**International Journal of Scientific Research and Management (IJSRM)**
**||Volume||14||Issue||03||Pages||2818-2828||2026||**
**Website: https://ijsrm.net ISSN (e): 2321-3418**
**DOI: 10.18535/ijsrm/v14i03.ec06**

# An improvement based on the hand pose model of YOLOv11-pose

**Ziru Zhang[1*]**

*School of Physics and Optoelectronic Engineering, Yangtze University*

**Abstract:** Existing one-stage pose estimation models, such as the YOLO series, face challenges in balancing real-time performance and accuracy. Traditional convolutional operations suffer from spatial insensitivity and channel redundancy when extracting spatial features, limiting the model's ability to adaptively model the spatial context of keypoints. Meanwhile, conventional feature pyramid networks often employ simple addition or concatenation when fusing multi-scale features, ignoring the differences in contributions across different scales, leading to suboptimal information fusion. To address these issues, this paper modifies the official Hand Keypoints dataset from Ultralytics, obtaining 5000 images after data augmentation and processing. It proposes an innovative Involution-BiFPN collaborative augmentation network, YOLOv11n-BMT, which is seamlessly integrated into the latest generation YOLOv11-pose framework. The core idea of this design is to introduce an Involution operator in the critical stages of the backbone network to enhance the model's specific perception of the local spatial structure of pose keypoints; and to employ a weighted bidirectional feature pyramid network (BiFPN) in the neck network to achieve efficient and adaptive multi-scale feature fusion. The two work together to improve the model's accuracy in detecting human posture and locating key points in complex scenarios.

**Keywords:** YOLO-Pose; handpose detection; deeplearning; keypoint detection; computer vision

## 1. Introduction

Human hand detection is increasingly widely used in mainstream fields such as virtual/augmented reality (VR/AR) [1] and robotics [2]. The hand detection task can be further subdivided into three sub-problems: palm detection, hand classification and hand pose estimation. Among them, palm detection aims to locate the palm region using bounding boxes; hand classification is used to determine whether the detected hand belongs to the left or right hand; and pose estimation requires identifying and locating the positions of key points such as fingertips. In addition, if combined with a depth-based camera system, pose estimation can also be extended to detect key points of the hand in three-dimensional space. This paper proposes a hand pose detection method based on a public hand dataset and the YOLO (You Only Look Once) convolutional neural network [3]. Due to its efficiency, the YOLO framework has great potential in real-time application scenarios, such as sign language translation, augmented/virtual reality and teaching-based robot learning.

Hand detection is essentially a category of object detection. Most current related studies regard hand detection, classification and pose estimation as a two-stage process [4]. Considering the structural symmetry of the left and right hands, the common approach is to first locate and distinguish the hands through human pose estimation, and then treat the left hand as the right hand for pose estimation [5]. Since ImageNet [6] promoted the significant achievements of deep learning based on convolutional neural networks in object classification, methods such as YOLO, SSD (Single Shot MultiBox Detector) and R-CNN (Region-based CNN) have further realized the joint localization and classification of objects. The study of YOLO-pose [7] shows that with bounding box and key point annotation information provided, an end-to-end single-stage network can be trained to complete human detection and pose estimation at the same time. Based on this idea, this paper assumes that if class labels, bounding boxes and key point annotations are provided for hand data, training an end-to-end network is expected to effectively achieve the goal of hand pose detection.

YOLO has attracted much attention due to its end-to-end network architecture and real-time processing capabilities [8]. YOLOv11 further expands the capabilities of this framework, enabling hand detection, classification and pose estimation to be completed synchronously in a single stage through an end-to-end network. In addition, the Hand Keypoints dataset provides ample support for the training of deep networks. To systematically study this problem, we extract and augment the images and their annotation information from the Hand Keypoints data and convert them into a format usable by YOLO [9]. Subsequently, we train on YOLOv11 pose networks of different scales and conduct comparative analysis on the models trained on different hand datasets. The work of this study verifies the feasibility of using end-to-end deep networks to solve the problem of hand pose detection. The following are the main contributions of this study:

(1) Modifications to the Hand Keypoints dataset include bounding box input box positions, and data enhancements include geometric transformations and color light deception enhancements.

(2) YOLO-Pose with different sizes was trained, and the network performance parameters were compared.

(3) Through quantitative and qualitative analysis, the advantages and disadvantages of the end-to-end method are identified, and the hand-box ratio is used as the evaluation index.

## 2.1. Improve the model architecture

Existing one-stage pose estimation models, such as the YOLO series, face challenges in balancing real-time performance and accuracy.Traditional convolutional operations suffer from spatial insensitivity and channel redundancy when extracting spatial features, limiting the model's ability to adaptively model the spatial context of keypoints. Meanwhile, conventional feature pyramid networks often employ simple addition or concatenation when fusing multi-scale features, ignoring the differences in contributions across different scales, leading to suboptimal information fusion. To address these issues in gesture recognition, this chapter proposes an innovative YOLO-BMT model and seamlessly integrates it into the latest generation YOLOv11-pose framework.

To address the high flexibility of hand joints in gesture recognition, the standard convolution (Conv) in the C3K2 block of Backbone is replaced with the Involution deconvolution operator. C3k2 primarily relies on traditional standard convolution, whose core characteristics are "spatial sharing and channel specificity." This means that the weights of the convolution kernel are fixed at various locations in the image, making it difficult to flexibly capture the extremely varied differences in finger poses. Involution, on the other hand, is "spatial specificity and channel sharing." It dynamically generates weights based on the current position of the input feature map (similar to a bottom-up spatial attention mechanism). This dynamic weight can adaptively align with the complex spatial arrangement of hand joints and more accurately extract local topological features of specific gestures, significantly improving the localization accuracy of keypoints. Thus, the spatial specificity of Involution enhances the modeling ability of complex non-rigid hand deformations, and overcomes self-occlusion with a large receptive field, achieving a balance between lightweight design and high accuracy.

To address the core pain points of insufficient multi-scale feature representation and rigid feature fusion weights, the traditional concat operation in the YOLO11n-pose Neck network was upgraded to BiFPN (Bidirectional Feature Pyramid Network) and integrated into a CBP module. In the field of hand keypoint detection, BiFPN introduces a fast normalized fusion mechanism. While the original PANet structure of YOLO11 has top-down and bottom-up paths, the information flow path is long, and high-frequency spatial details originally used to locate small-scale targets (such as bent finger joints) are easily diluted during multiple passes. BiFPN removes nodes with only one input edge (because they contribute little to feature fusion) and adds extra skip connections between input and output nodes at the same level. This design constructs a denser bidirectional information flow without significantly increasing computational cost, ensuring that high-resolution coordinate information from the lower layers can be more directly passed to the final prediction head, effectively preventing the loss of features from tiny keypoints.

2.2 BiFPN Bidirectional Feature Pyramid

Bidirectional Feature Pyramid Network[10] (BiFPN) is a highly efficient multi-scale feature fusion network that optimizes the structure and improves the mechanism of the traditional Feature Pyramid Network (FPN). Its main innovations include three aspects: First, it constructs efficient bidirectional cross-scale connections, allowing feature information to flow and interact fully between top-down and bottom-up paths, significantly improving the fusion effect of features at different scales[11]. Second, it introduces a learnable weight mechanism to adaptively adjust the importance of each input feature during feature fusion, making the network more focused on information-rich feature representations. Third, it reasonably simplifies the network structure by removing redundant nodes with only single inputs, adding connection edges between input and output nodes at the same level, and treating each bidirectional path as a feature network layer that can be repeatedly stacked, effectively optimizing the overall architecture of cross-scale connections. I will use the following diagram to compare and demonstrate the differences between BiFPN and four other different feature pyramid network designs, and how BiFPN more effectively integrates features:

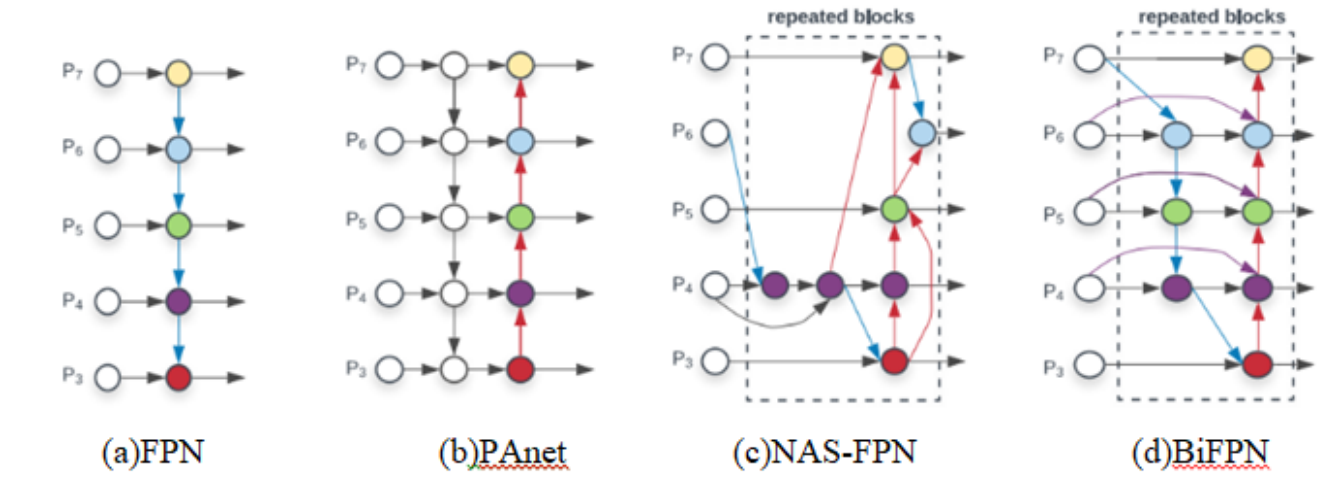



***Figure** 1?Feature Pyramid Network Structure Diagram*

(a):FPN (Feature Pyramid Network): A top-down path is introduced to fuse multi-scale features from layer 3 to layer 7 (P3 - P7).(b):PANet:It adds an extra bottom-up path on top of FPN.

(c):NAS-FPN:Use Neural Architecture Search (NAS) to find irregular feature network topologies, and then repeat the same blocks.

(d):BiFPN:The trade-off between accuracy and efficiency is improved through efficient bidirectional cross-scale connections and repetitive block structures.

As can be seen from the analysis in Figure 1 above, BiFPN achieves bidirectional flow and interaction of feature information across different scales by constructing bidirectional paths. This bidirectional flow can essentially be viewed as an effective exchange and integration of information between features at different scales. This design aims to improve the efficiency and effectiveness of multi-scale feature fusion by enhancing the bidirectional feature transfer mechanism across scales, thereby optimizing the overall performance of object detection.

**2.3 Involution (inverse convolution operator)**

In the DilateFormer paper [12], convolution is undoubtedly one of the foundational operators in the evolution of deep learning, especially in computer vision tasks where it has become a core component of network architecture. Classical two-dimensional convolution extracts and aggregates local features by performing a sliding window operation on the input feature map and using a shared convolution kernel. Before delving into a deeper theoretical discussion, it is necessary to re-examine the two essential properties of traditional convolution: spatial-agnostic and channel-specific.

Spatial invariance means that the same set of convolutional weights is forced to be applied to all spatial locations. While this design improves parameter efficiency, it limits the model's ability to adaptively represent differences in spatial distribution. On the other hand, channel specificity means that each output channel corresponds to an independent convolutional kernel, which can easily introduce parameter redundancy in deep, high-dimensional features, affecting the model's compactness and generalization performance.

To overcome the aforementioned limitations, this paper focuses on an inversely designed operator--Involution--whose basic idea contrasts sharply with convolution: it possesses spatial-specificity and channel-agnosticity. Specifically, Involution dynamically generates a unique kernel function for each spatial location and shares weights across the channel dimension, thereby achieving the following significant advantages:

Enhanced spatial adaptive modeling capabilities: Each pixel location has its own dedicated processing kernel, enabling more flexible capture of local structure and contextual information;

Significantly improved parameter efficiency: Through the channel sharing mechanism, the number of parameters is greatly reduced, which is especially beneficial for the lightweight deployment of deep high-channel-count networks;

Efficient modeling of long-range dependencies: By flexibly expanding the kernel size, Involution can establish long-distance spatial relationships without significantly increasing the computational burden, balancing the extraction of local details and global semantics.

In short, this operator generates a dynamic kernel in the spatial dimension and groups and shares it in the channel dimension, thereby achieving adaptive processing of features at different locations and effectively suppressing redundant responses between channels. Its design principles and implementation mechanism will be systematically explained below.

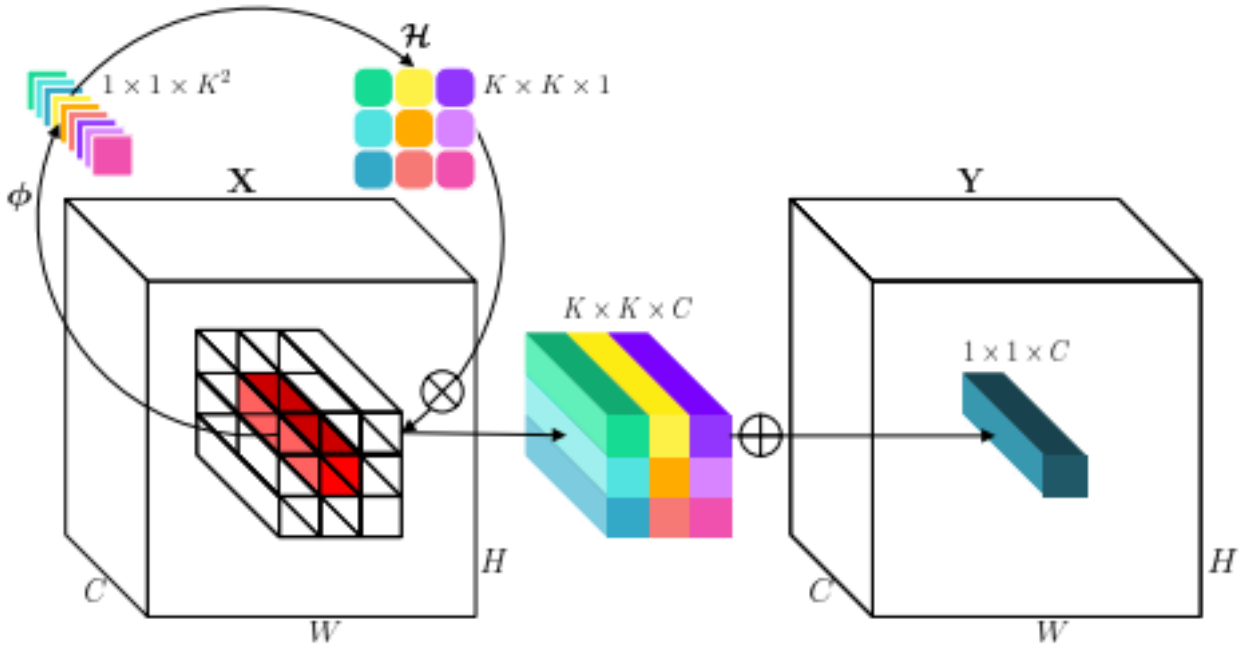



***Figure** 2?Involution Schematic diagram*

In this example, for demonstration purposes, involutionHij=RKxKx1(G = 1 is conditionalized on the function f at the single pixel (i, j))?Then, a new permutation of the channels into space is performed to generate the involution. The multiply-add operation of involution is decomposed into two steps.

Where N represents the multiplication broadcast on channel C and L represents the summation aggregation in the K x K spatial neighborhood.

In the original YOLO11n-pose model, the C3k2 module undertakes the main feature extraction task. However, the standard 3x3 convolutions used by its internal Bottleneck have spatially shared static weights, making it difficult to adaptively focus on the highly flexible topology of the hand.

This paper refactors the C3k2 module, designing the In_C3K2 module. The 1x1 convolutions in Bottleneck are retained to maintain information flow between channels, while the standard 3x3 convolutions are replaced with Involution operators. This modification allows the network to adaptively allocate spatial weights based on the specific content of the hand image during the core feature extraction stage, significantly enhancing the model's ability to represent complex gesture features with almost no additional computational overhead. Figure 3 shows the improved Bottleneck and its schematic diagram within the C3k structure.

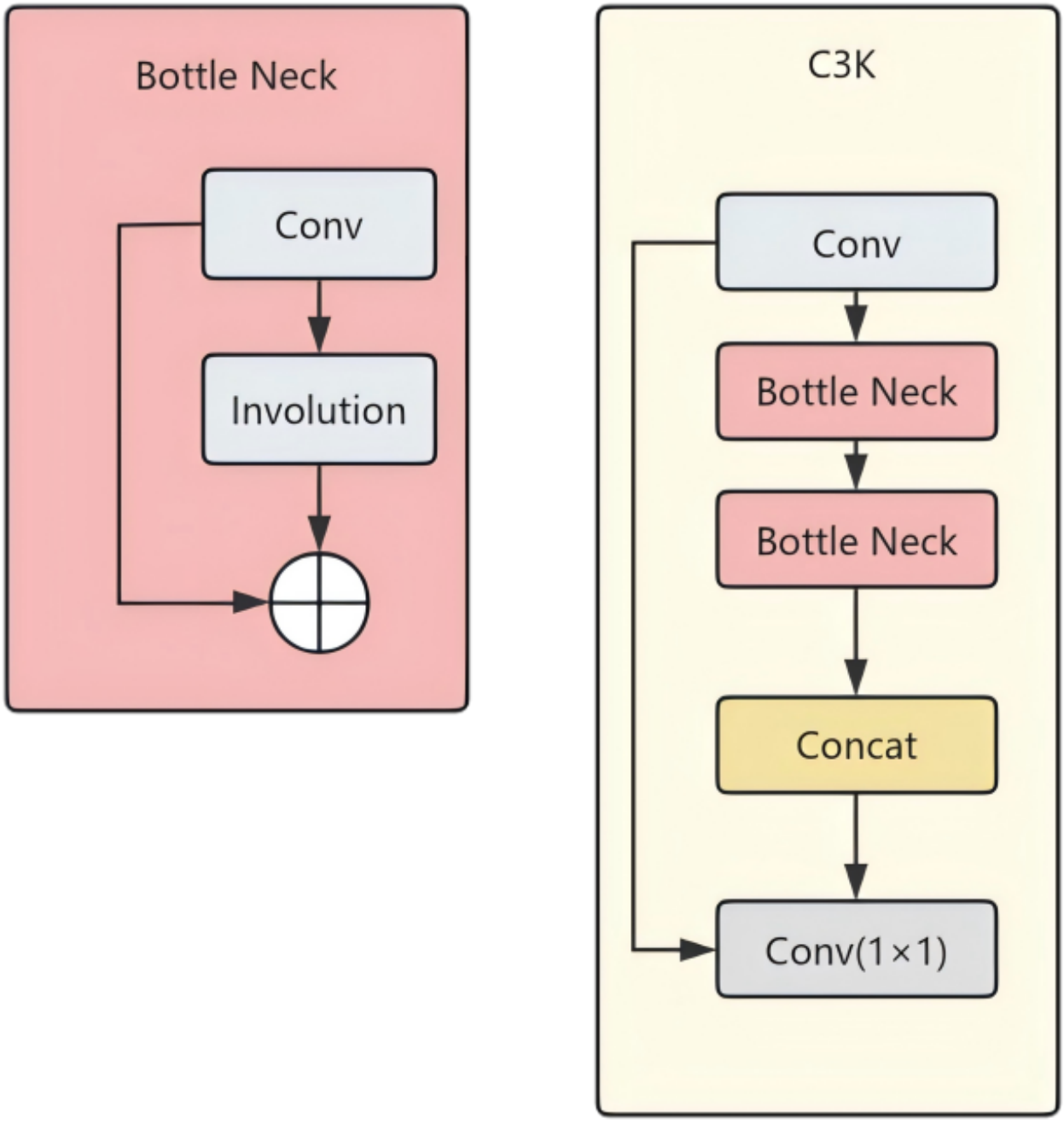



***Figure** 3?Bottleneck Structural diagram*

## 3. Design of a hand pose detection method based on YOLO-Pose

### 3.1 Overall architecture design

Our model follows the overall workflow of YOLOv11-pose, namely "Backbone -> Neck -> Head". We have made targeted enhancements to the backbone and neck networks.

Improved Backbone (Involution-Augmented Backbone):

In the deeper stages of the YOLOv11 backbone network (e.g., before the last two C2f modules or SPPF modules), we selectively replace or introduce involution layers in parallel. Specifically, we replace the original standard convolutions with spatially specific involution operations.

Design Motivation: Pose estimation heavily relies on the precise spatial location of joints and their local relationships (e.g., wrist and elbow). Involution generates a dynamic kernel for each pixel location, enabling the model to adaptively aggregate long-range contextual information based on image content. This is crucial for distinguishing overlapping human figures, occluded joints, and understanding limb orientation. This approach delivers a significant improvement in spatial modeling capabilities with minimal parameter increases.

Enhanced neck network (BiFPN-based Neck):

We abandoned the PANet structure used in the original YOLOv11 and adopted the efficient weighted bidirectional feature pyramid network (BiFPN) as our neck.

Design Motivation: Pose estimation requires fusing multi-scale features from high resolution (rich in detail) to low resolution (rich in semantics). BiFPN adaptively measures the importance of different input features through learnable weights and performs efficient cross-scale bidirectional (top-down + bottom-up) fusion. This mechanism ensures that the keypoint feature maps used for prediction possess both accurate local details and rich global semantics, which is particularly beneficial for detecting human targets at different scales.

Involution in the backbone network first produces feature maps with stronger spatial awareness, which more accurately encode the latent locations and local associations of keypoints. These enhanced multi-scale features are then fed into BiFPN. BiFPN's weighted fusion mechanism intelligently determines which type of spatial context information enhanced by Involution is more important at which scale, thus performing targeted fusion. Ultimately, the detection head receives a set of high-quality features that are both spatially adaptive and optimally fused across multiple scales, thereby simultaneously improving the accuracy of human bounding box detection and keypoint coordinate regression.

### 3.2 Key point detection strategy

This study's hand pose detection framework centers on keypoint detection, which directly determines the system's ability to recognize hand details. Based on the YOLOPose architecture, a multi-task learning framework is employed, unifying object detection and keypoint prediction into a single network. Specifically, the system predicts 21 keypoints for each hand region, including 5 fingertips, 14 knuckles, and 2 palm heels. These keypoints are represented by high-dimensional features consisting of coordinate values (x, y) and visibility confidence (v). To improve keypoint localization accuracy, a heatmap regression mechanism is introduced, transforming the keypoint prediction problem into pixel-level probability distribution prediction and combining it with coordinate regression to refine positional information. Furthermore, a decoupled head design is introduced, allowing the network to separately optimize keypoint location and visibility prediction tasks, effectively addressing partially occluded scenarios.

### 3.3 Bounding box generation

The Hand Keypoints dataset needs to be converted to YOLO pose format. For YOLO11-pose, each hand annotation must meet the requirements in (3-1).

L = {c?, x, y, w, h, kx?, ky?, ..., kx??, ky??} (3-1)

Here, cl represents the hand class, (x, y, w, h) provides the bounding box dimensions using the center coordinates (x, y) and the width and height (w, h), and (kxi, kyi) are the coordinates of the i-th hand keypoint. The 21 keypoints for hand detection were developed for pose estimation.

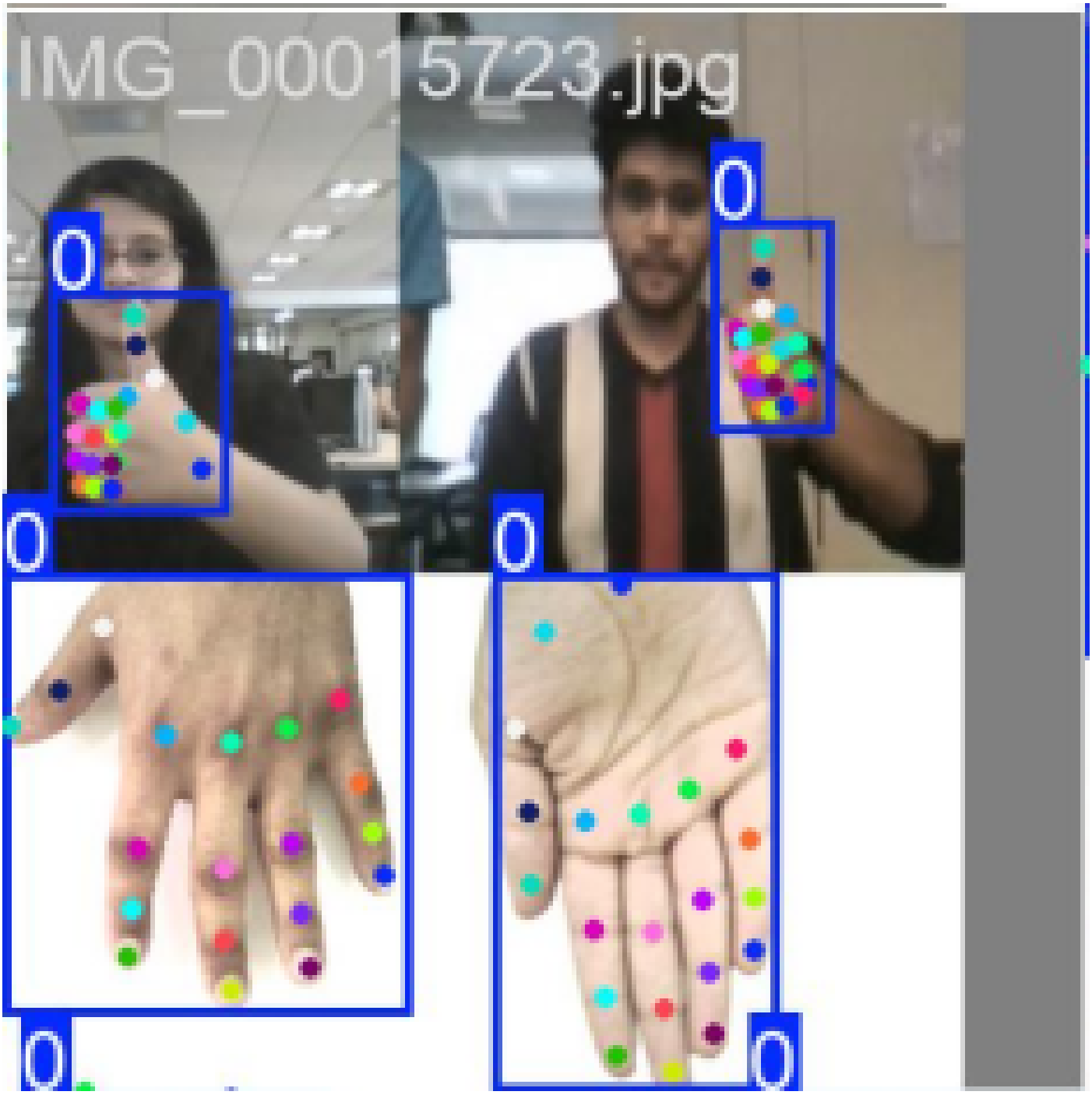


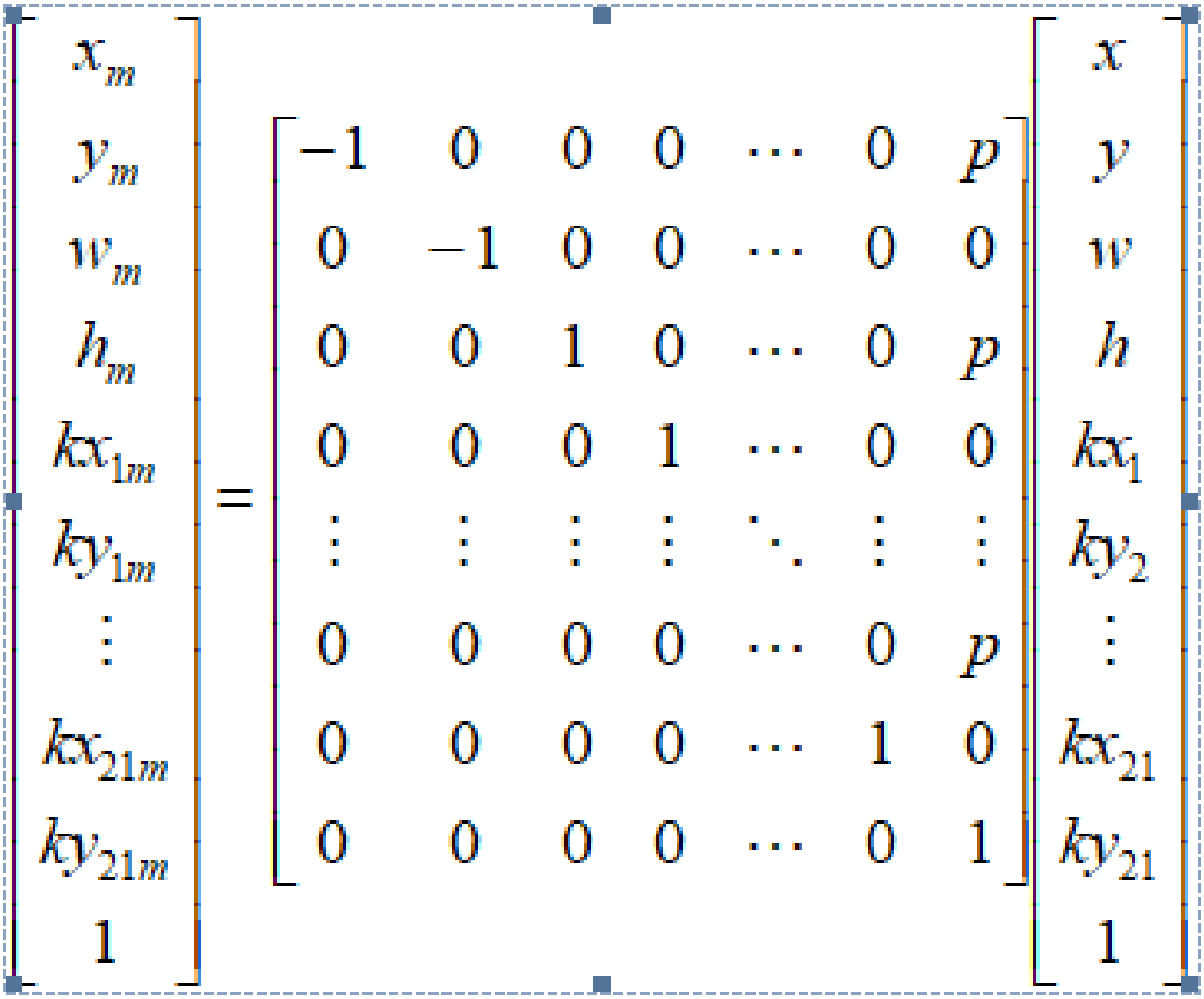

***Figure** 4?Bounding box generation*

**3.4 Hand key point extraction**

The Hand Keypoints dataset provides the camera coordinates and camera intrinsic parameter matrix for 21 keypoints. The homogeneous image plane coordinates are calculated using equation (3-2).

**? = M int x_c** (3-2)

To overcome the limitation of a right-hand-only dataset, a vertical flipping strategy is applied to both images and their corresponding labels. By horizontally shifting the flipped annotations according to equation (3-3), the model can effectively learn from synthetic left-hand data.



.(3-3)

The subscript m represents the mirror value of the coordinate. p = 1 helps with the translation of normalized annotations after mirroring along the vertical axis. Figure 5 shows rendered images of different types of hands, along with bounding boxes and keypoints.

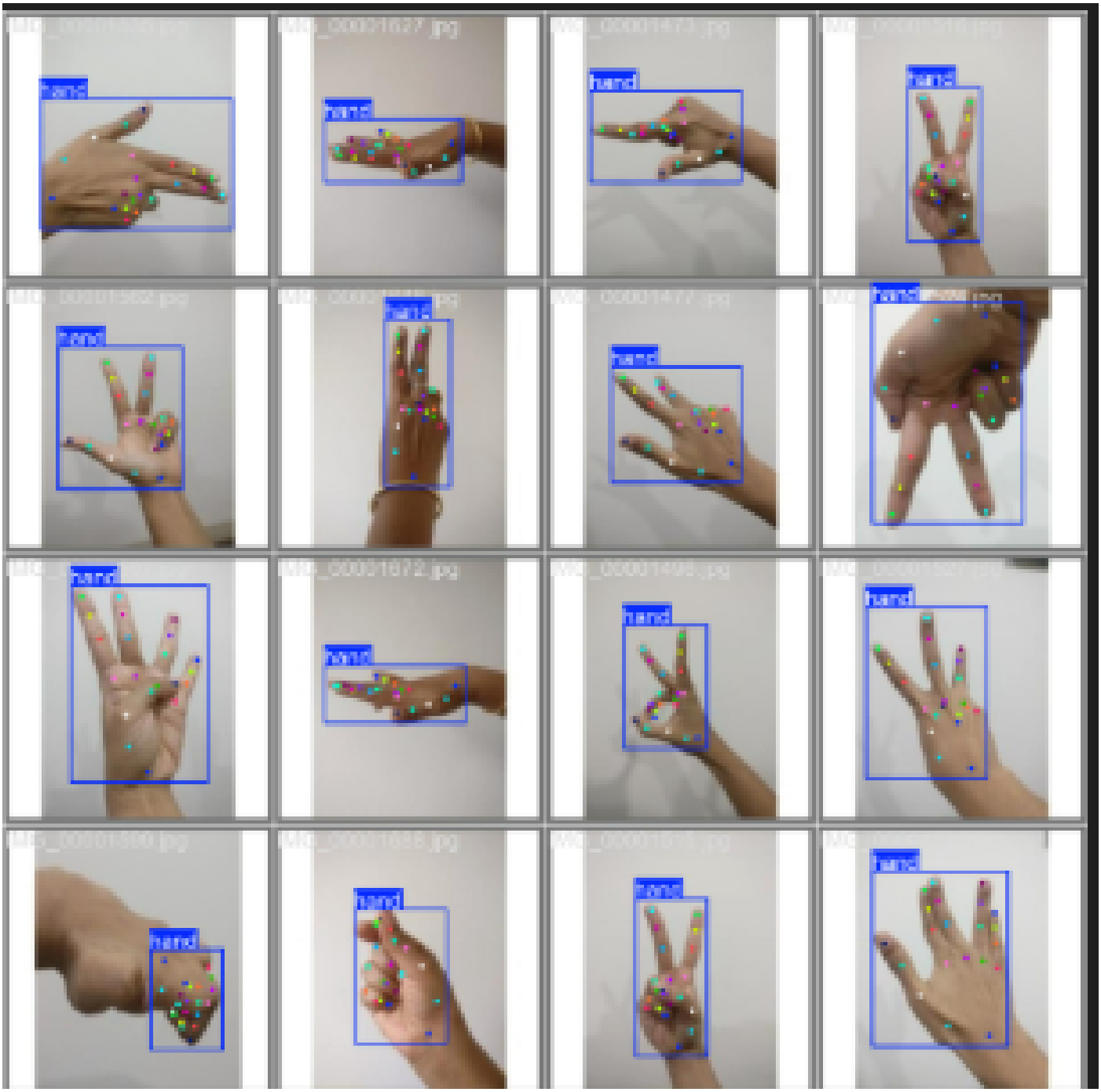


***Figure** 5?Rendered images of key points and bounding boxes*

### 3.5 Training optimization strategies and experimental environment configuration

We adopt an end-to-end learning approach based on the YOLOv11 pose architecture and use the software development kit (SDK) provided by Ultralytics. For the overall network architecture of YOLOv11, please refer to the detailed discussion by Terven et al. [13]. The training weights of the pose estimation part are pre-trained on the Hand Keypoints dataset.

We used the default training parameters provided in the Ultralytics SDK[14] to train the yolov8n-pose, yolov8s-pose, yolo11n-pose, yolo11s-pose and yolo12n-pose models for 400 rounds respectively. During the training process, the batch size of the YOLO pose model was set to 32. As for the loss function, we used the default settings: the bounding box regression used CIoU loss[32] and DFL loss[15], the classification task used binary cross-entropy (BCE) loss, and the key point detection used the loss function based on object key point similarity (OKS)[16].

The training environment was conducted on a computer running Windows 11, equipped with an AMD Ryzen 7 9800x3D processor and a 12GB NVIDIA GeForce RTX5070 GPU.

## 4. Experimental Design and Dataset Construction

### 4.1 Experimental Design and Dataset Construction

In this study, the resolution of all images in the dataset was first adjusted to 640x640 to achieve image standardization. Then, Gaussian filtering was used to remove noise and improve image quality. Keypoint annotation was completed by a semi-automatic annotation tool in conjunction with manual verification to ensure that the annotation accuracy reached more than 95%. In order to

improve the generalization ability of the model, data augmentation strategies such as random rotation (+-30), scaling (0.8-1.2), and color dithering (+-15%) were implemented to effectively expand the diversity of training samples. Finally, the dataset was divided into training set and validation set in a 3:1 ratio, and a complete evaluation system was established, which provided a solid foundation for subsequent model training and evaluation.

### 4.2 Comparative experimental design

This study designed a systematic comparative experimental framework to comprehensively evaluate the performance of hand pose detection algorithms based on YOLO-Pose. The experiments mainly evaluated the algorithm from four dimensions: model accuracy, detection speed, generalization ability, and robustness. First, the proposed algorithm was compared with current mainstream hand pose detection methods, such as the traditional YOLO series, in terms of performance. Then, comparative tests were designed under different scene conditions, such as complex backgrounds, low-light environments, partial occlusion, and multi-hand interaction, to verify the algorithm's adaptability in complex environments. Finally, deployment tests were conducted on different hardware platforms, including high-performance GPU servers, edge computing devices, and mobile terminals, to evaluate the algorithm's performance in real-world application scenarios. The use of these four indicators ensured the comprehensiveness and objectivity of the evaluation. All comparison algorithms were run on the same dataset and hardware environment, so the experimental results are fair and reproducible.

## 5. Experimental Results and Analysis

### 5.1 Comparative experiment

We conducted a comparative study, focusing on comparing the performance of various models in terms of precision, recall, mAP50, and mAP50 95 (mean precision). The design of these evaluation metrics is consistent with the evaluation criteria of the MS COCO dataset .

**Table** 1 presents the evaluation results for the bounding box detection and pose estimation tasks. Through comparative analysis of YOLO pose models of different sizes, we found that under the same input dataset conditions, there were no significant differences in the evaluation metrics among the models. Overall, yolo-BMT exhibits the best performance.

| Model | layers | Params/M | Precision | Recall | mAP50(P) | mAP50-95(P) |
|---|---|---|---|---|---|---|
| Yolov8n-pose | 259 | 3.2 | 0.832 | 0.821 | 0.812 | 0.521 |
| Yolov8s-pose | 287 | 11.5 | 0.843 | 0.816 | 0.821 | 0.537 |
| Yolo11n-pose | 266 | 3.0 | 0.841 | 0.815 | 0.818 | 0.557 |
| Yolo11s-pose | 292 | 10.1 | 0.857 | 0.828 | 0.825 | 0.585 |
| SSD | 132 | 27 | 0.741 | 0.695 | 0.713 | 0.435 |
| Faster-R-CNN | 189 | 53 | 0.852 | 0.816 | 0.824 | 0.567 |
| Yolo11n+A+B | 239 | 2.6 | 0.865 | 0.822 | 0.843 | 0.695 |

Table 1?Performance comparison of different detection models on the Hand Keypoints dataset

5.2 Ablation experiment

To verify the effectiveness of each key module in the proposed YOLO-BMT, we conducted ablation experiments on the Multi-Scale Expanded Attention (MSDA) module and the Bidirectional Feature Pyramid (BiFPN) module. The experimental results are shown in Table 2.

As shown in Table 2, based on the baseline model, the introduction of the BiFPN module alone improved the model's mAP50 from 81.8% to 83.9%, and mAP50-95 from 55.7% to 61.8%, while reducing the number of parameters by approximately 0.1M. This indicates that BiFPN significantly enhances the model's ability to detect targets at different scales through efficient bidirectional multi-scale feature fusion.

When the Involution module is introduced alone, the model achieves 82.9% and 67.1% accuracy on mAP50 and mAP50-95, respectively, which is better than the setting using BiFPN alone, while reducing the number of parameters by approximately 0.4M. The spatial specificity of Involution enhances the keypoint localization capability, thereby improving detection accuracy.

Integrating Involution and BiFPN into YOLOv11n yielded the best performance: mAP50 reached 84.3%, mAP50-95 reached 69.5%, and the total number of parameters was reduced by approximately 0.4M. This result demonstrates that deeply combining the spatial adaptation mechanism of Involution with the intelligent multi-scale fusion mechanism of BiFPN provides a powerful and efficient upgrade solution for the YOLO-pose real-time pose estimation model, pushing new boundaries on the Pareto frontier of accuracy and speed.

**Table** 2?Ablation Experiment Result Statistics Table

| Model | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Yolo11n-pose | V | V | V | V |
| Involution | | V | | V |
| BiFPN | | | V | V |
| Params/M | 3.0 | 2.6 | 2.9 | 2.6 |
| Precision | 0.841 | 0.847 | 0.873 | 0.865 |

| Model | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Recall | 0.815 | 0.801 | 0.799 | 0.822 |
| mAP50(P) | 0.818 | 0.829 | 0.839 | 0.843 |
| mAP50-95(P) | 0.557 | 0.671 | 0.618 | 0.695 |

## 6. Conclusion

This paper proposes an innovative hand detection and pose estimation model, YOLO-BMT. By introducing an involution module and a bidirectional feature pyramid (BiFPN) structure, this model significantly improves the overall performance of hand target detection and keypoint localization. Regarding module effectiveness verification, ablation experiments demonstrate that the BiFPN module effectively enhances the model's ability to perceive and integrate hand features at different scales by optimizing the bidirectional fusion mechanism of multi-scale features; Involution strengthens the model's collaborative modeling of local details and global contextual information; and the dual-module collaborative mechanism further exhibits a significant complementary enhancement effect, achieving a significant improvement in detection accuracy under conditions of moderate parameter increases.

In summary, YOLO-BMT, by integrating the aforementioned core modules, effectively addresses the challenges of multi-scale adaptation and feature representation in hand detection tasks while maintaining the efficient inference capabilities of the YOLO series models. This provides a feasible solution that balances accuracy and efficiency for real-time hand pose estimation applications. This study not only validates the effectiveness of the proposed method but also provides a valuable reference for subsequent research on hand interaction understanding in complex scenarios.

## References


1. Perimal, M.; Basah, S.N.; Safar, M.J.A.; Yazid, H. Hand-Gesture Recognition-Algorithm based on Finger .Counting. J. Telecommun. Electron. Comput. Eng. 2018, 10, 19- 24. P. Gil, C. Mateo, and F. Torres, "3d visual sensing of the human hand for the remote operation of a robotic hand," International journal of advanced robotic systems, vol. 11, no. 2, pp. 26-, 2014.
2. F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "Mediapipe hands: On-device real-time hand tracking," 6 2020.
3. T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," arXiv.org, 2017.
4. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Communications of the ACM, vol. 60, pp. 84-90, 5 2017.
5. D. Maji, S. Nagori, M. Mathew, and D. Poddar, "Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss," IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, vol. 2022-June, pp. 2636-2645, 4 2022. J.
6. G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO," Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics
7. T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan,P. Dolla´r, and C. L. Zitnick, "Microsoft coco: Common objects in context," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 8693 LNCS, pp. 740-755, 2014.
8. Jiao J, Tang Y M, Lin K Y, et al. Dilateformer: Multi-scale dilated transformer for visual recognition[J]. IEEE transactions on multimedia, 2023, 25: 8906-8919.
9. Li, Duo, et al. "Involution: Inverting the inherence of convolution for visual recognition." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.
10. C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. J. Argus, andT. Brox, "Freihand: A dataset for markerless capture of hand pose andshape from single rgb images," Proceedings of the IEEE InternationalConference on Computer Vision, vol. 2019-October, pp. 813-822, 102019.
11. G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO," Jan. 2023.[Online]. Available: https://github.com/ultralytics/ultralytics
12. X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang,"Generalized focal loss: learning qualified and distributed boundingboxes for dense object detection," in Proceedings of the 34th International Conference on Neural Information Processing Systems, ser. NIPS'20. Red Hook, NY, USA: Curran Associates Inc., 2020.
13. D. Maji, S. Nagori, M. Mathew, and D. Poddar, "Yolo-pose: Enhancingyolo for multi person pose estimation using object keypoint similarityloss," IEEE Computer Society Conference on Computer Vision andPattern Recognition Workshops, vol. 2022-June, pp. 2636-2645, 4 2022.
14. K. S. Yadav, A. M. Kirupakaran, and R. H. Laskar, "Gcr-net: A deep learning-based bare hand detection and gesticulated character recognition system for human-computer interaction," Concurrency and Computation: Practice and Experience, vol.

35, pp. 1-1, 3 2023

15. J. Terven, D. M. Co´rdova-Esparza, and J. A. Romero-Gonza´lez, "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas," Machine Learning and Knowledge Extraction 2023, Vol. 5, Pages 1680-1716, vol. 5, pp. 1680-1716, 11 2023
16. Terven, D. M. Co´rdova-Esparza, and J. A. Romero-Gonza´lez, "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas," Machine Learning and Knowledge Extraction 2023, Vol. 5, Pages 1680-1716, vol. 5, pp. 1680-1716, 11 2023