Neural Architecture Search for Joint Human Parsing and Pose Estimation

Dan Zeng[1] Yuhang Huang[†1] Qian Bao[2] Junjie Zhang[1] Chi Su[3] Wu Liu[*2]
[1]Shanghai University [2]AI Research of JD.com [3]Kingsoft Cloud
dzeng@shu.edu.cn, huangyuhang@shu.edu.cn, baoqian@jd.com
junjie_zhang@shu.edu.cn, suchi@kingsoft.com, liuwu1@jd.com

Abstract

Human parsing and pose estimation are crucial for the understanding of human behaviors. Since these tasks are closely related, employing one unified model to perform two tasks simultaneously allows them to benefit from each other. However, since human parsing is a pixel-wise classification process while pose estimation is usually a regression task, it is non-trivial to extract discriminative features for both tasks while modeling their correlation in the joint learning fashion. Recent studies have shown that Neural Architecture Search (NAS) has the ability to allocate efficient feature connections for specific tasks automatically. With the spirit of NAS, we propose to search for an efficient network architecture (NPPNet) to tackle two tasks at the same time. On the one hand, to extract task-specific features for the two tasks and lay the foundation for the further searching of feature interaction, we propose to search their encoder-decoder architectures, respectively. On the other hand, to ensure two tasks fully communicate with each other, we propose to embed NAS units in both multi-scale feature interaction and high-level feature fusion to establish optimal connections between two tasks. Experimental results on both parsing and pose estimation benchmark datasets have demonstrated that the searched model achieves state-of-the-art performances on both tasks. [1]

1. Introduction

Human parsing and pose estimation are two key tasks for analyzing human behaviors. Human parsing aims to segment the human body into different semantic regions, while pose estimation is to locate the keypoints of the human body and analyze structural information. Visually speaking, these two tasks are closely related. On the one hand, keypoints are often included within the different semantic regions, indicates that the semantic information from human parsing can help locate these points for accurate pose estimation. On the other hand, the group of keypoints naturally possesses rich structural information, which can guide the generation of semantic parts.

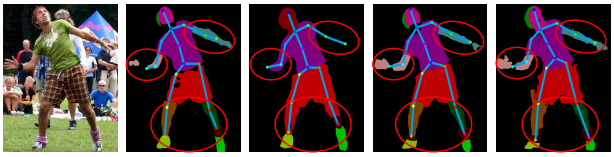

Figure 1. The illustration of our motivation. As we can see, manually designed frameworks [21,31] can not improve pose estimation and human parsing simultaneously. There is an obvious inconsistency between the two tasks, which suggests it is challenging for manually designed modules to extract task-specific features and enhance their correlation, *e.g.*, in MuLA [31], some keypoints are not surrounded by the corresponding semantic parts. This motivates us to search for a better network architecture that allows two tasks to fully benefit from each other.

With the advanced learning ability of the deep neural network, it becomes a general practice to deploy convolutional neural networks (CNN) to tackle these two tasks [28, 40]. The majority of existing models share a similar basic encoder-decoder architecture with different learning objectives. For human parsing, they [26, 36, 44] aim at mapping the up-sampled output to pixel-wise annotations, while the ground-truth of pose estimation [6, 30, 42] corresponds to the heatmaps of sparse keypoints. Given the correlation between two tasks, there are recent developments [21, 31, 41, 46] that attempt to perform joint inference via neural networks from the multi-task learning perspective. These models conduct the representation learning through one shared [21] or two separate encoder-decoder structure [31,46], and design the hand-crafted modules [31] to interact high-level features extracted for two tasks.

However, designing the suitable network architecture and optimal feature interaction for joint learning is challenging. On the one hand, though the two tasks are visually related, they still possess their own characteristics. The general solutions for pose estimation focus on aggregating information into small joint areas while human parsing needs to explore the pixel-wise context information. There-

†This work is done when Yuhang Huang is an intern at AI Research of JD.com.
* Wu Liu is the corresponding author.
[1]The code will be public at https://github.com/GuHuangAI/NPP.

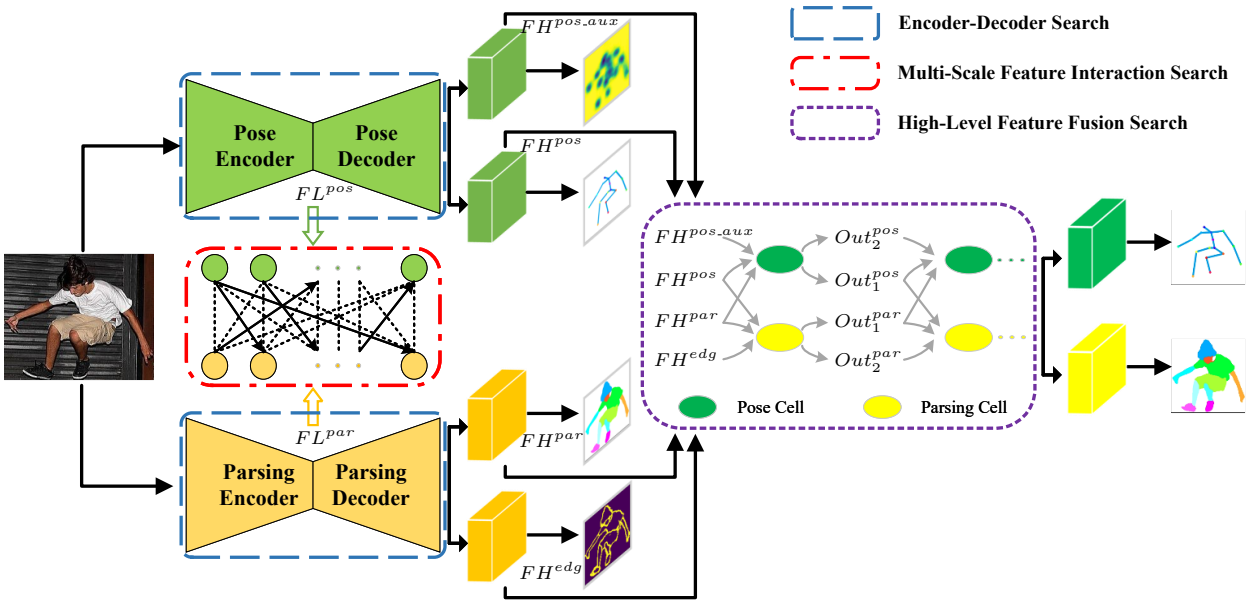

Figure 2. The overall framework of NPPNet. The input image passes through two task-specific encoder-decoder structures to extract two sets of features with multiple scales, each set absorbs information from the other one in the multi-scale feature interaction structure. The two encoder-decoder structures output two sets of high-level features including main and auxiliary features and generate the initial prediction. Then the high-level feature fusion structure fuses these features and commits the final prediction. The entire network architecture is searchable.

fore, it is challenging to extract discriminative features for both tasks when tackling them simultaneously. On the other hand, it is difficult to model the correlation between the two tasks. Existing works [31, 46] address this issue by manually designing the fusion modules to interact the high-level features from the two branches. Yet it is rather rigid and ignores the diverse intermediate features with multi-scale information, considering different levels of features require the more refined interaction. As shown in Fig. 1, existing frameworks for joint learning cannot obtain consistent results for both tasks, which motivates us to design a more efficient network architecture.

Recent studies [10, 18, 20, 24] have shown that Neural Architecture Search (NAS) is able to flexibly search for efficient architectures for various vision tasks [13,45], including parsing [23,29] and pose estimation [2]. Different from them, in this paper, we propose to employ search for the joint learning framework to tackle both tasks. We name our searched model as NPPNet. As shown in Fig. 2, on the one hand, to extract task-specific features for both tasks and lay the foundation for the further searching of feature interaction, we propose to search for their encoder-decoder architectures, respectively. On the other hand, to ensure both intermediate and high-level features of two tasks interact with each other, we propose to embed NAS units in both multi-scale feature interaction and high-level feature fusion to establish optimal connections between two tasks.

In this paper, we design three types of search spaces tailed for joint human parsing and pose estimation. By applying the different search spaces during feature extraction, interaction, and fusion, NPPNet improves the performance of both tasks simultaneously. In summary, the main contributions of this paper as follows:

- We propose an end-to-end network NPPNet entirely searched by NAS, which conducts human parsing and pose estimation simultaneously. To the best of our knowledge, this is one of the first attempts towards using NAS to tackle two tasks simultaneously.
- To extract the discriminative features for both tasks and allow them to benefit from each other, we design three searching spaces for constructing the task-specific encoder-decoder structure, the multi-scale feature interaction, and the high-level feature fusion.
- Extensive experiments are conducted on the LIP and the extended PASCAL-Person-Part datasets. Results show that our proposed NPPNet achieves state-of-the-art performances on both tasks.

2. Related works

Pose Estimation & Human Parsing. Early pose estimation methods [5, 34, 38] directly regress the coordinates of keypoints, but perform poorly when facing flexible human movements. Instead, CNN-based methods [1,3,19,25,37,40] predicts the 2D heatmap of each joint. An encoder-decoder structure is usually used to generate multi-scale feature maps. [30] proposes to use skip connections

to merge the same resolution features to fuse different scale features. Besides, [6, 19, 30, 40] add additional supervisions to refine the initial output. Our NPPNet is inspired by the pyramid feature fusion and utilizes the refinement layer to further enhance the prediction. Similar to pose estimation methods, many studies [12,36,44] on human parsing extract multi-scale features by reducing the resolution of feature maps gradually and then up-sample them through a decoder. Given human parsing is a fine-grained task, there are various methods to utilize the multi-task objectives to improve the parsing performance. For example, [36] adds an additional branch to predict the edge of different regions, [12] uses human body structure information to assist parsing. Moreover, [21,31,36] add the refinement branch to improve the initial accuracy. We also conduct edge prediction and parsing refinement in our framework. However, different from existing works, these learning objectives are guided by the interactions between two tasks.

Joint Learning of Two Tasks. Considering the correlation between two tasks, there is a line of work that explores the joint training paradigm. In [8], Dong *et al.* propose an And-Or graph model, which uses the spatial information of keypoints to assist the parsing and also constrain their locations with the regions generated from parsing. In the first neural network-based method [41], authors associate high-level features from two tasks through convolutions and conditional random field. However, the whole framework is trained stage-wisely. A shared backbone is used to extract common features for both tasks in [21]. However, the performance of pose estimation is far less satisfactory than parsing. In [31], two separate branches are utilized to extract pose and parsing features respectively, and hand-designed modules are proposed to interact two branches, while in [46], an attention-based module is used to fuse the pose, edge, and parsing features. Though these methods focus on explicitly modeling the correlations between tasks, the manually designed interactions are sub-optimal since the results of two tasks often present themselves in an inconsistent way. To address the above issues, we propose to design three types of search spaces and apply NAS to find the network architecture that is beneficial for both tasks, including the search of encoder-decoder, the multi-scale feature interaction, as well as the high-level feature fusion.

Neural Architecture Search. The recent developments of NAS-related algorithms can be roughly grouped into three types gradient-based [24], reinforcement learning [48], and evolutionary algorithm [35]. Among them, the gradient-based method PC-DARTS [43] proposes partial channel connection, which reduces the redundant space by sampling a small part of the supernet and significantly improves search efficiency. NAS has been applied to various vision tasks to search for the suitable network architecture, including image classification [10], object detection [13], semantic segmentation [23, 45], pose estimation [2], multi-task learning *etc.* In [45], a Pooled Conv operation is proposed to constrain the computation complexity for semantic segmentation. In [2], authors make the first attempt to employ NAS to search for the entire network architecture for pose estimation. Note that MTL-NAS [11] uses a two-branch architecture to solve joint learning but it only searches the interaction part and its search space is not flexible enough that it just searches which node of one task should be connected to the other but ignores which operations should be applied to the node. Different from prior works, we adopt the PC-DARTS as our main search algorithm and design an entire searchable framework to address the specific joint tasks.

3. Methods

3.1. Overview

In this section, we introduce the details of NPPNet, including the overall network architecture and its components. As shown in Fig. 2, we view the forward process of NPPNet in two steps.

Firstly, to ensure the discriminative features are extracted for each task, we adopt a two-branch structure. That is, each task has its own task-specific encoder-decoder. The pose branch generates both main and auxiliary pose features, while the parsing and edge features are obtained through the parsing branch. We formulate the cell-based space to search for the optimal network architecture. Secondly, since two tasks are also highly correlated, to build more interactions and free the interaction between tasks from manually designed modules, we propose to search the connections between intermediate features as well as between high-level features. Specifically, for intermediate feature interaction, we densely connecting multi-scale features from two branches to let two tasks fully communicate with each other. On the other hand, the searching for the fusion of high-level features is conducted by feeding four features, the main pose and parsing features as well as the auxiliary pose feature and edge feature, to the designed pose and parsing cells. With the optimization of multi-task loss, PC-DARTS search strategy, and the proposed search spaces, we obtain the optimal network architecture, which is then trained for further inference.

3.2. Search Algorithm

PC-DARTS [43] has demonstrated superior performances in searching for classification architecture while keeping memory usage constrained. Therefore, we employ it as the basic searching method in our framework and customize it to suit our purpose. Specifically, a searchable cell can be represented as a Directed Acyclic Graph (DAG) con-

taining N nodes. Each node is connected to all the nodes before it, while each connection represents a mixed operation. For the mixed operation, in order to reduce the memory cost, PC-DARTS performs the partial channel connection on the feature of the input node. It samples and extracts $1/k$ channels for the computation [2], in this way, the mixed operation can be formulated as:

$$\text{MixOP}(x) = [\sum_{o \in \mathcal{O}} \frac{\exp(\alpha^o)}{\sum_{o' \in \mathcal{O}} \exp(\alpha^{o'})} \cdot o(x_{k+}); x_{k-}] \quad (1)$$

where x_{k+} is the sampled $1/k$ channels of input tensor, and x_{k-} is the remaining channels. $\mathcal{O}$ is a predefined set of operations, and $o(\cdot)$ represents one of the operation from $\mathcal{O}$, such as 3x3 conv *etc.* α^o indicates attaching a weight to the given operation $o(\cdot)$, and $[;]$ stands for concatenating tensors on the channel dimension.

To stabilize the searching process, the partial channel connection adds a weight $\beta_{i,j}$ to each connection from node j to node i, the computation of node x_i is:

$$\begin{aligned} x_i &= \sum_{j<i} \frac{\exp(\beta_{i,j})}{\sum_{j'<i} \exp(\beta_{i,j'})} \cdot \text{MixOP}_{i,j}(x_j) \\ \gamma^o_{i,j} &= \frac{\exp(\beta_{i,j})}{\sum_{j'<i} \exp(\beta_{i,j'})} \cdot \frac{\exp(\alpha^o_{i,j})}{\sum_{o' \in \mathcal{O}} \exp(\alpha^{o'}_{i,j})} \end{aligned} \quad (2)$$

Here, $\text{MixOP}_{i,j}$ indicates attaching the operation weight vector $\alpha_{i,j}$. $\alpha_{i,j}$ and $\beta_{i,j}$ are updated by gradients, and the contribution of the operation $o(\cdot)$ in connection from node j to node i can be expressed as $\gamma^o_{i,j}$. After the search is done, we perform the pruning scheme to keep the operations that contribute the most. We adopt the above searching algorithm throughout the proposed framework and design the unique search space for each search type. We denote all architecture parameters as: $\mathcal{A} = \{\alpha, \beta\}$.

Inspired by the related works, the operation set we adopt is as follows:

- stand conv 3x3
- stand conv 1x1
- max pooling 3x3
- skip connect
- dilated conv 3x3 with rate 2
- SE connect
- dilated conv 3x3 with rate 4
- pooled conv

We use two types of dilated convolution with different dilation rates in the searching space since the larger receptive field is favorable for both parsing and pose estimation, and the SE connect [16] helps to enhance the spatial-wise and channel-wise information. The pooled conv [45] reduces the computation complexity while enlarges the receptive field, and the skip connection is selected to prevent the gradient from disappearing.

3.3. Encoder-Decoder Search

Though there are visual correlations between pose estimation and human parsing in extracting structural and semantic information from the human body, the differences between these two tasks inevitably require the discriminative features learned for their own. Therefore, it is challenging to improve performances for both of them simultaneously with existing encoder-decoder structures. As shown in Fig. 3, we have conducted a series of experiments to examine this issue. (a) shows the attention map when employing a shared encoder (ResNet50-based) for two tasks. It can be seen that the network is unable to focus on the human body, which indicates that each task requires its own task-specific feature. (b) illustrates the attention map after two separate encoders are applied. Though it is improved compared to (a), there are still many areas of ambiguity, which leaves room for further improvements.

To address the above issue and obtain a flexible architecture that is free from the manual design, given its potential to search for efficient network architectures, we propose to utilize NAS to search for the customized encoder-decoder

[2] We empirically set $k = 2$ in our experiments.

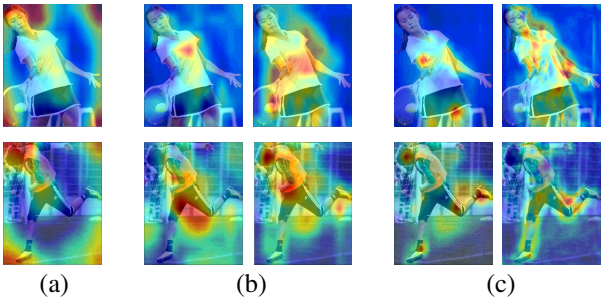

Figure 3. Feature maps generated by different methods. (a) is generated by shared encoder architecture, (b) is generated by manually designed unshared encoder architecture, and (c) is generated by an unshared encoder-decoder structure searched by our method. In both (b) and (c), the first column is generated from the pose branch, and the second column is generated by the parsing branch. Compared to other methods, our model learns more discriminative features for both tasks.

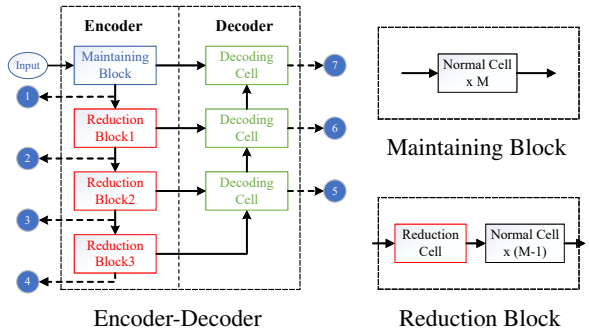

Figure 4. The details of encoder-decoder architecture. Our encoder design the normal cell and reduction cell by referring to DARTS [24]. And the decoding cell is modified from the normal cell and can process two features with different scales. We set M to 1/4 of the total number of cells in the encoder. Each encoder-decoder generates 7 intermediate features for further interaction.

structure of each task, which also lies the foundation for the feature interaction and fusion. More specifically, we use a cell-based search space to construct a searchable architecture. For the encoder, we use a stack of normal cells and reduction cells. As for the decoder, we design the decoding cell based on the normal cell, which accepts two input features with different scales, of which features with low scales are upsampled first. We use three decoding cells to build the search space which is similar to FPN [22]. Fig. 4 shows our encoder-decoder search space. It is worth noting that the search spaces of the two branches are the same, but we do not share their architecture parameters. Fig. 3 (c) shows the attention map after we use NAS to search the encoder-decoder structure for two branches. It can be seen that two branches focus on the respective areas that are effective for each task.

3.4. Multi-Scale Feature Interaction Search

The search of encoder-decoder structure extracts discriminative features for both tasks by maintaining a two-branch architecture. However, it requires necessary connections between two branches to model their correlation. Therefore, we propose a feature interaction search to allow multi-scale features generated by two branches to benefit from each other. As shown in Fig. 4, we denote the features generated by the pose branch in the encoder-decoder structure as below:

$$\{FL_i^{pos} \in \mathbb{R}^{j \cdot C \times H/j \times W/j}\} \tag{3}$$

where $j = 2^{i-1}, i \in [1, 4]$ among encoder outputs, and $j = 2^{7-i}, i \in [5, 7]$ among decoder outputs. (H, W) is the one fourth of the input size. Similarly, the features of the parsing branch can be expressed as $\{FL_i^{par}(\cdot), i \in [1-7]\}$, and the shape is the same as the pose feature. The two sets of multi-scale features are generated in a down-sampling and then up-sampling fashion.

Our objective here is to find the optimal connections between these two sets of features. On the one hand, we expect that the communication between two sets travel across scales so that the search space can be increased. On the other hand, we tend to avoid cycles when constructing the search space. Therefore, we design a densely connected search space, allowing the network to discover efficient connections by itself. As shown in the red dashed box in Fig. 2, We represent these features as two sets of nodes. Each node is connected with all the previous nodes from another branch, among which each connection indicates a mixed operation. Each node can be computed as:

$$FL_i^{pos'} = \sum_{j \le i} \beta l_{i,j}^{pos} \cdot \text{MixOP}_{i,j}^{pos}(FL_j^{par}) + FL_i^{pos} \tag{4}$$

$$FL_i^{par'} = \sum_{j \le i} \beta l_{i,j}^{par} \cdot \text{MixOP}_{i,j}^{par}(FL_j^{pos}) + FL_i^{par} \tag{5}$$

where the input of MixOP$(\cdot)$ in each branch comes from the other one, though the search space of the proposed interaction search is not cell-based, the searching algorithm described in Sec. 3.2 can still be applied.

3.5. High-Level Feature Fusion Search

As the input image passes through the encoder-decoder, there are four high-level features generated. Each branch contains two types: one is the main features for the given task, the other is the auxiliary features to help the prediction of the main task. With these four features, we consider utilizing the NAS to search for the way to fuse them, which helps them to benefit from each other's abundant structural and semantic information.

Specifically, we include the edge prediction as an auxiliary task to the parsing branch. Moreover, in order to balance the pose branch, we also add an auxiliary branch to it, the ground-truth of which is the heatmap after the Gaussian blurring. The two sets of high-level features can be expressed as: $\{FH^{pos}, FH^{pos\_aux}\}, \{FH^{par}, FH^{edg}\}$ respectively, and the sizes of FH^* are all (C, H, W). We concatenate $\{FL_i, i \in [5, 7]\}$ from the outputs of encoder-decoder, and use two 1×1 convolutions to generate the main and auxiliary features respectively:

$$\begin{aligned} FH^m &= \phi([FL_5^{pos}; FL_6^{pos}; FL_7^{pos}]) \\ FH^n &= \phi([FL_5^{par}; FL_6^{par}; FL_7^{par}]) \end{aligned} \tag{6}$$

where $\phi(\cdot)$ stands for the 1×1 convolution, and $m \in \{pos, pos\_aux\}$, $n \in \{par, edg\}$ represent two branches. At last, the two sets of features are used as the inputs of high-level feature fusion.

To allow them fully interacted, we design a parallel stacked cell-based search space. The basic normal cell only has two input nodes and one output node, which indicates a relatively smaller search space. To input more features and enlarge the search space, we design two new cells, named pose cell and parsing cell, which both have three input nodes, four intermediate nodes, and two output nodes. The search space is thereby expanded. As shown in purple dashed box in Fig. 2, we set the input nodes of pose cell as: $\{FH^{pos}, FH^{pos\_aux}, FH^{par}\}$, and the input nodes of parsing cell as: $\{FH^{par}, FH^{edg}, FH^{pos}\}$. The first output node of the pose cell is obtained by concatenating four intermediate nodes, while we generate the second output node by concatenating three input nodes. In this way, we utilize all available nodes and establish connections among them. Moreover, we fuse features from two branches iteratively by stacking the pose cell and the parsing cell to enhance the interaction among them.

3.6. Training & Optimization

We apply a Gaussian heatmap generated from keypoints to supervise the pose estimation and use the Mean Square

Error as the loss function. The same setting is adopted for the pose auxiliary task. For training the parsing branch, Cross-Entropy loss is employed for two sets of predictions. In order to adjust the proportions of these losses, we adopt the uncertainty loss [17] to learn the weights for each loss. More details can be referred to the supplementary materials.

In the searching phase, the network has two sets of parameters, the architecture parameters $\mathcal{A}$ and the weight parameters $\mathcal{W}$. We use an alternate optimization strategy in each epoch:

$$\begin{aligned} \mathcal{W}^* &= \arg\min_{\mathcal{W}} L_{tra\_w}(\mathcal{W}, \mathcal{A}^*) \\ \mathcal{A}^* &= \arg\min_{\mathcal{A}} L_{tra\_\alpha}(\mathcal{W}^*, \mathcal{A}) + E(\mathcal{A}) \end{aligned} \tag{7}$$

where tra_w and tra_α are two equal parts randomly divided in the training set, $E(\mathcal{A})$ is the entropy of architecture parameters, which is calculated by:

$$E(\mathcal{A}) = \sum_{\alpha_{i,j} \in \mathcal{A}} -\frac{\exp(\alpha_{i,j}^o)}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{i,j}^{o'})} \cdot \log(\frac{\exp(\alpha_{i,j}^o)}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{i,j}^{o'})}) \tag{8}$$

In order to address the instability of the search algorithm, we use $E(\mathcal{A})$ as a regularization to constrain the distribution of architecture parameters to be more concentrated so that after the network converges and prunes, the remaining connections will occupy a large proportion, and the performance gap between the searching phase and the retraining phase can be narrowed. In the retraining phase, we use the full training set, and we only need to update weight parameters $\mathcal{W}$.

4. Experiments

4.1. Datasets & Metrics

We evaluate our proposed NPPNet on two benchmark datasets that contain both human parsing and pose estimation annotations: the LIP [21] and the extended PASCAL-Person-Part [41]. LIP contains 50,462 images with elaborated pixel-wise annotations of 19 semantic human part labels and 2D human poses with 16 keypoints, of which 30,462 are used for training, 10,000 for validation, and 10,000 for testing. The extended PASCAL-Person-Part is a challenging multi-person dataset containing annotations for 14 body joints and 6 semantic parts. There are 3,533 images, which are split into 1,716 for training and 1,817 for testing. Since the extended PASCAL-Person-Part dataset is a multi-person dataset, we use Mask R-CNN [14] to generate single-person images. For the LIP dataset, following [21], we use Mean Intersection-over-Union (mIOU) to evaluate human parsing performance and Percentage of Correct Keypoints (PCK) to evaluate pose estimation. For the extended PASCAL-Person-Part, we use mIOU and Mean Average Precision (mAP), respectively.

4.2. Implementation Details

In the searching phase, we set C in Sec. 3.4 and Sec. 3.5 to 64. The total number of cells L in the encoder is set to 12, and the number of pose cells and the parsing cells are both 3. The entire search process is divided into two stages. First, we only search for the encoder-decoder structure for 50 epochs, then we fix its architecture and search for the feature interaction and fusion architectures for another 50 epochs. In both stages, we use the Adam optimizer with an initial learning rate of 0.001 and drop it to one-tenth at epoch 35 and 45. In the retraining phase, we set C and L to 128 and 16, respectively, and keep the number of pose cells and parsing cells unchanged. The input size of the image is 384×384, and we conduct a series of data augmentations, including random rotation in $[-40°, 40°]$, random scaling in [0.5, 1.5], and random left-right flipping. We employ Adam as the optimizer with an initial learning rate of 0.001, and we train NPPNet for 120 epochs and drop the learning rate to one-tenth at epoch 85 and 105. In the inference phase, by referring to [31], we use flipping and multi-scale with an interval of 0.25 from 0.5 to 1.5. We test our network on a 12GB TITAN V for a fair comparison and NPPNet achieves 7 FPS which is faster than JPPNet (2 FPS) and MuLA (5 FPS).

4.3. Search on LIP

We conduct searching experiments on the LIP dataset and search for 5 times. During each search, the training set is randomly divided into two equal parts, corresponding to $train\_w$ and $train\_\alpha$ in Sec. 3.6. And we select the best performing architecture from them. The entire searching takes 3 days on four P40 GPUs.

After the search, we combine two pruning schemes. First of all, since the search of encoder-decoder and feature fusion are both cell-based, we apply the pruning scheme in DARTS [24]. However, the search for the multi-scale feature interaction is not cell-based, and the connections are much denser. More valuable connections should be retained to ensure the efficiency of the network after pruning. Therefore, we design a new pruning scheme: for each node, we use $\gamma_{i,j}^o$ as the proportion of the operation $o(\cdot)$ in connection from node j to node i, and then select n operations with top n largest proportion. We constrain the sum of proportion over 0.7 and set $n \leq 4$ in our experiments. The pseudo-code of the above method and resulting architecture can be found in supplementary materials.

4.4. Experimental Results

Results on LIP. Tab. 1 shows the comparisons of human parsing between the proposed NPPNet and state-of-the-art models on the LIP validation set. As we can see, NPPNet achieves the highest mIOU of 58.56% as well as the best performance on 9 semantic parts. It is worth noting that

Table 1. The comparison of human parsing with state-of-the-art methods on the LIP validation set.

Method	bkg	hat	hair	glove	glass	u-clot	dress	coat	sock	pants	j-suit	scarf	skirt	face	l-arm	r-arm	l-leg	r-leg	l-shoe	r-shoe	mIOU
Attention [4]	84.00	58.87	66.78	23.32	19.48	63.20	29.63	49.70	35.23	66.04	24.73	12.84	20.41	70.58	50.17	54.03	38.35	37.70	26.20	27.09	42.92
MMAN [28]	84.75	57.66	65.63	30.07	20.02	64.15	28.39	51.98	41.46	71.03	23.61	9.65	23.20	69.54	55.30	58.13	51.90	52.17	38.58	39.05	46.81
SS-NAN [47]	88.67	63.86	70.12	30.63	23.92	70.27	33.51	56.75	40.18	72.19	27.68	16.98	26.41	75.33	55.24	58.93	44.01	41.87	29.15	32.64	47.92
CE2P [36]	87.67	65.29	72.54	39.09	32.73	69.46	32.52	56.28	49.67	74.11	27.23	14.19	22.51	75.5	65.14	66.59	60.1	58.59	46.63	46.12	53.1
HRNet [37]	87.55	68.69	73.31	40.41	34.27	70.41	32.42	57.89	47.31	74.79	25.01	25.52	29.77	76.11	66.11	67.95	59.77	60.29	44.59	45.79	54.40
BraidNet [26]	88.04	66.84	72.04	42.54	32.14	69.84	33.74	57.44	49.04	74.94	32.44	19.34	27.24	74.94	65.54	67.94	60.24	59.04	47.44	47.94	54.54
CorrPM [46]	87.77	66.20	71.56	41.06	31.09	70.20	37.74	57.95	48.40	75.19	32.37	23.79	29.23	74.36	66.53	68.61	62.80	62.81	49.03	49.82	55.33
PCNet [44]	88.68	69.32	73.08	44.72	34.21	**72.59**	36.02	**60.84**	51.03	76.66	**38.78**	31.60	**33.94**	76.65	67.07	68.74	60.22	60.16	47.65	48.67	57.03
CNIF [39]	87.99	69.55	73.45	45.17	**41.45**	70.57	38.52	57.94	**54.02**	75.07	28.00	**31.92**	30.20	76.38	68.28	69.49	65.52	65.51	52.67	53.38	57.74
DTCF [27]	**88.92**	**69.70**	**74.75**	45.66	40.73	71.50	36.52	59.16	52.39	76.97	30.93	29.36	31.33	**77.19**	68.57	70.25	64.52	64.04	51.57	52.30	57.82
MuLA [31]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	49.30
JPPNet [21]	86.25	63.55	70.20	36.16	23.48	68.15	31.42	55.65	44.56	72.19	28.39	18.76	25.14	73.36	61.97	63.88	58.21	57.99	44.02	44.09	51.37
NPPNet (ours)	88.38	66.43	72.34	**51.98**	31.59	71.88	**40.88**	60.54	49.81	**77.07**	27.55	26.58	33.54	75.31	**71.04**	**71.50**	**71.75**	**70.55**	**56.66**	**55.84**	**58.56**

Table 2. The comparisons of human pose estimation with state-of-the-art methods on the LIP test set.

Method	Head	Sho.	Elb.	Wri.	Hip	Knee	Ank.	PCK
CPM [40]	86.6	83.6	75.8	72.1	70.9	62.0	59.1	74.0
Hourglass [30]	86.4	84.7	77.5	73.9	74.0	63.3	58.4	75.2
Hybrid Pose Machine	71.7	87.1	82.3	78.2	69.2	77.0	73.5	77.2
BUPTMM-POSE	90.4	87.3	81.9	78.8	68.5	75.3	75.8	80.2
Pyramid Stream Network	91.1	88.4	82.2	79.4	70.1	80.8	81.2	82.1
Chou *et al.* [7]	94.9	93.1	89.1	86.5	75.7	85.5	85.7	87.4
GCCPM [33]	-	-	-	-	-	-	-	87.9
HRNet [37]	94.7	93.2	88.7	87.1	78.5	85.9	86.3	88.0
JPPNet [21]	93.3	89.3	84.4	82.54	70.0	78.3	77.7	82.7
MuLA [31]	94.9	93.1	89.9	87.6	75.9	84.9	84.4	87.5
NPPNet (ours)	**95.8**	**95.6**	**90.3**	**88.5**	**79.0**	**86.7**	**86.7**	**88.9**

NPPNet significantly outperforms other methods by 4-5% on glove, l-arm, r-arm, l-leg, r-leg, l-shoe, and r-shoe. It can be seen from Fig. 1, these classes often contain prominent keypoints, *e.g.* shoe areas contain ankles and arm areas include elbows, which validates that our model has captured the correlation between two tasks. Apart from models proposed only for human parsing, JPPNet [21] and MuLA [31] are two joint learning frameworks, yet unlike our model, they fail to improve the performances of both tasks.

Tab. 2 shows the comparison of the pose estimation performance on the LIP test set. The compared pose-only models include classic Hourglass network [30] and Convolutional Pose Machine [40], as well as their extensions BUPTMM-POSE and Hybrid Pose Machine. It can be seen that NPPNet surpasses them in a large margin, which validates the effectiveness of joint training of two tasks. Our model also outperforms the joint frameworks JPPNet and MuLA on the pose estimation and reports the best PCK 88.9%. We can see that NPPNet achieves the best PCK on each keypoint, which is benefited from the dense semantic information from the parsing branch.

More importantly, our model achieves the best performances on both tasks with 73.4M parameters and 113.7 GFlOPs that are lower than some joint frameworks (JPPNet [21]) and single-task frameworks (CNIF [39]), which shows its great potential to explore the interaction of the two branches. Please see the detail in the supplementary materials.

As shown at the top of Fig. 5, we compare NPPNet with several methods for the visualization of human parsing. It can be seen that NPPNet performs significantly better in areas containing prominent keypoints. This verifies that NPPNet can extract the structural information of keypoints to help infer the categories of the body regions. On the other hand, the bottom of Fig. 5 shows the comparison between NPPNet and related methods on pose estimation. As we can see, NPPNet can locate keypoints more accurately with the help of parsing information.

Table 3. The comparison of human parsing with state-of-the-art methods on the extended PASCAL-Person-Part validation set.

Method	bkg	head	torso	u-arm	l-arm	u-leg	l-leg	mIOU
Attention [4]	93.65	81.47	59.06	44.15	42.50	38.28	35.62	56.39
SS-NAN [47]	97.23	86.43	67.28	51.09	48.07	44.82	42.15	62.44
WSHP [9]	**97.72**	87.15	72.28	57.07	56.21	52.43	50.36	67.6
HRNet [37]	95.63	87.61	73.21	61.32	62.46	54.25	51.27	69.39
CNIF [39]	96.02	88.02	72.91	64.31	63.52	55.61	54.96	70.76
DTCF [27]	96.25	**88.32**	**73.54**	64.19	63.91	55.01	54.34	70.80
Xia *et al.* [41]	95.32	85.50	67.87	54.72	54.30	48.25	44.76	64.39
MuLA [31]	-	-	-	-	-	-	-	65.10
NPPNet (ours)	95.88	86.78	72.50	**66.33**	**63.95**	**58.06**	**58.59**	**71.73**

Table 4. The comparison of human pose estimation with state-of-the-art methods on the PASCAL-Person-Part validation set.

Method	Head	Sho.	Elb.	Wri.	Hip	Knee	Ank.	mAP
Chen and Yuille [40]	45.3	34.6	24.8	21.7	9.8	8.6	7.7	21.8
Insafutdinov *et al.*	41.5	39.3	34.0	27.5	16.3	21.3	20.6	28.6
PIL [32]	67.8	56.6	45.7	41.9	24.2	26.4	24.2	41.0
Xia *et al.* [41]	58.0	52.1	43.1	37.2	22.1	**30.8**	**31.1**	39.2
MuLA [31]	-	-	-	-	-	-	-	39.9
NPPNet (ours)	**68.1**	**57.9**	**46.1**	**42.0**	**27.3**	27.6	25.7	**42.1**

Results on extended PASCAL-Person-Part. For the extended PASCAL-Person-Part, we directly employ the architecture searched on the LIP dataset. Tab. 3 shows the performance of human parsing reported by related methods on the validation set. Among them, Xia *et al.* [41] and MuLA [31] are both joint learning methods, and PIL [32] utilize parsing to assist pose. Consistent with the observations on the LIP dataset, NPPNet outperforms other methods in areas where the joint points are prominent. Tab. 4 shows the comparison of the pose estimation performance on the validation set. NPPNet surpasses the best-compared model PIL by 1.1% with 42.1% mAP.

In summary, our NPPNet outperforms both joint learning and single-task models on two datasets, which verifies the effectiveness of the proposed searching method.

4.5. Ablation Studies

To verify the effectiveness of the three searches we proposed, we establish a baseline for comparison, where its

Table 5. Ablation studies on the LIP validation set.

Enc.-Dec.	Fea. Inter.	Fea. Fus.	mIOU	PCK
×	×	×	46.72	84.0
✓	×	×	51.13	85.6
×	✓	×	50.31	85.7
✓	✓	×	55.74	87.6
×	×	✓	48.05	85.2
✓	×	✓	53.58	86.9
×	✓	✓	52.03	86.5
✓	✓	✓	57.14	88.3

encoder-decoder is a fixed architecture, and conduct extensive ablation studies based on it. We adopt plain ResNet50 [15] as the encoder, and use FPN [22] as the decoder. Without losing generality, we perform studies on the LIP dataset and do not include flipping and multi-scale testing to speed up the process.

Encoder-Decoder. First of all, we report the performance of the baseline model. Although it has two branches to extract pose and parsing features, the parsing performance is still very low, which proves the existing encoder-decoder structure is not optimal for joint training. Compared to it, the searched structure obtains significantly better results, which validates that we can obtain reliable pose and parsing features for the subsequent interaction and fusion search. From Tab. 5, the searched basic architecture are 4.41% and 1.6% higher than the baseline in mIOU and PCK, respectively.

Feature Interaction. The role of feature interaction search is to allow multi-scale features to exchange information and enhance each other. Based on the experimental results, our search has greatly improved both branches, especially the parsing one. By adding interaction search to the baseline, it increases mIOU by 3.59% and PCK by 1.7%. When it is applied to the searched encoder-decoder structure, it also increases 4.61% and 2.0% on mIOU and PCK, respectively. Compared to the baseline, the improvements on the searched structure are much higher, which further confirms the searched encoder-decoder is more suitable for the joint training.

Feature Fusion. The searching of high-level feature fusion aims at allowing two branches to benefit from each other's structural and semantic information. We observe that it improves more on the searched encoder-decoder structure than the baseline. With both feature interaction and fusion search, the model improves by 6.01% mIOU and 2.7% PCK on the searched encode-decoder structure. And with all three proposed searches, the final model achieves the best performances on both tasks.

5. Conclusion

Joint training for human parsing and pose estimation is a non-trivial task. In order to model the correlation between the two tasks on top of extracting discriminative features, in this paper, we propose to search for the optimal network architecture from three aspects, namely the basic encoder-decoder structure, the multi-scale feature interaction, and the high-level feature fusion. Extensive experiments verify that with the proposed NPPNet, the structural information from the pose branch helps to allocate semantic body regions, and the abundant semantic information from the parsing branch, in turn, improves the pose estimation.

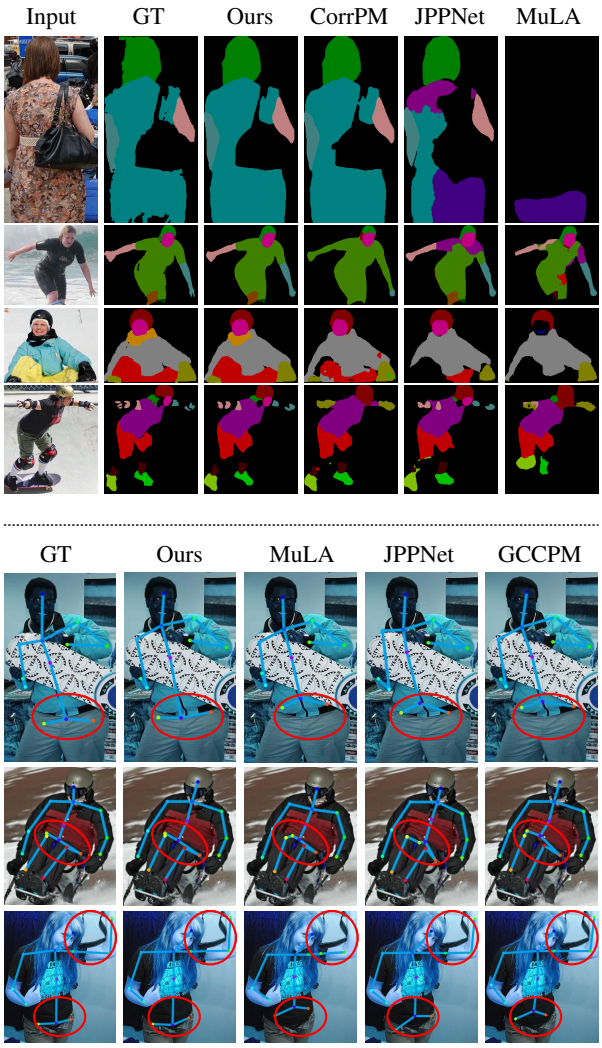

Figure 5. The visualization results on the LIP dataset. The top is for human parsing and the bottom is for pose estimation. We have highlighted visual improvements on hard joints (hips, pelvis) with red circles and our model obtains more accurate predictions against existing frameworks on both tasks.

Acknowledgment

This work was supported by the National Key R&D Program of China under Grand No. 2020AAA0103800 and the Beijing Nova Program under Grant Z201100006820023.

References

[1] Qian Bao, Wu Liu, Yuhao Cheng, Boyan Zhou, and Tao Mei. Pose-guided tracking-by-detection: Robust multi-person pose tracking. *IEEE Transactions on Multimedia*, 2020.

[2] Qian Bao, Wu Liu, Jun Hong, Lingyu Duan, and Tao Mei. Pose-native network architecture search for multi-person human pose estimation. In *ACM*, 2020.

[3] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[4] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, 2016.

[5] Xianjie Chen and Alan L. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. 2014.

[6] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *CVPR*, 2018.

[7] Chia-Jung Chou, Jui-Ting Chien, and Hwann-Tzong Chen. Self adversarial training for human pose estimation. In *APSIPA ASC*, 2018.

[8] Jian Dong, Qiang Chen, Xiaohui Shen, Jianchao Yang, and Shuicheng Yan. Towards unified human parsing and pose estimation. In *CVPR*, 2014.

[9] Hao-Shu Fang, Guansong Lu, Xiaolin Fang, Jianwen Xie, Yu-Wing Tai, and Cewu Lu. Weakly and semi supervised human body part parsing via pose-guided knowledge transfer. 2018.

[10] Jiemin Fang, Yuzhu Sun, Qian Zhang, Yuan Li, Wenyu Liu, and Xinggang Wang. Densely connected search space for more flexible neural architecture search. In *CVPR*, 2020.

[11] Yuan Gao, Haoping Bai, Zequn Jie, Jiayi Ma, Kui Jia, and Wei Liu. Mtl-nas: Task-agnostic neural architecture search towards general-purpose multi-task learning. In *CVPR*, 2020.

[12] Ke Gong, Xiaodan Liang, Dongyu Zhang, Xiaohui Shen, and Liang Lin. Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In *CVPR*, 2017.

[13] Jianyuan Guo, Kai Han, Yunhe Wang, Chao Zhang, Zhaohui Yang, Han Wu, Xinghao Chen, and Chang Xu. Hit-detector: Hierarchical trinity architecture search for object detection. In *CVPR*, 2020.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[16] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[17] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018.

[18] Guohao Li, Guocheng Qian, Itzel C Delgadillo, Matthias Muller, Ali Thabet, and Bernard Ghanem. Sgas: Sequential greedy architecture search. In *CVPR*, 2020.

[19] Wenbo Li, Zhicheng Wang, Binyi Yin, Qixiang Peng, Yuming Du, Tianzi Xiao, Gang Yu, Hongtao Lu, Yichen Wei, and Jian Sun. Rethinking on multi-stage networks for human pose estimation. *arXiv preprint arXiv:1901.00148*, 2019.

[20] Yingwei Li, Xiaojie Jin, Jieru Mei, Xiaochen Lian, Linjie Yang, Cihang Xie, Qihang Yu, Yuyin Zhou, Song Bai, and Alan L Yuille. Neural architecture search for lightweight non-local networks. In *CVPR*, 2020.

[21] Xiaodan Liang, Ke Gong, Xiaohui Shen, and Liang Lin. Look into person: Joint body parsing & pose estimation network and a new benchmark. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[22] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[23] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *CVPR*, 2019.

[24] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *ICLR*, 2019.

[25] Wu Liu, Qian Bao, Yu Sun, and Tao Mei. Recent advances in monocular 2d and 3d human pose estimation: A deep learning perspective. *arXiv preprint arXiv:2104.11536*, 2021.

[26] Xinchen Liu, Meng Zhang, Wu Liu, Jingkuan Song, and Tao Mei. Braidnet: Braiding semantics and details for accurate human parsing. In *Proceedings of the 27th ACM International Conference on Multimedia*, 2019.

[27] Yunan Liu, Liang Zhao, Shanshan Zhang, and Jian Yang. Hybrid resolution network using edge guided region mutual information loss for human parsing. In *ACM*, 2020.

[28] Yawei Luo, Zhedong Zheng, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Macro-micro adversarial network for human parsing. In *Proceedings of the European conference on computer vision (ECCV)*, 2018.

[29] Vladimir Nekrasov, Hao Chen, Chunhua Shen, and Ian Reid. Fast neural architecture search of compact semantic segmentation models via auxiliary cells. In *CVPR*, 2019.

[30] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[31] Xuecheng Nie, Jiashi Feng, and Shuicheng Yan. Mutual learning to adapt for joint human parsing and pose estimation. In *ECCV*, 2018.

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

[33] Daniil Osokin. Global context for convolutional pose machines. *arXiv preprint arXiv:1906.04104*, 2019.

[34] Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Strong appearance and expressive spatial models for human pose estimation. In *ICCV*, 2013.

[35] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *AAAI*, 2019.

[36] Tao Ruan, Ting Liu, Zilong Huang, Yunchao Wei, Shikui Wei, and Yao Zhao. Devil in the details: Towards accurate single and multiple human parsing. In *AAAI*, 2019.

[37] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[38] Raquel Urtasun and Trevor Darrell. Sparse probabilistic regression for activity-independent human pose inference. In *CVPR*, 2008.

[39] Wenguan Wang, Zhijie Zhang, Siyuan Qi, Jianbing Shen, Yanwei Pang, and Ling Shao. Learning compositional neural information fusion for human parsing. In *ICCV*, pages 5703–5713, 2019.

[40] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[41] Fangting Xia, Peng Wang, Xianjie Chen, and Alan L Yuille. Joint multi-person pose estimation and semantic part segmentation. In *CVPR*, 2017.

[42] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018.

[43] Yuhui Xu, Lingxi Xie, Xiaopeng Zhang, Xin Chen, Guo-Jun Qi, Qi Tian, and Hongkai Xiong. Pc-darts: Partial channel connections for memory-efficient architecture search. In *ICLR*, 2020.

[44] Xiaomei Zhang, Yingying Chen, Bingke Zhu, Jinqiao Wang, and Ming Tang. Part-aware context network for human parsing. In *CVPR*, 2020.

[45] Yiheng Zhang, Zhaofan Qiu, Jingen Liu, Ting Yao, Dong Liu, and Tao Mei. Customizable architecture search for semantic segmentation. In *CVPR*, 2019.

[46] Ziwei Zhang, Chi Su, Liang Zheng, and Xiaodong Xie. Correlating edge, pose with parsing. In *CVPR*, 2020.

[47] Jian Zhao, Jianshu Li, Xuecheng Nie, Fang Zhao, Yunpeng Chen, Zhecan Wang, Jiashi Feng, and Shuicheng Yan. Self-supervised neural aggregation networks for human parsing. In *CVPR*, 2017.

[48] Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. In *ICLR*, 2017.