ViPNAS: Efficient Video Pose Estimation via Neural Architecture Search – Supplementary Material –

1. Implementation Details

1.1. Spatial Search Space

We give implementation details of the spatial search space of ViPNAS in this section.

Elastic Depth. Elastic depth allows dynamic numbers of blocks in each stage. For example, the maximum number of the blacks in stage S is 4 as shown in Figure 1. When the depth D ($D \leq 4$) is selected, the first D blocks are activated and the rest $(4 - D)$ blocks are skipped. Note that the minimum depth of any stage should be no less than 1 ($D \geq 1$), as the first block may change the spatial resolution of the feature maps.

Elastic Width. Elastic width allows dynamic numbers of output channels in each block. For a convolutional layer, the shape of the filter is $O \times I \times K \times K$ given the input channels I, output channels O, and kernel size $K \times K$. When the output channel W ($W \leq O$) is selected, the filter is tailored to the shape of $W \times I \times K \times K$ as shown in Figure 2. We keep the first W out of O in the dimension of output channels.

Elastic Kernel Size. Elastic kernel size allows dynamic kernel sizes of convolutional layers in each block. The weights of the kernels are shared. As shown in Figure 3, we directly extract a $K \times K$ kernel filter from the centering of the super-network kernel filter, when the kernel size K is selected. This enables the weight sharing for kernels of different sub-networks, which has been shown simple but effective in our experiments. To avoid imbalance and biases of kernel extraction, we set the stride of the kernel size choice as 2, keeping all the selected kernels center-aligned.

Elastic Group Number. Elastic group number allows dynamic group numbers of convolutional layers in each block. A convolutional layer has a filter with the shape $O \times I \times K \times K$ given the input channels I, output channels O and kernel size $K \times K$. For example, when the group number is 2 (as shown in Figure 5), two filters with shape $\frac{O}{2} \times \frac{I}{2} \times K \times K$ are applied. In the figure, we concatenate the two groups of filters in the dimension of output channels for better illustration. We tailor the original filter to the shape of $O \times \frac{I}{2} \times K \times K$ and keep the first half in the dimension of input channels.

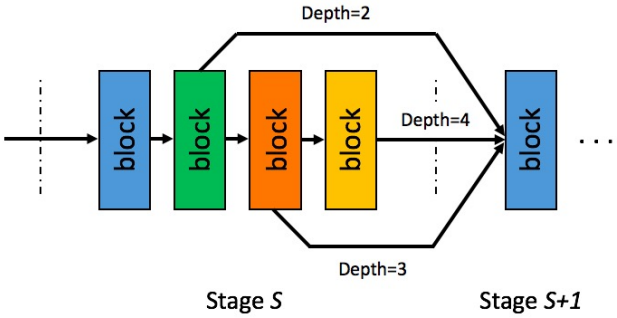

Figure 1. **Elastic Depth.** The first D blocks are activated if the depth D is selected in stage S.

Elastic Attention Module. Elastic attention module allows the network to choose whether or not to use the attention module in each block. As shown in Figure 4, the attention module is used if attention module is selected. We skip the attention module and identity mapping is applied if attention module is not selected. The attention module will keep both the spatial resolution and the feature channels the same before and after.

1.2. Super-Network Design

In this section, we introduce the structure of our super-network as well as the concrete search space designs for each super-network. As the search space increases with the exponential explosion, directly searching for block-level network architecture is hard. In our experiments, we explicitly enforce the same width, kernel size, group number and attention module for all the blocks in the same stage and search for stage-wise optimum.

MobileNet-V3 [3]. Our MobileNet-V3 based super-network consists of one convolutional layer, six stages, and three deconvolutional layers (followed by one 1×1 convolutional layer for output). Each stage contains a stack with mobile blocks [3], which consists of one 1×1 expansion convolution, a middle convolution and one 1×1 projection convolution. We search for the kernel size and the group number of the middle convolution in mobile blocks. The expansion convolution expands the input features to a higher-dimensional feature space. We search the expansion ratio,

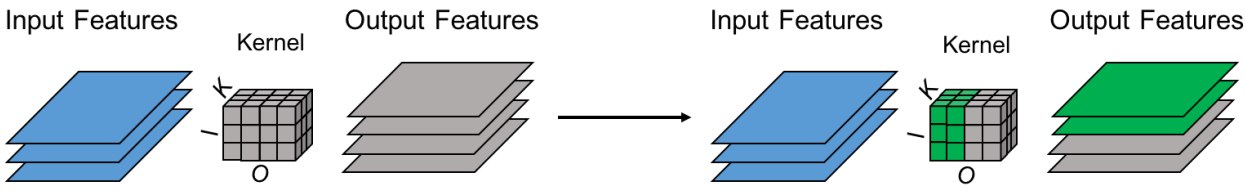

Figure 2. **Elastic Width.** Given the input channels I and kernel size $K \times K$, the first W output channels out of O is kept if the width W is selected. The filter is tailored from the shape $O \times I \times K \times K$ to $W \times I \times K \times K$.

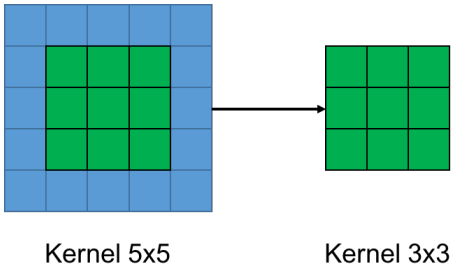

Figure 3. **Elastic Kernel Size.** The centering $K \times K$ kernel is reserved if the kernel size K is selected.

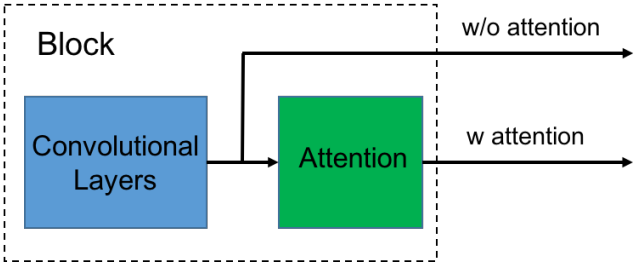

Figure 4. **Elastic Attention Module.** The attention module is applied if attention is selected, and is skipped if not.

which is similar to the elastic width. The detailed search space is summarized in Table 1.

ResNet-50 [2]. Following SBL [6], our ResNet-50 based super-network consists of one convolutional layer, four stages and three deconvolutional layers. Each block in stages is Bottleneck [2], which contains one 1×1 convolution followed by a middle convolution and another 1×1 convolution. Similar to our MobileNet-V3 based super-network design, we search for the kernel size and the group number of the middle convolution in the Bottleneck. We also search for whether to use a GC attention module [1] in each block. Table 2 specifies the search space of our ResNet-50 based super-network.

HRNet-W32[5]. We conduct experiments based on HRNet-W32 to further demonstrate the effectiveness of our proposed ViPNAS. Our HRNet-W32 based super-network consists of two convolutional layers followed by several Bottleneck blocks, three multi-resolution stages, and one 1×1 convolutional head for output. Each multi-resolution stage contains parallel branches with different spatial resolution, and each branch includes several BasicBlock [2]. Both the convolutions in BasicBlock apply the same width, kernel size, and group number. We search the configurations of each stage and each branch for the best performance. Table 3 displays the detailed search space of our HRNet-W32 based super-network.

Our search space is discrete. Take ResNet-50 backbone as an example, we set the search step to be 1 for depth, 16 for width, 2 for kernel size and 16 for group.

2. Qualitative Results

Figure 6 shows the qualitative results of our T-ViPNAS-Res50 on four adjacent frames. S-ViPNet localizes human poses on the first frame (key frame), and three different T-ViPNets propagate poses on the following frames (non-key frame). Our lightweight models keep the temporal consistency and are robust to occlusion, motion blur and unusual illumination. ViPNAS achieves state-of-the-art accuracy with CPU real-time performance.

References

[1] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Int. Conf. Comput. Vis. Worksh.*, 2019.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[3] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Int. Conf. Comput. Vis.*, 2019.

[4] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[5] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[6] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Eur. Conf. Comput. Vis.*, 2018.

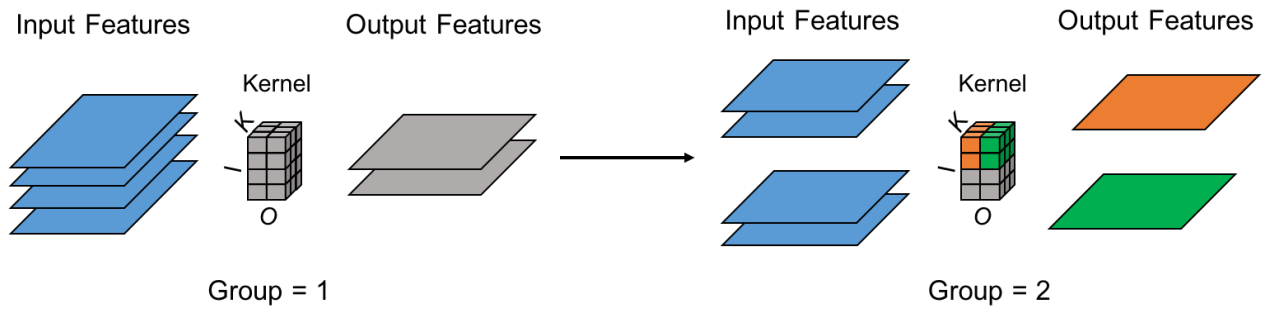

Figure 5. **Elastic Group Number.** An example of Group=2 is illustrated in the figure. Given the input channels I, output channels O, and kernel size $K \times K$, the filter is tailored from the shape $O \times I \times K \times K$ to $O \times \frac{I}{2} \times K \times K$. Two groups of filters with shape $\frac{O}{2} \times \frac{I}{2} \times K \times K$ are applied and are concatenated in the dimension of output channels.

Table 1. **MobileNet-V3 [3]** based search space. [min, max] indicates the range of each search space. Expansion ratio indicates the feature channel expansion rate in the middle of mobile blocks, and resolution indicates the ratio between the shapes of current features and those of input images. Kernel size and group number of the middle convolution in mobile blocks are searched.

Stage	Operator	Depth	Width	Kernel Size	Group	Attention (SE [4])	Expansion Ratio	Resolution
	Conv	-	[16, 16]	[3, 3]	[1, 1]	-	-	1/2
1	Mobile Block	[1, 1]	[16, 16]	[3, 3]	[2, 16]	[0, 1]	[1, 1]	1/2
2	Mobile Block	[2, 4]	[24, 24]	[3, 7]	[9, 144]	[0, 1]	[3, 6]	1/4
3	Mobile Block	[2, 4]	[40, 40]	[3, 7]	[15, 240]	[0, 1]	[3, 6]	1/8
4	Mobile Block	[2, 4]	[80, 80]	[3, 7]	[30, 480]	[0, 1]	[3, 6]	1/16
5	Mobile Block	[2, 4]	[112, 112]	[3, 7]	[42, 672]	[0, 1]	[3, 6]	1/16
6	Mobile Block	[2, 4]	[160, 160]	[3, 7]	[60, 960]	[0, 1]	[3, 6]	1/32
	Deconv	-	[256, 256]	[4, 4]	[32, 256]	-	-	1/4

Table 2. **ResNet-50 [2]** based search space. [min, max] indicates the range of each search space, and expansion ratio indicates the feature channel expansion rate in the middle of Bottleneck. The first convolution and max pooling with stride 2 down-sample the spatial resolution to 1/4 of the input image. Kernel size and group number of the middle convolution in Bottleneck are searched.

Stage	Operator	Depth	Width	Kernel Size	Group	Attention (GC [1])	Expansion Ratio	Resolution
	Conv+Pool	-	[32, 64]	[7, 7]	[1, 1]	-	-	1/4
1	Bottleneck	[3, 4]	[64, 80]	[3, 5]	[16, 64]	[0, 1]	[1, 1]	1/8
2	Bottleneck	[4, 6]	[128, 160]	[3, 5]	[16, 64]	[0, 1]	[1, 1]	1/8
3	Bottleneck	[6, 8]	[256, 320]	[3, 5]	[16, 64]	[0, 1]	[1, 1]	1/16
4	Bottleneck	[3, 4]	[512, 640]	[3, 5]	[16, 64]	[0, 1]	[1, 1]	1/32
	Deconv	-	[64, 256]	[4, 4]	[16, 64]	-	-	1/4

Table 3. **HRNet-W32 [5]** based search space. HRNet includes parallel branches with different resolution in stages, which indicates the ratio between the spatial shape of current features and input images. We search depth of each stage, and search width and attention of each branch. Kernel size and group number of the middle convolution in Bottleneck and both the convolutions in BasicBlock are searched.

Stage	Depth	Branch	Operator	Width	Kernel Size	Group	Attention (SE [4])	Resolution
	-		Conv	[16, 64]	[3, 3]	[1, 1]	-	1/4
1	[2, 4]	1	Bottleneck	[16, 64]	[3, 3]	[1, 16]	[0, 1]	1/4
2	[4, 4]	1	BasicBlock	[8, 32]	[3, 3]	[1, 32]	[0, 1]	1/4
		2	BasicBlock	[16, 64]	[3, 3]	[1, 64]	[0, 1]	1/8
3	[8, 16]	1	BasicBlock	[8, 32]	[3, 3]	[1, 32]	[0, 1]	1/4
		2	BasicBlock	[16, 64]	[3, 3]	[1, 64]	[0, 1]	1/8
		3	BasicBlock	[32, 128]	[3, 3]	[1, 128]	[0, 1]	1/16
4	[8, 12]	1	BasicBlock	[8, 32]	[3, 3]	[1, 32]	[0, 1]	1/4
		2	BasicBlock	[16, 64]	[3, 3]	[1, 64]	[0, 1]	1/8
		3	BasicBlock	[32, 128]	[3, 3]	[1, 128]	[0, 1]	1/16
		4	BasicBlock	[64, 256]	[3, 3]	[1, 256]	[0, 1]	1/32

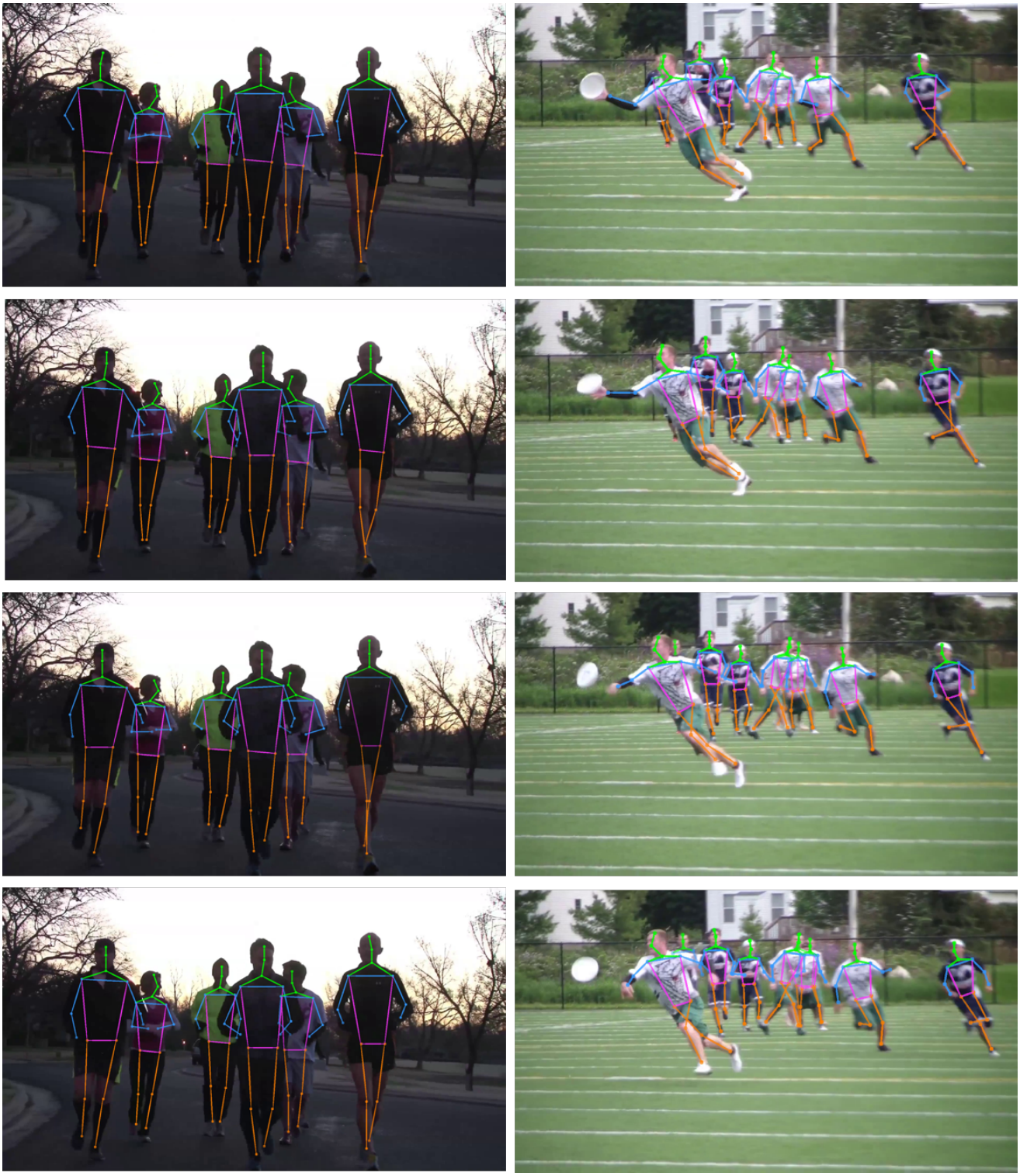

Figure 6. Qualitative results of T-ViPNAS-Res50 on four adjacent frames. S-ViPNet localizes human poses on the first frame, and three different T-ViPNets propagate poses on the following frames. Our proposed ViPNAS is robust to occlusion, motion blur and unusual illumination, and achieves state-of-art accuracy with CPU real-time performance.