WTPose: Waterfall Transformer for Multi-person Pose Estimation

Navin Ranjan Bruno Artacho Andreas Savakis
Rochester Institute of Technology
Rochester, New York 14623, USA

Abstract

Human pose estimation is an important problem with broad applications that can be particularly useful for privacy preservation when analyzing activities and human-object interactions. We propose the Waterfall Transformer architecture for Pose estimation (WTPose), a single-pass, end-to-end trainable framework designed for multi-person pose estimation. Our framework leverages a transformer-based waterfall module that generates multi-scale feature maps from various backbone stages. The module performs filtering in the cascade architecture to expand the receptive fields and to capture local and global context, therefore increasing the overall feature representation capability of the network. Our experiments on the COCO dataset demonstrate that the proposed WTPose architecture, with a modified Swin backbone and transformer-based waterfall module, outperforms other transformer architectures for multi-person pose estimation.

1. Introduction

Human pose estimation is a challenging computer vision task that deals with predicting the spatial locations of keypoints or joints of the human body. Challenges arise from several factors, including the intricate mechanics of human movement, frequent occlusions, diverse body appearances, and variations in scale and background. Deep learning methods based on Convolutional Neural Networks (CNNs) have increased state-of-the-art performance in pose estimation [3, 7, 35]. More recently, vision transformers [13, 15, 21, 32, 36, 40] have shown excellent performance in computer vision tasks, including pose estimation. This paper presents a new transformer architecture - called waterfall transformer - that can benefit pose estimation.

Human pose estimation has emerged as a transformative technology with a wide range of practical applications across domains such as healthcare, sports analytics, robotics, and retail. In the retail industry, it can be particularly impactful for analyzing activities and human-object interactions while preserving customer privacy. Skeletonization avoids display of the human body and face, addressing growing concerns about safety, privacy and security when using AI systems. Pose estimation enables retailers to gain insights into how customers engage with products, shelves, and displays, providing valuable data for optimizing store layouts and improving product placement. By relying on skeletal keypoints rather than biometric or facial data, pose estimation ensures privacy while providing actionable insight into customer behavior. Additionally, pose estimation plays a critical role in theft prevention by detecting suspicious behaviors, such as unusual movements near high-value items, enhancing security without compromising employee and customer privacy. Furthermore, employee tracking with pose estimation is a privacy preserving approach to enhancing safety in stores and warehouses by monitoring activities and preventing errors that could lead to accidents. With its potential to revolutionize customer engagement, operational efficiency, safety and security, pose estimation is becoming a cornerstone of innovation in the retail industry.

Figure 1. WTPose examples from the COCO dataset.

In this paper, we propose WTPose, a "Waterfall Transformer" architecture for pose estimation offering a flexible framework for improved performance over the baseline models. Pose estimation examples using WTPose are shown in Figure 1. A key innovation of our architecture is the integration of our multi-scale Waterfall Transformer Module (WTM), which enhances the performance of vision transformer models, such as the Shifted Window (Swin) transformer [21]. The WTM processes feature maps

from multiple levels of the backbone through its waterfall branches. The module performs filtering operations based on a dilated attention mechanism to increase the Field-of-View (FOV) and capture both local and global context, leading to significant performance improvements. The contributions of this paper are the following.

- We introduce the novel Waterfall Transformer architecture for pose estimation, a single-pass, end-to-end trainable, multi-scale approach for top-down multi-person 2D pose estimation.
- We propose a waterfall transformer module with multi-scale attention, that employs a dilated attention mechanism enabling a larger receptive field to capture global and local context.
- Our experiments on the COCO dataset demonstrate improved pose estimation performance over comparable transformer methods.

2. Related Work

2.1. CNNs for pose estimation

With the advancement of deep convolutional neural networks, human pose estimation has achieved superior results [3, 7, 31, 35]. The Convolutional Pose Machine (CPM) [35] architecture includes multiple stages, producing increasingly refined joint detection. The OpenPose method [7] included Part Affinity Fields to deal with pose of multiple people in an image. The Stacked Hour-glass network [24] uses repeated bottom-up and top-down processing with intermediate supervision to process across all scales and capture the best spatial relationship associated with the body for accurate pose estimation. Expanding on the stacked hourglass networks, the multi-context attention approach [11] designs Hourglass Residual Units (HRUs) with the goal of generating attentions maps with larger receptive fields at multiple resolutions and with various semantics. Additionally, post-processing with Conditional Random Fields (CRFs) is used to generate locally and globally consistent pose estimates.

The High-Resolution Network (HRNet) architecture [29, 33] connects high-to-low sub-networks in parallel, maintaining high-resolution representations throughout the process, and generating more accurate and spatially precise pose estimates. The Multi-Stage Pose Network [18] operates similarly with HRNet [33], but it employs a cross-stage feature aggregation strategy to propagate information from early stages to the latter ones and is equipped with coarse-to-fine supervision.

The UniPose+ [3], OmniPose [2], and BAPose [4, 5] methods propose multiple variants of the Waterfall Atrous Spatial Pooling (WASP) module for single person, multi-person top-down and multi-person bottom-up pose estimation. The WASP module is the inspiration for the waterfall transformer module in WTPose, as it significantly increases the multi-scale representations and field-of-view of the network and extracts features with a greater amount of contextual information, resulting in more precise pose estimates without the need for post-processing.

2.2. Vision Transformers for Pose Estimation

There is a recent surge of interest in models that employ transformer architectures for human pose estimation [19, 28, 36, 37, 40, 41]. In earlier works, a CNN backbone was used as a feature extractor and the transformer was treated as a superior decoder [19, 37]. The TransPose [37] architecture combines the initial parts of CNN-based backbones to extract features from images and the standard transformer architecture [32] to utilize attention layers for learning dependencies and predicting keypoints for 2D human pose estimation. However, TransPose has a limitation in modeling direct relationships between keypoints. TokenPose [19] explicitly embeds each keypoint as a token and simultaneously learns both visual clues and constraint relations through self-attention interactions. The HRFormer [40] is inspired by HRNet [33] and utilizes a multi-resolution parallel design. It adopts convolution in the stem and first stage, followed by transformer blocks. The transformer blocks perform self-attention on non-overlapping partitioned feature maps and use 3x3 depth-wise convolution for cross-attention among the partitioned maps. ViTPose [36] adopts the plain and non-hierarchical vision transformer [13] as a backbone to extract feature maps. The architecture then employs either deconvolutional layers or a bilinear upsampling-based decoder for 2D pose estimation. PoseFormer [41] proposed a pure transformer-based architecture for 3D pose estimation, based on 2D pose sequences in video frames.

2.3. Pose Representation

Early approaches to human pose estimation focused on directly predicting the coordinates of body joints from images [8, 23, 30, 31]. While these methods were efficient, their performance was limited due to the challenge of modeling the complex, non-linear mapping between raw image features and joint coordinates. To address these limitations, methods such as residual log-likelihood estimation [17] were proposed, aiming to better capture the underlying output distribution and improve the accuracy of joint localization. Recently, transformers have brought notable improvements due to their ability to capture long-range dependencies [22, 27].

The introduction of heatmap-based representations [1, 3, 18, 34] brought a major advancement in pose estimation by providing robust localization and generalization capabilities. These methods encode joint positions as Gaussian peaks in heatmaps, allowing models to learn spatial distri-

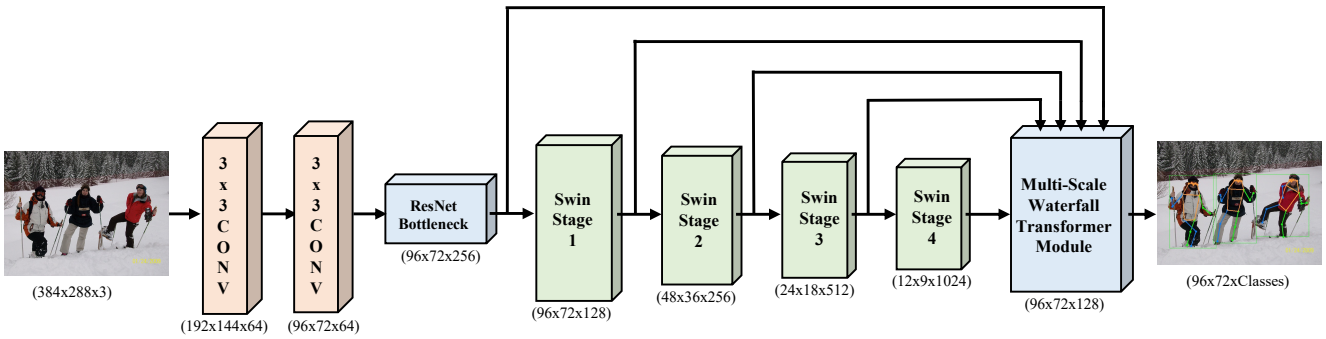

Figure 2. Waterfall transformer framework for multi-person pose estimation. The input image is fed through the modified Swin Transformer backbone and WTM module to obtain 128 feature channels at reduced resolution by a factor of 4. The decoder module generates K heatmaps, one per joint.

butions rather than precise coordinate mappings, which are prone to errors. These methods excel at handling occlusions and ambiguities, making them the dominant approach in the field. Subsequent research has focused on further enhancing heatmap-based methods by developing more powerful network architectures to estimate heatmaps with higher accuracy [3, 9, 10, 24, 29]. For instance, stacked hourglass networks [24] and convolutional pose machines [35] have been widely adopted for their ability to refine feature representations across multiple stages. However, heatmap representation methods suffer from quantization errors caused by the down-sampling operations in neural networks and do not model joint dependencies explicitly.

To address joint dependency modeling, earlier approaches [25, 38] relied on pictorial structures, where relationships between body joints were explicitly defined using anatomical priors. However, these methods had significant limitations, such as an inability to represent complex patterns and a lack of end-to-end trainability. Recent deep learning-based methods [6, 14, 39] have addressed these issues by implicitly modeling dependencies through the propagation of visual features between joints. For example, in [6], geometrical transform kernels were introduced to fuse features across different channels, capturing joint-specific characteristics. Similarly, in [14], a structured representation was proposed that defines M discrete tokens, each representing a sub-structure consisting of interdependent joints. Pose estimation is then framed as a classification task, where the model predicts the structure corresponding to these M tokens for a given image.

In this work, we adopted a top-down heatmap representation method for human pose estimation. Our WTPose architecture incorporates a multi-scale Waterfall Transformer Module, designed to learn feature map representations at both multi-scale and multi-stage levels. By capturing fine-grained local details as well as broader contextual information, our method effectively models joint dependencies and demonstrates robustness to quantization errors caused by down-sampling or occlusions.

3. Waterfall Transformer

The proposed waterfall transformer architecture, shown in Figure 2, is a single-pass, end-to-end trainable network that incorporates a modified Swin transformer [21] backbone and our multi-scale waterfall transformer module for multi-person pose estimation. The patch partition layer in Swin [21] is replaced by two 3×3 convolutions (Stem) followed by the first residual block of ResNet-101 [16], improving the feature representation of Swin.

The processing pipeline of WTPose is illustrated in Figure 2. The input image is fed to the transformer backbone which consists of our modified Swin transformer. The resulting multi-scale feature maps from multiple stages of Swin are processed by our waterfall transformer module and are fed to the decoder to generate K heatmaps, one heatmap per joint. The multi-scale WTM maintains high resolution of feature maps and generates accurate predictions for both visible and occluded joints.

The architecture of our waterfall transformer module is presented in Figure 3. The WTM takes inspiration from the Disentangled Waterfall Atrous Spatial Pooling (D-WASP) module [4, 5], which utilizes atrous blocks and the waterfall architecture to enhance the multi-scale representations. However, unlike D-WASP [4, 5], which expands the FOV through atrous convolution, our proposed approach employs a dilated neighborhood attention transformer block to expand the FOV.

The dilated transformer is built on the DiNAT [15] architecture, featuring both dilated and non-dilated neighbouring attention. The dilated neighborhood attention expands the local receptive fields by increasing the dilation rates and performs sparse global attention. On the other hand, the non-dilated neighborhood attention confines self-attention of each pixel to its nearest neighbors.

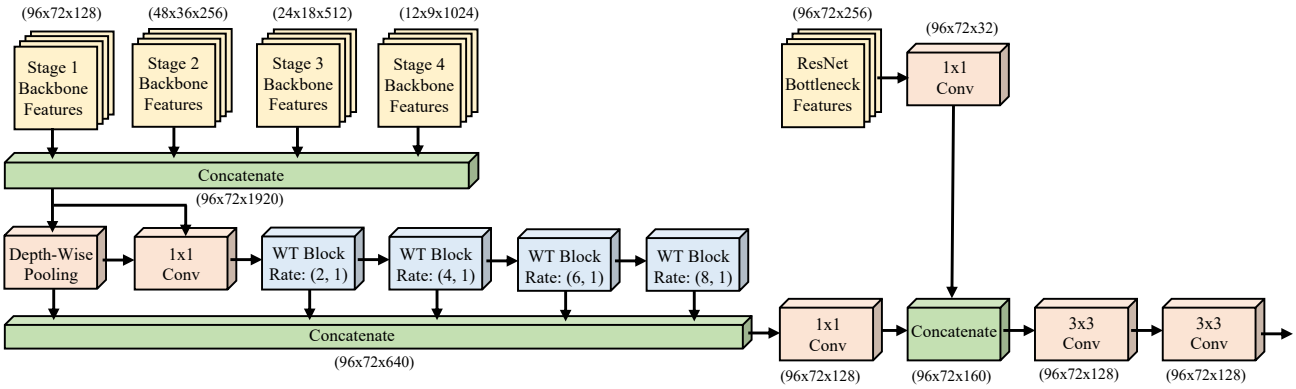

Figure 3. The proposed waterfall transformer module. The inputs are multi-scale feature maps from all four stages of the Swin backbone and low-level features from the ResNet bottleneck. The waterfall module creates a waterfall flow, initially processing the input and then creating a new branch. The feature dimensions (spatial and channel dimensions) output by various blocks are shown in parentheses.

To address contextual and spatial information loss resulting from the hierarchical backbone structure, the WTM processes multi-scale feature maps from all four stages of the Swin backbone through waterfall branches. The WTM module first performs upsampling operation using bilinear interpolation on the low-resolution feature maps from backbone stages 2, 3, and 4, to match them with the high-resolution feature maps from stage 1, and then combines all the feature maps to generate multi-scale feature representations for enhanced joint estimation. The multi-scale feature representation is then processed with 1×1 convolutions to reduce the channel size to 128. The concatenation (represented by the summation operator) of the feature maps is

$$g_0 = \sum_{i=1}^{4} (f_i), \tag{1}$$

where f_i represents the feature maps from the Swin backbone and the index i =1,2,3,4 indicates the Swin stages. The output after channel reduction is

$$z_0 = W_1 \circledast g_0 \tag{2}$$

where W_1 denotes the 1×1 convolution kernel and $\circledast$ represents convolution.

The output feature maps z_0 are then fed into the waterfall transformer blocks (WTB) which expand the FOV by performing a filtering cascade at increasing rates. Each WTB contains two types of attention, dilated multi-head neighborhood self-attention (D-MHSA) followed by multi-layer perceptron (MLP) to capture global context, and non-dilated multi-head neighborhood self-attention (N-MHSA) followed by MLP to capture local context.

$$\begin{aligned} \hat{z}_l &= \text{D-MHSA}(\text{LN}(z_{l-1})) + z_{l-1}, \\ z_l &= \text{MLP}(\text{LN}(\hat{z}_l)) + \hat{z}_l, \\ \hat{z}_{l+1} &= \text{N-MHSA}(\text{LN}(z_l)) + z_l, \\ z_{l+1} &= \text{MLP}(\text{LN}(\hat{z}_{l+1})) + \hat{z}_{l+1} \end{aligned}$$

where $\hat{z}_l$ and z_l denote the output features of the MHSA modules and MLP module for block l, respectively; D-MHSA and N-MSHA denote the multi-head self-attention based on dilated and non-dilated window, respectively.

The waterfall module is designed to create a waterfall flow, initially processing the input and then creating a new branch. The WTM goes beyond the cascade approach by combining all streams from all its WTB branches and the depth-wise pooling (DWP) layer from the multi-scale representation.

$$f_{\text{Waterfall}} = W_1 \circledast \left(\sum_{i=1}^{4} z_i + \text{DWP}(g_0) \right) \tag{3}$$

$$f_{\text{maps}} = W_3 \circledast (W_3 \circledast (W_1 \circledast f_{\text{LLF}} + f_{\text{Waterfall}})) \tag{4}$$

where, summation denotes concatenation, f_{LLF} are the low-level features from ResNet bottleneck, and W_1 denotes 1×1 convolution, and W_3 denotes 3×3 convolution with kernel size of 3 and strides of 1.

4. Experiments

4.1. Dataset

We conducted multi-person pose estimation experiments using the Common Objects in Context (COCO) dataset [20]. The COCO dataset consists of over 200k images captured in diverse real-world settings, containing more than 250k instances with annotated human keypoints. These keypoints include 17 landmarks representing major joints

Method	Input Size	Params (M)	GFLOPs	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
HRNet-W32 [29]	256×192	28.5	7.1	74.40	90.50	81.90	70.80	81.00	78.90
HRNet-W48 [29]	256×192	63.6	14.6	75.10	90.60	82.20	71.50	81.80	80.40
Swin-T [12]	256×192	32.8	6.3	72.44	90.09	80.59	68.99	79.05	78.20
WTPose-T (Ours)	256×192	30.0	12.8	**74.23**	**90.42**	**81.62**	**70.69**	**80.56**	**79.43**
Swin-B [12]	256×192	93.0	19.0	73.72	90.45	81.93	70.17	80.45	79.32
WTPose-B (Ours)	256×192	89.3	25.6	**74.96**	**90.54**	**82.16**	**71.71**	**81.74**	**80.51**
Swin-L [12]	256×192	203.0	41.0	74.30	90.56	82.09	70.58	81.22	79.82
WTPose-L (Ours)	256×192	198.0	47.9	**75.40**	**90.65**	**82.60**	**71.76**	**82.33**	**80.81**
HRNet-W32 [29]	384×288	28.5	17.3	75.80	90.60	82.70	71.90	82.80	81.00
HRNet-W48 [29]	384×288	63.60	35.4	76.30	90.80	82.90	72.30	83.40	81.20
Swin-T [12]	384×288	32.8	13.8	74.89	90.47	82.14	70.98	82.12	80.93
WTPose-T (Ours)	384×288	30.0	28.3	**76.36**	**90.80**	**82.95**	**72.33**	**83.40**	**81.43**
Swin-B [12]	384×288	93.2	41.6	75.81	90.92	83.10	71.61	82.97	80.99
WTPose-B (Ours)	384×288	89.3	55.8	**77.18**	**91.15**	**84.21**	**73.54**	**83.92**	**82.07**
Swin-L [12]	384×288	203.0	88.2	76.30	91.21	83.02	72.14	83.50	81.44
WTPose-L (Ours)	384×288	198.0	104.0	**77.56**	**91.43**	**84.42**	**73.93**	**84.34**	**82.61**

Table 1. WTPose results and comparison on the COCO validation dataset.

in the torso and limbs, as well as facial features such as the nose, eyes, and ears. The dataset presents a significant challenge due to its large scale, varied image contexts, diverse object sizes, and frequent occlusions.

4.2. Metrics

We adopt Object Keypoint Similarity metric (OKS) to evaluate our model, given as:

$$OSK = \frac{\left(\sum_i e^{-\frac{d_i^2}{2s^2k_i^2}}\right)\delta(v_i > 0)}{\sum_i \delta(v_i > 0)} \tag{5}$$

where, d_i is the euclidian distance between the estimated keypoint and its ground truth, v_i indicates if the keypoint is visible, s is the scale of the corresponding target, and k_i is the falloff control constant.

Following the evaluation framework set by [20], we report OKS as the Average Precision (AP) for the IOUs for all instances between 0.5 and 0.9 (AP), at 0.5 (AP^{50}) and 0.75 (AP^{75}), as well as instances of medium (AP^{M}) and large size (AP^{L}). We also report the Average Recall between 0.5 and 0.95 (AR).

4.3. Implementation details

We utilized the variants of Swin transformer as the backbone, initializing it with pre-trained weights from [21]. For the WTM module, we experimented with various rates of dilation and discovered that alternating between larger receptive field from dilated window and a small receptive field from non-dilated window results in improved prediction. We set the dilation rate for the WTB blocks to (2,1), (4,1), (6,1), (8,1), with window size of 7. Our models were trained on 4 A100 GPUs using the mmpose codebase [12], with a batch size of 32. We used the default training setting in mmpose to train WTPose, and employed the AdamW [26] optimizer with a learning rate of 5e-4. Our models were trained for 210 epochs, with a learning rate decay by 10 at the 170^{th} and 200^{th} epoch.

4.4. Experimental results on the COCO dataset

We performed training and testing on the COCO dataset and compared our WTPose architecture with the modified Swin backbone, as shown in Table 1. Our WTPose models consistently outperform their Swin counterparts in terms of average precision (AP) and average recall (AR) across all input sizes and model variants. For an input size of 256×192, WTPose achieves 1.79%, 1.24%, and 1.10% gains in AP over Swin for the tiny, base, and large variants, respectively. Similar performance improvements are observed across other metrics and for model variants with a larger input size of 384×288. For instance, WTPose-L achieves an AP of 77.56 at 384×288, surpassing Swin-L by 1.26 points, while maintaining higher precision for medium and large object detection.

In addition to accuracy improvements, WTPose achieves a balance between model size and computational complexity. While achieving superior performance, WTPose models maintain competitive parameter counts. For example, WTPose-T has 30.0M parameters, which is 2.8M fewer than Swin-T, while delivering a higher AP. Similarly, WTPose-B and WTPose-L show slight parameter reductions compared to their respective Swin variants while consistently outperforming them across all evaluation metrics. This reduction is achieved through the compact WTM module in WTPose, which operates on high-resolution feature

Figure 4. WTPose-Base examples from the COCO dataset.

Stem + ResNet BN	Model	Waterfall	Input Size	Dilation	AP	AR
-	Swin-B	-	256×192	-	73.72	79.32
✓	Swin-B	WTM	256×192	(1), (1), (1), (1)	74.46	79.99
✓	Swin-B	WTM	256×192	(2), (4), (6), (8)	73.79	79.35
-	Swin-B	WTM	256×192	(2,1), (4,1), (6,1), (8,1)	74.61	80.02
✓	Swin-B	WTM	256×192	(2,1), (4,1), (6,1), (8,1)	**74.96**	**80.51**
-	Swin-B	-	384×288	-	75.81	80.99
✓	Swin-B	WTM	384×288	(1), (1), (1), (1)	76.27	81.42
✓	Swin-B	WTM	384×288	(2), (4), (6), (8)	76.01	81.24
-	Swin-B	WTM	384×288	(2,1), (4,1), (6,1), (8,1)	76.78	81.54
✓	Swin-B	WTM	384×288	(2,1), (4,1), (6,1), (8,1)	**77.18**	**82.13**

Table 2. Results using different versions of WTPose on MS COCO validation dataset. All the models use Swin-B transformer as backbone. Stem + ResNet BN represents the Stem + ResNet bottleneck added at the initial layer of Swin and Waterfall indicates the use of the waterfall transformer module.

maps and requires only a single deconvolutional layer at decoder. In contrast, the vanilla Swin Transformer produces low-resolution heatmaps, requires multiple deconvolutional layers to recover the full resolution for pose estimation.

However, the accuracy gains of WTPose come with a trade-off in computational complexity. The majority of the WTM module is composed of transformer layers, which are computationally more expensive than convolutional layers. As a result, WTPose requires approximately 6 GFLOPs more at an input size of 256×192 and 15 GFLOPs more at 384×288 compared to Swin across all variants. Nevertheless, this increase in computational cost is justified by WTPose's consistent performance improvements across all evaluation metrics.

The higher computational complexity of WTPose is further justified when compared to Swin's larger model variants. For instance, at an input size of 384×288, our WTPose-B is approximately 56% smaller in model parameters and 37% less complex than Swin-L, yet still outperforms Swin-L by 0.88% in AP and 0.63% in AR. Similar trends are observed across all variants and input sizes. Furthermore, WTPose-T achieves comparable results to Swin-L, with significantly smaller model size and lower computational complexity. The waterfall transformer module enhances the feature maps, improving the accuracy of keypoint detection. Examples of pose estimation for subjects from COCO dataset are shown in Figure 4.

4.5. Ablation Study

We performed ablation studies to investigate individual components of WTPose. Table 2 shows results with various configurations using the Swin-B backbone and under

two input image sizes. We set WTM to have four Waterfall Transformer Blocks, with each block comprising a cascade of two transformer layers. These layers learn both global and local attention by employing a combination of dilated and non-dilated self-attention mechanisms. Each transformer layer uses a window size of 7×7 and performs multi-head self-attention with 8 heads.The dilation rates are varied to expand the receptive fields across different WTB blocks. We performed our analysis with dilation rates 1, 2, 4, 6, and 8, producing receptive field sizes of 7×7, 13×13, 25×25, 37×37, and 49×49, respectively.

First, we experimented with dilation rates of (1), (1), (1), (1) for each WTB, which involved performing a single non-dilated multi-head self-attention at each WTB. This configuration showed slight improvements over the baseline Swin-B backbone for both input sizes. Next, we used only dilated multi-head self-attention mechanisms for each WTB and set the dilation rates as (2), (4), (6), (8) for each WTB. Surprisingly, this configuration yielded minor gains, lower than those observed with the non-dilated transformer layers. Next, we combined both dilated and non-dilated multi-head self-attention mechanisms for each WTB and set the dilation rates as (2, 1), (4, 1), (4, 1), (8, 1) for successive WTBs. This configuration resulted in significant performance gains of 1.24% and 1.37% in terms of AP for smaller and larger input sizes, respectively. Similar improvements were observed for AR metrics. All the above WTPose architectures used a modified Swin backbone. To isolate the performance gains due to WTM, we incorporated WTM into the vanilla Swin transformer backbone. This configuration demonstrated gains of 0.89% and 0.97% in terms of AP for smaller and larger input sizes, respectively.

Our main observations are (i) integrating the waterfall transformer module with the modified Swin backbone improves the feature representations, and (ii) adding a Stem and ResNet Bottleneck at the start of the Swin-B further enhances the backbone's capability.

5. Conclusion

In this work, we introduced WTPose, a novel Waterfall Transformer framework for multi-person pose estimation. WTPose incorporates our proposed waterfall transformer module, which efficiently processes multi-scale feature maps extracted from various stages of the Swin transformer backbone. By utilizing a carefully designed cascade of dilated and non-dilated attention blocks, WTPose expands the receptive field, enabling the model to capture both fine-grained local details and broader global context.

With the modified Swin-B backbone and the WTM, WTPose achieves superior performance compared to other Swin-based models. Additionally, WTPose features a scalable and flexible architectural design that can be seamlessly integrated with other transformer-based backbones. This adaptability makes WTPose a valuable framework for advancing state-of-the-art pose estimation methods, particularly in real-world applications where precision and computational efficiency are critical.

6. Acknowledgment

The authors acknowledge Research Computing at the Rochester Institute of Technology for providing computational resources and support that have contributed to the research results reported in this publication.

References

[1] Bruno Artacho and Andreas Savakis. UniPose: Unified human pose estimation in single images and videos. In *CVPR*, pages 7035–7044, 2020. 2

[2] Bruno Artacho and Andreas Savakis. OmniPose: A multi-scale framework for multi-person pose estimation. *arXiv:2103.10180*, 2021. 2

[3] Bruno Artacho and Andreas Savakis. UniPose+: A unified framework for 2d and 3d human pose estimation in images and videos. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 44(12):9641–9653, 2021. 1, 2, 3

[4] Bruno Artacho and Andreas Savakis. BAPose: Bottom-up pose estimation with disentangled waterfall representations. In *Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pages 528–537, 2023. 2, 3

[5] Bruno Artacho and Andreas Savakis. Full-BAPose: Bottom up framework for full body pose estimation. *Sensors*, 23(7), 2023. 2, 3

[6] Adrian Bulat and Georgios Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *ECCV*, pages 717–732. Springer, 2016. 3

[7] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, pages 7291–7299, 2017. 1, 2

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

[9] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *CVPR*, pages 7103–7112, 2018. 3

[10] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *CVPR*, pages 5386–5395, 2020. 3

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

[12] MMPose Contributors. Openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose, 2020. 5

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner,

Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv:2010.11929*, 2020. 1, 2

[14] Zigang Geng, Chunyu Wang, Yixuan Wei, Ze Liu, Houqiang Li, and Han Hu. Human pose as compositional tokens. In *CVPR*, pages 660–671, 2023. 3

[15] Ali Hassani and Humphrey Shi. Dilated neighborhood attention transformer. *arXiv:2209.15001*, 2022. 1, 3

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3

[17] Jiefeng Li, Siyuan Bian, Ailing Zeng, Can Wang, Bo Pang, Wentao Liu, and Cewu Lu. Human pose regression with residual log-likelihood estimation. In *ICCV*, pages 11025–11034, 2021. 2

[18] 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*, abs/1901.00148, 2019. 2

[19] Yanjie Li, Shoukui Zhang, Zhicheng Wang, Sen Yang, Wankou Yang, Shu-Tao Xia, and Erjin Zhou. TokenPose: Learning keypoint tokens for human pose estimation. In *ICCV*, pages 11313–11322, October 2021. 2

[20] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 4, 5

[21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 1, 3, 5

[22] Weian Mao, Yongtao Ge, Chunhua Shen, Zhi Tian, Xinlong Wang, Zhibin Wang, and Anton van den Hengel. Poseur: Direct human pose regression with transformers. In *ECCV*, pages 72–88. Springer, 2022. 2

[23] Weian Mao, Zhi Tian, Xinlong Wang, and Chunhua Shen. Fcpose: Fully convolutional multi-person pose estimation with dynamic instance-aware convolutions. In *CVPR*, pages 9034–9043, 2021. 2

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

[25] Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Poselet conditioned pictorial structures. In *CVPR*, pages 588–595, 2013. 3

[26] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv:1904.09237*, 2019. 5

[27] Dahu Shi, Xing Wei, Liangqi Li, Ye Ren, and Wenming Tan. End-to-end multi-person pose estimation with transformers. In *CVPR*, pages 11069–11078, 2022. 2

[28] Michael Snower, Asim Kadav, Farley Lai, and Hans Peter Graf. 15 keypoints is all you need. In *Computer Vision and Pattern Recognition (CVPR)*, pages 6737–6747, 2020. 2

[29] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, pages 5693–5703, 2019. 2, 3, 5

[30] Zhi Tian, Hao Chen, and Chunhua Shen. Directpose: Direct end-to-end multi-person pose estimation. *arXiv:1911.07451*, 2019. 2

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

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NIPS*, 30, 2017. 1, 2

[33] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 43(10):3349–3364, 2020. 2

[34] Yihan Wang, Muyang Li, Han Cai, Wei-Ming Chen, and Song Han. Lite pose: Efficient architecture design for 2d human pose estimation. In *CVPR*, pages 13126–13136, 2022. 2

[35] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, pages 4724–4732, 2016. 1, 2, 3

[36] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. ViTPose: Simple vision transformer baselines for human pose estimation. *arXiv:2204.12484*, 2022. 1, 2

[37] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. TransPose: Keypoint localization via transformer. In *ICCV*, pages 11802–11812, October 2021. 2

[38] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, pages 1385–1392, 2011. 3

[39] Yiding Yang, Zhou Ren, Haoxiang Li, Chunluan Zhou, Xinchao Wang, and Gang Hua. Learning dynamics via graph neural networks for human pose estimation and tracking. In *(CVPR*, pages 8074–8084, 2021. 3

[40] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. HRFormer: High-resolution vision transformer for dense predict. In *NIPS*, 2021. 1, 2

[41] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. In *ICCV*, pages 11656–11665, 2021. 2