Research Report for YOLO-Pose: Paper analyze and Replication

BiCheng Weng

2025.1.3

Abstract

This report presents an analysis and implementation of YOLO-Pose, a novel approach for multi-person pose estimation that enhances the YOLO object detection framework. YOLO-Pose introduces a heatmap-free methodology that enables end-to-end training while optimizing the Object Keypoint Similarity (OKS) metric directly. OKS is the most popular metric for evaluating keypoints. The system achieves state-of-the-art results on the COCO dataset for AP50 metrics while maintaining computational efficiency. This work examines the key innovations, architecture, and performance characteristics of YOLO-Pose, highlighting its advantages over traditional top-down and bottom-up approaches.

Keywords: Pose Estimation, YOLO, Computer Vision, Deep Learning, Object Detection.

1 Introduction

Human pose estimation in computer vision involves detecting and localizing body joints for multiple persons in an image. Traditional approaches to this problem typically fall into two categories: top-down methods that first detect persons and then estimate poses, and bottom-up methods that detect all keypoints first and then group them into individual poses. YOLO-Pose introduces a novel unified approach that performs both tasks simultaneously, offering several key innovations.

The innovations of YOLO-Pose are comprehensive and transformative. First, it implements a heatmap-free approach that enables direct keypoint regression. Second, it features end-to-end training capability using the OKS metric. Third, it seamlessly integrates with the YOLO object detection framework. Finally, it provides efficient single-pass inference for multiple persons. This unique combination of features sets YOLO-Pose apart from existing solutions in the field.This report analyzes the YOLO-Pose architecture, implementation details, and performance characteristics, comparing it with existing state-of-the-art methods.

2 Related works

2.1 Top-down Approaches

Top-down methods [3] [9] [14] in pose estimation traditionally follow a two-stage process. First, they employ a person detector to identify and localize individuals in an image. Then, for each detected person, a single-person pose estimation is performed. While these approaches often achieve high accuracy, they suffer from computational complexity that scales linearly with the number of detected persons. Notable examples include Mask R-CNN and Simple Baselines, which utilize heavy backbone networks for detection.

2.2 Bottom-up Approaches

Bottom-up methods [8] [1] [10] take a different approach by first detecting all keypoints in an image and then grouping them into individual poses. These methods, including OpenPose [1] and HigherHRNet [2], typically use heatmaps to detect keypoints and employ various post-processing strategies for grouping. While they offer constant runtime regardless of the number of persons, they often struggle with accuracy and require complex post-processing steps.

3 Method

3.1 Overview

YOLO-Pose builds upon the YOLOv5 object detection framework [4], extending it to handle pose estimation. The system processes images in a single forward pass, producing both person detections and their corresponding pose keypoints. The architecture is comprehensive and includes several key components. At its core is a CSP-darknet53 backbone [13] for feature extraction. This is complemented by PANet for feature fusion across multiple scales. The system also incorporates parallel detection heads for boxes and keypoints, and notably implements a novel OKS-based loss function for direct keypoint optimization.

3.2 Architecture Details

The network architecture follows the YOLO framework, but introduces several significant modifications. The design includes an extended prediction head to handle keypoint coordinates and confidence scores. It implements multi-scale feature fusion using PANet and incorporates parallel box and keypoint prediction branches. Perhaps most importantly, it enables direct keypoint regression without requiring an intermediate heatmap representation.

3.3 Loss Function

One of the key innovations in YOLO-Pose is its loss function formulation. The total loss consists of several components:

$$\mathcal{L}_{total} = \lambda_{box}\mathcal{L}_{box} + \lambda_{kpts}\mathcal{L}_{kpts} + \lambda_{cls}\mathcal{L}_{cls} + \lambda_{kpts\_conf}\mathcal{L}_{kpts\_conf} \tag{1}$$

Where:

- $\mathcal{L}_{box}$ is the CIoU loss for bounding box detection
- $\mathcal{L}_{kpts}$ is the OKS-based loss for keypoint regression
- $\mathcal{L}_{cls}$ is the confidence loss for object detection

- $\mathcal{L}_{kpts\_conf}$ is the keypoint confidence loss
- $\lambda_{box} = 0.05, \lambda_{cls} = 0.5, \lambda_{kpts} = 0.1$ and $\lambda_{kpts\_conf} = 0.5$ are hyper-params chosen to balance between losses at different scales.

4 Implementation details

4.1 Comparing with Released Source Codes

The YOLO-Pose implementation builds upon the YOLOv5 codebase, which is available at https://github.com/ultra The implementation includes several significant enhancements to the original architecture. The detection head has been extended to accommodate keypoint prediction, and a sophisticated OKS loss function has been implemented. The system includes keypoint confidence prediction capabilities and features a modified post-processing pipeline. Throughout the implementation, the original YOLOv5 architecture was preserved where possible to maintain compatibility and leverage its optimizations.

4.2 Experimental Environment Setup

The experimental environment was set up on Windows Subsystem Linux(WSL), uses the Ubuntu 22.04 version. Pytorch was used to train the model. And the train2017 set in COCO dataset [6] is feed to the model. It consists of over 200,000 images with 250,000 person instances with 17 keypoints. The train2017 set includes 57K images, whereas val2017 and test-dev2017 set consists of 5K and 20K images, respectively. The model was trained on train2017 set and results were reported on both val2017 and test-dev2017 sets.

4.3 Main Contributions

The implementation makes several significant contributions to the field. The system successfully integrates pose estimation with object detection, creating a unified approach to human pose analysis. The implementation of the OKS loss function represents a significant advance in training methodology. The system achieves efficient single-pass inference, dramatically improving computational efficiency. Perhaps most importantly, it eliminates the need for complex post-processing steps, streamlining the entire pipeline.

5 Results and analysis

5.1 Performance Metrics

YOLO-Pose demonstrates impressive performance on the COCO dataset. The system achieves an AP50 of 90.2% on the COCO validation set, setting a new standard for accuracy. The YOLOv5l6-pose variant achieves an AP of 69.4%, demonstrating strong overall performance. Most notably, the system shows significantly reduced computational complexity compared to existing methods, making it particularly suitable for real-world applications.

Method	Backbone	Input size	#params	GMACS	AP	AP$_{50}$	AP$_{75}$	AP^{L}	AR
OpenPose [1]	-	-	-	-	61.8	84.9	67.5	68.2	66.5
Hourglass [9]	Hourglass	512	277.8M	413.8	56.6	81.8	61.8	67.0	-
PersonLab [10]	ResNet-152	1401	68.7M	911	66.5	88.0	72.6	72.3	71.0
PiPaf [5]	-	-	-	-	66.7	-	-	72.9	-
HRNet [11]	HRNet-W32	512	28.5M	77.8	64.1	86.3	70.4	73.9	-
EfficientHRNet-H$_0$ [7]	EfficientNetB0 [12]	512	23.3M	51.2	64.0	-	-	-	-
EfficientHRNet-H$_0^{\dagger}$ [7]	EfficientNetB0 [12]	512	23.3M	268.8	67.1	-	-	-	-
HigherHRNet [2]	HRNet-W32	512	28.6M	95.8	66.4	87.5	72.8	74.2	-
HigherHRNet [2]	HRNet-W48	640	63.8M	308.6	68.4	88.2	75.1	74.2	-
HigherHRNet† [2]	HRNet-W48	640	63.8M	1620.2	70.5	89.3	77.2	75.8	-
DEKR [3]	HRNet-W32	512	29.6M	90.8	67.3	87.9	74.1	76.1	72.4
DEKR [3]	HRNet-W48	640	65.7M	283.0	70.0	89.4	77.3	76.9	75.4
YOLOv5s6-pose	Darknet_csp-d53-s	960	15.1M	22.8	62.9	87.7	69.4	71.8	69.8
YOLOv5m6-pose	Darknet_csp-d53-m	960	41.4M	66.3	66.6	89.8	73.8	75.2	73.4
YOLOv5l6-pose	Darknet_csp-d53-l	960	87.0M	145.6	68.5	90.3	74.8	76.5	75.0

Table 1. Comparison with bottom-up methods on COCO2017 test-dev set. Complexity for results with flip-test and multi-scale testing are adjusted with 2x and 5.25x respectively. † indicates multi-scale testing.

5.2 Computational Efficiency

The model exhibits exceptional efficiency across multiple dimensions. It requires only a single forward pass for multiple person detection and pose estimation, dramatically reducing computational overhead. The system operates without requiring test-time augmentation, further improving efficiency. Runtime remains constant regardless of the number of persons in the scene, providing predictable performance characteristics. The model achieves a lower GMAC count compared to similarly performing models, demonstrating superior computational efficiency.

6 Conclusion and Future Work

YOLO-Pose represents a significant advancement in the field of pose estimation, successfully demonstrating that pose estimation can be effectively integrated with object detection. The system achieves state-of-the-art results while maintaining computational efficiency. The heatmap-free approach and direct OKS optimization represent significant advances that open new possibilities in the field.

Looking toward the future, several promising research directions emerge. The system could be extended to support 3D pose estimation, expanding its capabilities into new dimensions. Investigation of lighter backbone networks could further improve efficiency. Integration with real-time tracking systems could enable new applications in video analysis. Finally, the system shows promise for application in specific domains such as sports analysis and medical imaging, where precise pose estimation is crucial.

The implications of this work extend beyond immediate performance improvements, suggesting new directions for unified approaches to computer vision tasks. By demonstrating the feasibility of combining pose estimation with object detection, YOLO-Pose opens the door to further innovations in multi-task computer vision systems.

References

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

[2] 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, 2020.

[3] Zigang Geng, Ke Sun, Bin Xiao, Zhaoxiang Zhang, and Jingdong Wang. Bottom-up human pose estimation via disentangled keypoint regression, 2021.

[4] glenn jocher et al. yolov5. https://github.com/ultralytics/yolov5, 2021.

[5] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation, 2019.

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[7] Christopher Neff, Aneri Sheth, Steven Furgurson, and Hamed Tabkhi. Efficienthrnet: Efficient scaling for lightweight high-resolution multi-person pose estimation, 2020.

[8] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping, 2017.

[9] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation, 2016.

[10] 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, 2018.

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

[12] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020.

[13] Chien-Yao Wang, Hong-Yuan Mark Liao, I-Hau Yeh, Yueh-Hua Wu, Ping-Yang Chen, and Jun-Wei Hsieh. Cspnet: A new backbone that can enhance learning capability of cnn, 2019.

[14] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking, 2018.