A study of human pose estimation in low-light environments using YOLOv8 model

Kaiming Gu[1,3,†], **Boyu Su**[2,†]

[1]International Engineering College, Xi'an University of Technology, Xi'an, 710054, China
[2]School of Intelligent Engineering, Zhengzhou University of Aeronautics, Zhengzhou, 450046, China

[3]3222241013@stu.xaut.edu.cn
[†]All the authors contributed equally and their names were listed in alphabetical order.

Abstract. Human pose estimation is a formidable task in the field of computer vision., often constrained by limited training samples and various complexities encountered during target detection, including complex backgrounds, object occlusion, crowded scenes, and varying perspectives. The primary objective of this research paper is to explore the performance disparities of the recently introduced YOLOv8 model in the context of human pose estimation. We conduct a comprehensive evaluation of six different models with varying complexities on the same low-light photograph to assess their precision and speed. The objective is to determine the suitability of each model for specific environmental contexts. The experimental results reveal that our findings demonstrate a partial regression in accuracy for the yolov8s-pose and yolov8m-pose models when tested on our sampled images. The increase in model layers indicates enhanced complexity and expressive power, while additional parameters signify improved learning capabilities at the expense of increased computational resource requirements.

Keywords: human detection, pose estimation, YOLOv8, low-light environments.

1. Introduction

The advancements in human pose estimation and object detection have resulted in substantial progress within the domain of computer vision. Human pose estimation plays a pivotal role in the detection and localization of human keypoints in images, and it holds immense significance for the advancement of technologies like behavior recognition and pedestrian re-identification. In the field of human pose estimation, deep learning methods founded on convolutional neural networks have exhibited remarkable progress, achieving high accuracy in detecting and localizing human keypoints in both images and videos. As an illustrative example, the DeepPose model approaches the 2D human pose estimation task as a regression problem for keypoint coordinates [1]. Leveraging convolutional neural networks, it extracts pose features from images, thereby attaining elevated and more precise features to predict human keypoint coordinates [2]. This methodology has demonstrated improved performance in the accuracy of human pose estimation. Additionally, models like CPM and Hourglass have also achieved success in 2D human pose estimation, with wide applications not only in behavior recognition and pedestrian re-identification but also in pose analysis and motion tracking [3-4]. These models, when

© 2024 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

applied to the field of object detection, help to make human life more convenient by detecting human pose targets to achieve research objectives.

In the domain of object detection, conventional methods often depend on sequential steps, such as region extraction and feature matching, which can hinder the achievement of efficient real-time detection. Indeed, the emergence of YOLO (You Only Look Once) has effectively addressed this issue. By adopting a unified approach that processes the entire image at once, YOLO achieves real-time object detection without the need for complex intermediate steps like region extraction and feature matching [5]. This streamlined methodology has significantly improved the efficiency and accuracy of object detection tasks. The YOLO model revolutionizes object detection by transforming the problem into an end-to-end regression task. It accomplishes this by predicting both the object's class and bounding box coordinates through a single network. This unique approach enables high-speed and real-time object detection, making it a significant breakthrough in the field of computer vision. The introduction of the latest YOLO model, YOLOv8, brings about significant advancements with three distinct components. Firstly, it incorporates a new and improved backbone network, enhancing the model's overall performance. Secondly, the Anchor-Free detection head offers an innovative approach to object detection, contributing to improved accuracy [6]. Lastly, the utilization of new loss functions further refines the model's capabilities in detecting target objects, making it adaptable to various application scenarios. YOLOv8 effectively achieves enhanced detection performance and accuracy, catering to the diverse needs of different real-world applications.

Both human pose estimation and object detection have seen significant advancements with the utilization of deep learning techniques and YOLO series models, which have not only improved the accuracy of detection and estimation but also accelerated the development of related applications. Among these models, the YOLOv8-based human pose detection methods have attracted considerable attention due to their impressive efficiency and accuracy. However, in practical scenarios, the YOLOv8 model exhibits a large number of parameters and hyperparameters, and selecting different combinations can lead to variations in model performance. Therefore, it becomes imperative to thoroughly study and compare the performance of six models based on the YOLOv8 architecture for human pose detection.

2. Method

This section introduces the basic framework of YOLOv8 and discusses six pose models based on YOLOv8.

2.1. Introduction of YOLOv8

The YOLOv8 is an object detection algorithm that uses deep learning and Anchor-Free detection head to achieve fast detection and classification of objects in images [6]. YOLOv8 inherits the high-precision detection methods of the YOLO series models. Indeed, YOLOv8 models the object detection task as a regression problem, where the neural network directly predicts the bounding box coordinates and class of the detected objects. This streamlined approach simplifies the detection process and contributes to the model's high-speed and real-time capabilities, making it an efficient solution for object detection in various applications. Compared to previous models, YOLOv8 potentially achieves higher detection speed and more accurate results. The Network Structure of YOLOv8 is shown in figure 1.

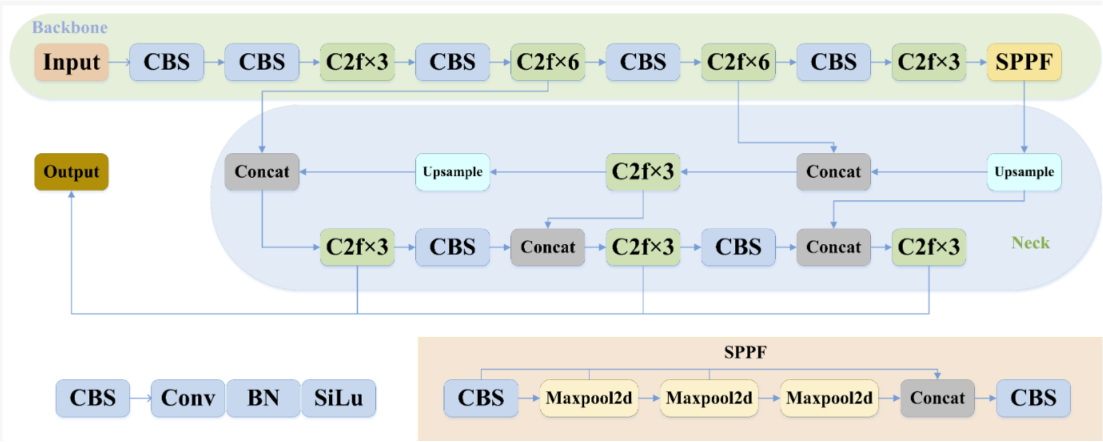

Figure 1. The network structure of YOLOv8 [7].

YOLOv8 uses the CSPDarkNet53 as its backbone network. The key highlight of the YOLOv8's backbone network is the implementation of a novel connection method known as Cross Stage Partial [8-9]. This technique optimally utilizes computational resources, ensuring both high accuracy and improved speed of the model. By intelligently leveraging these resources, YOLOv8 achieves a balance between accuracy and efficiency, making it a powerful tool for object detection tasks in real-world scenarios.

YOLOv8 also employs an Anchor-Free detection head, which directly predicts the object's bounding box from the feature map and adaptively adjusts the size and position of the bounding box [6]. This approach reduces the complexity of model training to some extent. In contrast to traditional detection heads, which adopt the Anchor-Based method and require predefined anchors of different sizes and positions for object localization and detection, the Anchor-Free detection head achieves better results, as adjusting anchor size and position in the Anchor-Based method can be complex [10].

In addition, YOLOv8 uses the CIOU and DFL loss functions [11]. The DFL loss function is defined as

$$DFL(S_i, S_{i+1}) = -\big((y_{i+1} - y)\log(s_i) + (y - y_i)\log(s_i + 1)\big) \tag{1}$$

In the DFL , s_i is the sigmoid function's output for the network. y_i and y_{i+1} are interval orders, and y denotes a label. This loss function handles overlapping between predicted bounding boxes and ground truth bounding boxes, improving the detection accuracy and robustness of small objects. The CIOU (Complete Intersection over Union) and DFL (Dynamic Focal Loss) loss functions in YOLOv8 are designed to measure the distance between predicted and ground truth bounding boxes, leveraging the IOU (Intersection over Union) value as a key metric. By incorporating a combination of cross-entropy loss and mean square error loss, these loss functions effectively optimize the model during training. This approach allows YOLOv8 to achieve higher precision and accuracy in detecting objects, further improving its performance in various object detection tasks.

2.2. Pose models based YOLOv8

In the YOLOv8 series, there are six pose estimation models (YOLOv8s-pose, YOLOv8n-pose, YOLOv8m-pose, YOLOv8l-pose, YOLOv8x-pose, YOLOv8x-pose-p6) that add the functionality of human pose estimation to the YOLOv8 model. These six models have different backbone network scales that balance accuracy and computational resources. Smaller models are suitable for resource-constrained environments, with lower computational and memory overhead, but may have limitations in pose estimation accuracy. Larger models offer higher pose estimation accuracy but require more computational resources and memory.

The pose estimation models in the YOLOv8 series combine the functionality of object detection and human pose estimation, exhibiting efficiency and accuracy. Absolutely, the selection of the appropriate model depends on the specific requirements of the task at hand. It involves striking a balance between computational resources and accuracy to achieve fast and precise object detection and pose estimation tasks. For scenarios where real-time performance is crucial, models like YOLOv8 that optimize speed while maintaining acceptable accuracy would be preferred. On the other hand, for applications where precision is of utmost importance and computational resources are less constrained, more sophisticated and accurate models might be chosen. The ultimate goal is to choose a model that best fits the practical needs and constraints of the given task.

The specific characteristics of each pose estimation model in the YOLOv8 series are as follows:

- YOLOv8s-pose

- Description of the backbone network: CSPDarkNet53.

- Characteristics of YOLOv8s-pose: small size, suitable for resource-constrained environments.

- Performance analysis: low computational and memory overhead, potential limitations in pose estimation accuracy.

- YOLOv8n-pose

- Description of the backbone network: CSPDarkNet53.

- Characteristics of YOLOv8n-pose: medium size, balance between accuracy and computational resources.

- Performance analysis: improved pose estimation accuracy compared to YOLOv8s-pose, increased computational and memory overhead.

- YOLOv8m-pose

- Description of the backbone network: CSPDarkNet53.

- Characteristics of YOLOv8m-pose: large size, better balance between accuracy and computational resources.

- Performance analysis: higher pose estimation accuracy compared to YOLOv8n-pose, increased computational resources and memory requirements.

- YOLOv8l-pose

- Description of the backbone network: CSPDarkNet53.

- Characteristics of YOLOv8l-pose: larger size, higher pose estimation accuracy.

- Performance analysis: improved accuracy and precision compared to YOLOv8m-pose, higher computational resources and memory requirements.

- YOLOv8x-pose

- Description of the backbone network: CSPDarkNet53.

- Characteristics of YOLOv8x-pose: largest size among the YOLOv8 series, highest pose estimation accuracy.

- Performance analysis: maximum computational resources and memory requirements.

- YOLOv8x-pose-p6

- Description of the backbone network: CSPDarkNet53.

- Characteristics of YOLOv8x-pose-p6: larger input resolution.

- Performance analysis: bette

3. Experimental analysis

3.1. Experimental details

Dataset: The coco128 dataset, widely used in computer vision, is utilized by the authors for training and evaluation purposes in this study. It includes images from 128 different categories, and each image may contain multiple object instances, thus possessing multi-label attributes.

Setup: The Google Colab experimental platform on GitHub is a cloud-based interactive computing environment that provides a free integrated Jupyter notebook for coding and running code directly in the browser. It supports GPU and TPU acceleration, seamlessly integrates with Google services, and has seamless connectivity with GitHub, facilitating user sharing and collaboration. The platform comes pre-installed with popular data science and machine learning libraries, providing convenient access to common datasets. By leveraging Google's computational resources, users can fully exploit the potential of cloud computing for conducting experiments and projects in the fields of data science, machine learning, and deep learning.

3.2. Experimental results analysis

From YOLOv8n-pose to yolov8x-pose-p6, there is an increase in the complexity of the model, leading to longer testing time requirements. The following figure 2 presents the test results of six different YOLOv8 models with varying complexities on the same photo.

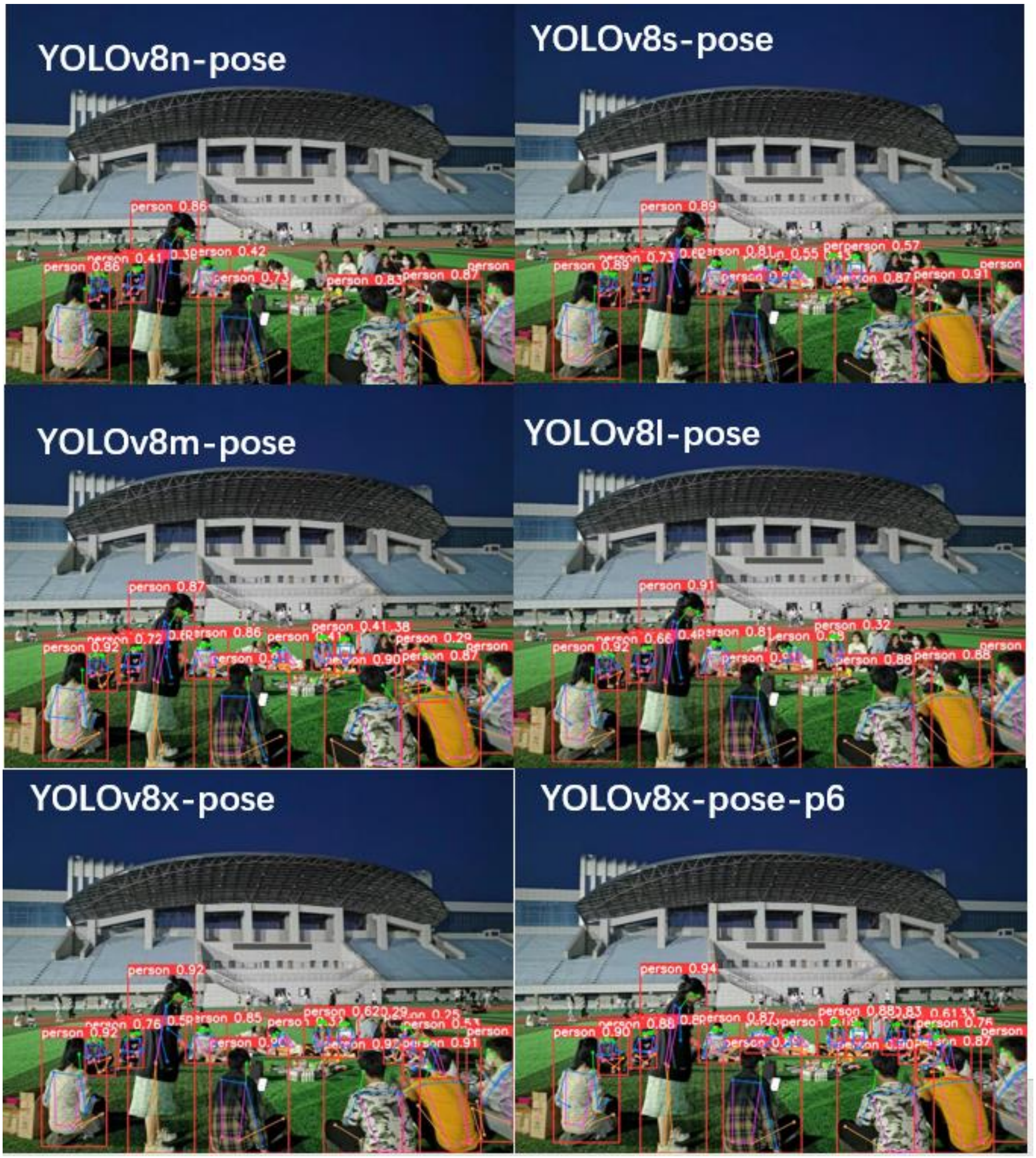

Figure 2. Experimental results in a low-light environment.

Overall, as the model complexity increases, the overall detection accuracy also improves. Nonetheless, when tested on images, the YOLOv8s-pose and YOLOv8m-pose models demonstrate a partial regression in accuracy. In particular, the accuracy of detecting the main subject on the left side decreased from 0.89 to 0.87, and the accuracy of detecting the person wearing a yellow shirt on the right side also decreased from 0.91 to 0.87. Additionally, the detection accuracy of the objects in the background, excluding the one in the middle, also experienced a decline. It is speculated that for the test targets, the results obtained by YOLOv8s-pose and YOLOv8m-pose are almost identical, with YOLOv8m-pose having slightly more layers and parameters than YOLOv8s-pose. Thus, within this range, increasing the model complexity does not necessarily lead to more accurate results in testing. On the contrary, it may result in longer model runtime, leading to repeated detections of a single target and consequently causing misidentification and decreased precision. Similarly, looking at the detection results of YOLOv8l-pose, a higher number of layers and parameters should ideally yield more accurate results, but it only detected eleven objects and had lower detection accuracy in the background.

Comparison of Metrics: In the provided results (Table 1), the preprocessing time refers to the duration taken to prepare and preprocess the images before feeding them into the model. The inference time represents the period consumed by the model to perform object detection predictions on the preprocessed images. Lastly, the post-processing time signifies the time spent on processing and organizing the model's outputs after the inference step, to present the final detected objects and their respective bounding boxes.

Table 1. Index comparison of six models.

Model Type	Number of Layers	Parameters (M)	Time Cost(ms)	Detected Persons	Preprocessing Procedure(ms)	Inference Process(ms)
YOLOv8n-pose	187	3.3	135.6	9	3.5	135.6
YOLOv8s-pose	187	11.6	375.4	13	2.4	375.4
YOLOv8m-pose	237	26.4	880.2	13	2.5	880.2
YOLOv8l-pose	287	44.5	1714.3	11	2.5	1714.3
YOLOv8x-pose	287	69.5	2555.4	14	2.6	2555.4
YOLOv8x-pose-p6	375	99.1	10373.2	16	12.6	10373.2

Analysis of Metrics: From the perspective of preprocessing and inference time, YOLOv8x-pose-p6 has the longest runtime. This indicates that it takes more time to handle issues such as scaling, cropping, and normalization due to the model's extensive layers and complex connections. Each input needs to pass through these layers, resulting in increased computational workload and correspondingly increased runtime. When performing higher-resolution scaling on images, employing more complex cropping strategies, or executing additional normalization steps, these additional preprocessing steps increase the model's processing time. Additionally, this model has the largest number of parameters, requiring more parameter calculations and storage during inference and training, leading to increased runtime. A larger number of parameters also increases the memory requirement, thereby slowing down the model's execution speed.

Increasing the number of layers in a model may imply greater complexity and expressive power, while parameters indicate more learning capacity but can also result in increased computational resource requirements. The above table displays the time required by each model for processing the photo. Among the models, YOLOv8n-pose is the fastest, with the highest inference speed and the smallest number of layers, but it does not yield the most accurate results. On the other hand, YOLOv8x-pose-p6 has the longest runtime, the most layers, and the slowest inference speed, but it delivers the most accurate results. The preprocessing, inference, and post-processing times represent the computational overhead at different stages of the model. The table clearly demonstrates that as the model type progresses to more advanced versions, both the model complexity and the number of parameters increase significantly. If the fastest inference speed is desired, YOLOv8n-pose can be selected. If higher detection accuracy is required, YOLOv8x-pose-p6 can be chosen.

4. Conclusion

To evaluate the results of different performance models on the same image, this paper analyzes six different pose models based on YOLOv8, discusses their principles, and focuses on analyzing the meanings of various parameter values after testing the six models on the same image. The analysis results indicate that increasing model complexity does not necessarily improve test accuracy and may instead lead to longer model runtimes, resulting in misidentification and decreased precision. Increasing the number of layers in a model implies increased complexity and expressive power, while increasing the number of parameters indicates increased learning capacity but can also result in higher computational resource requirements. Furthermore, the preprocessing, inference, and post-processing stages are associated with computational overhead, and different models handle these costs differently. The YOLOv8n-pose model is the fastest, with the highest inference speed, but it does not yield the most accurate results. On the other hand, the YOLOv8x-pose-p6 has the longest runtime, the most layers, and the slowest inference speed, but it achieves the highest accuracy. Indeed, from the aforementioned results, it is evident that both model complexity and the number of parameters increase as the model type advances.

In summary, as model complexity increases, detection accuracy generally improves, but within a certain range, further increasing model complexity may lead to longer runtimes, misidentification, and decreased precision. Selecting a suitable model requires considering inference speed and detection accuracy and striking a balance based on specific requirements. The preprocessing, inference, and post-processing stages also contribute to computational overhead, and for different application scenarios, it is possible to choose an appropriate model that achieves the optimal balance according to the specific needs.

References

[1] Toshev, A., & Szegedy, C. DeepPose: Human Pose Estimation via Deep Neural Networks. *In IEEE Conf. Comp. Vis. Patt. Recogn.* 2014, 1653-1660.

[2] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, 1998, 86(11), 2278-2324.

[3] Zhang, H., Cui, Y., Zhang, L., Wu, S., & Zhang, H. CPM: A Large-scale Generative Chinese Pre-trained Language Model. arXiv preprint arXiv:2012.00413,2020.

[4] Newell, A., Yang, K., & Deng, J. Stacked Hourglass Networks for Human Pose Estimation. *Euro. Conf. Comp. Vis.*, 2020, 483-499.

[5] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *In IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016, 779-788.

[6] Zhou, X., Wang, D., & Krähenbühl, P. Objects as Points. arXiv preprint arXiv:1904.07850, 2019.

[7] Haitong Lou, Xuehu Duan, Junmei Guo, DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor. *Electronics* 2023, 12(10), 2323.

[8] Y. Chen et al., "CSPDarkNet53: A Light-weight Convolutional Neural Network for Object Detection with a Comprehensive Evaluation," *IEEE Trans. Cir. Sys. Video Tech.*, 2020, 30 1, 1-14.

[9] He, K., Zhang, X., Ren, S., & Sun, J.. Deep Residual Learning for Image Recognition. In *In IEEE Conf. Comp. Vis. Patt. Recogn.*, 2021, 37-51.

[10] Ren, S., He, K., Girshick, R., & Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *In Adv. Neur. Inform. Proc. Sys.* 2015, 91-99.

[11] Zhu, X., He, C., & Zhang, J. Distribution Focal Loss for Dense Object Detection. *In IEEE Conf. Comp. Vis. Patt. Recogn.* 2019, 12814-12823.