The Center of Attention: Center-Keypoint Grouping via Attention for Multi-Person Pose Estimation –Supplementary Material –

Guillem Brasó Nikita Kister Laura Leal-Taixé
Technical University of Munich
{guillem.braso, n.kister, lealtaixe}@tum.de

Input Image HigherHRNet [3] CenterNet[17] Ours

Figure 1: Qualitative examples of our method's output in comparison to our baselines: HigherHRNet[3] and CenterNet[17]. Best viewed in color and in a screen. Additional results are shown at the end of this document.

Abstract

In this document, we provide: (i) an in-depth analysis of our results on COCO compared to bottom-up state-of-the-art methods, (ii) a detailed explanation of our procedure to match detected centers to ground truth centers, (iii) comprehensive training and inference implementation details, together with our exact architecture, and (iv) qualitative results of our method's output and visualizations of attention activations.

1. Extended COCO Comparison

In Table 1, we provide a detailed comparison of CenterGroup against published bottom-up approaches on the COCO test-dev dataset. For each method, we specify its backbone network, grouping procedure, input size, and parameter count. We observe that most top-performing methods rely on greedy decoding schemes, which often involve optimization in the form of solving a sequence of bipartite matching problems. Alternatively, SPM [13] uses offsets, but relies on top-down refinement to achieve competitive results [1], and HGG[6] uses a hierarchical clustering algorithm that operates on the output of graph network predictions.

CenterGroup outperforms all previous methods with our proposed attention-based grouping module, which does not rely on optimization and is end-to-end trainable. Note that this module only introduces a slight increase in the number of parameters with respect to HigherHRNet[3], and combined with our keypoint detector, yields a model with significantly fewer parameters than other methods.

Regarding performance, we note that the increase in accuracy is most significant for large persons, where our im-

[1] i.e. it applies a single person pose estimation model over the predicted poses.

Method	Backbone	Grouping	Input size	# Params	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}
w/o multi-scale test									
OpenPose* [1]	–	Greedy decoding w/ optimization	–	–	61.8	84.9	67.5	57.1	68.2
AE* [12]	Hourglass	Greedy decoding w/ optimization	512	277.8M	62.8	84.6	69.2	57.5	70.4
PersonLab[14]	ResNet152	Greedy decoding	1401	68.7M	66.5	88.0	72.6	62.4	72.3
PifPaf[9]	–	Greedy decoding w/ optimization	–	–	66.7	-	-	62.4	72.9
HigherHRNet[3, 12]	HRNet-W32	Greedy decoding w/ optimization	512	28.6M	66.4	87.5	72.8	61.2	74.2
HigherHRNet[3, 12]	HRNet-W48	Greedy decoding w/ optimization	640	63.8M	68.4	88.2	75.1	64.4	74.2
Ours	HRNet-W32	Attention	512	30.3M	67.6	88.7	73.6	61.9	75.6
Ours	HRNet-W48	Attention	640	65.5M	**69.6**	**89.7**	**76.0**	**64.9**	**76.3**
w/ multi-scale test									
AE* [12]	Hourglass	Greedy decoding w/ optimization	512	277.8M	65.5	86.8	72.3	60.6	72.6
SPM* [13]	Hourglass	Offsets (One-shot)	512	277.8M	66.9	88.5	72.9	62.6	73.1
HGG [6]	Hourglass	Graph Network + Graclus clustering [4]	512	–	67.6	85.1	73.7	62.7	74.6
PersonLab [14]	ResNet152	Greedy decoding	1401	68.7M	68.7	89.0	75.4	66.6	75.8
HrHRNet-W48 [3]	HRNet-W48	Greedy decoding w/ optimization	640	63.8M	70.5	89.3	77.2	66.6	75.8
Ours	HRNet-W32	Attention	512	30.3M	70.0	89.9	76.6	65.2	**77.1**
Ours	HRNet-W48	Attention	640	65.5M	**71.1**	**90.5**	**77.5**	**66.9**	76.7

Table 1: Comparison of published bottom-up methods on the **COCO2017 test-dev** split. * means top-down refinement. *w/ optimization* refers to the use of bipartite matching solvers during inference.

provement is of 2.1 AP points for single-scale, and 1.3 for multi-scale, which can be explained by the ability of our transformer to capture relationships among distant joints in the image. Overall, it outperforms the current state-of-the-art method, HigherHRNet[3] by approximately 1.2 AP for single-scale and 0.6 AP for multi-scale, while having the exact same backbone and input size, and being 2.5x faster, which confirms CenterGroup's increased efficiency.

2. Matching Centers

In order to train our grouping module, we need to determine which detected centers in the image correspond to a ground truth pose. As explained in Section 5.4 in the main paper, this allows us to define a target y^c_{center} for every detected center $c \in \mathcal{C}$ indicating whether it represents a ground truth pose (i.e., $y^c_{\text{center}} = 1$) or not ($y^c_{\text{center}} = 0$). These labels are used to train our center classification module. Moreover, for those detected centers that do correspond to a ground truth pose, we obtain the visibility of their corresponding keypoints as well as the locations of those that are visible by simply using the annotations of the ground truth center that the detected center is matched with.

In order to determine correspondences between detected centers ($\mathcal{C}$) and ground truth centers ($\mathcal{P}$), we compute the euclidean distance between every $c \in \mathcal{C}$ and $\bar{c} \in \mathcal{P}$, and normalize it by the scale of $\bar{c}$, $s_{\bar{c}}$:

$$\text{dist}(c, \bar{c}) := exp\left(-\frac{||\text{loc}_c - \text{loc}_{\bar{c}}||^2}{2s_{\bar{c}} * k^2}\right) \quad (1)$$

where k is a fixed constant set to 0.15[2], and the scale $s_{\bar{c}}$ is computed as 0.53 multiplied by $\bar{c}$'s bounding box height and width, following [10]. This formula is adapted from the OKS metric, and simply normalizes distances between 0 and 1 by using a pre-defined standard deviation that depends on the object size.

With the distances from Equation 1, we define an instance of a bipartite matching problem. For every $c \in \mathcal{C}$ and $\bar{c} \in \mathcal{P}$, their corresponding cost $\text{cost}(c, \bar{c}) := 1 - \text{dist}(c, \bar{c})$, whenever $\text{dist}(c, \bar{c}) < 0.5$ and ∞ otherwise. We obtain matches between centers and ground truth centers by solving the problem with the hungarian algorithm, similarly to [2]. Note that running this algorithm takes on average significantly less than 1ms since the cost matrix is, at most, of size 20x30, and therefore it adds no significant computational burden. Additionally, note that this procedure is only necessary at training time in order define ground truth assignments. At test-time, as explained in the main paper, we do not require any form of optimization.

3. Implementation Details

3.1. Training

We pretrain our backbone and keypoint detection module following HigherHRNet [3]. We then randomly initialize our encoding and grouping modules and train our entire model end-to-end for $27,000$ iterations with batch size 130,

[2]This number is determined by increasing by 50% the constant that the COCO dataset uses for hip joints for OKS computation.

which corresponds to approximately 50 epochs on COCO, and 270 epochs on CrowdPose, and use learning rate linear warm-up during the first $1,000$ iterations[5]. We use an Adam optimizer [7] with learning rate set to $1e-5$ for pretrained layers and $3e-4$ for the remaining parts of the network, which we drop by a factor of 10 at 10,000 and 20,000 iterations. In addition, use use automatic mixed precision for training [11], which reduces the memory requirements by approximately half, and allows training on 4 NVIDIA RTX6000 with 24GB of RAM memory in approximately 24 hours. We observe that our training loss shows high stability and allows training with mixed precision without any divergence problems, in contrast to Associative Embeddings[12]. For data augmentation, we use the same techniques as [3], which include random flipping, rotation, scale variation, and generating a random crop of size 512x512, when using an HRNet32 backbone, or 640x640 when using an HRNet48 backbone.

We add one grouping module at the output of every transformer encoder block and compute the location, visibility and center losses, and then average them over the output of every transformer encoder block. Loss terms are balanced as follows: the heatmap loss, $\mathcal{L}_{\text{heatmap}}$ is weighted by factor 10, the location loss, $\mathcal{L}_{\text{loc}}$ is averaged over all visible keypoints in the image and weighted by 0.02, the center and visibility losses, $\mathcal{L}_{\text{vis}}$ and $\mathcal{L}_{\text{center}}$, are both weighted by factor 1. The overall set of weights is determined by ensuring that each loss term has a comparable magnitude.

3.2. Inference

At inference, we resize images to preserve their aspect ratio and have their shorter side of size 512 if using a HRNet32 backbone, or 640 if using HRNet48. Following [3], predicted heatmaps are upsampled to full image resolution. We then extract peaks by applying heatmap Non-Maximum Suppression (NMS) with a max-pooling kernel of size 5x5 for keypoints and 17x17 for person centers, and select all peaks that either have score over 0.01 or are within the top-5 scoring peaks in the heatmap.

For every predicted center $c \in \mathcal{C}$, we build its pose by assigning it the keypoints with highest attention score according to the attention score corresponding to every type, as explained in Section 4.2 in the main paper. Formally, given center $c \in \mathcal{C}$ the location of each of its joint types $i \in \{1, \dots, J\}$ is determined as:

$$\widehat{\text{loc}}_c^i = \arg\max_{k \in \mathcal{K}} \text{attn}_i(c, k) \tag{2}$$

In order to score the resulting poses, we use the predicted visibility scores for every keypoint, $\widehat{\text{vis}}_c^i$, as well as the predicted probability that center c represents a true positive

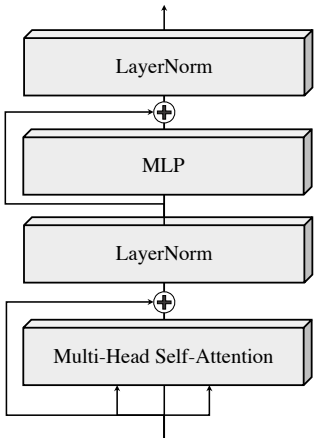


Figure 2: Overview of the architecture of a Transformer Encoder.

center, $\hat{y}^c_{\text{center}}$, as follows:

$$\text{score}_c = \begin{cases} \text{avg}\left(\{\widehat{\text{vis}}_c^i \mid \widehat{\text{vis}}_c^i \geq 0.5\}_{i=1}^J\right) & \text{if } \hat{y}^c_{\text{center}} \geq 0.5 \\ \hat{y}^c_{\text{center}} & \text{otherwise} \end{cases} \tag{3}$$

Intuitively, since visibility scores are only computed for those centers such that $y^c_{\text{center}} = 1$ during training (i.e. matched centers), we only use them whenever our network predicts centers to represent true pose centers with probability over 0.5. In that case, the overall pose score is the average visibility confidence score of keypoints that are predicted to be visible (i.e., $\widehat{\text{vis}}_c^i \geq 0.5$).

Unlike [12, 13, 1], we do not perform top-down refinement, nor ensembling [8], and all results are reported with flip-testing as it is common practice [15, 12, 14]. For postprocessing, following [12, 3], keypoint coordinates are shifted by 0.25 towards the contiguous second maximal activation in each heatmap, to account for quantization errors.

3.3. Exact Architecture

Our keypoint detection network is minimally modified from HigherHRNet, as explained in Section 5.2 in the main paper. Our newly added modules include an additional residual block and a multi-layer perceptron (MLP) to generate initial keypoint and person features, a transformer encoder and the grouping module. Our transformer encoder has 3 blocks, each with input dimension 128, 4 self-attention heads and MLP hidden dimension set to 512. We found no significant performance benefits from further increasing the transformer's size. The architecture of each transformer encoder block is not modified from the original one [16], and shown in Figure 2.

All of the MLPs in the grouping module, as well as the one generating the transformer's input contain two hidden

Layer Name	# Parameters
Keypoint Detection	
Backbone	28.5M (63.7M)
Keypoint Heads	110K
Encoding	
Residual Block	595K
Initial MLP	33K
Transformer Encoder	594K
Grouping	
Multi-Head Attention	420K
$\text{MLP}_{\text{center}}$	33K
MLP_{vis}	41K
Overall	
–	30.3M

Table 2: Parameter count breakdown among components in each stage of our model's pipeline. For the backbone, 28.5M refers to a HRNet32 backbone, and 63.7M refers to a HRNet48. Note that the overall number of parameters of our proposed encoding and grouping modules combined is relatively small, at 1.7M.

layers. We detail the number of parameters of each component in Table 2. The overall parameter count of our proposed keypoint encoding and grouping module is below 2M, which is relatively small, and only accounts for $<6\%$ (resp. $<3\%$) of the overall count when using an HRNet32 (resp. HRNet48) backbone.

4. Qualitative Results

4.1. Qualitative Examples

In Figure 3, we visualize results produced by our method in comparison to those from our baselines: HigherHRNet[3] and CenterNet [17]. As explained in the main paper, we reimplement CenterNet to use an HRNet[15] backbone and HigherHRNet's scale-aware heatmaps [3] for keypoint heatmap regression for a fair comparison.

We observe that our method's performance is robust under severe occlusion and challenging conditions. In comparison, CenterNet often fails whenever there is significant overlap among different poses, as can be seen in rows 1, 4, 5, 6 and 7. Moreover, since it always predicts joint locations for a given pose regardless of whether they are visible or not, it often hallucinates joints and produces unfeasible pose estimates (all rows).

HigherHRNet generally does a better job at grouping, as can be seen in rows 1, 4, 5, and 6, but this comes at a significantly increased computational cost of 2.5x inference time. Moreover, we observe that it tends to miss or assign very low confidence to large-sized poses (rows 2, 4, 5, 6).

Our method, instead, has a runtime inference time comparable to CenterNet's, due to its fast optimization-free test-time procedure, and has increased robustness where our baselines fail. Namely, it performs well in images with heavy occlusion, and, due to its ability to capture long-rage connections with our attention mechanism, it does not struggle with large-sized poses.

4.2. Visualizing Attention Activations

In Figures 4 and 5 we visualize the attention output scores with which the results in Figure 3 were obtained. We observe that despite the large amount of keypoints over which each center attends, particularly in crowded scenes, attention scores are heavily concentrated over a small subset of keypoints, for each center. Indeed, most attention scores for a given type have magnitude over 0.95%, which can be seen from the dark color of most lines. This can be explained due to our loss formulation: to achieve low training error, our model must concentrate attention weights in the most promising keypoint locations, as otherwise it'd incur in large L1 loss values. Overall, Figures 4 and 5 show how our model is able to consider a large number of center-keypoint association candidates but still focus on those keypoints belonging to each pose, even in highly challenging scenarios.

References

[1] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017. 2, 3

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 2

[3] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 3, 4, 6

[4] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE transactions on pattern analysis and machine intelligence*, 29(11):1944–1957, 2007. 2

[5] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour, 2018. 3

[6] Sheng Jin, Wentao Liu, Enze Xie, Wenhai Wang, Chen Qian, Wanli Ouyang, and Ping Luo. Differentiable hierarchical

graph grouping for multi-person pose estimation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 718–734, Cham, 2020. Springer International Publishing. 1, 2

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 3

[8] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Multiposenet: Fast multi-person pose estimation using pose residual network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 417–433, 2018. 3

[9] 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*, pages 11977–11986, 2019. 2

[10] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Hao-Shu Fang, and Cewu Lu. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[11] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. In *International Conference on Learning Representations*, 2018. 3

[12] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 2277–2287. Curran Associates, Inc., 2017. 2, 3

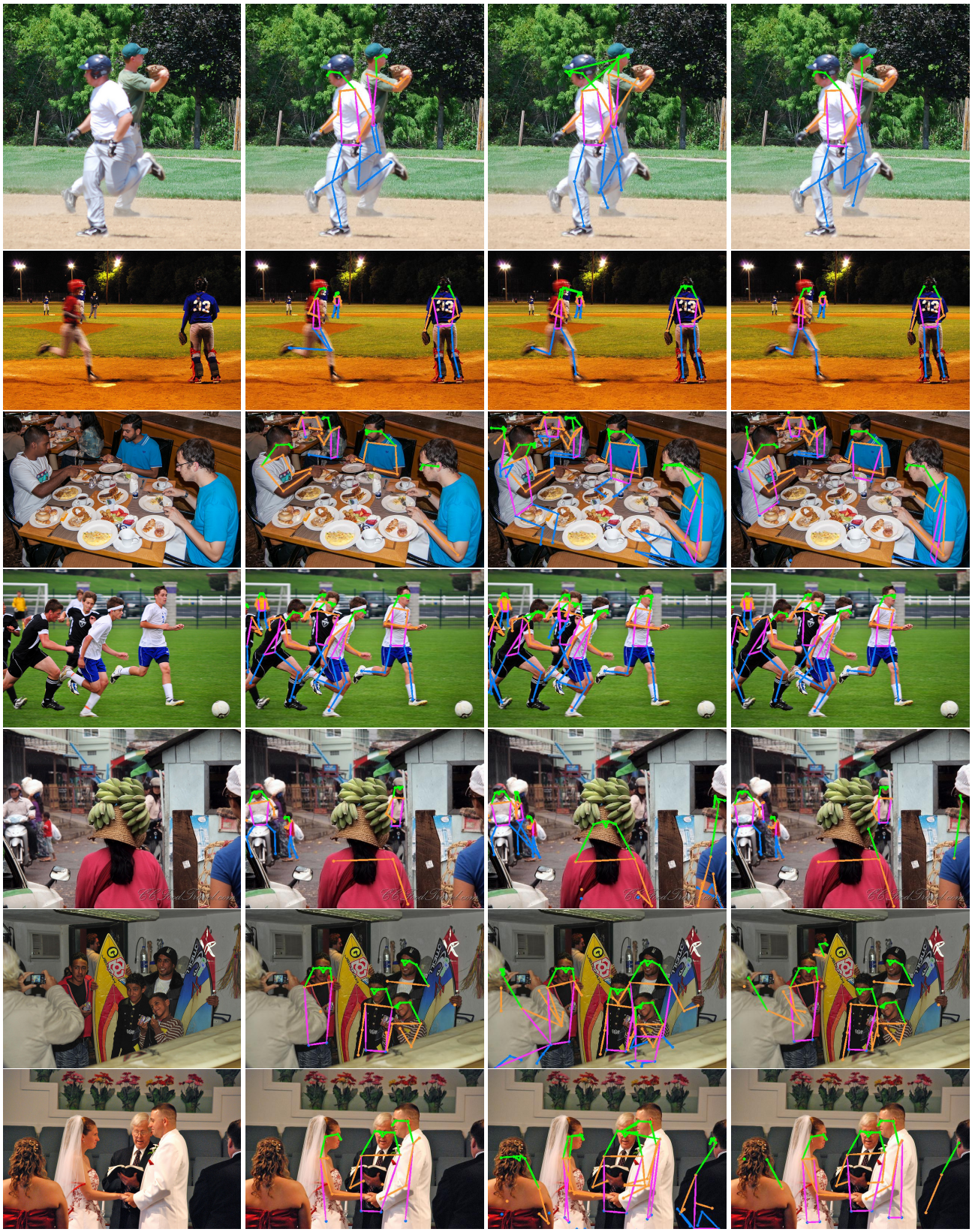
[13] Xuecheng Nie, Jiashi Feng, Jianfeng Zhang, and Shuicheng Yan. Single-stage multi-person pose machines. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6951–6960, 2019. 1, 2, 3

[14] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–286, 2018. 2, 3

[15] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019. 3, 4

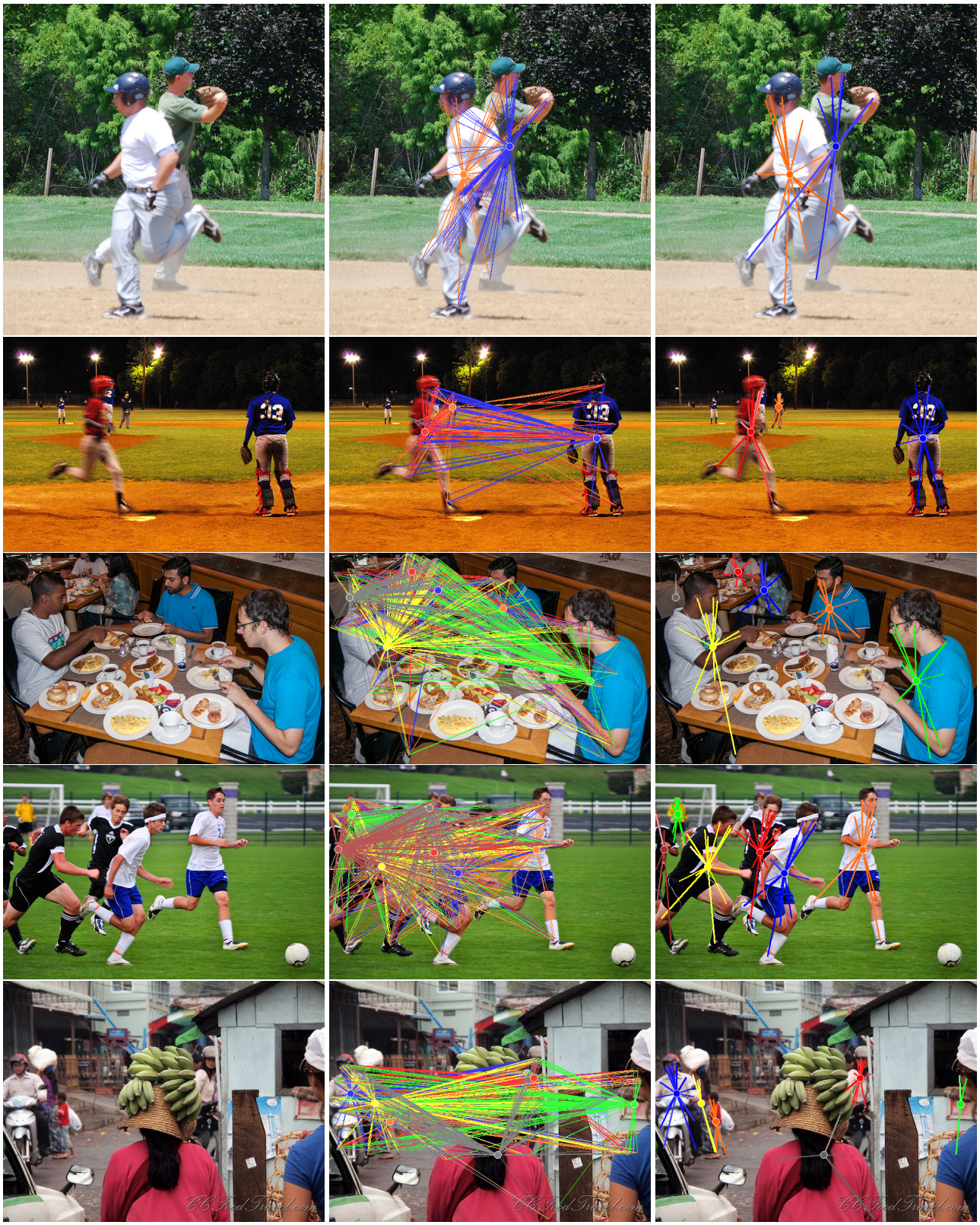
[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 3

[17] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 1, 4, 6



(a) Input Image (b) HigherHRNet [3] (c) CenterNet[17] (d) Ours

Figure 3: Qualitative examples of our method's performance in comparison to HigherHRNet[3] and CenterNet[17]. Best viewed in color and in a screen.



(a) Input Image (b) Center Keypoint Connections (c) Predicted Attention Scores

Figure 4: Visualization of predicted attention scores by our grouping module. In (b) we show all pairwise connections between detected keypoints and centers classified as true positives. In (c) we show all final attention scores predicted with attention weight over 0.5 and as visible. The attention weight is color-coded in the color's intensity. Best viewed in color and in a screen.

(a) Input Image (b) Center Keypoint Connections (c) Predicted Attention Scores

Figure 5: Visualization of predicted attention scores by our grouping module. In (b) we show all pairwise connections between detected keypoints and centers classified as true positives. In (c) we show all final attention scores predicted with attention weight over 0.5 and as visible. The attention weight is color-coded in the color's intensity. Best viewed in color and in a screen.