Transpose convolution (padding = 1, stride = 2)

2	3
4	5

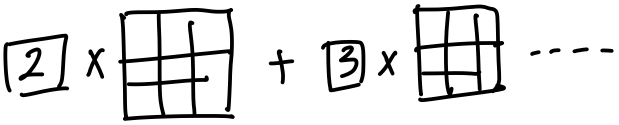
Input feature

⊛

1	2	3
4	5	6
7	8	9

filter

계산을 쉽기위해서



이런식으로 진행한다.

$z = s - 1$

$p' = k - p - 1$

$s' = 1$

다음과 같이 반영하면

z : row & col 사이의 0 개수 ⇒ (2−1)

p' : padding수 (3−1−1 =1)

s' : stride. =1

2	0	3
0	0	0
4	0	5

z만큼 0 대입

⇒

0	0	0	0	0
0	2	0	3	0
0	0	0	0	0
0	4	0	5	0
0	0	0	0	0

p' 만큼 padding

⊛

1	2	3
4	5	6
7	8	9

st=1

→ 단순 conv 연산진행

Dilated convolution

수용영역 확대 dilation rate = 2인 경우 3x3 filter는 5x5 filter 처럼 작용

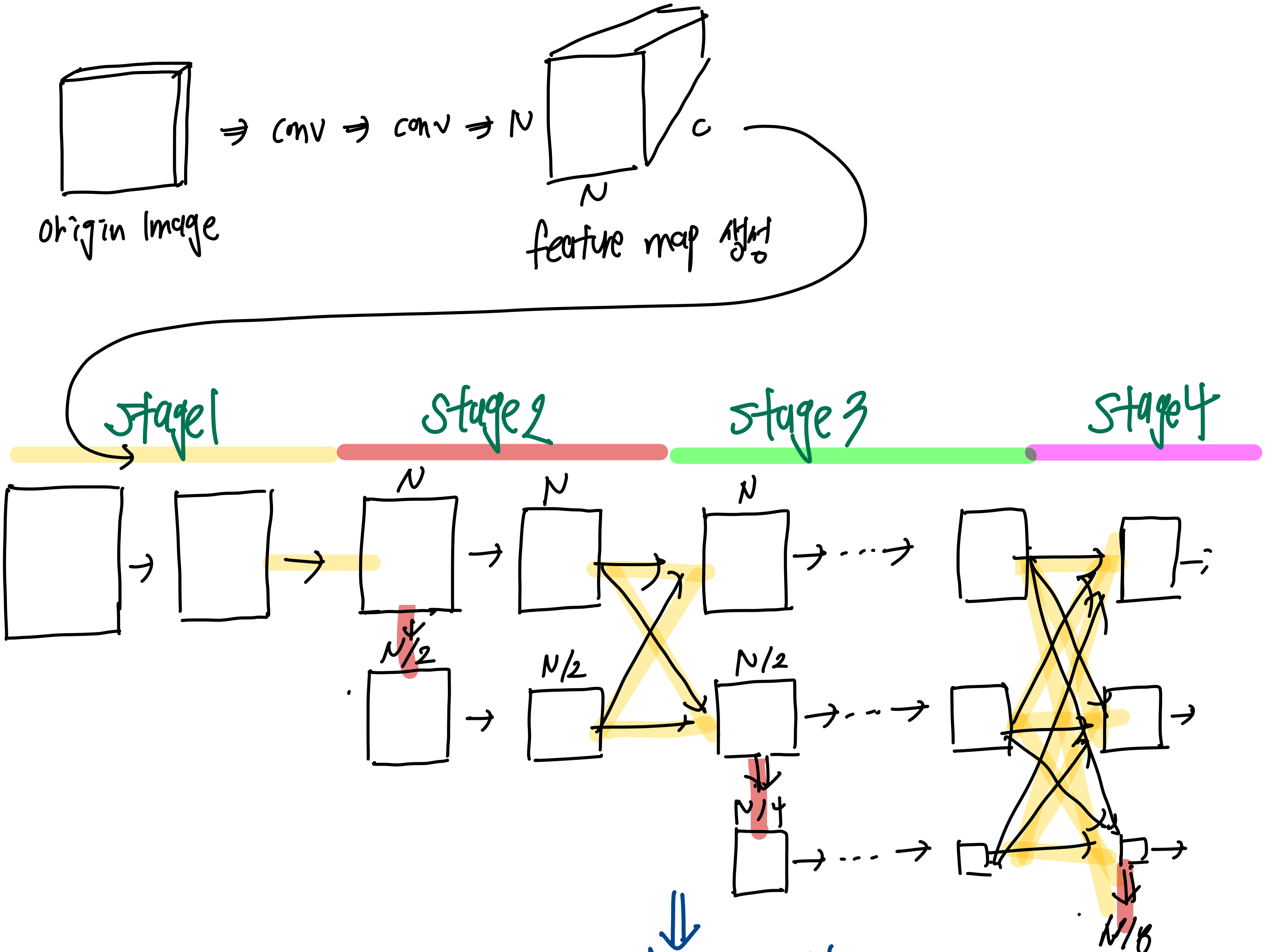
1	2	3
4	5	6
7	8	9

3x3

→

1	0	2	0	3
0	0	0	0	0
4	0	5	0	6
0	0	0	0	0
7	0	8	0	9

5x5



exchange unit

$$Y_k = \sum_{i=1}^{S} a(X_i, k)$$

Deep High-Resolution Representation Learning for Human Pose Estimation

Ke Sun[1,2*†] Bin Xiao[2*] Dong Liu[1] Jingdong Wang[2‡]
[1]University of Science and Technology of China [2]Microsoft Research Asia
sunk@mail.ustc.edu.cn, dongleiu@ustc.edu.cn, {Bin.Xiao,jingdw}@microsoft.com

Abstract

In this paper, we are interested in the human pose estimation problem with a focus on learning reliable high-resolution representations. Most existing methods recover high-resolution representations from low-resolution representations *produced by a high-to-low resolution network. Instead, our proposed network* maintains high-resolution representations *through the whole process.*

We start from a high-resolution subnetwork as the first stage, gradually add high-to-low resolution subnetworks one by one to form more stages, and connect the mutli-resolution subnetworks in parallel. *We conduct* repeated multi-scale fusions *such that each of the high-to-low resolution representations receives information from other parallel representations over and over, leading to rich high-resolution representations. As a result, the predicted keypoint heatmap is potentially more accurate and spatially more precise. We empirically demonstrate the effectiveness of our network through the superior pose estimation results over two benchmark datasets: the COCO keypoint detection dataset and the MPII Human Pose dataset. In addition, we show the superiority of our network in pose tracking on the PoseTrack dataset. The code and models have been publicly available at* `https://github.com/leoxiaobin/deep-high-resolution-net.pytorch`.

1. Introduction

2D human pose estimation has been a fundamental yet challenging problem in computer vision. The goal is to localize human anatomical keypoints (e.g., elbow, wrist, etc.) or parts. It has many applications, including human action recognition, human-computer interaction, animation, etc. This paper is interested in single-person pose estimation, which is the basis of other related problems, such as multi-person pose estimation [6, 26, 32, 38, 46, 55, 40, 45, 17, 69], video pose estimation and tracking [48, 70], etc.

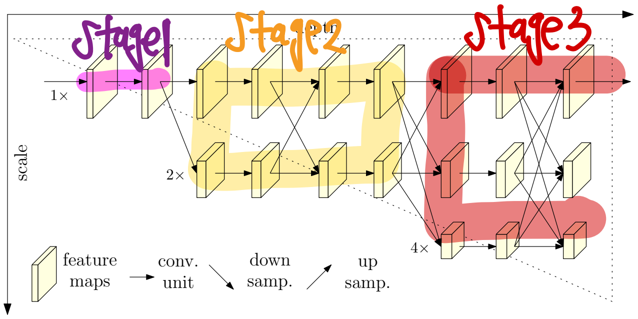


Figure 1. Illustrating the architecture of the proposed HRNet. It consists of parallel high-to-low resolution subnetworks with repeated information exchange across multi-resolution subnetworks (multi-scale fusion). The horizontal and vertical directions correspond to the depth of the network and the scale of the feature maps, respectively.

The recent developments show that deep convolutional neural networks have achieved the state-of-the-art performance. Most existing methods pass the input through a network, typically consisting of high-to-low resolution subnetworks that are connected in series, and then *raise the resolution*. For instance, Hourglass [39] recovers the high resolution through a symmetric low-to-high process. SimpleBaseline [70] adopts a few transposed convolution layers for generating high-resolution representations. In addition, dilated convolutions are also used to blow up the later layers of a high-to-low resolution network (e.g., VGGNet or ResNet) [26, 74].

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 subnetwork as the first stage, gradually add high-to-low resolution subnetworks one by one to form more stages, and connect the multi-resolution subnetworks in parallel. We conduct repeated multi-scale fusions by exchanging the information across the parallel multi-resolution subnetworks over and over through the whole process. We estimate the keypoints over the high-resolution representations output by our network. The resulting network is illustrated in Figure 1.

[*] Equal contribution.
[†] This work is done when Ke Sun was an intern at Microsoft Research, Beijing, P.R. China
[‡] Corresponding author

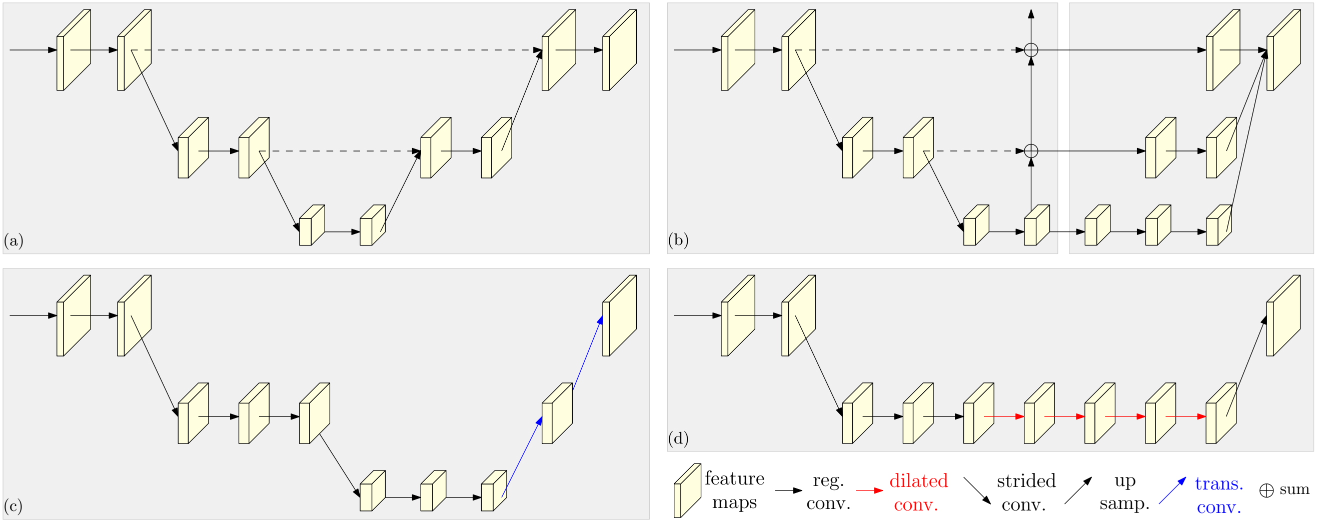


Figure 2. Illustration of representative pose estimation networks that rely on the high-to-low and low-to-high framework. (a) Hourglass [39]. (b) Cascaded pyramid networks [11]. (c) SimpleBaseline [70]: transposed convolutions for low-to-high processing. (d) Combination with dilated convolutions [26]. Bottom-right legend: reg. = regular convolution, dilated = dilated convolution, trans. = transposed convolution, strided = strided convolution, concat. = concatenation. In (a), the high-to-low and low-to-high processes are symmetric. In (b), (c) and (d), the high-to-low process, a part of a classification network (ResNet or VGGNet), is *heavy*, and the low-to-high process is *light*. In (a) and (b), the skip-connections (dashed lines) between the same-resolution layers of the high-to-low and low-to-high processes mainly aim to fuse low-level and high-level features. In (b), the right part, refinenet, combines the low-level and high-level features that are processed through convolutions.

Our network has two benefits in comparison to existing widely-used networks [39, 26, 74, 70] for pose estimation. (i) Our approach connects high-to-low resolution subnetworks in parallel rather than in series as done in most existing solutions. Thus, our approach is able to maintain the high resolution instead of recovering the resolution through a low-to-high process, and accordingly the predicted heatmap is potentially spatially more precise. (ii) Most existing fusion schemes aggregate low-level and high-level representations. Instead, we perform repeated multi-scale fusions to boost the high-resolution representations with the help of the low-resolution representations of the same depth and similar level, and vice versa, resulting in that high-resolution representations are also rich for pose estimation. Consequently, our predicted heatmap is potentially more accurate.

We empirically demonstrate the superior keypoint detection performance over two benchmark datasets: the COCO keypoint detection dataset [35] and the MPII Human Pose dataset [2]. In addition, we show the superiority of our network in video pose tracking on the PoseTrack dataset [1].

2. Related Work

Most traditional solutions to single-person pose estimation adopt the probabilistic graphical model or the pictorial structure model [76, 49], which is recently improved by exploiting deep learning for better modeling the unary and pair-wise energies [9, 63, 44] or imitating the iterative inference process [13]. Nowadays, deep convolutional neural network provides dominant solutions [20, 34, 60, 41, 42, 47, 56, 16]. There are two mainstream methods: regressing the position of keypoints [64, 7], and estimating keypoint heatmaps [13, 14, 75] followed by choosing the locations with the highest heat values as the keypoints.

Most convolutional neural networks for keypoint heatmap estimation consist of a stem subnetwork similar to the classification network, which decreases the resolution, a main body producing the representations with the same resolution as its input, followed by a regressor estimating the heatmaps where the keypoint positions are estimated and then transformed in the full resolution. The main body mainly adopts the high-to-low and low-to-high framework, possibly augmented with multi-scale fusion and intermediate (deep) supervision.

High-to-low and low-to-high. The high-to-low process aims to generate low-resolution and high-level representations, and the low-to-high process aims to produce high-resolution representations [4, 11, 22, 70, 39, 60]. Both the two processes are possibly repeated several times for boosting the performance [74, 39, 14].

Representative network design patterns include: (i) Symmetric high-to-low and low-to-high processes. Hourglass and its follow-ups [39, 14, 74, 30] design the low-to-high process as a mirror of the high-to-low process. (ii) Heavy high-to-low and light low-to-high. The high-to-low process is based on the ImageNet classification network, e.g., ResNet adopted in [11, 70], and the low-to-high process is simply a few bilinear-upsampling [11] or transpose convolution [70] layers. (iii) Combination with dilated convolutions. In [26, 50, 34], dilated convolutions are adopted in the last two stages in the ResNet or VGGNet to eliminate the spatial resolution loss, which is followed by a light low-to-high process to further increase the resolution, avoiding expensive computation cost for only using dilated convolutions [11, 26, 50]. Figure 2 depicts four representative pose estimation networks.

Multi-scale fusion. The straightforward way is to feed multi-resolution images separately into multiple networks and aggregate the output response maps [62]. Hourglass [39] and its extensions [74, 30] combine low-level features in the high-to-low process into the same-resolution high-level features in the low-to-high process progressively through skip connections. In cascaded pyramid network [11], a globalnet combines low-to-high level features in the high-to-low process progressively into the low-to-high process, and then a refinenet combines the low-to-high level features that are processed through convolutions. Our approach repeats multi-scale fusion, which is partially inspired by deep fusion and its extensions [65, 71, 57, 77, 79].

Intermediate supervision. Intermediate supervision or deep supervision, early developed for image classification [33, 59], is also adopted for helping deep networks training and improving the heatmap estimation quality, e.g., [67, 39, 62, 3, 11]. The hourglass approach [39] and the convolutional pose machine approach [67] process the intermediate heatmaps as the input or a part of the input of the remaining subnetwork.

Our approach. Our network connects high-to-low subnetworks in parallel. It maintains high-resolution representations through the whole process for spatially precise heatmap estimation. It generates reliable high-resolution representations through repeatedly fusing the representations produced by the high-to-low subnetworks. Our approach is different from most existing works, which need a separate low-to-high upsampling process and aggregate low-level and high-level representations. Our approach, without using intermediate heatmap supervision, is superior in keypoint detection accuracy and efficient in computation complexity and parameters.

There are related multi-scale networks for classification and segmentation [5, 8, 72, 78, 29, 73, 53, 54, 23, 80, 53, 51, 18]. Our work is partially inspired by some of them [54, 23, 80, 53], and there are clear differences making them not applicable to our problem. Convolutional neural fabrics [54] and interlinked CNN [80] fail to produce high-quality segmentation results because of a lack of proper design on each subnetwork (depth, batch normalization) and multi-scale fusion. The grid network [18], a combination of many weight-shared U-Nets, consists of two separate fusion processes across multi-resolution representations: on the first stage, information is only sent from high resolution to low resolution; on the second stage, information is only sent from low resolution to high resolution, and thus less competitive. Multi-scale densenets [23] does not target and cannot generate reliable high-resolution representations.

3. Approach

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 $W \times H \times 3$. The state-of-the-art methods transform this problem to estimating K heatmaps of size $W^{'} \times H^{'}$, $\{\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_K\}$, where each heatmap $\mathbf{H}_k$ indicates the location confidence of the kth keypoint.

We follow the widely-adopted pipeline [39, 70, 11] to predict human keypoints using a convolutional network, which is composed of a stem consisting of two strided convolutions decreasing the resolution, a main body outputting the feature maps with the same resolution as its input feature maps, and a regressor estimating the heatmaps where the keypoint positions are chosen and transformed to the full resolution. We focus on the design of the main body and introduce our High-Resolution Net (HRNet) that is depicted in Figure 1.

Sequential multi-resolution subnetworks. Existing networks for pose estimation are built by connecting high-to-low resolution subnetworks in series, where each subnetwork, forming a stage, is composed of a sequence of convolutions and there is a down-sample layer across adjacent subnetworks to halve the resolution.

Let $\mathcal{N}_{sr}$ be the subnetwork in the sth stage and r be the resolution index (Its resolution is $\frac{1}{2^{r-1}}$ of the resolution of the first subnetwork). The high-to-low network with S (e.g., 4) stages can be denoted as: stage별 이미지는 1/2배 된다.

$$\mathcal{N}_{11} \rightarrow \mathcal{N}_{22} \rightarrow \mathcal{N}_{33} \rightarrow \mathcal{N}_{44}. \tag{1}$$

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

An example network structure, containing 4 parallel subnetworks, is given as follows,

$$\begin{tikzcd}
\mathcal{N}_{11} \arrow[r] \arrow[dr] & \mathcal{N}_{21} \arrow[r] & \mathcal{N}_{31} \arrow[r] & \mathcal{N}_{41} \arrow[r, Rightarrow] & \text{resolution } R \\
 & \mathcal{N}_{22} \arrow[r] \arrow[dr] & \mathcal{N}_{32} \arrow[r] & \mathcal{N}_{42} \arrow[r, Rightarrow] & 〃 \quad R/2 \\
 & & \mathcal{N}_{33} \arrow[r] \arrow[dr] & \mathcal{N}_{43} \arrow[r, Rightarrow] & 〃 \quad R/2^2 \\
 & & & \mathcal{N}_{44}. \arrow[r, Rightarrow] & . \quad R/2^3
\end{tikzcd} \tag{2}$$

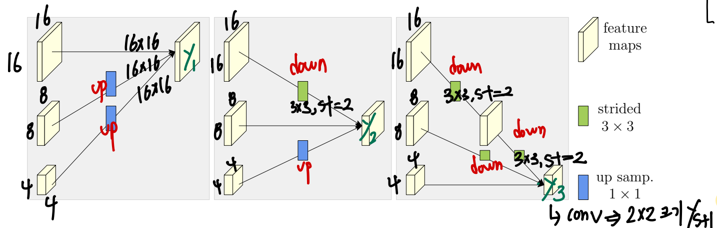


Figure 3. Illustrating how the exchange unit aggregates the information for high, medium and low resolutions from the left to the right, respectively. Right legend: strided 3×3 = strided 3×3 convolution, up samp. 1×1 = nearest neighbor up-sampling following a 1×1 convolution.

Repeated multi-scale fusion. We introduce *exchange units* across parallel subnetworks such that each subnetwork repeatedly receives the information from other parallel subnetworks. Here is an example showing the scheme of exchanging information. We divided the third stage into several (e.g., 3) exchange blocks, and each block is composed of 3 parallel convolution units with an exchange unit across the parallel units, which is given as follows,

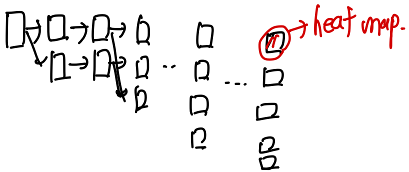
$$\begin{array}{ccccccccccc} \mathcal{C}_{31}^1 & \searrow & & \nearrow & \mathcal{C}_{31}^2 & \searrow & & \nearrow & \mathcal{C}_{31}^3 & \searrow & \\ \mathcal{C}_{32}^1 & \rightarrow & \mathcal{E}_3^1 & \rightarrow & \mathcal{C}_{32}^2 & \rightarrow & \mathcal{E}_3^2 & \rightarrow & \mathcal{C}_{32}^3 & \rightarrow & \mathcal{E}_3^3, \\ \mathcal{C}_{33}^1 & \nearrow & & \searrow & \mathcal{C}_{33}^2 & \nearrow & & \searrow & \mathcal{C}_{33}^3 & \nearrow & \end{array} \tag{3}$$

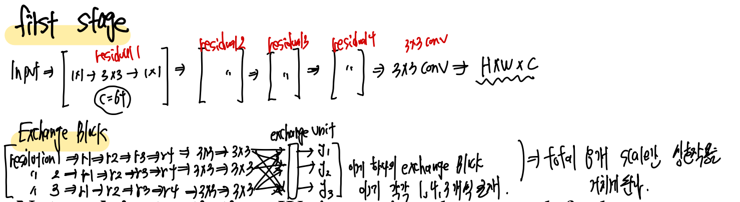
where $\mathcal{C}_{sr}^b$ represents the convolution unit in the rth resolution of the bth block in the sth stage, and $\mathcal{E}_s^b$ is the corresponding exchange unit.

We illustrate the exchange unit in Figure 3 and present the formulation in the following. We drop the subscript s and the superscript b for discussion convenience. The inputs are s response maps: $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_s\}$. The outputs are s response maps: $\{\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_s\}$, whose resolutions and widths are the same to the input. Each output is an aggregation of the input maps, $\mathbf{Y}_k = \sum_{i=1}^{s} a(\mathbf{X}_i, k)$. The exchange unit across stages has an extra output map $\mathbf{Y}_{s+1}$: $\mathbf{Y}_{s+1} = a(\mathbf{Y}_s, s+1)$.

The function $a(\mathbf{X}_i, k)$ consists of upsampling or downsampling $\mathbf{X}_i$ from resolution i to resolution k. We adopt strided 3×3 convolutions for downsampling. For instance, one strided 3×3 convolution with the stride 2 for $2\times$ downsampling, and two consecutive strided 3×3 convolutions with the stride 2 for $4\times$ downsampling. For upsampling, we adopt the simple nearest neighbor sampling following a 1×1 convolution for aligning the number of channels. If $i = k$, $a(\cdot, \cdot)$ is just an identify connection: $a(\mathbf{X}_i, k) = \mathbf{X}_i$.

Heatmap estimation. We regress the heatmaps simply from the high-resolution representations output by the last exchange unit, which empirically works well. The loss function, defined as the mean squared error, is applied for comparing the predicted heatmaps and the groundtruth heatmaps. The groundtruth heatmpas are generated by applying 2D Gaussian with standard deviation of 1 pixel centered on the grouptruth location of each keypoint.




first stage
residual1
Input ⇒
1×1 → 3×3 → 1×1
(C=64)
residual2
residual3
residual4
3×3 conv
⇒ 3×3 conv ⇒ H×W×C
Exchange Block
exchange unit
resolution 1 ⇒ r1 ⇒ r2 ⇒ r3 ⇒ r4 ⇒ 3×3 ⇒ 3×3
2 → r1 ⇒ r2 → r3 ⇒ r4 ⇒ 3×3 ⇒ 3×3
3 ⇒ r1 → r2 ⇒ r3 ⇒ r4 → 3×3 ⇒ 3×3
y_1
y_2
y_3
여기 하나의 exchange block
여기 각각 1, 4, 3 개씩 존재.
⇒ total 8개 scale인 상호작용을 거치게 된다.

Network instantiation. We instantiate the network for keypoint heatmap estimation by following the design rule of ResNet to distribute the depth to each stage and the number of channels to each resolution.

The main body, i.e., our HRNet, contains four stages with four parallel subnetworks, whose the resolution is gradually decreased to a half and accordingly the width (the number of channels) is increased to the double. The first stage contains 4 residual units where each unit, the same to the ResNet-50, is formed by a bottleneck with the width 64, and is followed by one 3×3 convolution reducing the width of feature maps to C. The 2nd, 3rd, 4th stages contain 1, 4, 3 exchange blocks, respectively. One exchange block contains 4 residual units where each unit contains two 3×3 convolutions in each resolution and an exchange unit across resolutions. In summary, there are totally 8 exchange units, i.e., 8 multi-scale fusions are conducted.

해상도 1/2
채널수 2

In our experiments, we study one small net and one big net: HRNet-W32 and HRNet-W48, where 32 and 48 represent the widths (C) of the high-resolution subnetworks in last three stages, respectively. The widths of other three parallel subnetworks are 64, 128, 256 for HRNet-W32, and 96, 192, 384 for HRNet-W48.

4. Experiments

4.1. COCO Keypoint Detection

Dataset. The COCO dataset [35] contains over 200,000 images and 250,000 person instances labeled with 17 keypoints. We train our model on COCO train2017 dataset, including $57K$ images and $150K$ person instances. We evaluate our approach on the val2017 set and test-dev2017 set, containing 5000 images and $20K$ images, respectively.

Evaluation metric. The standard evaluation metric is based on Object Keypoint Similarity (OKS): $\text{OKS} = \frac{\sum_i \exp(-d_i^2/2s^2k_i^2)\delta(v_i>0)}{\sum_i \delta(v_i>0)}$. 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[1]: AP^{50} (AP at OKS $= 0.50$) AP^{75}, AP (the mean of AP scores at 10 positions, OKS $= 0.50, 0.55, \dots, 0.90, 0.95$; AP^M for medium objects, AP^L for large objects, and AR at OKS $= 0.50, 0.55, \dots, 0.90, 0.95$.

Training. We extend the human detection box in height or width to a fixed aspect ratio: height : 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 augmentation includes random rotation ($[-45°, 45°]$), random scale

[1] http://cocodataset.org/#keypoints-eval

$\underline{I=30,\ s=1,\ k=3,\ p=1}$

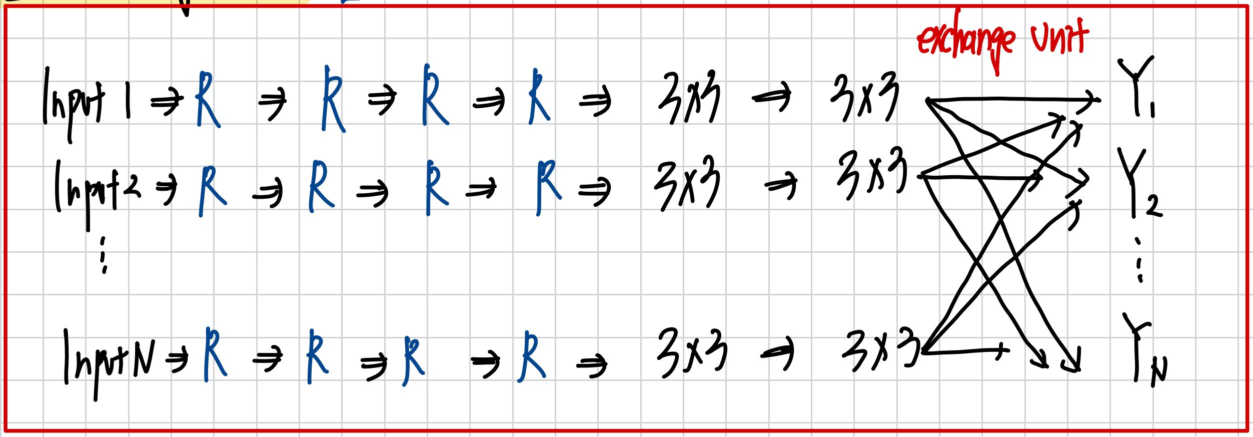
$\underline{30-3+2}+1=$

▨ Residual Block R

HRNet의 경우 3x3시 C=64로 설정.

Input ⇒ $\boxed{1\times1 \Rightarrow 3\times3 \Rightarrow 1\times1}$ ⇒ 최종 해상도의 변화X.

▨ Exchange Unit E


exchange unit
Input 1 ⇒ R ⇒ R ⇒ R ⇒ R ⇒ 3x3 ⇒ 3x3
Input 2 ⇒ R ⇒ R ⇒ R ⇒ R ⇒ 3x3 ⇒ 3x3
⋮
Input N ⇒ R ⇒ R ⇒ R ⇒ R ⇒ 3x3 ⇒ 3x3
Y1
Y2
⋮
YN

▨ Model Architecture $\left(Y_{ij}\ \begin{cases} i : \text{stage} \\ j : \text{index} \end{cases}\right)$

<stage1>

Input Image ⇒ R ⇒ R ⇒ R ⇒ R ⇒ 3x3 ⇒ Y_{11}

$\downarrow$ 해상도 1/2 & 너비 X2

Y_{12}

<stage2>

$Y_{11} \rightarrow E_1 \rightarrow Y_{21}$

$Y_{12} \rightarrow E_2 \rightarrow Y_{22}$

$\downarrow$

Y_{23}

<stage3>

$Y_{21} \Rightarrow E_3 \Rightarrow E_3 \Rightarrow E_3 \Rightarrow E_3 \rightarrow Y_{31}$

$Y_{22} \rightarrow E_3 \rightarrow E_3 \Rightarrow E_3 \Rightarrow E_3 \rightarrow Y_{32}$

$Y_{23} \Rightarrow E_3 \Rightarrow E_3 \Rightarrow E_3 \Rightarrow E_3 \rightarrow Y_{33}$

$\downarrow$

Y_{34}

<stage4>

$Y_{31} \Rightarrow E \Rightarrow E \Rightarrow E \rightarrow Y_{41}$ → 1x1 ⇒ Final heatmap.

$Y_{32} \Rightarrow E \Rightarrow E \Rightarrow E \rightarrow Y_{42}$

$Y_{33} \Rightarrow E \Rightarrow E \Rightarrow E \rightarrow Y_{43}$

$Y_{34} \Rightarrow E \Rightarrow E \Rightarrow E \rightarrow Y_{44}$

NN. ModuleList()

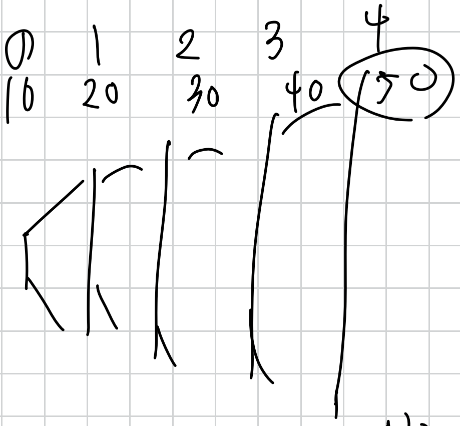
pytorch module 들을 리스트로 관리. ⇒ "동적으로 Module 추가 & 삭제가 가능하다"

example)

```
class Model(nn.Module):
      def __init__(self):
          super().__init__()
          self.linears= nn.ModuleList()
          for i in range(5):
                self.linears.append(nn.Linear(10,20))
      def forward(self,x):
           for layer in self.linears:
                x=layer(x)

           return(x).
```

1, 4, 3



① Input 1 ⇒ R 거번호를

feature map 생성

② Input (1, 2) ⇒

Input_Info (Channel 정보)

self.stem ⇒ 이미지 해상도 줄이고 stage1 Input 준비.

self.stag1 ⇒ [R , R , R , R]

C2 , C3 , C4 =

Table 1. Comparisons on the COCO validation set. Pretrain = pretrain the backbone on the ImageNet classification task. OHKM = online hard keypoints mining [11].

Method	Backbone	Pretrain	Input size	#Params	GFLOPs	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
8-stage Hourglass [39]	8-stage Hourglass	N	256 × 192	25.1M	14.3	66.9	–	–	–	–	–
CPN [11]	ResNet-50	Y	256 × 192	27.0M	6.20	68.6	–	–	–	–	–
CPN + OHKM [11]	ResNet-50	Y	256 × 192	27.0M	6.20	69.4	–	–	–	–	–
SimpleBaseline [70]	ResNet-50	Y	256 × 192	34.0M	8.90	70.4	88.6	78.3	67.1	77.2	76.3
SimpleBaseline [70]	ResNet-101	Y	256 × 192	53.0M	12.4	71.4	89.3	79.3	68.1	78.1	77.1
SimpleBaseline [70]	ResNet-152	Y	256 × 192	68.6M	15.7	72.0	89.3	79.8	68.7	78.9	77.8
HRNet-W32	HRNet-W32	N	256 × 192	28.5M	7.10	73.4	89.5	80.7	70.2	80.1	78.9
HRNet-W32	HRNet-W32	Y	256 × 192	28.5M	7.10	74.4	90.5	81.9	70.8	81.0	79.8
HRNet-W48	HRNet-W48	Y	256 × 192	63.6M	14.6	75.1	90.6	82.2	71.5	81.8	80.4
SimpleBaseline [70]	ResNet-152	Y	384 × 288	68.6M	35.6	74.3	89.6	81.1	70.5	79.7	79.7
HRNet-W32	HRNet-W32	Y	384 × 288	28.5M	16.0	75.8	90.6	82.7	71.9	82.8	81.0
HRNet-W48	HRNet-W48	Y	384 × 288	63.6M	32.9	**76.3**	**90.8**	**82.9**	**72.3**	**83.4**	**81.2**

Table 2. Comparisons on the COCO test-dev set. #Params and FLOPs are calculated for the pose estimation network, and those for human detection and keypoint grouping are not included.

Method	Backbone	Input size	#Params	GFLOPs	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
Bottom-up: keypoint detection and grouping										
OpenPose [6]	–	–	–	–	61.8	84.9	67.5	57.1	68.2	66.5
Associative Embedding [38]	–	–	–	–	65.5	86.8	72.3	60.6	72.6	70.2
PersonLab [45]	–	–	–	–	68.7	89.0	75.4	64.1	75.5	75.4
MultiPoseNet [32]	–	–	–	–	69.6	86.3	76.6	65.0	76.3	73.5
Top-down: human detection and single-person keypoint detection										
Mask-RCNN [21]	ResNet-50-FPN	–	–	–	63.1	87.3	68.7	57.8	71.4	–
G-RMI [46]	ResNet-101	353 × 257	42.6M	57.0	64.9	85.5	71.3	62.3	70.0	69.7
Integral Pose Regression [58]	ResNet-101	256 × 256	45.0M	11.0	67.8	88.2	74.8	63.9	74.0	–
G-RMI + extra data [46]	ResNet-101	353 × 257	42.6M	57.0	68.5	87.1	75.5	65.8	73.3	73.3
CPN [11]	ResNet-Inception	384 × 288	–	–	72.1	91.4	80.0	68.7	77.2	78.5
RMPE [17]	PyraNet [74]	320 × 256	28.1M	26.7	72.3	89.2	79.1	68.0	78.6	–
CFN [24]	–	–	–	–	72.6	86.1	69.7	78.3	64.1	–
CPN (ensemble) [11]	ResNet-Inception	384 × 288	–	–	73.0	91.7	80.9	69.5	78.1	79.0
SimpleBaseline [70]	ResNet-152	384 × 288	68.6M	35.6	73.7	91.9	81.1	70.3	80.0	79.0
HRNet-W32	HRNet-W32	384 × 288	28.5M	16.0	74.9	92.5	82.8	71.3	80.9	80.1
HRNet-W48	HRNet-W48	384 × 288	63.6M	32.9	**75.5**	**92.5**	**83.3**	**71.9**	**81.5**	**80.5**
HRNet-W48 + extra data	HRNet-W48	384 × 288	63.6M	32.9	**77.0**	**92.7**	**84.5**	**73.4**	**83.1**	**82.0**

($[0.65, 1.35]$), and flipping. Following [66], half body data augmentation is also involved.

We use the Adam optimizer [31]. The learning schedule follows the setting [70]. The base learning rate is set as $1\mathrm{e}{-3}$, and is dropped to $1\mathrm{e}{-4}$ and $1\mathrm{e}{-5}$ at the 170th and 200th epochs, respectively. The training process is terminated within 210 epochs.

Testing. The two-stage top-down paradigm similar as [46, 11, 70] is used: detect the person instance using a person detector, and then predict detection keypoints.

We use the same person detectors provided by SimpleBaseline[2] for both validation set and test-dev set. Following [70, 39, 11], we compute the heatmap by averaging the headmaps 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 validation set. We report the results of our method and other state-of–the-art methods in Table 1. Our small network - HRNet-W32, trained from scratch with the input size 256 × 192, achieves an 73.4 AP score, outperforming other methods with the same input size. (i) Compared to Hourglass [39], our small network improves AP by 6.5 points, and the GFLOPs of our network is much lower and less than half, while the number of parameters are similar and ours is slightly larger. (ii) Compared to CPN [11]

[2] https://github.com/Microsoft/human-pose-estimation.pytorch

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 SimpleBaseline [70], our HRNet-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 nets can benefit from (i) training from the model pretrained on the ImageNet: The gain is 1.0 points for HRNet-W32; (ii) increasing the capacity by increasing the width: HRNet-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 HRNet-W32 and HRNet-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 the SimpleBaseline [70] that uses ResNet-152 as the backbone, our HRNet-W32 and HRNet-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, HRNet-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, HRNet-W48, achieves the highest 75.5 AP. Compared to the SimpleBaseline [70] with the same input size, our small and big networks receive 1.2 and 1.8 improvements, respectively. With additional data from AI Challenger [68] for training, our single big network can obtain an AP of 77.0.

4.2. MPII Human Pose Estimation

Dataset. The MPII Human Pose dataset [2] consists of images taken from a wide-range of real-world activities with full-body pose annotations. There are around $25K$ images with $40K$ subjects, where there are $12K$ subjects for testing and the remaining subjects for the training set. The data augmentation and the training strategy are the same to MS COCO, except that the input size is cropped to 256×256 for fair comparison with other methods.

Testing. The testing procedure is almost the same to that in COCO except that we adopt the standard testing strategy to use the provided person boxes instead of detected person boxes. Following [14, 74, 60], a six-scale pyramid testing procedure is performed.

Evaluation metric. The standard metric [2], the PCKh (head-normalized probability of correct keypoint) score, is used. A joint is correct if it falls within αl pixels of the groundtruth position, where α is a constant and l is the head size that corresponds to 60% of the diagonal length of the ground-truth head bounding box. The PCKh@0.5 ($\alpha = 0.5$) score is reported.

Table 3. Comparisons on the MPII test set (PCKh@0.5).

Method	Hea	Sho	Elb	Wri	Hip	Kne	Ank	Total
Insafutdinov et al. [26]	96.8	95.2	89.3	84.4	88.4	83.4	78.0	88.5
Wei et al. [67]	97.8	95.0	88.7	84.0	88.4	82.8	79.4	88.5
Bulat et al. [4]	97.9	95.1	89.9	85.3	89.4	85.7	81.7	89.7
Newell et al. [39]	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9
Sun et al. [56]	98.1	96.2	91.2	87.2	89.8	87.4	84.1	91.0
Tang et al. [61]	97.4	96.4	92.1	87.7	90.2	87.7	84.3	91.2
Ning et al. [43]	98.1	96.3	92.2	87.8	90.6	87.6	82.7	91.2
Luvizon et al. [36]	98.1	96.6	92.0	87.5	90.6	88.0	82.7	91.2
Chu et al. [14]	98.5	96.3	91.9	88.1	90.6	88.0	85.0	91.5
Chou et al. [12]	98.2	96.8	92.2	88.0	91.3	89.1	84.9	91.8
Chen et al. [10]	98.1	96.5	92.5	88.5	90.2	**89.6**	86.0	91.9
Yang et al. [74]	98.5	96.7	92.5	88.7	91.1	88.6	86.0	92.0
Ke et al. [30]	98.5	96.8	92.7	88.4	90.6	89.3	**86.3**	92.1
Tang et al. [60]	98.4	**96.9**	92.6	88.7	**91.8**	89.4	86.2	**92.3**
SimpleBaseline [70]	98.5	96.6	91.9	87.6	91.1	88.1	84.1	91.5
HRNet-W32	**98.6**	**96.9**	**92.8**	**89.0**	91.5	89.0	85.7	**92.3**

Table 4. #Params and GFLOPs of some top-performed methods reported in Table 3. The GFLOPs is computed with the input size 256×256.

Method	#Params	GFLOPs	PCKh@0.5
Insafutdinov et al. [26]	42.6M	41.2	88.5
Newell et al. [39]	25.1M	19.1	90.9
Yang et al. [74]	28.1M	21.3	92.0
Tang et al. [60]	15.5M	15.6	92.3
SimpleBaseline [70]	68.6M	20.9	91.5
HRNet-W32	28.5M	9.5	92.3

Results on the test set. Tables 3 and 4 show the PCKh@0.5 results, the model size and the GFLOPs of the top-performed methods. We reimplement the SimpleBaseline [70] by using ResNet-152 as the backbone with the input size 256×256. Our HRNet-W32 achieves a 92.3 PCKh@0.5 score, and outperforms the stacked hourglass approach [39] and its extensions [56, 14, 74, 30, 60]. Our result is the same as the best one [60] among the previously-published results on the leaderboard of Nov. 16th, 2018[3]. We would like to point out that the approach [60], complementary to our approach, exploits the compositional model to learn the configuration of human bodies and adopts multi-level intermediate supervision, from which our approach can also benefit. We also tested our big network - HRNet-W48 and obtained the same result 92.3. The reason might be that the performance in this dataset tends to be saturate.

4.3. Application to Pose Tracking

Dataset. PoseTrack [27] is a large-scale benchmark for human pose estimation and articulated tracking in video. The

[3] http://human-pose.mpi-inf.mpg.de/#results

dataset, based on the raw videos provided by the popular MPII Human Pose dataset, contains 550 video sequences with 66, 374 frames. The video sequences are split into 292, 50, 208 videos for training, validation, and testing, respectively. The length of the training videos ranges between $41-151$ frames, and 30 frames from the center of the video are densely annotated. The number of frames in the validation/testing videos ranges between $65-298$ frames. The 30 frames around the keyframe from the MPII Pose dataset are densely annotated, and afterwards every fourth frame is annotated. In total, this constitutes roughly 23, 000 labeled frames and 153, 615 pose annotations.

Evaluation metric. We evaluate the results from two aspects: frame-wise multi-person pose estimation, and multi-person pose tracking. Pose estimation is evaluated by the mean Average Precision (mAP) as done in [50, 27]. Multi-person pose tracking is evaluated by the multi-object tracking accuracy (MOTA) [37, 27]. Details are given in [27].

Training. We train our HRNet-W48 for single person pose estimation on the PoseTrack2017 training set, where the network is initialized by the model pre-trained on COCO dataset. We extract the person box, as the input of our network, from the annotated keypoints in the training frames by extending the bounding box of all the keypoints (for one single person) by 15% in length. The training setup, including data augmentation, is almost the same as that for COCO except that the learning schedule is different (as now it is for fine-tuning): the learning rate starts from $1e{-}4$, drops to $1e{-}5$ at the 10th epoch, and to $1e{-}6$ at the 15th epoch; the iteration ends within 20 epochs.

Testing. We follow [70] to track poses across frames. It consists of three steps: person box detection and propagation, human pose estimation, and pose association cross nearby frames. We use the same person box detector as used in SimpleBaseline [70], and propagate the detected box into nearby frames by propagating the predicted keypoints according to the optical flows computed by FlowNet 2.0 [25][4], followed by non-maximum suppression for box removing. The pose association scheme is based on the object keypoint similarity between the keypoints in one frame and the keypoints propagated from the nearby frame according to the optical flows. The greedy matching algorithm is then used to compute the correspondence between keypoints in nearby frames. More details are given in [70].

Results on the PoseTrack2017 test set. Table 5 reports the results. Compared with the second best approach, the FlowTrack in SimpleBaseline [70], that uses ResNet-152 as the backbone, our approach gets 0.3 and 0.1 points gain in terms of mAP and MOTA, respectively. The superiority over the FlowTrack [70] is consistent to that on the COCO keypoint detection and MPII human pose estimation datasets. This further implies the effectiveness of our pose estimation network.

Table 5. Results of pose tracking on the PoseTrack2017 test set.

Entry	Additional training Data	mAP	MOTA
ML-LAB [81]	COCO+MPII-Pose	70.3	41.8
SOPT-PT [52]	COCO+MPII-Pose	58.2	42.0
BUTD2 [28]	COCO	59.2	50.6
MVIG [52]	COCO+MPII-Pose	63.2	50.7
PoseFlow [52]	COCO+MPII-Pose	63.0	51.0
ProTracker [19]	COCO	59.6	51.8
HMPT [52]	COCO+MPII-Pose	63.7	51.9
JointFlow [15]	COCO	63.6	53.1
STAF [52]	COCO+MPII-Pose	70.3	53.8
MIPAL [52]	COCO	68.8	54.5
FlowTrack [70]	COCO	74.6	57.8
HRNet-W48	COCO	**74.9**	**57.9**

Table 6. Ablation study of exchange units that are used in repeated multi-scale fusion. Int. exchange across = intermediate exchange across stages, Int. exchange within = intermediate exchange within stages.

Method	Final exchange	Int. exchange across	Int. exchange within	AP
(a)	✓			70.8
(b)	✓	✓		71.9
(c)	✓	✓	✓	73.4

4.4. Ablation Study

We study the effect of each component in our approach on the COCO keypoint detection dataset. All results are obtained over the input size of 256×192 except the study about the effect of the input size.

Repeated multi-scale fusion. We empirically analyze the effect of the repeated multi-scale fusion. We study three variants of our network. (a) W/o intermediate exchange units (1 fusion): There is no exchange between multi-resolution subnetworks except the last exchange unit. (b) W/ across-stage exchange units (3 fusions): There is no exchange between parallel subnetworks within each stage. (c) W/ both across-stage and within-stage exchange units (totally 8 fusion): This is our proposed method. All the networks are trained from scratch. The results on the COCO validation set given in Table 6 show that the multi-scale fusion is helpful and more fusions lead to better performance.

Resolution maintenance. We study the performance of a variant of the HRNet: all the four high-to-low resolution subnetworks are added at the beginning and the depth are the same; the fusion schemes are the same to ours. Both our HRNet-W32 and the variant (with similar #Params and GFLOPs) are trained from scratch and tested on the COCO validation set. The variant achieves an AP of 72.5, which is lower than the 73.4 AP of HRNet-W32. We believe that the reason is that the low-level features extracted from the

[4] https://github.com/NVIDIA/flownet2-pytorch

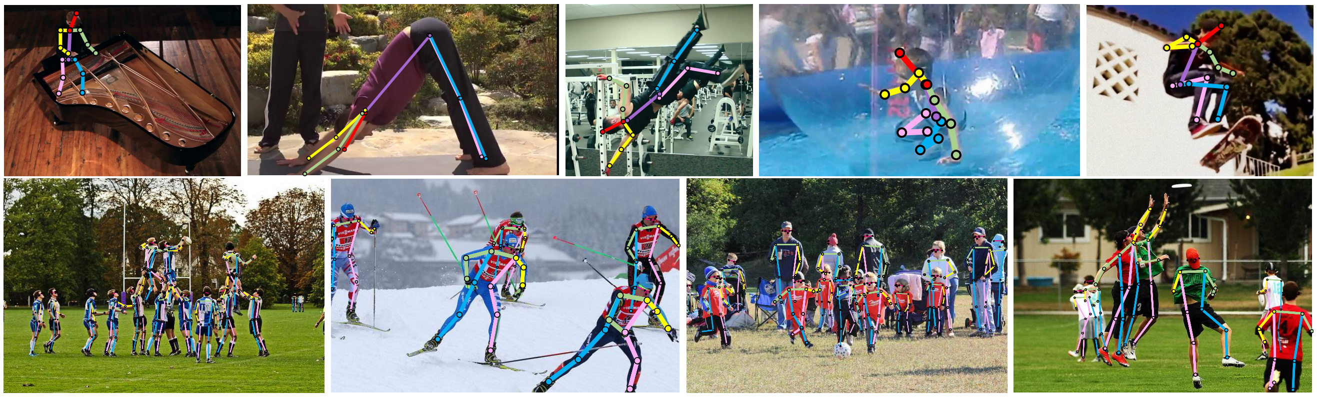


Figure 4. Qualitative results of some example images in the MPII (top) and COCO (bottom) datasets: containing viewpoint and appearance change, occlusion, multiple persons, and common imaging artifacts.

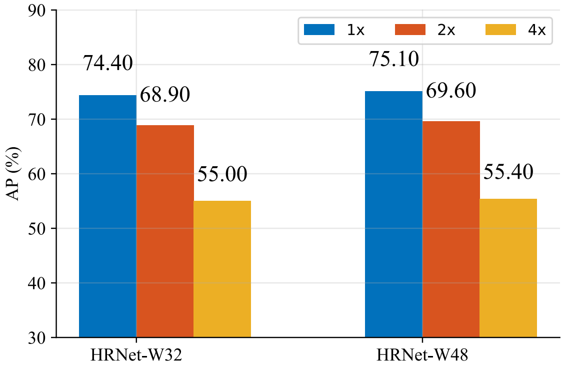


Figure 5. Ablation study of high- and low-resolution representations. 1×, 2×, 4× correspond to the representations of the high, medium, low resolutions, respectively.

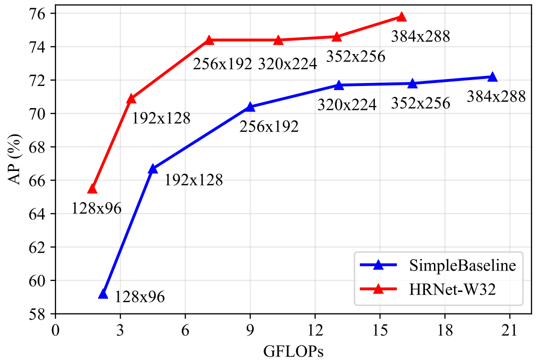


Figure 6. Illustrating how the performances of our HRNet and SimpleBaseline [70] are affected by the input size.

early stages over the low-resolution subnetworks are less helpful. In addition, the simple high-resolution network of similar parameter and GFLOPs without low-resolution parallel subnetworks shows much lower performance .

Representation resolution. We study how the representation resolution affects the pose estimation performance from two aspects: check the quality of the heatmap estimated from the feature maps of each resolution from high to low, and study how the input size affects the quality.

We train our small and big networks initialized by the model pretrained for the ImageNet classification. Our network outputs four response maps from high-to-low solutions. 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 5. The comparison implies that the resolution does impact the keypoint prediction quality.

Figure 6 shows how the input size affects the performance in comparison with SimpleBaseline (ResNet-50). Thanks to maintaining the high resolution through the whole process, the improvement for the smaller input size is more significant than the larger input size, e.g., the improvement is 4.0 points for 256 × 192 and 6.3 points for 128 × 96. This implies that our approach is more advantageous in the real applications where the computation cost is also an important factor. On the other hand, our approach with the input size 256 × 192 outperforms the SimpleBaseline with the large input size of 384 × 288.

5. Conclusion and Future Works

In this paper, we present a high-resolution network for human pose estimation, yielding accurate and spatially-precise keypoint heatmaps. The success stems from two aspects: (i) maintain the high resolution through the whole process without the need of recovering the high resolution; and (ii) fuse multi-resolution representations repeatedly, rendering reliable high-resolution representations.

The future works include the applications to other dense prediction tasks[5], e.g., face alignment, object detection, semantic segmentation, as well as the investigation on aggregating multi-resolution representations in a less light way.

Acknowledgements. The authors thank Dianqi Li and Lei Zhang for helpful discussions. Dr. Liu was partially supported by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDB06040900.

[5] The project page is https://jingdongwang2017.github.io/Projects/HRNet/

References

[1] Mykhaylo Andriluka, Umar Iqbal, Anton Milan, Eldar Insafutdinov, Leonid Pishchulin, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking. In *CVPR*, pages 5167–5176, 2018. 2

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter V. Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, pages 3686–3693, 2014. 2, 6

[3] Vasileios Belagiannis and Andrew Zisserman. Recurrent human pose estimgation. In *FG*, pages 468–475, 2017. 3

[4] Adrian Bulat and Georgios Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *ECCV*, volume 9911 of *Lecture Notes in Computer Science*, pages 717–732. Springer, 2016. 2, 6

[5] Zhaowei Cai, Quanfu Fan, Rogério Schmidt Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *ECCV*, pages 354–370, 2016. 3

[6] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, pages 1302–1310, 2017. 1, 5

[7] João Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *CVPR*, pages 4733–4742, 2016. 2

[8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan 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. 3

[9] Xianjie Chen and Alan L. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *NIPS*, pages 1736–1744, 2014. 2

[10] Yu Chen, Chunhua Shen, Xiu-Shen Wei, Lingqiao Liu, and Jian Yang. Adversarial posenet: A structure-aware convolutional network for human pose estimation. In *ICCV*, pages 1221–1230, 2017. 6

[11] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. *CoRR*, abs/1711.07319, 2017. 2, 3, 5

[12] Chia-Jung Chou, Jui-Ting Chien, and Hwann-Tzong Chen. Self adversarial training for human pose estimation. *CoRR*, abs/1707.02439, 2017. 6

[13] Xiao Chu, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Structured feature learning for pose estimation. In *CVPR*, pages 4715–4723, 2016. 2

[14] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L. Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. In *CVPR*, pages 5669–5678, 2017. 2, 6

[15] Andreas Doering, Umar Iqbal, and Juergen Gall. Joint flow: Temporal flow fields for multi person tracking, 2018. 7

[16] Xiaochuan Fan, Kang Zheng, Yuewei Lin, and Song Wang. Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation. In *CVPR*, pages 1347–1355, 2015. 2

[17] Haoshu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. RMPE: regional multi-person pose estimation. In *ICCV*, pages 2353–2362, 2017. 1, 5

[18] Damien Fourure, Rémi Emonet, Élisa Fromont, Damien Muselet, Alain Trémeau, and Christian Wolf. Residual conv-deconv grid network for semantic segmentation. In *British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017*, 2017. 3

[19] Rohit Girdhar, Georgia Gkioxari, Lorenzo Torresani, Manohar Paluri, and Du Tran. Detect-and-track: Efficient pose estimation in videos. In *CVPR*, pages 350–359, 2018. 7

[20] Georgia Gkioxari, Alexander Toshev, and Navdeep Jaitly. Chained predictions using convolutional neural networks. In *ECCV*, pages 728–743, 2016. 2

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988, 2017. 5

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

[23] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens van der Maaten, and Kilian Q. Weinberger. Multi-scale dense convolutional networks for efficient prediction. *CoRR*, abs/1703.09844, 2017. 3

[24] Shaoli Huang, Mingming Gong, and Dacheng Tao. A coarse-fine network for keypoint localization. In *ICCV*, pages 3047–3056. IEEE Computer Society, 2017. 5

[25] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, pages 1647–1655, 2017. 7

[26] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *ECCV*, pages 34–50, 2016. 1, 2, 3, 6

[27] Umar Iqbal, Anton Milan, and Juergen Gall. Posetrack: Joint multi-person pose estimation and tracking. In *CVPR*, pages 4654–4663, 2017. 6, 7

[28] Sheng Jin, Xujie Ma, Zhipeng Han, Yue Wu, Wei Yang, Wentao Liu, Chen Qian, and Wanli Ouyang. Towards multi-person pose tracking: Bottom-up and top-down methods. In *ICCV PoseTrack Workshop*, 2017. 7

[29] Angjoo Kanazawa, Abhishek Sharma, and David W. Jacobs. Locally scale-invariant convolutional neural networks. *CoRR*, abs/1412.5104, 2014. 3

[30] Lipeng Ke, Ming-Ching Chang, Honggang Qi, and Siwei Lyu. Multi-scale structure-aware network for human pose estimation. *CoRR*, abs/1803.09894, 2018. 2, 3, 6

[31] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 5

[32] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Multiposenet: Fast multi-person pose estimation using pose residual network. In *ECCV*, volume 11215 of *Lecture Notes in Computer Science*, pages 437–453. Springer, 2018. 1, 5

[33] Chen-Yu Lee, Saining Xie, Patrick W. Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *AISTATS*, 2015. 3

[34] Ita Lifshitz, Ethan Fetaya, and Shimon Ullman. Human pose estimation using deep consensus voting. In *ECCV*, pages 246–260, 2016. 2, 3

[35] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755, 2014. 2, 4

[36] Diogo C. Luvizon, Hedi Tabia, and David Picard. Human pose regression by combining indirect part detection and contextual information. *CoRR*, abs/1710.02322, 2017. 6

[37] Anton Milan, Laura Leal-Taixé, Ian D. Reid, Stefan Roth, and Konrad Schindler. MOT16: A benchmark for multi-object tracking. *CoRR*, abs/1603.00831, 2016. 7

[38] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *NIPS*, pages 2274–2284, 2017. 1, 5

[39] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, pages 483–499, 2016. 1, 2, 3, 5, 6

[40] Xuecheng Nie, Jiashi Feng, Junliang Xing, and Shuicheng Yan. Pose partition networks for multi-person pose estimation. In *ECCV*, September 2018. 1

[41] Xuecheng Nie, Jiashi Feng, and Shuicheng Yan. Mutual learning to adapt for joint human parsing and pose estimation. In *ECCV*, September. 2

[42] Xuecheng Nie, Jiashi Feng, Yiming Zuo, and Shuicheng Yan. Human pose estimation with parsing induced learner. In *CVPR*, June 2018. 2

[43] Guanghan Ning, Zhi Zhang, and Zhiquan He. Knowledge-guided deep fractal neural networks for human pose estimation. *IEEE Trans. Multimedia*, 20(5):1246–1259, 2018. 6

[44] Wanli Ouyang, Xiao Chu, and Xiaogang Wang. Multi-source deep learning for human pose estimation. In *CVPR*, pages 2337–2344, 2014. 2

[45] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *ECCV*, September 2018. 1, 5

[46] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *CVPR*, pages 3711–3719, 2017. 1, 5

[47] Xi Peng, Zhiqiang Tang, Fei Yang, Rogerio S. Feris, and Dimitris Metaxas. Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation. In *CVPR*, June 2018. 2

[48] Tomas Pfister, James Charles, and Andrew Zisserman. Flowing convnets for human pose estimation in videos. In *ICCV*, pages 1913–1921, 2015. 1

[49] Leonid Pishchulin, Mykhaylo Andriluka, Peter V. Gehler, and Bernt Schiele. Poselet conditioned pictorial structures. In *CVPR*, pages 588–595, 2013. 2

[50] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V. Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *CVPR*, pages 4929–4937, 2016. 3, 7

[51] Tobias Pohlen, Alexander Hermans, Markus Mathias, and Bastian Leibe. Full-resolution residual networks for semantic segmentation in street scenes. In *CVPR*, 2017. 3

[52] PoseTrack. PoseTrack Leader Board. https://posetrack.net/leaderboard.php. 7

[53] Mohamed Samy, Karim Amer, Kareem Eissa, Mahmoud Shaker, and Mohamed ElHelw. Nu-net: Deep residual wide field of view convolutional neural network for semantic segmentation. In *CVPRW*, June 2018. 3

[54] Shreyas Saxena and Jakob Verbeek. Convolutional neural fabrics. In *NIPS*, pages 4053–4061, 2016. 3

[55] Taiki Sekii. Pose proposal networks. In *ECCV*, September 2018. 1

[56] Ke Sun, Cuiling Lan, Junliang Xing, Wenjun Zeng, Dong Liu, and Jingdong Wang. Human pose estimation using global and local normalization. In *ICCV*, pages 5600–5608, 2017. 2, 6

[57] Ke Sun, Mingjie Li, Dong Liu, and Jingdong Wang. IGCV3: interleaved low-rank group convolutions for efficient deep neural networks. In *BMVC*, page 101. BMVA Press, 2018. 3

[58] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, pages 536–553, 2018. 5

[59] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 3

[60] Wei Tang, Pei Yu, and Ying Wu. Deeply learned compositional models for human pose estimation. In *ECCV*, September 2018. 2, 6

[61] Zhiqiang Tang, Xi Peng, Shijie Geng, Lingfei Wu, Shaoting Zhang, and Dimitris N. Metaxas. Quantized densely connected u-nets for efficient landmark localization. In *ECCV*, pages 348–364, 2018. 6

[62] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *CVPR*, pages 648–656, 2015. 3

[63] Jonathan J. Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, pages 1799–1807, 2014. 2

[64] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, pages 1653–1660, 2014. 2

[65] Jingdong Wang, Zhen Wei, Ting Zhang, and Wenjun Zeng. Deeply-fused nets. *CoRR*, abs/1605.07716, 2016. 3

[66] Zhicheng Wang, Wenbo Li, Binyi Yin, Qixiang Peng, Tianzi Xiao, Yuming Du, Zeming Li, Xiangyu Zhang, Gang Yu, and Jian Sun. Mscoco keypoints challenge 2018. In *Joint Recognition Challenge Workshop at ECCV 2018*, 2018. 5

[67] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, pages 4724–4732, 2016. 3, 6

[68] Jiahong Wu, He Zheng, Bo Zhao, Yixin Li, Baoming Yan, Rui Liang, Wenjia Wang, Shipei Zhou, Guosen Lin, Yanwei Fu, et al. Ai challenger: A large-scale dataset for going deeper in image understanding. *arXiv preprint arXiv:1711.06475*, 2017. 6

[69] Fangting Xia, Peng Wang, Xianjie Chen, and Alan L. Yuille. Joint multi-person pose estimation and semantic part segmentation. In *CVPR*, pages 6080–6089, 2017. 1

[70] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, pages 472–487, 2018. 1, 2, 3, 5, 6, 7, 8

[71] Guotian Xie, Jingdong Wang, Ting Zhang, Jianhuang Lai, Richang Hong, and Guo-Jun Qi. Interleaved structured sparse convolutional neural networks. In *CVPR*, pages 8847–8856. IEEE Computer Society, 2018. 3

[72] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *ICCV*, pages 1395–1403, 2015. 3

[73] Yichong Xu, Tianjun Xiao, Jiaxing Zhang, Kuiyuan Yang, and Zheng Zhang. Scale-invariant convolutional neural networks. *CoRR*, abs/1411.6369, 2014. 3

[74] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. In *ICCV*, pages 1290–1299, 2017. 1, 2, 3, 5, 6

[75] Wei Yang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *CVPR*, pages 3073–3082, 2016. 2

[76] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, pages 1385–1392, 2011. 2

[77] Ting Zhang, Guo-Jun Qi, Bin Xiao, and Jingdong Wang. Interleaved group convolutions. In *ICCV*, pages 4383–4392, 2017. 3

[78] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 6230–6239, 2017. 3

[79] Liming Zhao, Mingjie Li, Depu Meng, Xi Li, Zhaoxiang Zhang, Yueting Zhuang, Zhuowen Tu, and Jingdong Wang. Deep convolutional neural networks with merge-and-run mappings. In *IJCAI*, pages 3170–3176, 2018. 3

[80] Yisu Zhou, Xiaolin Hu, and Bo Zhang. Interlinked convolutional neural networks for face parsing. In *ISNN*, pages 222–231, 2015. 3

[81] Xiangyu Zhu, Yingying Jiang, and Zhenbo Luo. Multi-person pose estimation for posetrack with enhanced part affinity fields. In *ICCV PoseTrack Workshop*, 2017. 7