A Simple Pyramid Vision Transformer for Human Pose Estimation in Crowds

Mickael Cormier

Vision and Fusion Laboratory
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
mickael.cormier@kit.edu

Abstract

Multi-person Pose Estimation is essential for several computer vision tasks related to motion analysis and anomaly detection. The impressive and continual progress in this field leads to application in uncooperative real-world scenarios such as detecting anomalous and dangerous behavior from individuals or groups within dense crowds in public places. However, reliably detecting poses within crowds in surveillance footage remains a very challenging task, due to diverse occlusions, illumination changes and long processing time. In this work, we present a simple Pyramid Vision Transformer for Human Pose Estimation achieving competitive results on the COCO Keypoints 2017 [16] while requiring significantly less parameters and thus computation time. A significant improvement is reported over the baselines on the more crowded OCHuman [33], PoseTrack 2018 [2], and CrowdPose [14] datasets.

1 Introduction

Human Pose Estimation (HPE) is a computer vision task which has made impressive progress over the last few years [5, 13, 8, 27, 30, 31, 15]. Applications include pedestrian gait recognition [26] and more generally action recognition [11]. While HPE in controlled environment delivers convincing results, multiple challenges arise for application in real-world uncontrolled scenarios, such as computation time for larger crowds, elevated view on persons, partial or almost total occlusion by diverse buildings of infrastructures, other persons or even self-occlusion [10].

In this work, we leverage the power of emerging Transformer architectures and based a on several best practices and experiments we propose a simple yet efficient model to reduce computation time while improving results, especially on occluded detections.

2 Related work

Mutiple datasets for HPE have been released in the last years [16, 1, 2, 14, 33]. One the most popular, the COCO Keypoints 2017 [16], offers over 200,000 images and 250,000 poses in single images with common poses and a frontal view. PoseTrack18 [2] features video frames with more complex real life scenarios in controlled environments, such as sport events, and is based on the MPII dataset [1]. Smaller datasets such as OCHuman [33] and CrowdPose [14] specifically address (self-)occlusion with similar frontal views on single images with two subjects for the former, and crowds in controlled environments such as group photos or sport events for the latter.

Different topologies of the human pose are proposed with different number of keypoints. In COCO a human pose is represented by 17 keypoints, of which five (nose, eyes, ears) are on the head. The MPII and Posetrack18 topologies simplify the pose by reducing the head keypoints to two and three, respectively, i.e. Posetrack18 has three keypoints for the head: top of head, nose and neck.

Two main categories of approaches have been presented in recent years to tackle HPE. First, bottom-up methods [5, 13, 8] detect all body parts in an image and fuse the retrieved keypoints to create a human pose. Since these methods detect keypoints independently from the actual person count on the image, the inference time is independent of the amount of people present. Second, top-down methods composed of a person detector and a pose estimator predict the bounding boxes and poses separately. Recently Transformer-based approaches [31, 15] challenged the mainly CNN [27, 30] dominated field. The quality of a top-down method is, however, highly dependent on the quality of the person detection and the inference time increases relatively to the person count. In this work we attempt to reduce this inference time significantly while improving the quality of the predictions.

3 Methods

Vision Transformers have been recently applied to HPE with impressive results and significantly less parameters [31, 15]. The work propose by Yang et al. [31] proposed an hybrid model which relies on a shorted CNN model and a Zransformer head. Following this work, we propose a simple and flexible full Transformer model, compose of three main components: a Transformer-based backbone instead of a CNN for feature extraction, a Transformer Encoder to model long-range relationship between feature vectors, and a head for keypoints heatmaps prediction. The architecture is illustrated in Figure 3.1. In the reminder of this section, we described each part.

3.1 Transformer Backbone

While hybrid architectures combining CNNs and Transformers have shown impressive results, recent works have shown that Transformer-based backbones improve performance on several vision tasks [28, 17] and seem more robust to severe occlusions, perturbations, and domain shifts [18]. We argue that these properties are beneficial to HPE and therefore, we adopt the recent Pyramid Vision Transformer (PVT)v2 [28] designed for pixel-level dense prediction tasks as our backbone. Following the idea of shortening the backbone from [31], we chose to reduce the original four stages from our backbones to only two stages.

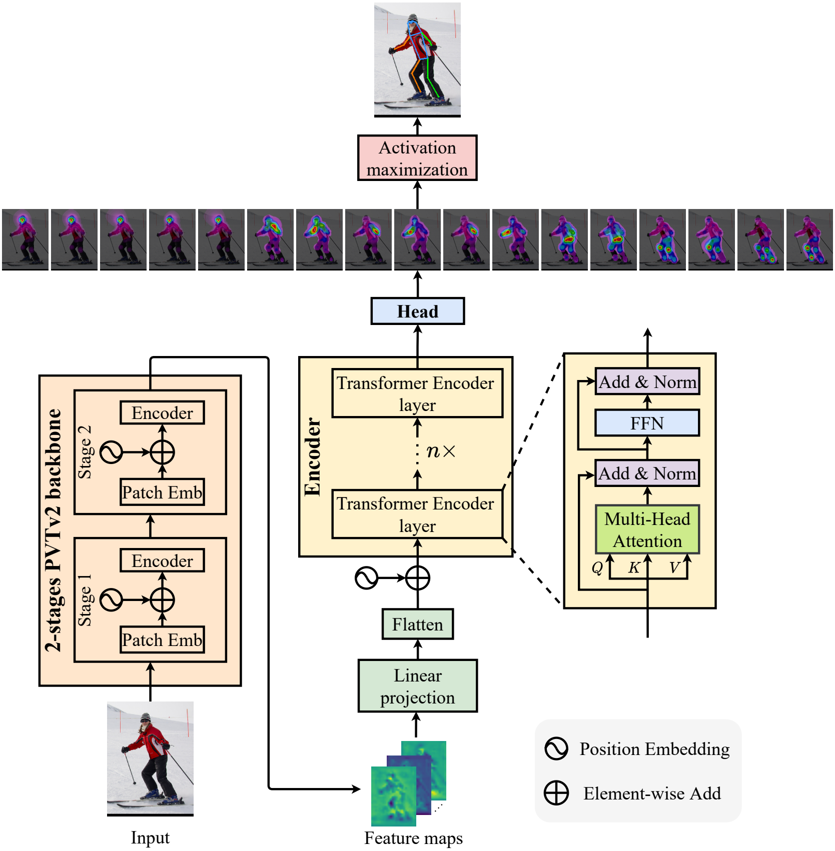


Figure 3.1: Overview of our model architecture. For an input detection image, a shortened 2-stage PVTv2 [28] backbone extracts feature maps, which are then flattened into fixed-size feature vectors and added with position embeddings. Subsequently, the dependencies between feature vectors in sequence are modeled by Transformer Encoder layers. Finally, a lightweight head is attached to predict the keypoint heatmaps.

3.2 Transformer Encoder

Following [31], we choose to encode the long-range relationships between the rich features with a Transformer Encoder.

Given an input image $\mathbf{I} \in \mathbb{R}^{3 \times H_I \times W_I}$, the backbone extracts the low-level features and outputs feature maps $\mathbf{X}_f$ of size $d_f \times H \times W$, in this case, $(H, W) = (H_I/8, W_I/8)$. The feature maps are then linearly embedded and their dimension is transformed to d. The transformed feature maps are finally flattened into a 1D fixed-size feature vector sequence $\mathbf{X} \in \mathbb{R}^{L \times d}$, where $L = H \cdot W$. To retain positional information, a fixed 2D position encoding $\mathbf{E}_{\text{pos}}$ is added to the sequence as proposed in recent works [21, 6, 31, 15]. To retain positional information, a fixed 2D position encoding $\mathbf{E}_{\text{pos}}$ [31, 15] is added to the sequence (Eq. (3.1)).

$$\mathbf{Z}_0 = \mathbf{X} + \mathbf{E}_{\text{pos}} \tag{3.1}$$

Subsequently, $\mathbf{Z}_0$ enters the Transformer Encoder, which consists of n Transformer Encoder layers. Concretely, each Transformer Encoder layer comprises a Multi-head Self-Attention (MSA) sub-layer (Eq. (3.2)) and a feed-forward network (FFN) sub-layer (Eq. (3.3)). The FFN contains two linear transformations with a ReLU non-linearity in between. Moreover, residual connection followed by LayerNorm (LN) [3] is applied around each of the two sub-layers.

$$\mathbf{Z}'_i = \text{LN}(\text{MSA}(\mathbf{Z}_{i-1}) + \mathbf{Z}_{i-1}), \qquad i = 1, \ldots, n \tag{3.2}$$

$$\mathbf{Z}_i = \text{LN}(\text{FFN}(\mathbf{Z}'_i) + \mathbf{Z}'_i), \qquad i = 1, \ldots, n \tag{3.3}$$

3.3 Regression Head

Heatmaps predictions are obtained for each keypoint by a regression head following the output of the Encoder. For an input sequence $\mathbf{X} \in \mathbb{R}^{L \times d}$, the Encoder outputs a sequence $\mathbf{E} \in \mathbb{R}^{L \times d}$. The output $\mathbf{E}$ is then reshaped back to the shape of $d \times H \times W$, where here $(H, W) = (H_I/8, W_I/8)$. Following common practice [30, 25, 31, 15], the resolution of the heatmap is set to a quarter of the input image, *i.e.* $(H', W') = (H_I/4, W_I/4)$. Hence, a deconvolution layer is added for upsampling [30, 8]. Finally, for the heatmaps prediction $\mathbf{H} \in \mathbf{R}^{K \times H' \times W'}$ of K different keypoints, the channel dimension of $\mathbf{E}$ is reduced from d to K via 1×1 convolution.

3.4 Model Variants

Since the Pyramid Vision Transformerv2 backbone is originally proposed in different scales, we also use seven backbone variations for our model, as described in Table 3.1. We follow the model naming convention in [31]. The name of our model is composed of three parts: the prefix ”TP“, the name of the backbone, and the number of Transformer Encoder layers. For instance a model called “TP-P-B0-A4” is composed of a Pyramid Vision Transformerv2-B0 backbone (abbreviated as P-B0) and a Transformer Encoder containing 4 Encoder layers.

All model variants produce 128 channels feature maps except for PVTv2-B0 with 64 channels. Following [28], the resolution of the feature map is always 1/8 of the input image. For the Transformer Encoder of all variants, we simply follow the setting of TransPose-R [31]. The dimension of the Transformer Encoder is $d = 256$. We employ $n = 4$ Transformer Encoder layers in total. In each layer, the number of heads for MSA and the number of hidden units for FFN is set to 8 and $h = 1024$, respectively. In the head, the upsampling is achieved via a 4×4 deconvolution.

Model Name	**Backbone**			**Transformer Encoder**				**Head**	
	Backbone	d_f	**Downsampling**	**#Encoder layers**	**#Heads**	d	h	**#DECONV layers**	**Kernel size**
TP-P-B0-A4	PVTv2-B0*	64	1/8	4	8	256	1024	1	4
TP-P-B1-A4	PVTv2-B1*	128	1/8	4	8	256	1024	1	4
TP-P-B2-A4	PVTv2-B2*	128	1/8	4	8	256	1024	1	4
TP-P-B2_Li-A4	PVTv2-B2_Li*	128	1/8	4	8	256	1024	1	4
TP-P-B3-A4	PVTv2-B3*	128	1/8	4	8	256	1024	1	4
TP-P-B4-A4	PVTv2-B4*	128	1/8	4	8	256	1024	1	4
TP-P-B5-A4	PVTv2-B5*	128	1/8	4	8	256	1024	1	4

Table 3.1: Architecture configuration details of different variants of our propose model. The star symbol (*) indicates that the Pyramid Vision Transformerv2 backbone [28] is reduced from the original four stages to two. d_f, d, and h are the dimension of feature maps, the dimension of Transformer Encoder, and the dimension of hidden layer in FFN of Transformer Encoder layer.

4 Evaluation

We first evaluate different variations of our models regarding architectural choices. We then quantitatively evaluate our models on four different datasets.

4.1 Ablation Study

We perform ablation studies on the COCO [16] dataset. As in [30, 25, 31, 15], we first extend the ground truth human bounding boxe to a fixed aspect ratio $height : width = 4 : 3$. Then we crop and resize the bounding boxe from the original image to a fixed size 256×192. To reduce overfitting, we apply standard data augmentation techniques, including random scale ($\pm 30\%$), random rotation ($[-45°, 45°]$), and flipping. We also use half body augmentation [29]. The reduced Pyramid Vision Transformerv2 [28] backbone network is initialized with the weights pre-trained on ImageNet-1K classification task [24].

All models are trained for 230 epochs on two NVIDIA GeForce RTX 2080 Ti GPUs using Adam [12] as optimizer and a cosine annealing learning rate schedule from $2e - 4$ to $2e - 5$.

4.1.1 Number of Backbone Stages

We analyze the number of stages of Pyramid Vision Transformerv2 [28] backbone using TP-P-B0-A4 with originally 4 stages. We reduce to 3 and 2 stages and report the results in Table 4.1. As suggested in [31], reducing the number of stages yields better results with fewer parameters, e.g. the model with 2 stages backbone achieves the best AP. We also observe the similar tendency for other variants, as shown in Figure 4.1.

4.1.2 Number of Transformer Encoder Layers

We then evaluate the influence of the number of Transformer layers in the encoder on the performance of the model. To this aim, TP-P-B0 and TP-P-B2_Li are used which represent models with small and medium size respectively. For

Model	Backbone	#Stages	AP ↑	Params (M) ↓
TP-P-B0-A4	Pyramid Vision Transformerv2-B0 [28]	2	**74.2**	**4.74**
		3	74.0	6.74
		4	72.6	9.79

Table 4.1: Ablation study on number of stages of Pyramid Vision Transformerv2 [28] backbone on COCO [16] validation set with ground truth human bounding boxes. ↑/↓ indicates that the higher/lower, the better. The best value in each column is marked in bold.

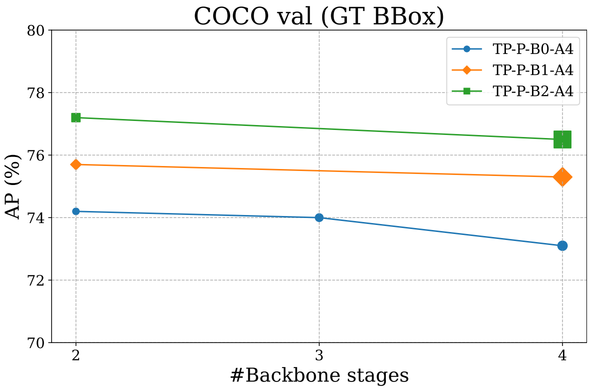


Figure 4.1: Effect of number of stages of Pyramid Vision Transformerv2 [28] backbone. AP is measured on COCO [16] validation set with ground truth human bounding boxes. Model size is indicated by marker size. For all models, AP drops with increasing number of backbone stages.

Model	#Layers	AP ↑	AP$_{0.5}$ ↑	AP$_{0.75}$ ↑	AP$_{\mathrm{M}}$ ↑	AP$_{\mathrm{L}}$ ↑	AR ↑	Params (M) ↓
TP-P-B0	2	69.1	90.3	76.3	66.0	73.5	72.2	**3.1**
	4	74.2	92.5	81.6	71.1	78.8	76.8	4.7
	6	**75.7**	**92.6**	**82.7**	**72.7**	**80.4**	**78.5**	6.3
TP-P-B2_Li	2	75.6	92.5	82.5	72.7	80.3	78.4	**4.5**
	4	77.1	**93.5**	**84.8**	74.2	81.7	79.8	6.0
	6	**77.6**	**93.5**	84.7	**74.6**	**82.2**	**80.2**	7.6

Table 4.2: Ablation study on Transformer Encoder size on COCO [16] validation set with ground truth human bounding boxes. "#Layers" refers to the number of Transformer Encoder layers. ↑/↓ indicates that the higher/lower, the better. For each model, the best value in each column is marked in bold.

a fair comparison, all models are trained on COCO [16] dataset using the same configuration and strategy. Results are evaluated on the COCO validation set with ground truth human bounding boxes and summarized in Table 4.2.

For both models the overall performance indicated by AP improves accordingly as the number of Transformer Encoder layers increases. We observe in more depth that the $\mathrm{AP_M}$ (AP for medium objects) and $\mathrm{AP_L}$ (AP for large objects) benefit largely from more Transformer layers. For example, when increasing the number of Encoder layers of TP-P-B0 from two to four, $\mathrm{AP_M}$ climbs rapidly $(+5.1)$.

In addition, the impact of scaling size of Transformer Encoder varies for backbones of different sizes, as compared in Table 4.2. For TP-P-B0, whose backbone is relatively small (0.5M), enlarging the Transformer Encoder from 2 layers to 4 layers leads to a noticeable performance improvement (+5.1AP). In contrast, it only brings about a slight enhancement for TP-P-B2_Li which is a larger backbone (1.8M). These results seem to concur with similar experiments in [31], in which ResNet based models require more encoder layers than HrNet based models. Therefore, there is a clear trade-off between backbone and encoder layers. While the backbone is usually responsible for large part of the inference time, increasing the number of Transformer layers in the encoder seem to compensate to some extent in term of quality.

4.2 Quantitative Results

We further conduct extensive performance studies on four popular datasets with different topologies and challenges, described in Table 4.2.

4.2.1 COCO

Unless otherwise specified, the models are trained using the same settings as mentioned in Section 4.1. Similar as [30, 25, 7], we apply a two-stage top-down paradigm. We use the same person detection result as in [30, 25], which is generated by an off-the-shell faster-RCNN detector [23] with person detection AP 56.4 on COCO validation set. Following the common practice [30, 25, 19,

Dataset	**#Images**	**#Labeled Person**	**#Keypoints**
COCO Keypoints 2017 [16]	> 200,000	> 250,000	17
PoseTrack 2018 [2]	66,374	153,615	15
OCHuman [33]	4731	8110	17
CrowdPose [14]	20,000	80,000	14

Table 4.3: Comparison between COCO Keypoints [16], PoseTrack [2], OCHuman [33], and CrowdPose [14] in terms of number of images, number of labeled person instances, and number of keypoints annotation of an individual person.

Model	**AP** ↑	**AP**$_{0.5}$ ↑	**AP**$_{0.75}$ ↑	**AP**$_{\mathbf{M}}$ ↑	**AP**$_{\mathbf{L}}$ ↑	**AR** ↑	**Params (M)** ↓	**FPS** ↑
TP-R-A4 [31]	75.1	92.6	82.6	71.9	79.6	77.8	5.8	**135.9**
TP-H-A6 [31]	**78.1**	**93.6**	84.6	**74.9**	**82.6**	**80.5**	17.2	32.3
TP-P-B0-A4	74.2	92.5	81.6	71.1	78.8	76.8	**4.7**	126.3
TP-P-B1-A4	75.7	92.5	82.7	72.6	80.6	78.6	6.2	120.9
TP-P-B2-A4	77.2	93.5	**84.8**	74.1	82.0	79.8	7.8	91.0
TP-P-B2_Li-A4	77.1	93.5	**84.8**	74.2	81.7	79.8	6.0	88.2
TP-P-B3-A4	77.1	**93.6**	83.8	74.2	81.8	79.8	7.8	91.8
TP-P-B4-A4	77.7	93.5	84.7	74.6	82.3	80.3	10.2	67.4
TP-P-B5-A4	76.7	93.5	82.8	73.5	81.4	79.3	8.1	78.3

Table 4.4: Results on COCO [16] validation set with ground truth bounding boxes. The input size is 256×192. ↑/↓ indicates that the higher/lower, the better. The best value in each column is marked in bold.

Model	**AP** ↑	**AP**$_{0.5}$ ↑	**AP**$_{0.75}$ ↑	**AP**$_{\mathbf{M}}$ ↑	**AP**$_{\mathbf{L}}$ ↑	**AR** ↑	**Params (M)** ↓	**FPS** ↑
TP-R-A4 [31]	72.6	89.1	79.9	68.8	79.8	78.0	5.8	**135.9**
TP-H-A6 [31]	**75.8**	**90.1**	**82.1**	**71.9**	**82.8**	**80.8**	17.2	32.3
TP-P-B0-A4	71.9	88.9	79.0	68.2	78.9	77.2	**4.7**	126.3
TP-P-B1-A4	73.6	89.6	80.3	69.9	80.6	78.7	6.2	120.9
TP-P-B2-A4	74.4	89.7	81.2	70.7	81.6	79.6	7.8	91.0
TP-P-B2_Li-A4	74.7	89.8	81.5	71.0	81.6	79.7	6.0	88.2
TP-P-B3-A4	74.8	90.0	81.5	71.0	81.8	79.9	7.8	91.8
TP-P-B4-A4	75.2	89.9	**82.1**	71.2	82.4	80.3	10.2	67.4
TP-P-B5-A4	74.2	89.7	80.7	70.4	81.4	79.5	8.1	78.3

Table 4.5: Results on COCO [16] validation set with detected human boxes generated by faster-RCNN [23] detector having human AP of 56.4. The input size is 256×192. ↑/↓ indicates that the higher/lower, the better. The best value in each column is marked in bold.

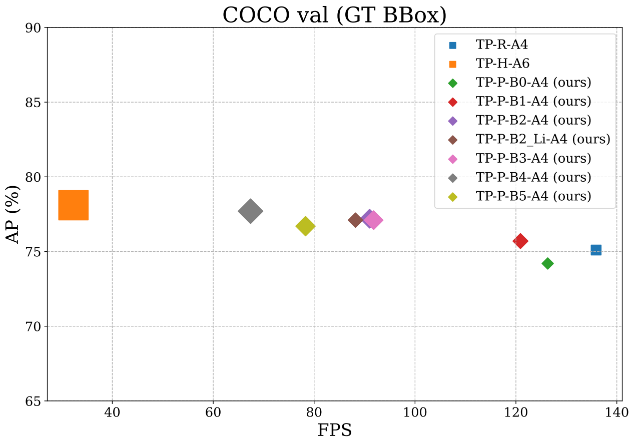
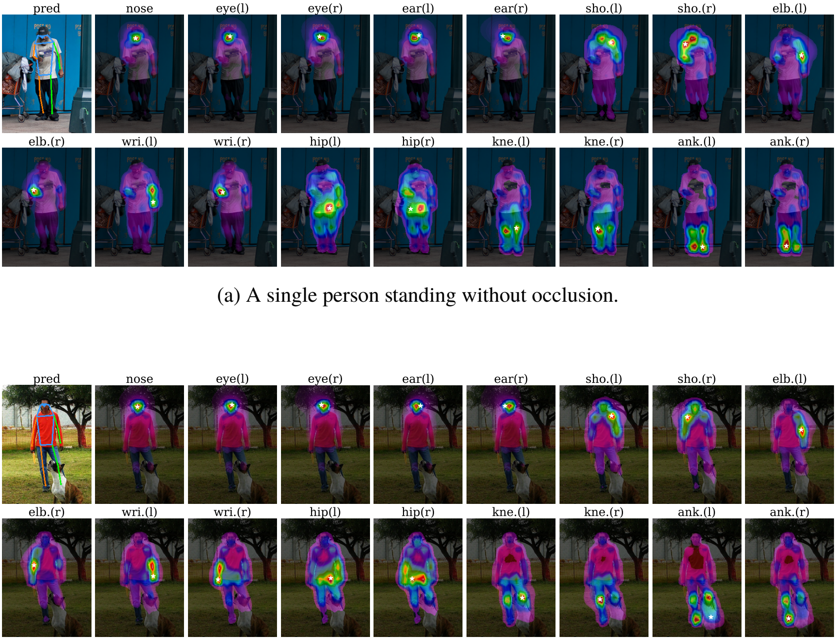


Figure 4.2: Model comparison on COCO [16] validation set with ground truth bounding boxe in aspects of model size, accuracy, and speed. □ and ◊ correspond to TransPose models [31] and our proposed models, respectively. Larger symbol indicates model with larger number of parameters.

7] to generate final heatmap prediction, we run the input image as well as its horizontally flipped version through the network and average the results. To alleviate error when decoding the predicted downscaled heatmaps into the final joint coordinates in the original image, we adopt Distribution-Aware coordinate Representation of Keypoint (DARK) [32] and its decoding strategy. Pose rescoring strategy and OKS-based non maximal suppression (NMS) [20] are also employed.

Finally, we visualize the position prediction and attention maps for different keypoints in Figure 4.3, for a single person with and without occlusion. While the non-occluded case seems straightforward, we observe that the model learns context information, especially for the shoulders, hips, knees and ankles. In the occluded case, the left knee and left ankle are occluded by a dog in the front. The model is still able to predict the accurate location using the context. For the partly occluded left knee, the model is able to pay attention to the relatively accurate area, with the help of the symmetrical joint (right knee). For the completely occluded left ankle, the attention focuses mainly to its nearby joints on the same side (left knee) and its symmetrical joint (right ankle). Based on these spatial clues, the model predicts the possible location where the left ankle is probably located.



(a) A single person standing without occlusion.

(b) A single person standing with occlusion.

Figure 4.3: Comparison of attention maps of the last Transformer Encoder layer between a single person standing *without* occlusion and a single person standing *with* occlusion. In each subfigure, the top left image is the input image annotated with the predicted pose. Pose prediction and attention maps are generated by TP-P-B2_Li-A4.

4.2.2 OCHuman

Following the setting from [33], we first train all models on COCO as in Section 4.1. The robustness of our models against strong occlusion is validated on the validation and test set of OCHuman with ground truth bounding boxes. The TransPose [31] models are tested and compared to as a baseline. We report the results on the validation set and test set in Table 4.6 and Table 4.7, respectively.

Model	AP ↑	$\text{AP}_{0.5}$ ↑	$\text{AP}_{0.75}$ ↑	AP_{L} ↑	AR ↑	Params (M) ↓	FPS ↑
TP-R-A4 [31]	62.0	80.3	66.7	62	66.2	5.8	**135.9**
TP-H-A6 [31]	62.3	77.2	67.9	62.4	66.6	17.2	32.3
TP-P-B0-A4	60.9	81.4	66.4	60.9	65.4	**4.7**	126.3
TP-P-B1-A4	62.8	81.6	68.1	62.8	66.7	6.2	120.9
TP-P-B2-A4	65.0	**81.9**	70.4	65.0	68.8	7.8	91.0
TP-P-B2_Li-A4	64.7	81.7	70.1	64.7	68.8	6.0	88.2
TP-P-B3-A4	65.0	81.7	70.1	65.0	68.9	7.8	91.8
TP-P-B4-A4	**65.9**	81.8	**71.4**	**65.9**	**69.5**	10.2	67.4
TP-P-B5-A4	65.0	81.7	71.2	65.0	69.1	8.1	78.3

Table 4.6: Results on OCHuman [33] validation set with ground truth bounding boxes. The input size is 256×192. ↑/↓ indicates that the higher/lower, the better. The best value in each column is marked in bold.

Model	AP ↑	$\text{AP}_{0.5}$ ↑	$\text{AP}_{0.75}$ ↑	AP_{L} ↑	AR ↑	Params (M) ↓	FPS ↑
TP-R-A4 [31]	61.8	78.5	67.2	61.8	65.9	5.8	**135.9**
TP-H-A6 [31]	62.0	76.6	66.9	62.1	66.3	17.2	32.3
TP-P-B0-A4	61.2	80.4	67.0	61.2	65.3	**4.7**	126.3
TP-P-B1-A4	63.0	80.6	68.4	63.0	66.8	6.2	120.9
TP-P-B2-A4	65.0	**81.7**	**70.4**	65.0	68.9	7.8	91.0
TP-P-B2_Li-A4	65.1	**81.7**	70.4	65.1	**69.0**	6.0	88.2
TP-P-B3-A4	64.6	81.4	69.3	64.6	68.4	7.8	91.8
TP-P-B4-A4	**65.2**	80.4	70.3	**65.2**	68.8	10.2	67.4
TP-P-B5-A4	64.8	81.6	**70.4**	64.8	68.7	8.1	78.3

Table 4.7: Results on OCHuman [33] test set with ground truth bounding boxes. The input size is 256×192. ↑/↓ indicates that the higher/lower, the better. The best value in each column is marked in bold.

As stated earlier, one of our motivation for replacing the CNN backbone with a Transformer backbone is to improve the robustness of our model against occlusion. This assumption is here largely proven. While our models are beaten by the HrNet backbone in TP-H-A6 [31] on the COCO dataset, our models surpass it largely when conducting evaluation on the largely occluded OCHuman dataset. The best-performing model TP-P-B4-A4 greatly outperforms TP-H-A6 on the validation set (+3.6AP) and on the test set (+3.2AP) with much fewer parameters and twice its speed. Moreover, almost all variants of our model achieve better performance than TP-R-A4 and TP-H-A6 on both validation set and test set. This is mainly due to more than 30% persons in OCHuman being under heavy occlusion (MaxIoU > 0.75), compared to less than 0.1% for COCO [33].

4.2.3 PoseTrack18

Model	Head AP ↑	Shou AP ↑	Elb AP ↑	Wri AP ↑	Hip AP ↑	Knee AP ↑	Ankl AP ↑	Total AP ↑
TP-R-A4 [31]	86.8	88.9	83.9	78.2	82.3	81.8	78.0	83.1
TP-H-A6 [31]	87.0	89.3	84.8	79.6	82.5	82.7	78.9	82.8
TP-P-B0-A4	86.5	88.4	83.0	76.8	81.0	80.6	76.9	82.2
TP-P-B1-A4	87.2	89.9	84.1	78.7	81.8	81.6	78.5	83.3
TP-P-B2-A4	86.9	89.2	85.1	80.0	82.5	82.6	79.5	83.9
TP-P-B2_Li-A4	87.5	**90.6**	85.6	80.5	82.1	83.3	79.7	84.3
TP-P-B3-A4	87.5	**90.6**	85.5	80.4	82.5	83.6	80.5	84.4
TP-P-B4-A4	**88.1**	90.2	**85.7**	**81.3**	**82.9**	**84.2**	**81.0**	**85.0**
TP-P-B5-A4	87.3	89.7	85.5	80.3	82.0	83.0	79.7	84.2

Table 4.8: Results on PoseTrack18 [2] validation set with ground truth bounding boxes. The input size is 256×192. ↑/↓ indicates that the higher/lower, the better. The best value in each column is marked in bold.

We further focus on the PoseTrack18 dataset [2] and its multi person pose estimation task with a topology of 15 keypoints. To this aim, we reuse our models pre-trained on COCO. The training setup as well as the data augmentation are almost the same as those for COCO, described in Section 4.1. We start with re-initializing the final layer uniformly. We train only the new final layer with initial learning rate $1e - 4$ for 30 epochs, while freezing other parts of the model. Finally, we finetune the entire model for another 30 epochs using a smaller starting learning rate ($5e - 5$). The cosine annealing learning rate scheduler is involved in both steps. For testing we adopt the person detection results provided by `mmpose` [9], which are generated by a Cascade R-CNN

(X-101-64x4d-FPN) [4] human detector. Other testing configurations remain the same as for COCO [16]. We report the results on the validation set with ground truth bounding boxes in Table 4.8 and report not only AP but also APs of different keypoints. Most of our models surpass the baseline models, due to main AP gain from the wrists and knees, which are more volatile joints at the far ends, often subjects to occlusions.

4.2.4 CrowdPose

Model	AP	AP$_{0.5}$ ↑	AP$_{0.75}$ ↑	AR ↑	AP$_{\mathrm{E}}$ ↑	AP$_{\mathrm{M}}$ ↑	AP$_{\mathrm{H}}$ ↑	Params (M) ↓	FPS ↑
TP-R-A4 [31]	69.8	83.7	75.7	78.8	79.7	71.2	56.4	5.8	**135.9**
TP-H-A6 [31]	71.3	83.6	76.5	80.3	80.5	72.7	58.3	17.2	32.3
TP-P-B0-A4	68.2	83.1	73.7	77.5	78.4	69.5	54.2	**4.7**	126.3
TP-P-B1-A4	70.3	83.6	75.8	79.5	80.0	71.8	56.7	6.2	120.9
TP-P-B2-A4	71.7	83.8	76.9	80.8	81.3	73.2	58.0	7.8	91.0
TP-P-B2_Li-A4	71.7	83.7	76.9	81.0	81.3	73.3	58.3	6.0	88.2
TP-P-B3-A4	71.8	83.7	76.9	81.0	81.3	73.4	58.2	7.8	91.8
TP-P-B4-A4	**72.7**	**84.2**	**77.6**	**82.1**	**81.8**	**74.3**	**59.2**	10.2	67.4
TP-P-B5-A4	71.4	83.6	76.7	80.6	81.0	73.0	57.6	8.1	78.3

Table 4.9: Results on CrowdPose [14] test set with detected human boxes generated by YOLOv3 [22] detector. The input size is 256×192. ↑/↓ indicates that the higher/lower, the better. The best value in each column is marked in bold. E, M, and H of AP stand for crowding levels easy, medium, and hard, as defined in [14].

For the CrowdPose dataset [14], we use the same two-stage finetuning strategy as for PoseTrack18 (see Section 4.2.3). All models are evaluated on CrowdPose test set with detected human bounding boxes generated by a YOLOv3 detector. Our results are reported in Table 4.9. We also report the results on three crowding levels, *i.e.*, uncrowded (easy), medium crowded, and extremely crowded (hard), as defined in [14]. Almost all of our proposed models surpass the baselines, with the exception of two. Our best-performing model TP-P-B4-A4, outperforms TP-H-A6 by a large margin of $+1.4$ AP with significantly fewer parameters and more than twice its speed. The strongest difference comes from AP$_{\mathrm{E}}$ ($+1.3$AP) and AP$_{\mathrm{M}}$ ($+1.6$AP) while AP$_{\mathrm{H}}$ is also moderately improved ($+0.9$AP). The results demonstrate that our models are able to handle not only simple daily but also crowded cases, probably due to better performance in occluded cases.

5 Conclusion

We engineer a full Transformer-based model for top-down HPE. The recipe is simple, flexible and can be applied with several variations: a reduced Transformer-based backbone without convolutions for feature extraction, a Transformer Encoder to model long-range relationship between feature vectors, and a simple head for keypoint heatmap estimation. Our results show that our model performs competitively and even outperforms the much heavier baseline on three out of four datasets, with heavy occlusions and higher levels of crowdedness. Future works should focus on difficult occlusions for which multiple person are visible in a detected bounding boxe often leading to predictions of the right keypoint belonging to the wrong person.

Acknowledgments

This work is based on joint works resulting from a very close collaboration between Mickael Cormier with his Master Thesis student Haobin Tan. Both the author and the corresponding student have contributed substantially to this research. While it is difficult to set a precise boundary, Mickael Cormier was rather in charge of the idea, while the student focused on the implementation.

References

[1] Mykhaylo Andriluka et al. “2D Human Pose Estimation: New Benchmark and State of the Art Analysis”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[2] Mykhaylo Andriluka et al. “Posetrack: A benchmark for human pose estimation and tracking”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 5167–5176.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. “Layer normalization”. In: *arXiv preprint arXiv:1607.06450* (2016).

[4] Zhaowei Cai and Nuno Vasconcelos. “Cascade r-cnn: Delving into high quality object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 6154–6162.

[5] Z. Cao et al. “OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[6] Nicolas Carion et al. *End-to-End Object Detection with Transformers*. 2020. arXiv: `2005.12872 [cs.CV]`.

[7] Yilun Chen et al. “Cascaded pyramid network for multi-person pose estimation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7103–7112.

[8] Bowen Cheng et al. “Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 5386–5395.

[9] MMPose Contributors. *OpenMMLab Pose Estimation Toolbox and Benchmark*. `https://github.com/open-mmlab/mmpose`. 2020.

[10] Mickael Cormier et al. “Where Are We With Human Pose Estimation in Real-World Surveillance?” In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022, pp. 591–601.

[11] Haodong Duan et al. “Revisiting Skeleton-based Action Recognition”. In: *arXiv preprint arXiv:2104.13586* (2021).

[12] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: `1412.6980 [cs.LG]`.

[13] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. “Pifpaf: Composite fields for human pose estimation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 11977–11986.

[14] Jiefeng Li et al. *CrowdPose: Efficient Crowded Scenes Pose Estimation and A New Benchmark*. 2019. arXiv: `1812.00324 [cs.CV]`.

[15] Yanjie Li et al. *TokenPose: Learning Keypoint Tokens for Human Pose Estimation*. 2021. arXiv: `2104.03516 [cs.CV]`.

[16] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[17] Ze Liu et al. *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. 2021. arXiv: `2103.14030 [cs.CV]`.

[18] Muzammal Naseer et al. *Intriguing Properties of Vision Transformers*. 2021. arXiv: `2105.10497 [cs.CV]`.

[19] Alejandro Newell, Kaiyu Yang, and Jia Deng. "Stacked hourglass networks for human pose estimation". In: *European conference on computer vision*. Springer. 2016, pp. 483–499.

[20] George Papandreou et al. "Towards accurate multi-person pose estimation in the wild". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4903–4911.

[21] Niki Parmar et al. *Image Transformer*. 2018. arXiv: `1802.05751 [cs.CV]`.

[22] Joseph Redmon and Ali Farhadi. *YOLOv3: An Incremental Improvement*. 2018. arXiv: `1804.02767 [cs.CV]`.

[23] Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015), pp. 91–99.

[24] Olga Russakovsky et al. "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3 (2015), pp. 211–252.

[25] Ke Sun et al. "Deep high-resolution representation learning for human pose estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5693–5703.

[26] Torben Teepe et al. "GaitGraph: Graph Convolutional Network for Skeleton-Based Gait Recognition". In: *arXiv preprint arXiv:2101.11228* (2021).

[27] Jingdong Wang et al. "Deep high-resolution representation learning for visual recognition". In: *IEEE transactions on pattern analysis and machine intelligence* (2020).

[28] Wenhai Wang et al. *PVTv2: Improved Baselines with Pyramid Vision Transformer*. 2021. arXiv: `2106.13797 [cs.CV]`.

[29] Zhicheng Wang et al. “Mscoco keypoints challenge 2018”. In: *Joint Recognition Challenge Workshop at ECCV 2018*. Vol. 5. 2018.

[30] Bin Xiao, Haiping Wu, and Yichen Wei. “Simple baselines for human pose estimation and tracking”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 466–481.

[31] Sen Yang et al. “TransPose: Keypoint localization via transformer”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 11802–11812.

[32] Feng Zhang et al. “Distribution-aware coordinate representation for human pose estimation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 7093–7102.

[33] Song-Hai Zhang et al. *Pose2Seg: Detection Free Human Instance Segmentation*. 2019. arXiv: `1803.10683 [cs.CV]`.