme [139] further improves the segmentation quality. D-ResNet-101 = Dilated-ResNet-101.

	backbone	extra.	pixel acc.	avg. acc.	mIoU
Attention+SSL [34]	VGG16	Pose	84.36	54.94	44.73
DeepLabV3+ [18]	D-ResNet-101	-	84.09	55.62	44.80
MMAN [82]	D-ResNet-101	-	-	-	46.81
SS-NAN [150]	ResNet-101	Pose	87.59	56.03	47.92
MuLA [86]	Hourglass	Pose	**88.50**	60.50	49.30
JPPNet [69]	D-ResNet-101	Pose	86.39	62.32	51.37
CE2P [80]	D-ResNet-101	Edge	87.37	63.20	53.10
HRNetV2	HRNetV2-W48	N	88.21	67.43	55.90
HRNetV2 + OCR [139]	HRNetV2-W48	N	88.24	**67.84**	**56.48**

5 Semantic Segmentation

Semantic segmentation is a problem of assigning a class label to each pixel. Some example results by our approach are given in Figure 7. We feed the input image to the HRNetV2 (Figure 4 (b)) and then pass the resulting $15C$-dimensional representation at each position to a linear classifier with the softmax loss to predict the segmentation maps. The segmentation maps are upsampled (4 times) to the input size by bilinear upsampling for both training and testing. We report the results over two scene parsing datasets, PASCAL-Context [83] and Cityscapes [23], and a human parsing dataset, LIP [34]. The mean of class-wise intersection over union (mIoU) is adopted as the evaluation metric.

Cityscapes. The Cityscapes dataset [23] contains $5,000$ high quality pixel-level finely annotated scene images. The finely-annotated images are divided into $2,975/500/1,525$ images for training, validation and testing. There are 30 classes, and 19 classes among them are used for evaluation. In addition to the mean of class-wise intersection over union (mIoU), we report other three scores on the test set: IoU category (cat.), iIoU class (cla.) and iIoU category (cat.).

We follow the same training protocol [148], [149]. The data are augmented by random cropping (from 1024×2048 to 512×1024), random scaling in the range of $[0.5, 2]$, and random horizontal flipping. We use the SGD optimizer with the base learning rate of 0.01, the momentum of 0.9 and the weight decay of 0.0005. The poly learning rate policy with the power of 0.9 is used for dropping the learning rate. All the models are trained for $120K$ iterations with the batch size of 12 on 4 GPUs and syncBN.

Table 3 provides the comparison with several representative methods on the Cityscapes `val` set in terms of parameter and computation complexity and mIoU class. (i) HRNetV2-W40 (40 indicates the width of the high-resolution convolution), with similar model size to DeepLabv3+ and much lower computation complexity, gets better performance: 4.7 points gain over UNet++, 1.7 points gain over DeepLabv3 and about 0.5 points gain over PSPNet, DeepLabv3+. (ii) HRNetV2-W48, with similar model size to PSPNet and much lower computation complexity, achieves much significant improvement: 5.6 points gain over UNet++, 2.6 points gain over DeepLabv3 and about 1.4 points gain over PSPNet, DeepLabv3+. In the following comparisons, we adopt HRNetV2-W48 that is pretrained on ImageNet and has similar model size as most Dilated-ResNet-101 based methods.

Table 4 provides the comparison of our method with state-of-the-art methods on the Cityscapes `test` set. All the results are with six scales and flipping. Two cases w/o using coarse data are evaluated: One is about the model learned on the `train` set, and the other is about the model learned on the `train+val` set. In both cases, HRNetV2-W48 achieves the superior performance.

PASCAL-Context. The PASCAL-Context dataset [83] includes $4,998$ scene images for training and $5,105$ images for testing with 59 semantic labels and 1 background label.

The data augmentation and learning rate policy are the same as Cityscapes. Following the widely-used training strategy [26], [141], we resize the images to 480×480 and set the initial learning rate to 0.004 and weight decay to 0.0001. The batch size is 16 and the number of iterations is $60K$.

We follow the standard testing procedure [26], [141]. The image is resized to 480×480 and then fed into our network. The resulting 480×480 label maps are then resized to the original image size. We evaluate the performance of our approach and other approaches using six scales and flipping.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.2983686, IEEE Transactions on Pattern Analysis and Machine Intelligence

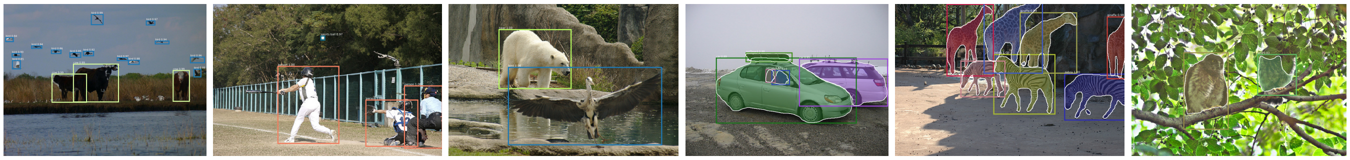


Fig. 8. Qualitative examples for COCO object detection (left three) and instance segmentation (right three).

TABLE 7
GFLOPs and #parameters for COCO object detection. The numbers are obtained with the input size 800×1200 and if applicable 512 proposals fed into R-CNN except the numbers for CenterNet are obtained with the input size 511×511. R-x = ResNet-x-FPN, X-101 = ResNeXt-101-64×4d, H-x = HRNetV2p-Wx, and HG-52 = Hourglass-52.

	Faster R-CNN [39]						Cascade R-CNN [10]						FCOS [112]				CenterNet [28]			
	R-50	H-18	R-101	H-32	X-101	H-48	R-50	H-18	R-101	H-32	X-101	H-48	R-50	H-18	R-101	H-32	HG-52	H-48	HG-104	H-64
#param. (M)	39.8	26.2	57.8	45.0	94.9	79.4	69.4	55.1	88.4	74.9	127.3	111.0	32.0	17.5	51.0	37.3	104.8	73.6	210.1	127.7
GFLOPs	172.3	159.1	239.4	245.3	381.8	399.1	226.2	207.8	298.7	300.8	448.3	466.5	190.0	180.3	261.2	273.3	227.0	217.1	388.4	318.5
	Cascade Mask R-CNN [10]						Hybrid Task Cascade [12]						Mask R-CNN [39]							
	R-50	H-18	R-101	H-32	X-101	H-48	R-50	H-18	R-101	H-32	X-101	H-48	R-50	H-18	R-101	H-32				
#param. (M)	77.3	63.1	96.3	82.9	135.2	118.9	80.3	66.1	99.3	85.9	138.2	121.9	44.4	30.1	63.4	49.9				
GFLOPs	431.7	413.1	504.1	506.2	653.7	671.9	476.9	458.3	549.2	551.4	698.9	717.0	266.5	247.9	338.8	341.0				

TABLE 8
Object detection results on COCO `val` in the Faster R-CNN and Cascade R-CNN frameworks. LS = learning schedule. $1\times = 12e$, $2\times = 24e$. Our approach performs better than ResNet and ResNeXt. Our approach gets more significant improvement for $2\times$ than $1\times$ and for small objects (AP$_S$) than medium (AP$_M$) and large objects (AP$_L$).

backbone	LS	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
Faster R-CNN [74]							
ResNet-50-FPN	1×	36.7	58.3	39.9	20.9	39.8	47.9
HRNetV2p-W18	1×	36.2	57.3	39.3	20.7	39.0	46.8
ResNet-50-FPN	2×	37.6	58.7	41.3	21.4	40.8	49.7
HRNetV2p-W18	2×	38.0	58.9	41.5	22.6	40.8	49.6
ResNet-101-FPN	1×	39.2	61.1	43.0	22.3	42.9	50.9
HRNetV2p-W32	1×	39.6	61.0	43.3	23.7	42.5	50.5
ResNet-101-FPN	2×	39.8	61.4	43.4	22.9	43.6	52.4
HRNetV2p-W32	2×	40.9	61.8	44.8	24.4	43.7	53.3
X-101-64×4d-FPN	1×	41.3	63.4	45.2	24.5	45.8	53.3
HRNetV2p-W48	1×	41.3	62.8	45.1	25.1	44.5	52.9
X-101-64×4d-FPN	2×	40.8	62.1	44.6	23.2	44.5	53.7
HRNetV2p-W48	2×	41.8	62.8	45.9	25.0	44.7	54.6
Cascade R-CNN [10]							
ResNet-50-FPN	$20e$	41.1	59.1	44.8	22.5	44.4	54.9
HRNetV2p-W18	$20e$	41.3	59.2	44.9	23.7	44.2	54.1
ResNet-101-FPN	$20e$	42.5	60.7	46.3	23.7	46.1	56.9
HRNetV2p-W32	$20e$	43.7	61.7	47.7	25.6	46.5	57.4
X-101-64×4d-FPN	$20e$	44.7	63.1	49.0	25.8	48.3	58.8
HRNetV2p-W48	$20e$	44.6	62.7	48.7	26.3	48.1	58.5

Table 5 provides the comparison of our method with state-of-the-art methods. There are two kinds of evaluation schemes: mIoU over 59 classes and 60 classes (59 classes + background). In both cases, HRNetV2-W48 achieves state-of-the-art results except that the result from [37] is higher than ours without using the OCR scheme [139].

LIP. The LIP dataset [34] contains $50,462$ elaborately annotated human images, which are divided into $30,462$ training images, and $10,000$ validation images. The methods are evaluated on 20 categories (19 human part labels and 1 background label). Following the standard training and testing settings [80], the images are resized to 473×473 and the performance is evaluated on the average of the segmentation maps of the original and flipped images.

The data augmentation and learning rate policy are the same as Cityscapes. The training strategy follows the recent setting [80]. We set the initial learning rate to 0.007 and the momentum to 0.9 and the weight decay to 0.0005. The batch

TABLE 9
Object detection results on COCO `val` in the FCOS and CenterNet frameworks. The results are obtained using the implementations provided by the authors. Our approach performs superiorly to ResNet and Hourglass for similar parameter and computation complexity. Our HRNetV2p-W64 performs slightly worse than Hourglass-104, and the reason is that Hourglass-104 is much more heavier than HRNetV2p-W64. See Table 7 for #parameters and GFLOPs.

backbone	LS	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
FCOS [112]							
ResNet-50-FPN	2×	37.1	55.9	39.8	21.3	41.0	47.8
HRNetV2p-W18	2×	37.7	55.3	40.2	22.0	40.8	48.8
ResNet-101-FPN	2×	41.4	60.3	44.8	25.0	45.6	53.1
HRNetV2p-W32	2×	41.9	60.3	45.0	25.1	45.6	53.2
CenterNet [28]							
Hourglass-52	-	41.3	59.2	43.9	23.6	43.8	55.8
HRNetV2p-W48	-	43.4	61.8	45.6	23.8	47.1	59.3
Hourglass-104	-	44.8	62.4	48.2	25.9	48.9	58.8
HRNetV2p-W64	-	44.0	62.5	47.3	23.9	48.2	60.2

size is 40 and the number of iterations is 110K.

Table 6 provides the comparison of our method with state-of-the-art methods. The overall performance of HRNetV2-W48 performs the best with fewer parameters and lighter computation cost. We also would like to mention that our networks do not use extra information such as pose or edge.

6 COCO Object Detection

We perform the evaluation on the MS COCO 2017 detection dataset, which contains about 118k images for training, 5k for validation (`val`) and $\sim$ 20k testing without provided annotations (`test-dev`). The standard COCO-style evaluation is adopted. Some example results by our approach are given in Figure 8.

We apply our multi-level representations (HRNetV2p)[4], shown in Figure 4 (c), for object detection. The data is augmented by standard horizontal flipping. The input images are resized such that the shorter edge is 800 pixels [74]. Inference is performed on a single image scale.

We compare our HRNet with the standard models: ResNet [40] and ResNeXt [129]. We evaluate the detection

4. Same as FPN [75], we also use 5 levels.

TABLE 10
Object detection results on COCO val in the Mask R-CNN and its extended frameworks. The overall performance of our approach is superior to ResNet except that HRNetV2p-W18 sometimes performs worse than ResNet-50. Similar to detection (bbox), the improvement for small objects (AP_S) in terms of mask is also more significant than medium (AP_M) and large objects (AP_L). The results are obtained from MMDetection [13].

backbone	LS	mask				bbox			
		AP	AP_S	AP_M	AP_L	AP	AP_S	AP_M	AP_L
Mask R-CNN [39]									
ResNet-50-FPN	1×	34.2	15.7	36.8	50.2	37.8	22.1	40.9	49.3
HRNetV2p-W18	1×	33.8	15.6	35.6	49.8	37.1	21.9	39.5	47.9
ResNet-50-FPN	2×	35.0	16.0	37.5	52.0	38.6	21.7	41.6	50.9
HRNetV2p-W18	2×	35.3	16.9	37.5	51.8	39.2	23.7	41.7	51.0
ResNet-101-FPN	1×	36.1	16.2	39.0	53.0	40.0	22.6	43.4	52.3
HRNetV2p-W32	1×	36.7	17.3	39.0	53.0	40.9	24.5	43.9	52.2
ResNet-101-FPN	2×	36.7	17.0	39.5	54.8	41.0	23.4	44.4	53.9
HRNetV2p-W32	2×	37.6	17.8	40.0	55.0	42.3	25.0	45.4	54.9
Cascade Mask R-CNN [10]									
ResNet-50-FPN	$20e$	36.6	19.0	37.4	50.7	42.3	23.7	45.7	56.4
HRNetV2p-W18	$20e$	36.4	17.0	38.6	52.9	41.9	23.8	44.9	55.0
ResNet-101-FPN	$20e$	37.6	19.7	40.8	52.4	43.3	24.4	46.9	58.0
HRNetV2p-W32	$20e$	38.5	18.9	41.1	56.1	44.5	26.1	47.9	58.5
X-101-64×4d-FPN	$20e$	39.4	20.8	42.7	54.1	45.7	26.2	49.6	60.0
HRNetV2p-W48	$20e$	39.5	19.7	41.8	56.9	46.0	27.5	48.9	60.1
Hybrid Task Cascade [12]									
ResNet-50-FPN	$20e$	38.1	20.3	41.1	52.8	43.2	24.9	46.4	57.8
HRNetV2p-W18	$20e$	37.9	18.8	39.9	55.2	43.1	26.6	46.0	56.9
ResNet-101-FPN	$20e$	39.4	21.4	42.4	54.4	44.9	26.4	48.3	59.9
HRNetV2p-W32	$20e$	39.6	19.1	42.0	57.9	45.3	27.0	48.4	59.5
X-101-64×4d-FPN	$20e$	40.8	22.7	44.2	56.3	46.9	28.0	50.7	62.1
HRNetV2p-W48	$20e$	40.7	19.7	43.4	59.3	46.8	28.0	50.2	61.7
X-101-64×4d-FPN	$28e$	40.7	20.0	44.1	59.9	46.8	27.5	51.0	61.7
HRNetV2p-W48	$28e$	41.0	20.8	43.9	59.9	47.0	28.8	50.3	62.2

performance on COCO val. under two anchor-based frameworks: Faster R-CNN [96] and Cascade R-CNN [9], and two recently-developed anchor-free frameworks: FCOS [112] and CenterNet [28]. We train the Faster R-CNN and Cascade R-CNN models for both our HRNetV2p and the ResNet on the public MMDetection platform [13] with the provided training setup, except that we use the learning rate schedule suggested in [38] for 2×, and FCOS [112] and CenterNet [28] from the implementations provided by the authors. Table 7 summarizes #parameters and GFLOPs. Table 8 and Table 9 report detection scores.

We also evaluate the performance of joint detection and instance segmentation, under three frameworks: Mask R-CNN [39], Cascade Mask R-CNN [10], and Hybrid Task Cascade [12]. The results are obtained on the public MMDetection platform [13] and are in Table 10.

There are several observations. On the one hand, as shown in Tables 8 and 9, the overall object detection performance of HRNetV2 is better than ResNet under similar model size and computation complexity. In some cases, for 1×, HRNetV2p-W18 performs worse than ResNet-50-FPN, which might come from insufficient optimization iterations. On the other hand, as shown in Table 10, the overall object detection and instance segmentation performance is better than ResNet and ResNeXt. In particular, under the Hybrid Task Cascade framework, the HRNet performs slightly worse than ResNeXt-101-64×4d-FPN for $20e$, but better for $28e$. This implies that our HRNet benefits more from longer training.

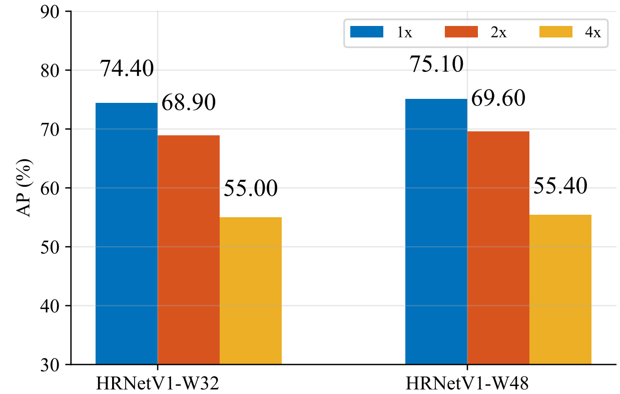


Fig. 9. Ablation study about the resolutions of the representations for human pose estimation. 1×, 2×, 4× correspond to the representations of the high, medium, low resolutions, respectively. The results imply that higher resolution improves the performance.

Table 11 reports the comparison of our network to state-of-the-art single-model object detectors on COCO test-dev without using multi-scale training and multi-scale testing that are done in [67], [79], [90], [95], [103], [104]. In the Faster R-CNN framework, our networks perform better than ResNets with similar parameter and computation complexity: HRNetV2p-W32 vs. ResNet-101-FPN, HRNetV2p-W40 vs. ResNet-152-FPN, HRNetV2p-W48 vs. X-101-64 × 4d-FPN. In the Cascade R-CNN and CenterNet framework, our HRNetV2 also performs better. In the Cascade Mask R-CNN and Hybrid Task Cascade frameworks, the HRNet gets the overall better performance.

7 Ablation Study

We perform the ablation study for the components in HRNet over two tasks: human pose estimation on COCO validation and semantic segmentation on Cityscapes validation. We mainly use HRNetV1-W32 for human pose estimation, and HRNetV2-W48 for semantic segmentation. All results of pose estimation are obtained over the input size 256 × 192. We also present the results for comparing HRNetV1 and HRNetV2.

Representations of different resolutions. We study how the representation resolution affects the pose estimation performance by checking the quality of the heatmap estimated from the feature maps of each resolution from high to low.

We train two HRNetV1 networks initialized by the model pretrained for the ImageNet classification. Our network outputs four response maps from high-to-low resolutions. The quality of heatmap prediction over the lowest-resolution response map is too low and the AP score is below 10 points. The AP scores over the other three maps are reported in Figure 9. The comparison implies that the resolution does impact the keypoint prediction quality.

Repeated multi-resolution fusion. We empirically analyze the effect of the repeated multi-resolution fusion. We study three variants of our network. (a) W/o intermediate fusion units (1 fusion): There is no fusion between multi-resolution streams except the final fusion unit. (b) W/ across-stage fusion units (3 fusions): There is no fusion between parallel streams within each stage. (c) W/ both across-stage

TABLE 11
Comparison with the state-of-the-art single-model object detectors on COCO `test-dev` with BN parameters fixed and without mutli-scale training and testing. * means that the result is from the original paper [9]. GFLOPs and #parameters of the models are given in Table 7. The observations are similar to those on COCO `val`, and show that the HRNet performs better than ResNet and ResNeXt under state-of-the-art object detection and instance segmentation frameworks.

	backbone	size	LS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MLKP [117]	VGG16	-	-	28.6	52.4	31.6	10.8	33.4	45.1
STDN [153]	DenseNet-169	513	-	31.8	51.0	33.6	14.4	36.1	43.4
DES [146]	VGG16	512	-	32.8	53.2	34.6	13.9	36.0	47.6
CoupleNet [157]	ResNet-101	-	-	33.1	53.5	35.4	11.6	36.3	50.1
DeNet [114]	ResNet-101	512	-	33.8	53.4	36.1	12.3	36.1	50.8
RFBNet [78]	VGG16	512	-	34.4	55.7	36.4	17.6	37.0	47.6
DFPR [59]	ResNet-101	512	1×	34.6	54.3	37.3	-	-	-
PFPNet [55]	VGG16	512	-	35.2	57.6	37.9	18.7	38.6	45.9
RefineDet [144]	ResNet-101	512	-	36.4	57.5	39.5	16.6	39.9	51.4
Relation Net [42]	ResNet-101	600	-	39.0	58.6	42.9	-	-	-
C-FRCNN [20]	ResNet-101	800	1×	39.0	59.7	42.8	19.4	42.4	53.0
RetinaNet [75]	ResNet-101-FPN	800	1.5×	39.1	59.1	42.3	21.8	42.7	50.2
Deep Regionlets [132]	ResNet-101	800	1.5×	39.3	59.8	-	21.7	43.7	50.9
FitnessNMS [115]	ResNet-101	768	-	39.5	58.0	42.6	18.9	43.5	54.1
DetNet [68]	DetNet59-FPN	800	2×	40.3	62.1	43.8	23.6	42.6	50.0
CornerNet [62]	Hourglass-104	511	-	40.5	56.5	43.1	19.4	42.7	53.9
M2Det [152]	VGG16	800	$\sim 10\times$	41.0	59.7	45.0	22.1	46.5	53.8
Faster R-CNN [74]	ResNet-101-FPN	800	1×	39.3	61.3	42.7	22.1	42.1	49.7
Faster R-CNN	HRNetV2p-W32	800	1×	39.5	61.2	43.0	23.3	41.7	49.1
Faster R-CNN [74]	ResNet-101-FPN	800	2×	40.3	61.8	43.9	22.6	43.1	51.0
Faster R-CNN	HRNetV2p-W32	800	2×	41.1	62.3	44.9	24.0	43.1	51.4
Faster R-CNN [74]	ResNet-152-FPN	800	2×	40.6	62.1	44.3	22.6	43.4	52.0
Faster R-CNN	HRNetV2p-W40	800	2×	42.1	63.2	46.1	24.6	44.5	52.6
Faster R-CNN [13]	X-101-64×4d-FPN	800	2×	41.1	62.8	44.8	23.5	44.1	52.3
Faster R-CNN	HRNetV2p-W48	800	2×	42.4	63.6	46.4	24.9	44.6	53.0
Cascade R-CNN [9]*	ResNet-101-FPN	800	$\sim 1.6\times$	42.8	62.1	46.3	23.7	45.5	55.2
Cascade R-CNN	ResNet-101-FPN	800	$\sim 1.6\times$	43.1	61.7	46.7	24.1	45.9	55.0
Cascade R-CNN	HRNetV2p-W32	800	$\sim 1.6\times$	43.7	62.0	47.4	25.5	46.0	55.3
Cascade R-CNN	X-101-64 × 4d-FPN	800	$\sim 1.6\times$	44.9	63.7	48.9	25.9	47.7	57.1
Cascade R-CNN	HRNetV2p-W48	800	$\sim 1.6\times$	44.8	63.1	48.6	26.0	47.3	56.3
FCOS [112]	ResNet-50-FPN	800	2×	37.3	56.4	39.7	20.4	39.6	47.5
FCOS	HRNetV2p-W18	800	2×	37.8	56.1	40.4	21.6	39.8	47.4
FCOS [112]	ResNet-101-FPN	800	2×	39.2	58.8	41.6	21.8	41.7	50.0
FCOS	HRNetV2p-W32	800	2×	40.5	59.3	43.3	23.4	42.6	51.0
CenterNet [28]	Hourglass-52	511	–	41.6	59.4	44.2	22.5	43.1	54.1
CenterNet	HRNetV2-W48	511	–	43.5	62.1	46.5	22.2	46.5	57.8
Cascade Mask R-CNN [10]	ResNet-101-FPN	800	$\sim 1.6\times$	44.0	62.3	47.9	24.3	46.9	56.7
Cascade Mask R-CNN	HRNetV2p-W32	800	$\sim 1.6\times$	44.7	62.5	48.6	25.8	47.1	56.3
Cascade Mask R-CNN [10]	X-101-64 × 4d-FPN	800	$\sim 1.6\times$	45.9	64.5	50.0	26.6	49.0	58.6
Cascade Mask R-CNN	HRNetV2p-W48	800	$\sim 1.6\times$	46.1	64.0	50.3	27.1	48.6	58.3
Hybrid Task Cascade [12]	ResNet-101-FPN	800	$\sim 1.6\times$	45.1	64.3	49.0	25.2	48.0	58.2
Hybrid Task Cascade	HRNetV2p-W32	800	$\sim 1.6\times$	45.6	64.1	49.4	26.7	47.7	58.0
Hybrid Task Cascade [12]	X-101-64 × 4d-FPN	800	$\sim 1.6\times$	47.2	66.5	51.4	27.7	50.1	60.3
Hybrid Task Cascade	HRNetV2p-W48	800	$\sim 1.6\times$	47.0	65.8	51.0	27.9	49.4	59.7
Hybrid Task Cascade [12]	X-101-64 × 4d-FPN	800	$\sim 2.3\times$	47.2	66.6	51.3	27.5	50.1	60.6
Hybrid Task Cascade	HRNetV2p-W48	800	$\sim 2.3\times$	47.3	65.9	51.2	28.0	49.7	59.8

TABLE 12
Ablation study for multi-resolution fusion units on COCO `val` human pose estimation (AP) and Cityscapes `val` semantic segmentation (mIoU). Final = final fusion immediately before representation head, Across = intermediate fusions across stages, Within = intermediate fusions within stages. We can see that the three fusions are beneficial for both human pose estimation and semantic segmentation.

Method	Final	Across	Within	Pose (AP)	Segmentation (mIoU)
(a)	✓			70.8	74.8
(b)	✓	✓		71.9	75.4
(c)	✓	✓	✓	73.4	76.4

and within-stage fusion units (totally 8 fusions): This is our proposed method. All the networks are trained from scratch. The results on COCO human pose estimation and Cityscapes semantic segmentation (validation) given in Table 12 show that the multi-resolution fusion unit is helpful and more fusions lead to better performance.

We also study other possible choices for the fusion design: (i) use bilinear downsample to replace strided convolutions, and (ii) use the multiplication operation to replace the sum operation. In the former case, the COCO pose estimation AP score and the Cityscapes segmentation mIoU score are reduced to 72.6 and 74.2. The reason is that downsampling reduces the volume size (width × height × #channels) of the representation maps, and strided convolutions learn better volume size reduction than bilinear downsampling. In the later case, the results are much worse: 54.7 and 66.0, respectively. The possible reason might be that multiplication increases the training difficulty as pointed in [120].

Resolution maintenance. We study the performance of a variant of the HRNet: all the four high-to-low resolution streams are added at the beginning and the depths of the four streams are the same; the fusion schemes are the same to ours. Both the HRNets and the variants (with similar #Params and GFLOPs) are trained from scratch.

The human pose estimation performance (AP) on COCO `val` for the variant is 72.5, which is lower than 73.4 for HRNetV1-W32. The segmentation performance (mIoU) on Cityscapes `val` for the variant is 75.7, which is lower than 76.4 for HRNetV2-W48. We believe that the reason is that the low-level features extracted from the early stages over the low-resolution streams are less helpful. In addition, another simple variant, only the high-resolution stream of similar #parameters and GFLOPs without low-resolution parallel streams shows much lower performance on COCO and Cityscapes.

V1 vs. V2. We compare HRNetV2 and HRNetV2p, to HRNetV1 on pose estimation, semantic segmentation and COCO object detection. For human pose estimation, the performance is similar. For example, HRNetV2-W32 (w/o ImageNet pretraining) achieves the AP score 73.6, which is slightly higher than 73.4 HRNetV1-W32.

The segmentation and object detection results, given in Figure 10 (a) and Figure 10 (b), imply that HRNetV2 outperforms HRNetV1 significantly, except that the gain is minor in the large model case (1×) in segmentation for Cityscapes. We also test a variant (denoted by HRNetV1h), which is built by appending a 1×1 convolution to align the dimension of the output high-resolution representation with the dimension of HRNetV2. The results in Figure 10 (a) and Figure 10 (b) show that the variant achieves slight improvement to HRNetV1, implying that aggregating the representations from low-resolution parallel convolutions in our HRNetV2 is essential for improving the capability.

8 Conclusions

In this paper, we present a high-resolution network for visual recognition problems. There are three fundamental differences from existing low-resolution classification networks and high-resolution representation learning networks: (i) Connect high and low resolution convolutions in parallel other than in series; (ii) Maintain high resolution through the whole process instead of recovering high resolution from low resolution; and (iii) Fuse multi-resolution representations repeatedly, rendering rich high-resolution representations with strong position sensitivity.

The superior results on a wide range of visual recognition problems suggest that our proposed HRNet is a stronger backbone for computer vision problems. Our research also encourages more research efforts for designing network architectures directly for specific vision problems

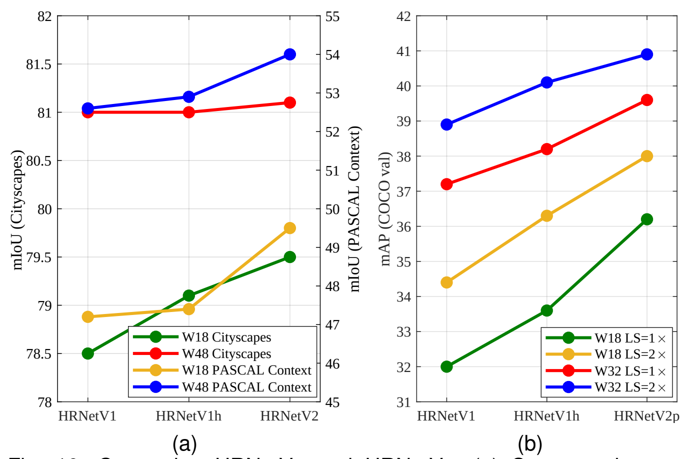


Fig. 10. Comparing HRNetV1 and HRNetV2. (a) Segmentation on Cityscapes `val` and PASCAL-Context for comparing HRNetV1 and its variant HRNetV1h, and HRNetV2 (single scale and no flipping). (b) Object detection on COCO `val` for comparing HRNetV1 and its variant HRNetV1h, and HRNetV2p (LS = learning schedule). We can see that HRNetV2 is superior to HRNetV1 for both semantic segmentation and object detection.

other than extending, remediating or repairing representations learned from low-resolution networks (e.g., ResNet or VGGNet).

Discussions. There is a possible misunderstanding: the memory cost of the HRNet is larger as the resolution is higher. In fact, the memory cost of the HRNet for all the three applications, human pose estimation, semantic segmentation and object detection, is comparable to state-of-the-arts except that the training memory cost in object detection is a little larger.

In addition, we summarize the runtime cost comparison on the PyTorch 1.0 platform. The training and inference time cost of the HRNet is comparable to previous state-of-the-arts except that (1) the inference time of the HRNet for segmentation is much smaller and (2) the training time of the HRNet for pose estimation is a little larger, but the cost on the MXNet 1.5.1 platform, which supports static graph inference, is similar as SimpleBaseline. We would like to highlight that for semantic segmentation the inference cost is significantly smaller than PSPNet and DeepLabv3. Table 13 summarizes the memory and time cost comparisons [5].

Future and followup works. We will study the combination of the HRNet with other techniques for semantic segmentation and instance segmentation. Currently, we have results (mIoU), which are depicted in Tables 3 4 5 6, by combining the HRNet with the object-contextual representation (OCR) scheme [139] [6], a variant of object context [45], [140]. We will conduct the study by further increasing the resolution of the representation, e.g., to $\frac{1}{2}$ or even a full resolution.

The applications of the HRNet are not limited to the above that we have done, and are suitable to other position-sensitive vision applications, such as facial landmark de-

5. The detailed comparisons are given in the supplementary file.
6. We empirically observed that the HRNet combined with ASPP [16] or PPM [148] did not get a performance improvement on Cityscape, but got a slight improvement on PASCAL-Context and LIP.

TABLE 13
Memory and time cost comparisons for pose estimation, semantic segmentation and object detection (under the Faster R-CNN framework) on PyTorch 1.0 in terms of training/inference memory and training/inference time. We also report inference time (in ()) for pose estimation on MXNet $1.5.1$, which supports static graph inference that mutli-branch convolutions used in the HRNet benefits from. The numbers for training are obtained on a machine with 4 V100 GPU cards. During training, the input sizes are 256×192, 512×1024, and 800×1333, and the batch sizes are 128, 8 and 8 for pose estimation, segmentation and detection respectively. The numbers for inference are obtained on a single V100 GPU card. The input sizes are 256×192, 1024×2048, and 800×1333, respectively. The score means AP for pose estimation on COCO `val` (Table 1) and detection on COCO `val` (Table 8) , and mIoU for cityscapes segmentation (Table 3). Several observations are highlighted. Memory: The HRNet consumes similar memory for both training and inference except that it consumes smaller memory for training in human pose estimation. Time: The training and inference time cost of the HRNet is comparable to previous state-of-the-arts except that the inference time of the HRNet for segmentation is much smaller. SB-ResNet-152 = SimpleBaseline with the backbone of ResNet-152. PSPNet and DeepLabV3 use dilated ResNet-101 as the backbone (Table 3).

	Pose estimation		Segmentation			Detection			
	SB-ResNet-152	HRNetV1-W48	PSPNet	DeepLabV3	HRNetV2-W48	ResNet-101	ResNeXt-101	HRNetV2p-W32	HRNetV2p-W48
training memory	14.8G	7.3G	14.4G	13.3G	13.9G	5.4G	9.5G	8.5G	11.3G
inference memory/image	0.29G	0.27G	1.60G	1.15G	1.79G	0.62G	0.77G	0.51G	0.79G
training second/iteration	1.085	1.231	0.837	0.850	0.692	0.550	1.183	0.690	0.965
inference second/image	0.030 (0.012)	0.058 (0.017)	0.397	0.411	0.150	0.087	0.144	0.101	0.116
score	72.0	75.1	79.7	78.5	81.1	39.8	40.8	40.9	41.8

tection[7], super-resolution, optical flow estimation, depth estimation, and so on. There are already followup works, e.g., image stylization [65], inpainting [36], image enhancement [48], image dehazing [1], temporal pose estimation [4], and drone object detection [156].

It is reported in [21] that a slightly-modified HRNet combined with ASPP achieved the best performance for Mapillary panoptic segmentation in the single model case. In the COCO + Mapillary Joint Recognition Challenge Workshop at ICCV 2019, the COCO DensePose challenge winner and almost all the COCO keypoint detection challenge participants adopted the HRNet. The OpenImage instance segmentation challenge winner (ICCV 2019) also used the HRNet.

References

[1] C. O. Ancuti, C. Ancuti, R. Timofte, L. Van Gool, L. Zhang, and M.-H. Yang. Ntire 2019 image dehazing challenge report. In *CVPR*. 12

[2] A. Arnab, S. Jayasumana, S. Zheng, and P. H. S. Torr. Higher order conditional random fields in deep neural networks. In *ECCV*, pages 524–540, 2016. 7

[3] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(12):2481–2495, 2017. 1, 2

[4] G. Bertasius, C. Feichtenhofer, D. Tran, J. Shi, and L. Torresani. Learning temporal pose estimation from sparsely-labeled videos. *CoRR*, abs/1906.04016, 2019. 12

[5] A. Bulat and G. Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *ECCV*, pages 717–732, 2016. 2

[6] A. Bulat and G. Tzimiropoulos. Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. In *ICCV*, pages 3726–3734, 2017. 2

[7] A. Bulat and G. Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230, 000 3d facial landmarks). In *ICCV*, pages 1021–1030, 2017. 2

[8] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *ECCV*. 2

[9] Z. Cai and N. Vasconcelos. Cascade R-CNN: delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018. 2, 9, 10

[10] Z. Cai and N. Vasconcelos. Cascade R-CNN: high quality object detection and instance segmentation. *CoRR*, abs/1906.09756, 2019. 8, 9, 10

7. We provide the facial landmark detection results in the supplementary file.

[11] Z. Cao, T. Simon, S. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, pages 1302–1310, 2017. 5

[12] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, C. C. Loy, and D. Lin. Hybrid task cascade for instance segmentation. *CoRR*, abs/1901.07518, 2019. 2, 8, 9, 10

[13] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin. Mmdetection: Open mmlab detection toolbox and benchmark. *CoRR*, abs/1906.07155, 2019. 9, 10

[14] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *CoRR*, abs/1412.7062, 2014. 2

[15] 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.*, 40(4):834–848, 2018. 1, 2, 6, 7

[16] L. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017. 6, 11

[17] L. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, pages 3640–3649, 2016. 2

[18] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 833–851, 2018. 2, 6, 7

[19] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun. Cascaded pyramid network for multi-person pose estimation. *CoRR*, abs/1711.07319, 2017. 2, 5, 6

[20] Z. Chen, S. Huang, and D. Tao. Context refinement for object detection. In *ECCV*, pages 74–89, 2018. 10

[21] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L. Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. *CoRR*, abs/1911.10194, 2019. 12

[22] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang. Multi-context attention for human pose estimation. In *CVPR*, pages 5669–5678, 2017. 2

[23] 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*, pages 3213–3223, 2016. 7

[24] J. Dai, K. He, and J. Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, pages 1635–1643, 2015. 7

[25] J. Deng, G. Trigeorgis, Y. Zhou, and S. Zafeiriou. Joint multi-view face alignment in the wild. *CoRR*, abs/1708.06023, 2017. 2

[26] H. Ding, X. Jiang, B. Shuai, A. Q. Liu, and G. Wang. Context contrasted feature and gated multi-scale aggregation for scene segmentation. In *CVPR*, pages 2393–2402, 2018. 7

[27] H. Ding, X. Jiang, B. Shuai, A. Q. Liu, and G. Wang. Semantic correlation promoted shape-variant context for segmentation. In *CVPR*, June 2019. 6, 7

[28] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian. Centernet:

Keypoint triplets for object detection. *CoRR*, abs/1904.08189, 2019. 2, 8, 9, 10

[29] H. Fang, S. Xie, Y. Tai, and C. Lu. RMPE: regional multi-person pose estimation. In *ICCV*, pages 2353–2362, 2017. 5

[30] D. Fourure, R. Emonet, É. Fromont, D. Muselet, A. Trémeau, and C. Wolf. Residual conv-deconv grid network for semantic segmentation. In *BMVC*, 2017. 2, 6

[31] J. Fu, J. Liu, H. Tian, Z. Fang, and H. Lu. Dual attention network for scene segmentation. *CoRR*, abs/1809.02983, 2018. 6, 7

[32] J. Fu, J. Liu, Y. Wang, and H. Lu. Stacked deconvolutional network for semantic segmentation. *CoRR*, abs/1708.04943, 2017. 2

[33] G. Ghiasi and C. C. Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *ECCV*, pages 519–534, 2016. 6

[34] K. Gong, X. Liang, X. Shen, and L. Lin. Look into person: Self-supervised structure-sensitive learning and A new benchmark for human parsing. *CoRR*, abs/1703.05446, 2017. 7, 8

[35] J. Guo, J. Deng, N. Xue, and S. Zafeiriou. Stacked dense u-nets with dual transformers for robust face alignment. In *BMVC*, page 44, 2018. 2

[36] Z. Guo, Z. Chen, T. Yu, J. Chen, and S. Liu. Progressive image inpainting with full-resolution residual network. *CoRR*, abs/1907.10478, 2019. 12

[37] J. He, Z. Deng, L. Zhou, Y. Wang, and Y. Qiao. Adaptive pyramid context network for semantic segmentation. In *CVPR*, June 2019. 7, 8

[38] K. He, R. B. Girshick, and P. Dollár. Rethinking imagenet pre-training. *CoRR*, abs/1811.08883, 2018. 9

[39] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988, 2017. 2, 5, 8, 9

[40] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 2, 6, 8

[41] S. Honari, J. Yosinski, P. Vincent, and C. J. Pal. Recombinator networks: Learning coarse-to-fine feature aggregation. In *CVPR*, pages 5743–5752, 2016. 2

[42] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei. Relation networks for object detection. In *CVPR*, pages 3588–3597, 2018. 10

[43] P. Hu and D. Ramanan. Bottom-up and top-down reasoning with hierarchical rectified gaussians. In *CVPR*, pages 5600–5609, 2016. 2

[44] G. Huang, D. Chen, T. Li, F. Wu, L. van der Maaten, and K. Q. Weinberger. Multi-scale dense convolutional networks for efficient prediction. *CoRR*, abs/1703.09844, 2017. 2

[45] L. Huang, Y. Yuan, J. Guo, C. Zhang, X. Chen, and J. Wang. Interlaced sparse self-attention for semantic segmentation. *CoRR*, abs/1907.12273, 2019. 11

[46] S. Huang, M. Gong, and D. Tao. A coarse-fine network for keypoint localization. In *ICCV*, pages 3047–3056. IEEE Computer Society, 2017. 5

[47] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu. Ccnet: Criss-cross attention for semantic segmentation. *CoRR*, abs/1811.11721, 2018. 6

[48] A. Ignatov and R. Timofte. Ntire 2019 challenge on image enhancement: Methods and results. In *CVPRW*, June 2019. 12

[49] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *ECCV*, pages 34–50, 2016. 2

[50] M. A. Islam, M. Rochan, N. D. B. Bruce, and Y. Wang. Gated feedback refinement network for dense image labeling. In *CVPR*, pages 4877–4885, 2017. 2

[51] X. Jin, X. Li, H. Xiao, X. Shen, Z. Lin, J. Yang, Y. Chen, J. Dong, L. Liu, Z. Jie, J. Feng, and S. Yan. Video scene parsing with predictive feature learning. In *ICCV*, pages 5581–5589, 2017. 6

[52] A. Kanazawa, A. Sharma, and D. W. Jacobs. Locally scale-invariant convolutional neural networks. *CoRR*, abs/1412.5104, 2014. 2

[53] L. Ke, M. Chang, H. Qi, and S. Lyu. Multi-scale structure-aware network for human pose estimation. *CoRR*, abs/1803.09894, 2018. 2

[54] T. Ke, J. Hwang, Z. Liu, and S. X. Yu. Adaptive affinity fields for semantic segmentation. In *ECCV*, pages 605–621, 2018. 6

[55] S. Kim, H. Kook, J. Sun, M. Kang, and S. Ko. Parallel feature pyramid network for object detection. In *ECCV*, pages 239–256, 2018. 10

[56] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 6

[57] M. Kocabas, S. Karagoz, and E. Akbas. Multiposenet: Fast multi-person pose estimation using pose residual network. In *ECCV*, pages 437–453. Springer, 2018. 5

[58] S. Kong and C. C. Fowlkes. Recurrent scene parsing with perspective understanding in the loop. In *CVPR*, pages 956–965, 2018. 6

[59] T. Kong, F. Sun, W. Huang, and H. Liu. Deep feature pyramid reconfiguration for object detection. In *ECCV*, pages 172–188, 2018. 10

[60] M. Kowalski, J. Naruniec, and T. Trzcinski. Deep alignment network: A convolutional neural network for robust face alignment. *CoRR*, abs/1706.01789, 2017. 2

[61] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012. 1

[62] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, pages 765–781, 2018. 10

[63] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998. 1

[64] H. Li, P. Xiong, J. An, and L. Wang. Pyramid attention network for semantic segmentation. In *BMVC*, page 285, 2018. 6

[65] M. Li, C. Ye, and W. Li. High-resolution network for photorealistic style transfer. *CoRR*, abs/1904.11617, 2019. 12

[66] X. Li, Z. Liu, P. Luo, C. C. Loy, and X. Tang. Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade. In *CVPR*, pages 6459–6468, 2017. 6

[67] Z. Li, Y. Chen, G. Yu, and Y. Deng. R-FCN++: towards accurate region-based fully convolutional networks for object detection. In *AAAI*, pages 7073–7080, 2018. 9

[68] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun. Detnet: Design backbone for object detection. In *ECCV*, pages 339–354, 2018. 2, 10

[69] X. Liang, K. Gong, X. Shen, and L. Lin. Look into person: Joint body parsing & pose estimation network and A new benchmark. *CoRR*, abs/1804.01984, 2018. 7

[70] X. Liang, H. Zhou, and E. Xing. Dynamic-structured semantic propagation network. In *CVPR*, pages 752–761, 2018. 6

[71] I. Lifshitz, E. Fetaya, and S. Ullman. Human pose estimation using deep consensus voting. In *ECCV*, pages 246–260, 2016. 2

[72] G. Lin, A. Milan, C. Shen, and I. D. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, pages 5168–5177, 2017. 2, 6, 7

[73] G. Lin, C. Shen, A. van den Hengel, and I. D. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, pages 3194–3203, 2016. 6, 7

[74] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 936–944, 2017. 2, 8, 10

[75] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *ICCV*, pages 2999–3007, 2017. 8, 10

[76] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755, 2014. 1, 5

[77] C. Liu, L.-C. Chen, F. Schroff, H. Adam, W. Hua, A. Yuille, and L. Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. *arXiv preprint arXiv:1901.02985*, 2019. 6

[78] S. Liu, D. Huang, and Y. Wang. Receptive field block net for accurate and fast object detection. In *ECCV*, pages 404–419, 2018. 10

[79] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. In *CVPR*, pages 8759–8768, 2018. 9

[80] T. Liu, T. Ruan, Z. Huang, Y. Wei, S. Wei, Y. Zhao, and T. Huang. Devil in the details: Towards accurate single and multiple human parsing. *CoRR*, abs/1809.05996, 2018. 7, 8

[81] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 2

[82] Y. Luo, Z. Zheng, L. Zheng, T. Guan, J. Yu, and Y. Yang. Macro-micro adversarial network for human parsing. In *ECCV*, pages 424–440, 2018. 7

[83] R. Mottaghi, X. Chen, X. Liu, N. Cho, S. Lee, S. Fidler, R. Urtasun, and A. L. Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, pages 891–898, 2014. 7

[84] A. Newell, Z. Huang, and J. Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *NIPS*, pages 2274–2284, 2017. 5

[85] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, pages 483–499, 2016. 1, 2, 5, 6

[86] X. Nie, J. Feng, and S. Yan. Mutual learning to adapt for joint human parsing and pose estimation. In *ECCV*, pages 519–534, 2018. 7

[87] H. Noh, S. Hong, and B. Han. Learning deconvolution network

for semantic segmentation. In *ICCV*, pages 1520–1528, 2015. 1, 2

[88] G. Papandreou, T. Zhu, L.-C. Chen, S. Gidaris, J. Tompson, and K. Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *ECCV*, 2018. 5

[89] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy. Towards accurate multi-person pose estimation in the wild. In *CVPR*, pages 3711–3719, 2017. 5, 6

[90] C. Peng, T. Xiao, Z. Li, Y. Jiang, X. Zhang, K. Jia, G. Yu, and J. Sun. Megdet: A large mini-batch object detector. In *CVPR*, pages 6181–6189, 2018. 9

[91] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun. Large kernel matters - improve semantic segmentation by global convolutional network. In *CVPR*, pages 1743–1751, 2017. 2, 6

[92] X. Peng, R. S. Feris, X. Wang, and D. N. Metaxas. A recurrent encoder-decoder network for sequential face alignment. In *ECCV (1)*, volume 9905, pages 38–56, 2016. 1, 2

[93] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *CVPR*, pages 4929–4937, 2016. 2

[94] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe. Full-resolution residual networks for semantic segmentation in street scenes. In *CVPR*, pages 3309–3318, 2017. 2, 6

[95] L. Qi, S. Liu, J. Shi, and J. Jia. Sequential context encoding for duplicate removal. In *NeurIPS*, pages 2053–2062, 2018. 9

[96] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017. 9

[97] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015. 1, 2

[98] M. Samy, K. Amer, K. Eissa, M. Shaker, and M. ElHelw. Nu-net: Deep residual wide field of view convolutional neural network for semantic segmentation. In *CVPRW*, June 2018. 2

[99] S. Saxena and J. Verbeek. Convolutional neural fabrics. In *NIPS*, pages 4053–4061, 2016. 2

[100] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013. 2

[101] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):640–651, 2017. 7

[102] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1, 2

[103] B. Singh and L. S. Davis. An analysis of scale invariance in object detection SNIP. In *CVPR*, pages 3578–3587, 2018. 9

[104] B. Singh, M. Najibi, and L. S. Davis. SNIPER: efficient multi-scale training. In *NeurIPS*, pages 9333–9343, 2018. 9

[105] K. Sun, M. Li, D. Liu, and J. Wang. IGCV3: interleaved low-rank group convolutions for efficient deep neural networks. In *BMVC*, page 101. BMVA Press, 2018. 3

[106] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 3

[107] 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*, abs/1904.04514, 2019. 3

[108] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei. Integral human pose regression. In *ECCV*, pages 536–553, 2018. 5

[109] 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*, pages 1–9, 2015. 1

[110] W. Tang, P. Yu, and Y. Wu. Deeply learned compositional models for human pose estimation. In *ECCV*, September 2018. 2

[111] Z. Tang, X. Peng, S. Geng, L. Wu, S. Zhang, and D. N. Metaxas. Quantized densely connected u-nets for efficient landmark localization. In *ECCV*, pages 348–364, 2018. 2

[112] Z. Tian, C. Shen, H. Chen, and T. He. FCOS: fully convolutional one-stage object detection. *CoRR*, abs/1904.01355, 2019. 2, 8, 9, 10

[113] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. In *CVPR*, pages 648–656, 2015. 2

[114] L. Tychsen-Smith and L. Petersson. Denet: Scalable real-time object detection with directed sparse sampling. In *ICCV*, pages 428–436, 2017. 10

[115] L. Tychsen-Smith and L. Petersson. Improving object localization with fitness NMS and bounded iou loss. In *CVPR*, pages 6877–6885, 2018. 10

[116] R. Valle, J. M. Buenaposada, A. Valdés, and L. Baumela. A deeply-initialized coarse-to-fine ensemble of regression trees for face alignment. In *ECCV*, pages 609–624, 2018. 2

[117] H. Wang, Q. Wang, M. Gao, P. Li, and W. Zuo. Multi-scale location-aware kernel representation for object detection. In *CVPR*, pages 1248–1257, 2018. 10

[118] J. Wang, Z. Wei, T. Zhang, and W. Zeng. Deeply-fused nets. *CoRR*, abs/1605.07716, 2016. 3

[119] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. W. Cottrell. Understanding convolution for semantic segmentation. In *WACV*, 2018. 6

[120] Y. Wang, L. Xie, C. Liu, S. Qiao, Y. Zhang, W. Zhang, Q. Tian, and A. L. Yuille. SORT: second-order response transform for visual recognition. In *ICCV*, pages 1368–1377, 2017. 11

[121] Z. Wang, W. Li, B. Yin, Q. Peng, T. Xiao, Y. Du, Z. Li, X. Zhang, G. Yu, and J. Sun. Mscoco keypoints challenge 2018. In *Joint Recognition Challenge Workshop at ECCV 2018*, 2018. 6

[122] Z. Wojna, J. R. R. Uijlings, S. Guadarrama, N. Silberman, L. Chen, A. Fathi, and V. Ferrari. The devil is in the decoder. In *BMVC*, 2017. 2

[123] J. Wu, H. Zheng, B. Zhao, Y. Li, B. Yan, R. Liang, W. Wang, S. Zhou, G. Lin, Y. Fu, et al. Ai challenger: A large-scale dataset for going deeper in image understanding. *arXiv preprint arXiv:1711.06475*, 2017. 6

[124] W. Wu, C. Qian, S. Yang, Q. Wang, Y. Cai, and Q. Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *CVPR*, pages 2129–2138, 2018. 2

[125] Z. Wu, C. Shen, and A. van den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *CoRR*, abs/1611.10080, 2016. 6

[126] B. Xiao, H. Wu, and Y. Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, pages 472–487, 2018. 1, 2, 5, 6

[127] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. Unified perceptual parsing for scene understanding. In *ECCV*, pages 432–448, 2018. 2

[128] G. Xie, J. Wang, T. Zhang, J. Lai, R. Hong, and G. Qi. Interleaved structured sparse convolutional neural networks. In *CVPR*, pages 8847–8856. IEEE Computer Society, 2018. 3

[129] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 5987–5995, 2017. 8

[130] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, pages 1395–1403, 2015. 2

[131] D. Xu, W. Ouyang, X. Wang, and N. Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *CVPR*, pages 675–684, 2018. 6

[132] H. Xu, X. Lv, X. Wang, Z. Ren, N. Bodla, and R. Chellappa. Deep regionlets for object detection. In *ECCV*, pages 827–844, 2018. 10

[133] Y. Xu, T. Xiao, J. Zhang, K. Yang, and Z. Zhang. Scale-invariant convolutional neural networks. *CoRR*, abs/1411.6369, 2014. 2

[134] J. Yang, Q. Liu, and K. Zhang. Stacked hourglass network for robust facial landmark localisation. In *CVPR*, pages 2025–2033, 2017. 2

[135] W. Yang, S. Li, W. Ouyang, H. Li, and X. Wang. Learning feature pyramids for human pose estimation. In *ICCV*, pages 1290–1299, 2017. 2, 5

[136] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *ECCV*, pages 334–349, 2018. 6

[137] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang. Learning a discriminative feature network for semantic segmentation. In *CVPR*, pages 1857–1866, 2018. 6

[138] F. Yu, V. Koltun, and T. A. Funkhouser. Dilated residual networks. *CoRR*, abs/1705.09914, 2017. 2

[139] Y. Yuan, X. Chen, and J. Wang. Object-contextual representations for semantic segmentation. *CoRR*, abs/1909.11065, 2019. 6, 7, 8, 11

[140] Y. Yuan and J. Wang. Ocnet: Object context network for scene parsing. *CoRR*, abs/1809.00916, 2018. 11

[141] H. Zhang, K. J. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal. Context encoding for semantic segmentation. In *CVPR*, pages 7151–7160, 2018. 7

[142] H. Zhang, H. Zhang, C. Wang, and J. Xie. Co-occurrent features in semantic segmentation. In *CVPR*, June 2019. 6, 7

[143] R. Zhang, S. Tang, Y. Zhang, J. Li, and S. Yan. Scale-adaptive convolutions for scene parsing. In *ICCV*, pages 2050–2058, 2017. 6

[144] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li. Single-shot refinement neural network for object detection. In *CVPR*, pages 4203–4212, 2018. 10

[145] T. Zhang, G. Qi, B. Xiao, and J. Wang. Interleaved group convolutions. In *ICCV*, pages 4383–4392, 2017. 3, 4

[146] Z. Zhang, S. Qiao, C. Xie, W. Shen, B. Wang, and A. L. Yuille. Single-shot object detection with enriched semantics. In *CVPR*, pages 5813–5821, 2018. 10

[147] Z. Zhang, X. Zhang, C. Peng, X. Xue, and J. Sun. Exfuse: Enhancing feature fusion for semantic segmentation. In *ECCV*, pages 273–288, 2018. 2

[148] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, pages 6230–6239, 2017. 1, 2, 6, 7, 11

[149] 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*, pages 270–286, 2018. 6, 7

[150] J. Zhao, J. Li, X. Nie, F. Zhao, Y. Chen, Z. Wang, J. Feng, and S. Yan. Self-supervised neural aggregation networks for human parsing. In *CVPRW*, pages 1595–1603, 2017. 7

[151] L. Zhao, M. Li, D. Meng, X. Li, Z. Zhang, Y. Zhuang, Z. Tu, and J. Wang. Deep convolutional neural networks with merge-and-run mappings. In *IJCAI*, pages 3170–3176, 2018. 3

[152] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling. M2det: A single-shot object detector based on multi-level feature pyramid network. *CoRR*, abs/1811.04533, 2018. 10

[153] P. Zhou, B. Ni, C. Geng, J. Hu, and Y. Xu. Scale-transferrable object detection. In *CVPR*, pages 528–537, 2018. 10

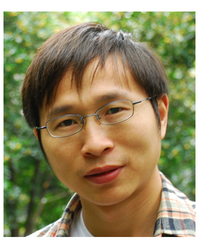
[154] Y. Zhou, X. Hu, and B. Zhang. Interlinked convolutional neural networks for face parsing. In *ISNN*, pages 222–231, 2015. 2

[155] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang. Unet++: A nested u-net architecture for medical image segmentation. In *MICCAI*, pages 3–11, 2018. 2, 6, 7

[156] P. Zhu, D. Du, L. Wen, X. Bian, H. Ling, Q. Hu, T. Peng, J. Zheng, X. Wang, Y. Zhang, L. Bo, H. Shi, R. Zhu, B. Dong, D. Reddy Pailla, F. Ni, G. Gao, G. Liu, H. Xiong, J. Ge, J. Zhou, J. Hu, L. Sun, L. Chen, M. Lauer, Q. Liu, S. Saketh Chennamsetty, T. Sun, T. Wu, V. Alex Kollerathu, W. Tian, W. Qin, X. Chen, X. Zhao, Y. Lian, Y. Wu, Y. Li, Y. Li, Y. Wang, Y. Song, Y. Yao, Y. Zhang, Z. Pi, Z. Chen, Z. Xu, Z. Xiao, Z. Luo, and Z. Liu. Visdrone-vid2019: The vision meets drone object detection in video challenge results. In *ICCV Workshop*, Oct. 12

[157] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, and H. Lu. Couplenet: Coupling global structure with local parts for object detection. In *ICCV*, pages 4146–4154, 2017. 10

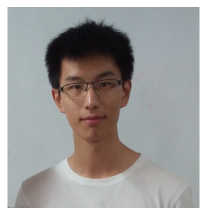
[158] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai. Asymmetric non-local neural networks for semantic segmentation. *CoRR*, abs/1908.07678, 2019. 6, 7



Jingdong Wang is a Senior Principal Research Manager with the Visual Computing Group, Microsoft Research, Beijing, China. He received the B.Eng. and M.Eng. degrees from the Department of Automation, Tsinghua University, Beijing, China, in 2001 and 2004, respectively, and the PhD degree from the Department of Computer Science and Engineering, the Hong Kong University of Science and Technology, Hong Kong, in 2007. His areas of interest include deep learning, large-scale indexing, human understanding, and person re-identification. He is an Associate Editor of IEEE TPAMI, IEEE TMM and IEEE TCSVT, and is an area chair (or SPC) of some prestigious conferences, such as CVPR, ICCV, ECCV, ACM MM, IJCAI, and AAAI. He is a Fellow of IAPR and an ACM Distinguished Member.

Ke Sun is a Ph.D candidate in electrical engineering, the University of Science and Technology of China (USTC), Hefei, China. He received the B.S. degrees from Jilin University. His research interest is in computer vision fields, including human pose estimation, semantic segmentation and image classification.

Tianheng Cheng received his B.S degree in Electronic Information and Communications from Huazhong University of Science and Technology (HUST) in 2019. He is currently pursuing the Ph.D. degree at HUST. He was a research intern at Microsoft Research Asia and worked on object detection. His research interests include computer vision and machine learning.



Borui Jiang is a Ph.D candidate in Academy for Advanced Interdisciplinary Studies, Peking University. He obtained the B.S. degree from Peking University. During the undergraduate period, he worked as an intern in Megvii inc. and MSRA, and he joined the Machine Intelligence Lab of Yadong Mu at Institute of Computer Science & Technology, Peking University. His research interest is in computer vision fields, particularly object detection and image classification.

Chaorui Deng is a Master student in School of Software Engineering, South China University of Technology, Guangzhou, China. His research interests include computer vision and reinforcement learning.

Yang Zhao received the B.Sc. degree in computer science and technology from Wuhan University of Technology, China. She is currently working toward the Ph.D. degree in the School of Engineering, Griffith University, Australia. She is also a visiting student with Australian Institute for Machine Learning (AIML), the University of Adelaide. Her current research interests include deep learning and its applications in computer vision.

Dong Liu received the B.S. and Ph.D. degrees in electrical engineering from the University of Science and Technology of China (USTC), Hefei, China, in 2004 and 2009, respectively. He was a member of the research staff with Nokia Research Center, Beijing, China, from 2009 to 2012. He joined USTC as an Associate Professor in 2012. He has authored or co-authored more than 100 papers in international journals and conferences. He has 16 granted patents, and 1 technical proposal adopted by AVS. His research interests include image and video coding, multimedia signal processing, and multimedia data mining.

Dr. Liu received the 2009 IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY Best Paper Award, and the Best 10% Paper Award at VCIP 2016. He and his team were winners of four technical challenges held in ACM MM 2018, ECCV 2018, CVPR 2018, and ICME 2016, respectively. Dr. Liu is a Senior Member of CSIG. He served as a Registration Co-Chair for ICME 2019, a Symposium Co-Chair for WCSP 2014. He is an elected member of the MSA Technical Committee of IEEE CAS Society.

Yadong Mu is an Assistant Professor and leading the Machine Intelligence Lab at Institute of Computer Science & Technology, Peking University. He obtained both the B.S. and Ph.D. degrees from Peking University. Before joining Peking University, he worked as research fellow at National University of Singapore, research scientist at the DVMM lab of Columbia University, researcher at the data mining team of Huawei Noah's Ark Lab in Hong Kong, and senior scientist at Multimedia Department of AT &T Lab s, New Jersey, U.S.A.. His research interest is in broad research topics in computer vision and machine learning, particularly large scale image and video computing (search, indexing, event detection etc), autonomous driving techniques, and distributed/approximate large scale machine learning.

Mingkui Tan received his Bachelor Degree in Environmental Science and Engineering in 2006 and Master degree in Control Science and Engineering in 2009, both from Hunan University in Changsha, China. He received the PhD degree in Computer Science from Nanyang Technological University, Singapore, in 2014. From 2014-2016, he worked as a Senior Research Associate on computer vision in the School of Computer Science, University of Adelaide, Australia. Since 2016, he has been with the School of Software Engineering, South China University of Technology, China, where he is currently a Professor. His research interests include machine learning, sparse analysis, deep learning and large-scale optimization.

Xinggang Wang received the B.S. and Ph.D. degrees in Electronics and Information Engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2009 and 2014, respectively. He is currently an Associate Professor with the School of Electronic Information and Communications, HUST. His research interests include computer vision and machine learning.

Wenyu Liu received the B.S. degree in Computer Science from Tsinghua University, Beijing, China, in 1986, and the M.S. and Ph.D. degrees, both in Electronics and Information Engineering, from Huazhong University of Science and Technology (HUST), Wuhan, China, in 1991 and 2001, respectively. He is now a professor and associate dean of the School of Electronic Information and Communications, HUST. His current research areas include computer vision, multimedia, and machine learning.

Bin Xiao received the Bachelor and Master degrees from the School of Electronic and Information Engineer, South China University of Technology, in 2009 and 2012, respectively. He is a researcher with AI Lab of ByteDance, Beijing. His main research interests include deep learning and computer vision.