

Article

Research on Human Posture Estimation Algorithm Based on YOLO-Pose

Jing Ding ¹, Shanwei Niu ², Zhigang Nie ^{2,3,*} and Wenyu Zhu ⁴

¹ Department of Physical Education, Gansu Agricultural University, Lanzhou 730070, China; dingj@gsau.edu.cn

² College of Information Science and Technology, Gansu Agricultural University, Lanzhou 730070, China; niusw@st.gsau.edu.cn

³ Key Laboratory of Opto-Technology and Intelligent Control, Ministry of Education, Lanzhou Jiaotong University, Lanzhou 730070, China

⁴ Intelligent Sensing and Control Laboratory, Shandong University of Petrochemical Technology, Dongying 257000, China; 2015017@sdipct.edu.cn

* Correspondence: niezg@gsau.edu.cn

Abstract: In response to the numerous challenges faced by traditional human pose recognition methods in practical applications, such as dense targets, severe edge occlusion, limited application scenarios, complex backgrounds, and poor recognition accuracy when targets are occluded, this paper proposes a YOLO-Pose algorithm for human pose estimation. The specific improvements are divided into four parts. Firstly, in the Backbone section of the YOLO-Pose model, lightweight GhostNet modules are introduced to reduce the model's parameter count and computational requirements, making it suitable for deployment on unmanned aerial vehicles (UAVs). Secondly, the ACmix attention mechanism is integrated into the Neck section to improve detection speed during object judgment and localization. Furthermore, in the Head section, key points are optimized using coordinate attention mechanisms, significantly enhancing key point localization accuracy. Lastly, the paper improves the loss function and confidence function to enhance the model's robustness. Experimental results demonstrate that the improved model achieves a 95.58% improvement in mAP50 and a 69.54% improvement in mAP50-95 compared to the original model, with a reduction of 14.6 M parameters. The model achieves a detection speed of 19.9 ms per image, optimized by 30% and 39.5% compared to the original model. Comparisons with other algorithms such as Faster R-CNN, SSD, YOLOv4, and YOLOv7 demonstrate varying degrees of performance improvement.

Keywords: deep learning; target detection; YOLO-Pose; human posture; drones



Citation: Ding, J.; Niu, S.; Nie, Z.; Zhu, W. Research on Human Posture Estimation Algorithm Based on YOLO-Pose. *Sensors* **2024**, *24*, 3036. <https://doi.org/10.3390/s24103036>

Academic Editors: Eui Chul Lee, Irene Amerini, Paolo Russo and Fabiana Di Ciaccio

Received: 3 March 2024

Revised: 7 April 2024

Accepted: 9 May 2024

Published: 10 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Background

The study of human posture is of great significance for understanding movement mechanisms, improving motor skills, and optimizing training programs. China has always been committed to improving the health of its people, and has emphasized adherence to the principle of prevention as the mainstay, optimization of the health service system, and enhancement of the non-medical health service model. The healthy growth of the body has a profound impact on the overall health of college students and is an important manifestation of the country's comprehensive strength, as well as a key resource for the country's sustainable development. Health education in colleges and universities, as an important component for the implementation of such a program, has the responsibility to promote correct health concepts and methods.

The main cause of poor body posture is an abnormal change in the alignment of the bones, which affects range of motion and places abnormal stress on muscles, joints, ligaments, and other tissues. In such unbalanced postures, the rest of the body "compensates" for the imbalance, but the balance is fragile and gradually leads to discomfort and

pain. With increased awareness of the importance of managing body shape and health, the “pre-intervention” model of correction and rehabilitation is becoming more widely recognized, and effective methods such as exercise are being used to address the problem proactively, thereby reducing the risk of disease associated with poor posture.

Human posture estimation is an important computer vision task aimed at detecting human posture from images or videos, including the body’s joint positions and relative angles. This task is of great significance. Traditional 2D human pose estimation methods are mainly classified as bottom-up and top-down. Top-down methods first detect all human positions in the image and then estimate the pose information and size for each detected person independently. Bottom-up methods, on the other hand, first utilize a convolutional neural network (CNN) or its variants to extract features from the input image and then use a dense key point detector to detect all relevant key points and combine the human pose information based on the relationship between them. Human body posture estimation must judge and locate the target and interpret the target’s rotation angle in three-dimensional space. In traditional two-dimensional image recognition in the process of practical application, there are low light intensity, dense targets, serious edge occlusion, single application scenes, complex background of the target object, the target is occluded, recognition accuracy is poor, etc. All of these problems in human body posture estimation algorithms’ recognition speed and accuracy are challenging.

2. Introduction

2.1. Research Work by Relevant Scholars

With the advancement of deep learning and the increase in computational power, significant progress has been made in human pose estimation network models. Zheng et al. [1] (2021) proposed a purely Transformer-based method called PoseFormer for 3D human pose estimation in videos. This method comprehensively models the intra-frame human joint relationships and inter-frame temporal correlations to output accurate 3D human poses for the central frame. Liu et al. [2] (2021) introduced a novel multi-frame human pose estimation framework that leverages rich temporal cues between video frames to enhance key point detection. This method encodes the spatiotemporal context of key points through pose–time merging to generate an effective search range and computes bidirectional weighted pose residuals through a pose residual fusion module, effectively improving pose estimation. Li et al. [3] (2021) proposed an efficient and effective regression-based approach utilizing maximum likelihood estimation (MLE) for developing human pose estimation, modeling the output distribution using likelihood heatmaps. Zhang et al. [4] (2021) proposed a method that utilizes the YOLOv3 model to create a human pose estimation network, combining the squeeze-and-excitation network structure in High-resolution network (HRNet) residual architecture and improving the HRNet algorithm’s output of human key points. They designed a pose classification algorithm based on support vector machines (SVMs) to classify human poses in a classroom setting.

Li et al. [5] (2022) proposed a strided Transformer architecture to efficiently convert a long sequence of 2D joint positions into a single 3D pose. This method combines single-frame supervision and applies additional temporal smoothness constraints to generate smoother and more accurate 3D poses. Liu et al. [6] (2022) introduced an anisotropic Gaussian coordinate encoding method to describe the skeletal orientation cues between adjacent key points. This is the first time skeletal orientation cues have been incorporated into heat map encoding for human pose estimation (HPE) tasks. They also introduced multiple loss functions to constrain the output and prevent overfitting. They use Kullback–Leibler divergence to measure the difference between predicted labels and ground truth labels. This method demonstrates significant advantages over existing state-of-the-art models for human pose estimation, but it suffers from algorithmic complexity and poor robustness, making it difficult to apply in real-life scenarios. Yuan et al. [7] (2022) proposed a video-based fall detection and orientation estimation method based on human pose estimation. They predict the coordinates of key points for each person using a pose estimation network and then use an SVM classifier

to detect falls. This approach can effectively be applied to fall detection and orientation estimation in videos. Lee et al. [8] (2022) proposed an OpenPose network and applied the DeepSort algorithm for multi-person tracking. This algorithm can identify the poses of each individual based on the single-frame joints obtained from OpenPose. However, the algorithm exhibits poor robustness and struggles to cope with the challenges of complex human pose estimation in current scenarios.

Su et al. [9] (2022) proposed a motion pose estimation algorithm based on OpenPose and trained it using the COCO dataset. Through comparison with standard poses, the study demonstrated the algorithm's ability to accurately recognize various badminton action poses, with a recognition rate of up to 94%. Amadi et al. [10] (2023) introduced a novel and fully differentiable pose consistency loss method. This method is unaffected by camera direction and has shown improvements in single-view human pose estimators trained using limited labeled 3D pose data. Manesco et al. [11] (2023) proposed a novel approach called the domain unified method, aiming to address pose misalignment in cross-dataset scenarios through a combination of three modules on top of a pose estimator, including a pose transformer, uncertainty estimator, and domain classifier. Li et al. [12] (2023) presented a hybrid model that combines convolution and transformation models to address the inconsistency between the performances of key point localization with higher accuracy and overall performance.

2.2. Contribution of This Article

Building on the academic research of the aforementioned scholars and the practical engineering applications, this paper proposes a novel approach to address the challenges of severe occlusion at the edges, complex background, and low recognition accuracy caused by target occlusion in human pose estimation. The aim is to improve the detection speed, accuracy, and robustness of the model.

Regarding the YOLOv5 model, which is currently a milestone in the industrial world, its lightweight and precision improvement give full play to its simplicity, ease of use, and high efficiency [13] and lay the foundation for its wide application in actual production, life, and industrial engineering. The summary diagram of this work is shown in Figure 1.

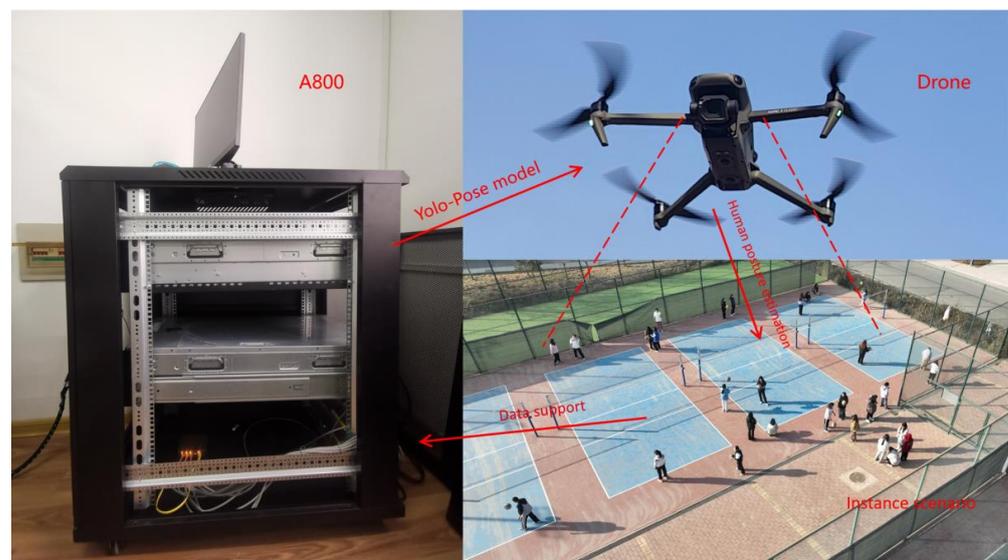


Figure 1. A summary of the research work in this paper.

The main contributions of the research presented in this paper are as follows:

- (1) The Backbone section introduces a lightweight GhostNet module to complete the generation of redundant features with a more economical linear transformation, thus greatly reducing the computational cost of convolution to lower the number of

- parameters of the model and reduce the arithmetic demand, making it more suitable for deployment on UAVs.
- (2) The Neck part introduces the ACmix attention mechanism, which captures local features by convolution in the task of judgement and localization of the target by the model, so that it focuses on judging the human body's bounding box convolution of local features to improve the detection speed.
 - (3) The key points in the Head part and the decoupling information of key points are optimized through the coordinate attention mechanism in order to solve the problems of complex target background and poor target occlusion detection accuracy and to improve the positioning accuracy of key points.
 - (4) The loss function and confidence function are improved to guarantee the robustness of the projection of the bounding box (BBox) for human pose estimation in complex scenes in order to improve the robustness of the model and prevent the occurrence of lagging, frame dropping, and video blurring problems [14].

3. Experimental Data

The resources for this experimental study consist of three components: the study object, the data collection and dataset, and the A800 deep learning GPU arithmetic server.

3.1. Research Object

A total of 227 undergraduate female students from Gansu Agricultural University, who were enrolled in the academic year of 2022, were randomly selected as the subjects of this study. Initially, a questionnaire survey was conducted to collect basic information regarding their height, weight, body fat percentage, daily routines, medical history, pain history, allergy history, lifestyle habits, and self-perception of body posture. It is worth noting that students diagnosed with definite spinal disorders, thoracic deformities, developmental abnormalities, as well as those exempt from physical education classes due to medical reasons, were not included in the scope of this research.

3.2. Data Collection and Dataset

The sample dataset collected in this study is derived from the daily learning and living scenarios of students at Gansu Agricultural University in Lanzhou, China. To ensure the diversity of images in the dataset, the experiment collected various complex scenarios including classroom learning, campus strolls, physical exercise, and laboratory activities. The methods of image collection in the dataset include the use of mobile devices and drones. The mobile devices utilize the Sony IMX866 large sensor camera, which significantly enhances color performance and image quality, with a resolution of 1279×1706 pixels, and a total of 600 images were collected. The unmanned aerial vehicle employs the Mavic 3 Cine model, equipped with a one-inch sensor and 20 million pixels, capable of capturing high-dynamic-range images, and possessing outstanding stability and endurance performance. The image resolution is 5280×2970 pixels, and 600 images were also collected. Both datasets, totaling 1200 images, were divided into training, testing, and validation sets in an 8:1:1 ratio.

In this study, we annotated human body pose key points using the Labelme software (version 1.8.6) and employed the COCO format for data annotation. The annotation was specifically performed on instances of the class "Person". A total of 17 key points on the human body were annotated, including the nose, left eye, right eye, left ear, right ear, left shoulder, right shoulder, left elbow, right elbow, left wrist, right wrist, left hip, right hip, left knee, right knee, left ankle, and right ankle [15]. Each key point (C1, C2, C3) is stored in a JSON data format. Specifically, C1 and C2 represent 2D plane coordinate data of the human body pose estimation key points, while C3 serves as a decision identifier, indicating the presence or absence of the key point in the image. The annotated key points are illustrated in Figure 2.

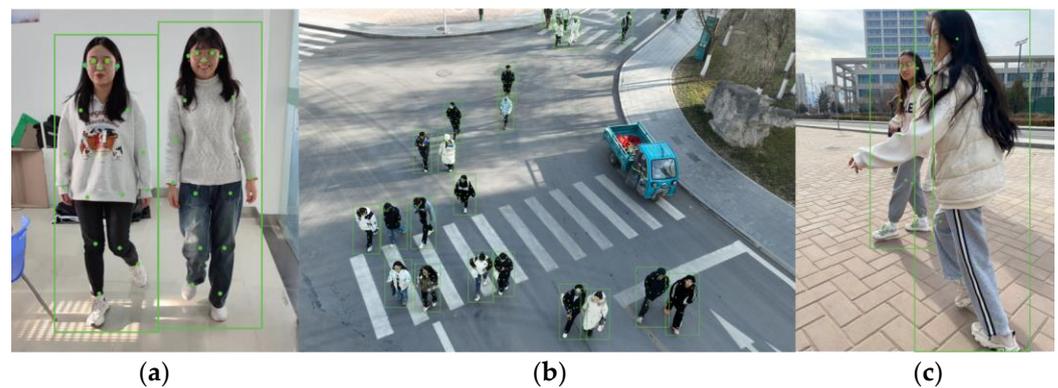


Figure 2. The schematic diagram of human pose key point annotation. (a) Laboratory; (b) Drone; (c) Physical Education Class.

3.3. Details of the A800 Deep Learning GPU Computing Power Server

The computational resource employed in this study was the A800 deep learning GPU server at the Intelligent Sensing and Control Laboratory of Shandong University of Petroleum and Chemical Technology in China. The server utilized for this purpose is the Wave Computing's AI Server NF548M6. Its hardware configuration includes an Intel® (Santa Clara, CA, USA) Xeon(R) Silver 4314 CPU @ 2.4 GHz \times 64 as the CPU processor, equipped with 8 NVIDIA A100 GPUs. Graphical rendering utilized llvmpipe (LLVM 7.0, 256 bits), while the operating system was CentOS Linux 7 (3.28.2), with 128 GB of memory capacity and 2.048 TB of disk storage space. Python version 3.8 was employed, setting the learning rate for the neural network to 0.01, and using a batch size of 16 for image training. All computational experiments were conducted on this powerful computing platform.

4. YOLO-Pose Human Posture Estimation Algorithm

The YOLO-Pose network structure, as illustrated in Figure 3, is built upon the foundation of the YOLOv5 network structure. The YOLOv5 network structure comprises three main components: Backbone, Neck, and Head [16], with the detailed flowchart depicted in Figure 3. The significance of YOLOv5 lies in its concise, user-friendly, and efficient characteristics, which have swiftly established its position in the industrial sector and enabled its widespread application in practical production fields. The scenario of this study involves conducting human pose detection tasks on unmanned aerial vehicles (UAVs), hence the utilization of the lightweight and low-computational-resource YOLOv5s.

In the Backbone, 4 C3_1s and 5 CBSs are utilized, with SPPF incorporated into the Backbone. The stride of all five CBSs is 2, resulting in a halving of both the height and width of the image after passing through the CBS. As shown in Figure 4, a CBS consists of a 2D convolutional layer, a BN layer, and a SiLU activation function. The distinction between BottleNeck2 and BottleNeck1 lies in the removal of the connection from input to output. Additionally, the difference between C3_1_X and C3_2_X lies in the use of BottleNeck1 in C3_1_X and BottleNeck2 in C3_2_X. C3_1 and C3_2 are collectively referred to as the C3 module, each employing three CBSs. Compared to CSPX, C3_X entails a smaller computational load. In YOLOv5, the authors transformed the BottleneckCSP module into the C3 module, which serves as the primary module for residual feature learning. It is composed of two structures: one acting on the Backbone main network and the other on the Neck module branch.

The spatial pyramid pooling (SPP) consists of SPP and SPPF, both of which are fundamentally similar modules that perform multi-scale transformations and fusion on feature maps. However, they differ slightly in structure. In SPPF, the input first passes through a CBS, followed by three layers of MaxPool, and eventually four output channels are merged to produce the final output through another CBS. It should be noted that, as opposed to SPP's three pooling operations using window sizes of 5×5 , 9×9 , and 13×13 ,

SPPF's input for each pooling layer is derived from the previous layer's output, with all three layers utilizing a 5×5 pooling window. Experimental results demonstrate that the computational load of the SPPF model is significantly smaller, leading to a substantial speed improvement.

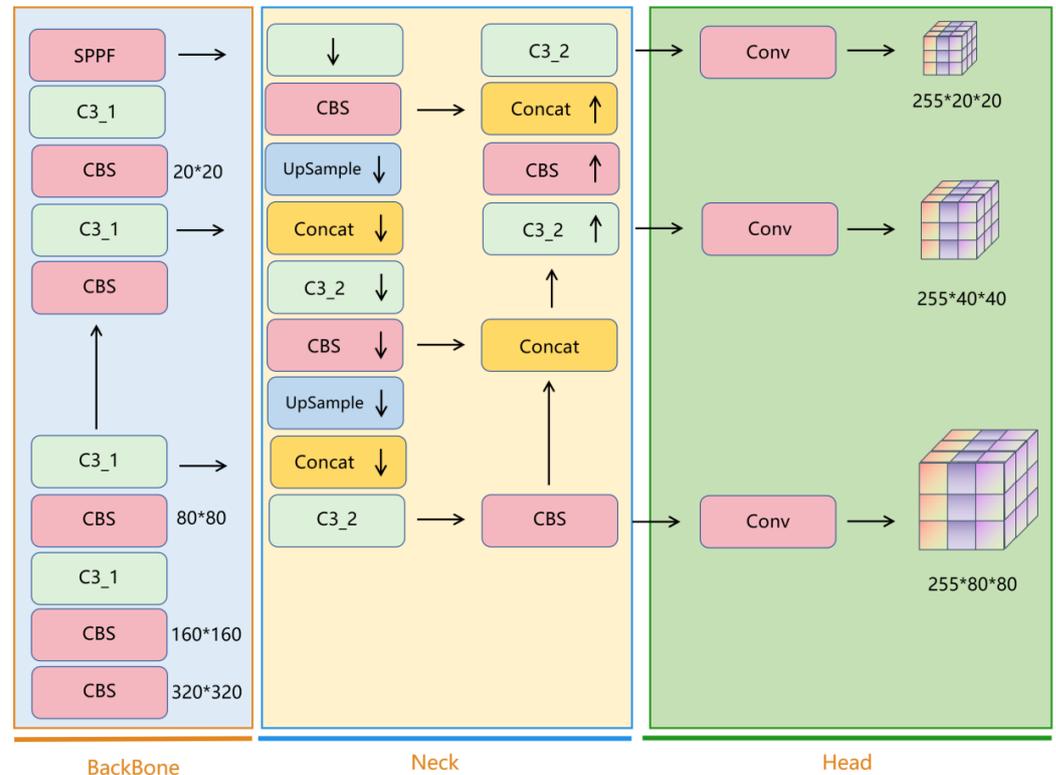


Figure 3. The YOLOv5 Network Model. Notes: * represents convolution.

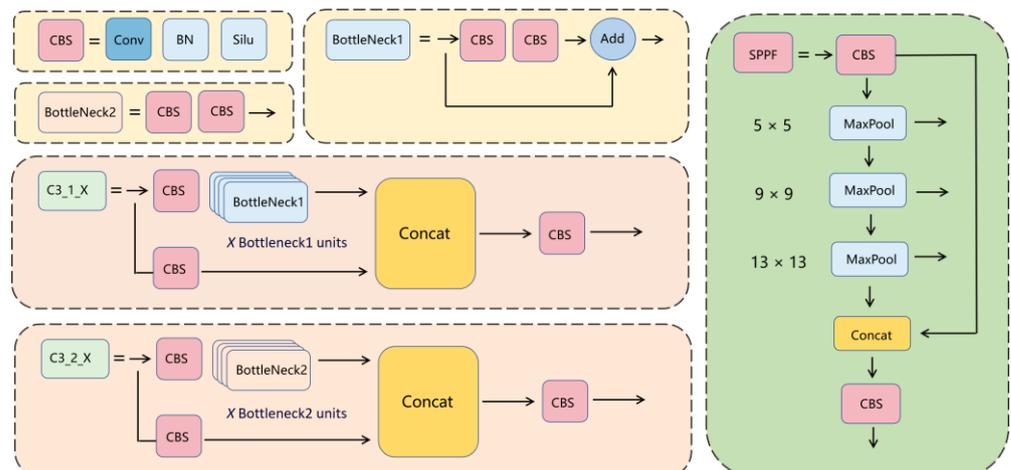


Figure 4. The Composition Structure of the Modules in the YOLOv5 Network Model.

The Neck section consists of 4 C3_1s, 4 CBSs, 4 Concats, and 2 UpSample modules. In the fusion pathways from top to bottom on the left side, there are two CBSs and two C3_2 modules; while in the fusion pathways from bottom to top on the right side, there are also two CBSs and one C3_2 module. This bidirectional feature fusion pathway is referred to as PANet. As shown in Figure 5, PANet merges high-dimensional features into low-dimensional features from top to bottom on the left side. Compared to the feature pyramid network (FPN), the bidirectional fusion structure of PANet is more conducive to

comprehensive feature integration. The Concat module is used for dimension concatenation, integrating feature maps with high dimensions lacking semantic information and low dimensions lacking detailed information. The UpSample module is used for upsampling, which effectively detects small objects and details. In the Head section, only one Conv module is retained, primarily to adapt the number of channels, uniformly transform dimensional information, and parse the channel information of the feature map into corresponding detection boxes and categories.

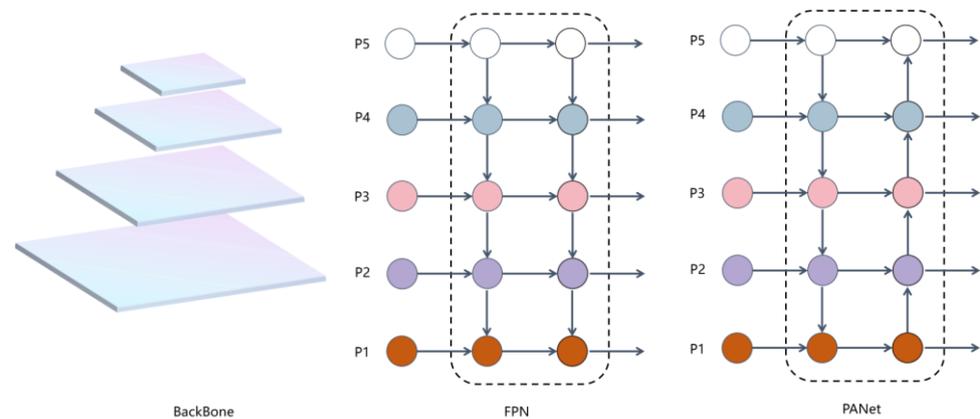


Figure 5. Schematic Diagram of Feature Fusion between FPN and PANet.

5. Improvement of YOLO-Pose Human Pose Estimation Algorithm

5.1. Backbone Section Introduces Lightweight GhostNet Module

In practical application scenarios, standard convolution modules may generate a large number of approximate features, resulting in significant computational resource consumption. This becomes particularly problematic when deploying the model on unmanned aerial vehicles (UAVs) for human pose estimation tasks, as mobile devices on UAVs often have limited computational power, which can lead to stuttering and unsmooth model performance. To address this issue, we introduce the GhostNet module into the YOLO-Pose model. By utilizing more cost-effective linear transformations to generate redundant features, we greatly reduce the computational cost of convolution. Initially, we employ standard convolutions to generate m layers of original features, as illustrated in Figure 6a and computed using Equation (1).

$$\gamma' = X * f + b \quad (1)$$

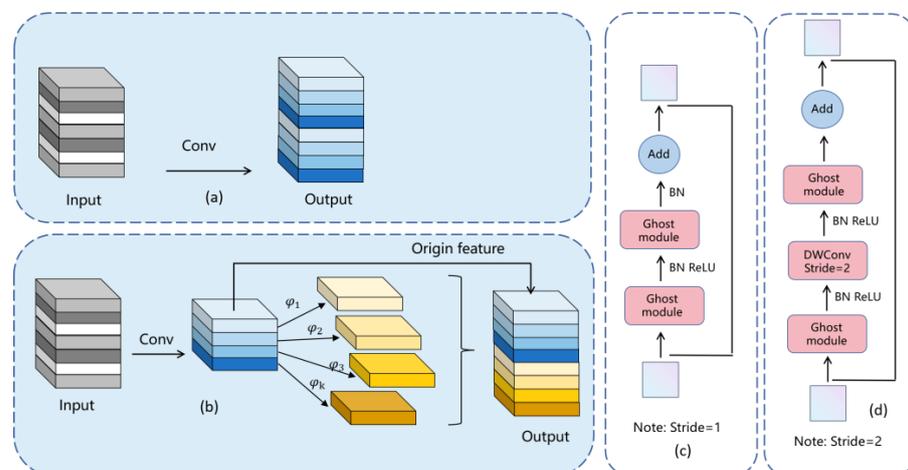


Figure 6. Lightweight Ghost Module Diagram. (a) Standard convolution. (b) Ghost Convolution. (c) Ghost module structure. (d) New Ghost module structure.

In the formula, $\gamma' \in R^{h' \times \omega' \times m}$ represents the output feature map, b represents the bias term [17], $*$ signifies the convolution operation, and subsequently, γ' undergoes an inexpensive mapping. As shown in Formula (2), $y'_i \in Y'$ and $\varphi_{i,j}$ denote the j -th linear transformation of the source feature i . The schematic diagram is depicted in Figure 6b, where it is evident that $\varphi_{i,j}$ generates multiple corresponding Ghost features y_{ij} .

$$y_{ij} = \varphi_{i,j}(y'_i), \forall i, \dots, m, j = 1, \dots, s \quad (2)$$

The standard convolution floating-point operation is denoted as $n \times h' \times \omega' \times c \times k \times k$, wherein c represents the number of input channels. In contrast, the Ghost convolution combines $m(s-1) = n/s + (s-1)$ linear computations [18] with the standard convolution. The linear transformation convolves the kernel of size $d \times d$. Hence, the computational ratio between the two can be expressed as Formula (3).

$$R = \frac{nh\omega'cck}{(n/s)h\omega'cck + (s-1)(n/s)h\omega'dd} = \frac{cck}{s^{-1}cck + ((s-1)/s)dd} = \frac{sc}{s+c-1} \approx s \quad (3)$$

The Ghost convolution, compared to the standard convolution, increases the theoretical number of operations by a factor of c given $d \times d = k \times k$ and $s \ll c$. Leveraging the performance advantages of the Ghost module, two Ghost modules are combined to construct a new Ghost module structure, as illustrated in Figure 6c. The Backbone is formed by concatenating two Ghost modules in series. The role of the first module is to increase the feature dimension and expand the number of channels. The second Ghost module reduces the number of channels to match the number of input channels [19] and connects with the input through a shortcut to obtain the final output. Thus, the input and output dimensions of the new Ghost structure are the same, facilitating its integration into neural networks. When the stride is 2, a DWConv convolution layer with a stride of 2 is added between the two Ghost modules of the Backbone, which reduces the output feature map size to half of the input feature map size [20]. Two types of stride handling innovations provide greater flexibility for models to adapt to tasks of varying sizes and complexities. This study reconstructs the entire fusion network using the novel lightweight Ghost module to reduce model parameters and decrease computational requirements, making it more suitable for deployment on mobile devices, significantly enhancing the model's usability and portability.

5.2. Neck Partially Introduces the ACmix Attention Mechanism

After introducing the lightweight Ghost module, the YOLO-Pose algorithm for human pose estimation has targeted detection and localization tasks. To further enhance performance, this study has also introduced the ACmix attention mechanism, which allows the network model to focus on feature information that is crucial for model performance, while ignoring irrelevant information and facilitating effective information exchange and propagation with other modules. The ACmix attention mechanism is a hybrid model that combines the advantages of self-attention and convolutional operations [21]. The core concept of this attention mechanism is to utilize 1×1 convolutions to perform most of the computations for both self-attention and convolutional operations, thereby enabling both global perception capability and the capture of local features through convolution.

According to Figure 7, the processing of the feature maps with a size of $H \times W \times C$ is first performed through three 1×1 convolution projections, resulting in three sets of feature maps with sizes of $3 \times N$. Subsequently, convolution and self-attention operations are separately applied to these feature maps [22]. The convolution operation can be divided into two stages, namely Stage 1 represented by Equation (4) and Stage 2 represented by Equations (5) and (6).

$$g_{ij}^{\sim(p,g)} = K_{p,q} f_{ij} \quad (4)$$

$$g_{ij} = \sum_{p,q} g_{ij}^{(p,q)} \quad (5)$$

$$g_{ij}^{(p,q)} = \text{Shift}\left(g_{ij}^{(p,q)}, p - \lfloor k/2 \rfloor, q - \lfloor k/2 \rfloor\right) \quad (6)$$

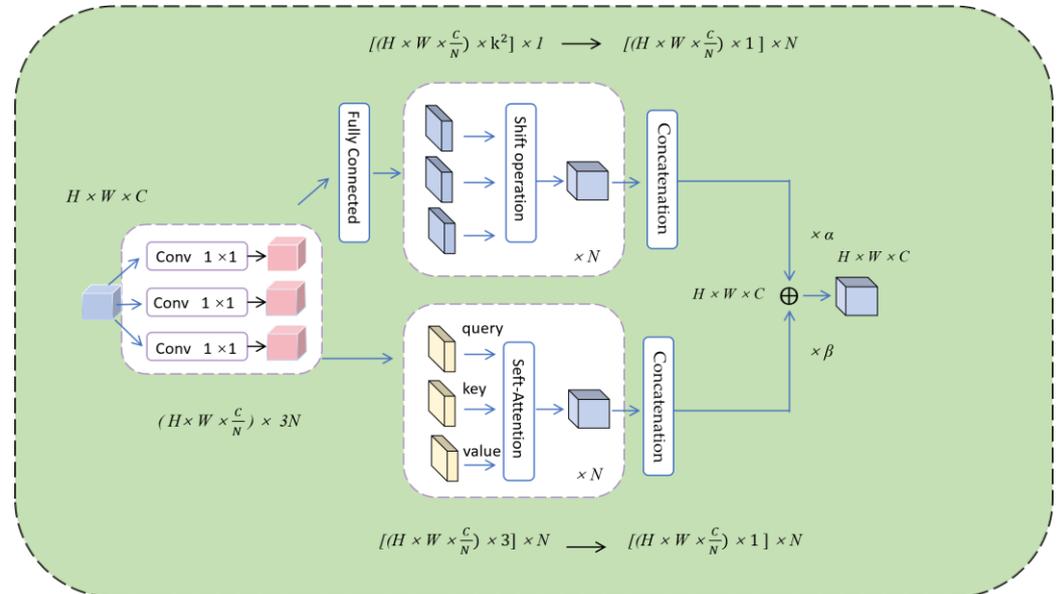


Figure 7. Structural diagram of lightweight ACmix attention mechanism.

In the above equations, f_{ij} represents the feature vector of the input pixel, $\lfloor \cdot \rfloor$ denotes the positional operation, k represents the kernel size, $g_{ij}^{(p,q)}$ represents the feature map before projection, $g_{ij}^{(p,q)}$ represents the feature map after projection, Shift represents the shift transformation, p, q represent the linear projection, $K_{p,q}$ represents the linear projection on each position, and g_{ij} represents the total sum of the feature obtained after the aggregation operation.

The first stage involves projecting the input features onto different coordinate positions (p, q) according to the weight K . In the second stage, the projected mappings undergo horizontal and vertical shift operations separately based on $K_{p,q}$, and finally, all the mapped feature information is aggregated together [23].

Similarly, the self-attention operation can also be divided into 2 stages. The first stage is represented by Equation (7), and the second stage is represented by Equation (8).

$$\begin{cases} q_{ij}^{(l)} = W_k^{(l)} f_{ij} \\ k_{ij}^{(l)} = W_k^{(l)} f_{ij} \\ v_{ij}^{(l)} = W_k^{(l)} f_{ij} \end{cases} \quad (7)$$

$$g_{ij} = \parallel_{I=1}^N \left(\sum_{a,b \in N_k(i,j)} A(q_{ij}^{(l)}, k_{ab}^{(l)}) v_{ab}^{(l)} \right) \quad (8)$$

In the aforementioned equation, $W_q^{(l)}$ represents the input feature map matrix of the query at pixel (i, j) , $W_k^{(l)}$ represents the input feature map matrix of the key at pixel (i, j) , and $W_v^{(l)}$ represents the input feature map matrix of the value at pixel (i, j) [24]. $k_{ij}^{(l)}$ is the feature mapping after key projection, $v_{ij}^{(l)}$ is the feature mapping after value projection, $q_{ij}^{(l)}$ is the feature mapping after query projection, and \parallel is the cascade of N attention head outputs. $N_k(i, j)$ denotes the region centered at pixel (i, j) with spatial width k , and $A(q_{ij}^{(l)}, k_{ab}^{(l)})$

denotes the corresponding weights in region $N_k(i, j)$. The feature mapping after the first stage of projection through three 1×1 convolutions is noted as query, key, and value, and finally the paths of the two operations are merged and summed to output as:

$$F_{out} = \alpha F_{conv} + \beta F_{att} \tag{9}$$

In Formula (9), F_{out} represents the final output of the path [25], F_{conv} represents the output of the convolutional attention branch, and F_{att} represents the output of the self-attention branch, which is used to measure the output weights. In order to achieve a balance between global and local feature information in the convolution and self-attention operations, this paper sets the models of α and β to 1, thereby enhancing the aggregation capability of intermediate mapping information for both modes and making the network more suitable for detecting small target information.

5.3. Optimizing the Head Section Key Points

Key point detection is a task that is highly sensitive to position information, while human pose estimation is often affected by external factors such as lighting, resulting in missed and false detections of key points. In the original YOLOv5 network, the design of the key point decoupling head involves independent two-dimensional convolutional operations, as shown in Figure 8, predicting at three different scales (80×80 , 40×40 , 20×20). Each scale corresponds to three anchors, which in turn predict feature boxes at different scales of 80×80 , 40×40 , and 20×20 . Therefore, the entire network predicts a total of 25,200 feature boxes. This design enables the network to more accurately identify and decode key point information. However, the large number of 25,200 feature boxes greatly wastes computational resources. For tasks involving human pose estimation by drones, computational resources are already scarce, thus non-maximum suppression (NMS) must be applied to filter out low-accuracy detection boxes, retaining only high-accuracy ones.



Figure 8. Schematic of YOLOv5 key point decoupling head. Notes: * represents convolution.

Each feature point in the sample images captured by drones has 8 feature channels, which include parameters representing the detection box, confidence (conf), 2D screen coordinates (C1 and C2) of the key points for human pose estimation, and an identification indicator for the existence of key points (C3). The detection box has 4 parameters, namely the center point (bx, by), width (bw), and height (bh), where the center point of the detection box falls within the grid at the center of the feature map [26]. During the computation process, the center point coordinates of the detection box are first calculated, with $grid_i$ representing the i-th column and $grid_j$ representing the j-th row. YOLOv3, YOLOv4, and

YOLOv5 all employ anchor-based methods to compute the position of the detection box, although the formulas for calculating the center point coordinates (bx, by) as well as the width (bw) and height (bh) may differ slightly.

This study further introduces the coordinate attention mechanism to optimize the decoupling of key point information and improve the accuracy of key point localization. The schematic diagram of the coordinate attention structure is shown in Figure 9, where H , W , and C represent the height, width, and channel number of the feature map [27], respectively.

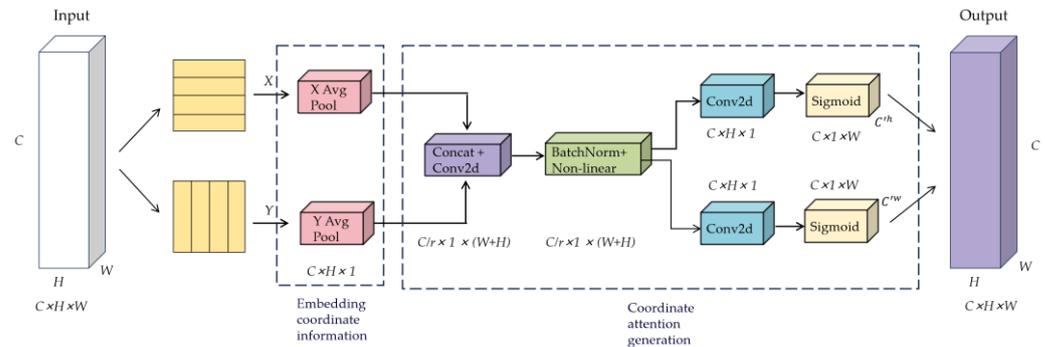


Figure 9. Schematic diagram of the structure of the key point attention mechanism.

The coordinate attention mechanism encodes the horizontal and vertical positional [28] information into channel attention, enabling the network to capture not only inter-channel information but also directional perception and position-sensitive information. Specifically, this mechanism consists of two steps: coordinate position embedding and coordinate attention generation. Firstly, we apply global average pooling with pooling kernels of size $[H, 1]$ and $[1, W]$ to transform the feature map from a matrix of size $[H, W, C]$ into a vector of size $[1, 1, C]$. Following the global average pooling layer, we utilize 1D 1×1 convolutions to acquire inter-channel mutual information, with the size of the convolution kernels adjusted by an adaptive function. This adaptive function allows layers with more channels to engage in more inter-channel interactions. The specific adaptive function is described in Formula (10).

$$k = \left\lfloor \frac{\log_2(c)}{\gamma} + \frac{\beta}{\gamma} \right\rfloor \quad (10)$$

The channel adaptation performs optimally when $\gamma = 2$ and $\beta = 1$. We apply the adaptive function to the 1D 1×1 convolutions to obtain the weights for each channel in the feature pattern. Finally, by multiplying the normalized weights with the initial input feature pattern channels, we obtain the feature output $Z_c^h(h)$ and $Z_c^h(w)$ for the c -th channel at height h and width w .

$$\begin{cases} Z_c^h(h) = \frac{1}{W} \sum_{0 \leq i < W} x_c(h, j) \\ Z_c^h(w) = \frac{1}{H} \sum_{0 \leq j < H} x_c(j, w) \end{cases} \quad (11)$$

In Equation (11), x_c represents the input for channel c . The feature map obtained from Equation (11) is subjected to dimension concatenation and transformed into intermediate feature maps through operations like 1×1 convolution, batch normalization, and non-linear activation functions. This process yields intermediate feature mappings as shown in Equation (12).

$$f = \delta \left(F_1 \left(\left[Z^h, Z^w \right] \right) \right) \quad (12)$$

In the equation, $f \in R^{C/r \times (H+W)}$ represents the intermediate feature containing both horizontal and vertical spatial information. φ represents the non-linear activation function. Z^h and Z^w represent the outputs of the concatenated feature map in terms of height and width, respectively. r denotes the reduction factor, R represents the set of real numbers, C represents the number of channels in the feature map, and F_1 represents the convolution op-

eration with a kernel size of 1. Subsequently, the feature tensor f is split into two independent tensors, $f \in R^{C/r \times (r+H)}$ and $f \in R^{C/r \times (r+W)}$, along the height and width dimensions. Additionally, two 1×1 convolutions are employed to transform $f \in R^{C/r \times (r+H)}$ and $f \in R^{C/r \times (r+W)}$ into F_h and F_w , respectively, ensuring that f^h and f^w have the same number of channels as the input feature tensor X . Afterward, the sigmoid activation function, σ , is separately applied to g^h and g^w to obtain attention weights along the height and width dimensions, as depicted in Equation (13):

$$\begin{cases} g^h = \sigma(F_h(f^h)) \\ g^w = \sigma(F_w(f^w)) \end{cases} \quad (13)$$

Finally, the input feature map X is weighted by the attention weights g^h and g^w through a multiplication operation, resulting in the output of the coordinate attention module, denoted as $Y \in R^{C \times H \times W}$, as shown in the following equation.

$$y_c(i, j) = x_c(i, j) \times g_c^h(i) \times g_c^w(j) \quad (14)$$

In Equation (14), g_c^h and g_c^w represent the attention weights of the feature map along the height and width dimensions, respectively, in the c -th channel. In this study, we have incorporated the coordinate attention mechanism into the conventional 2D convolution key point decoupling head. This mechanism enhances the sensitivity to the position of key points during the feature enhancement and prediction processes. It effectively addresses the challenges of accurate recognition and prediction in scenarios with complex backgrounds and occluded objects. Ultimately, it improves the accuracy of recognition and prediction.

5.4. Introduction of New Loss Function and Confidence Function

The YOLO-Pose network model outputs information including target class probabilities, coordinates of 17 key points, and confidence scores. In this study, the network training is conducted using the following loss function.

$$L = \lambda_{pt} L_{pt} + \lambda_{conf} L_{conf} + \lambda_{id} L_{id} \quad (15)$$

In Equation (15), L represents the loss function, L_{pt} is the coordinate loss, L_{conf} is the confidence loss, and L_{id} is the class loss. λ_{pt} is the weight for the coordinate loss function, λ_{conf} is the weight for the confidence loss function, and λ_{id} is the weight for the class loss function. The loss function, L , is composed of three components: the coordinate loss function, L_{pt} , the confidence loss function, L_{conf} , and the class loss function, L_{id} [29]. The coordinate loss and confidence loss are computed using the mean square error function, while the class loss function is computed using the cross-entropy function.

In the early stages of training the network model, the precision of confidence prediction is low. At the beginning, it is necessary to set λ_{conf} as 0 and gradually increase it for the units containing the target objects as training progresses. When dealing with units that do not contain the target object class, λ_{conf} is set to 0.1. The weight for the coordinate loss function, λ_{pt} , is set as 1, and the weight for the class loss function, λ_{id} , is also set as 1.

When training the network with the aforementioned loss functions, it has been demonstrated through empirical evidence that computing the IoU for calculating the loss function is extremely time-consuming. Therefore, this network proposes Equation (16) as a substitute for approximating the computation of IoU.

$$c(x) = \begin{cases} \frac{e^a \left(1 - \frac{D_T(x)}{d_{th}}\right) - 1}{e^a - 1}, & D(x) < d_{th} \\ 0, & otherwise \end{cases} \quad (16)$$

where d_{th} represents the predicted distance of key points from the ground truth distance, $D(x)$ represents the average error of key points in various bounding boxes (BBox) for human pose estimation, a denotes the hyperparameter scale factor for the current target, and $C(x)$

represents the approximate IoU for the bounding box and predicted box. This approximation calculation greatly reduces the time consumption while mostly not sacrificing accuracy.

Building upon this foundation, the proposed model incorporates the calculation of variance for the error of each key point to ensure the robustness of the bounding box (BBox) projection for human pose estimation in complex scenes. The calculation formula is as follows:

$$c'(x) = \frac{c(x) + \sum_{i=1}^9 (d_i - \bar{d})^2}{0, otherwise}, D(x) < d_{th} \quad (17)$$

In our network model prediction, filtering is performed based on the confidence score and intersection-over-union (IoU) of the objects [30]. Similarly, in the three-dimensional space, it is necessary to analyze the confidence of the target objects. This network model employs the confidence function $f(x)$ based on Euclidean distance to evaluate the deviation distance between the predicted pose of the target object and the ground truth pose. The updated formula is as follows:

$$f(x) = \begin{cases} c(x) + \sum_{i=1}^9 (d_i - \bar{d})^2, & D(x) < d_{th} \\ 0, & otherwise \end{cases} \quad (18)$$

5.5. Improved YOLO-Pose Model

The improved YOLO-Pose model consists of four components: Input, Backbone, Neck, and Prediction [31], as shown in Figure 10. The Input component includes adaptive scaling, mosaic data augmentation, and anchor box calculation, where the adaptive image size is set as the default size of 640×640 . YOLOv5 computes the optimal anchor box values for different training iterations. The mosaic data augmentation utilizes four images and combines them through random scaling, cropping, and arrangement. The purpose of anchor box calculation is to adjust the size and position of the correct targets in object detection. The Backbone component incorporates the GhostNet module, the Neck component introduces the ACmix attention mechanism to optimize the key point prediction in the Head component, and new loss functions and confidence functions are introduced. Through improvements in each module, the new YOLO-Pose model is formed.

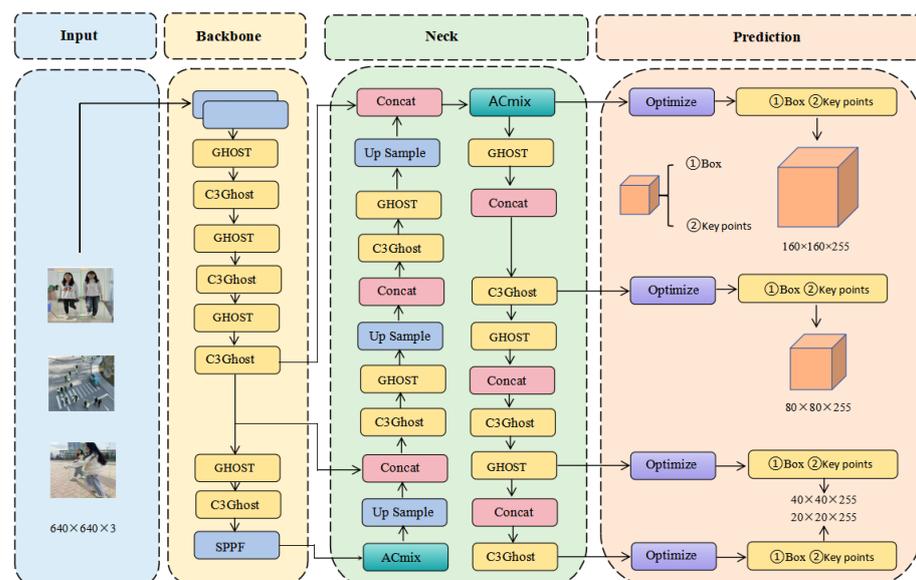


Figure 10. YOLO-Pose Network model diagram.

6. Experimental Methods and Results

6.1. Network Training

This study adopts the stochastic gradient descent (SGD) as the core algorithm for network optimization. To maintain the stability of the deep layers in the model, a warm-up strategy is implemented during the training process. The initial learning rate is set to 0.0001 for predictive training, and it is decayed by a factor of 0.1 after every 150 epochs. The weight decay is set to 0.01. The transfer learning technique is employed, utilizing models trained on the ImageNet-1K and COCO datasets as pre-trained models. The gradient accumulation strategy is employed, with a batch size step of 4 and parameter updates performed every 16 steps. The training is conducted for 500 epochs, and loss and accuracy are sampled every 5 min per epoch.

6.2. Evaluation Indicators

In order to validate the speed, accuracy, and robustness of the improved YOLO-Pose model, the human pose estimation algorithm adopts the average precision based on the object key point similarity L_{oks} , as defined by the official MS COCO evaluation criteria. Specifically, L_{oks} refers to:

$$L_{oks} = \frac{\sum_i \exp\left(-\frac{d_i^2}{2s^2k_i^2}\right) \delta(v_i > 0)}{\sum_i \delta(v_i > 0)} \quad (19)$$

In Equation (19), i represents the annotated key point index, d_i^2 represents the squared Euclidean distance between the detected key point position and the ground truth key point position, s^2 represents the area occupied by the detected human body in the image, k_i represents the decay constant used to control the key point category i . δ is the impulse function, indicating that the L_{oks} value is only computed for visible relationship points in the ground truth annotations. v_i represents the visibility of the i key point, where 0 signifies unannotated, 1 signifies annotated but occluded, and 2 signifies annotated and visible.

The evaluation of the algorithm's object recognition accuracy for object category detection is conducted using precision (P), recall (R), and mean average precision (mAP) [32]. Precision (P) represents the proportion of correctly predicted samples among samples predicted as positive, as shown in Equation (20).

$$P = \frac{TP}{TP + FP} \times 100\% \quad (20)$$

The recall (R) represents the proportion of correctly predicted samples among the actual positive samples, as shown in Equation (21).

$$R = \frac{TP}{TP + FN} \times 100\% \quad (21)$$

The average precision (AP) is the area under the precision–recall curve, as indicated in Equation (12). The mean average precision (mAP) is defined as the average of AP values [33].

$$mAP = \frac{\sum_{i=1}^N \int_0^1 P(R) dr}{N} \times 100\% \quad (22)$$

In the aforementioned equation, TP represents the number of samples correctly detected as the target class in the image, FP represents the number of falsely detected samples in the image, and FN represents the number of samples in the image where the target class was not correctly detected. The similarity losses of the target key points in the training and validation datasets, as well as the accuracy curves of various indicators in the training dataset, are shown in Figure 11. From the figure, we can observe that when the model is

iterated 500 times, all the losses tend to stabilize and reach their minimum. At this point, all the accuracy metrics achieve their optimal values.

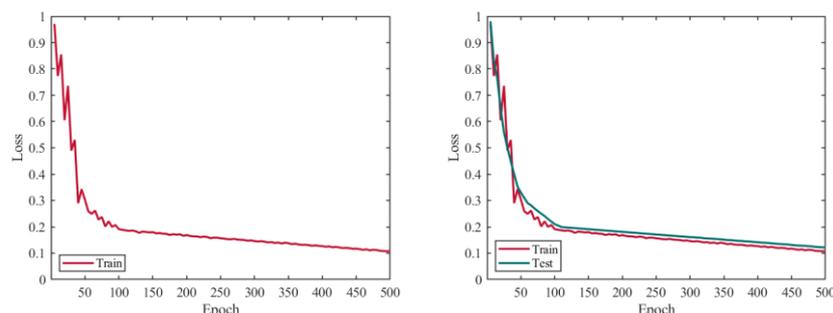


Figure 11. YOLO-Pose model training loss curve.

In key point detection, we adopt mAP50 and mAP50-95 as evaluation metrics. Here, mAP50 represents the evaluation metric for single-object class detection accuracy when the threshold L_{oks} is ≥ 0.5 . mAP50-95 represents the average detection accuracy over 10 different thresholds, including 0.5, 0.55, ..., 0.90, and 0.95, when using L_{oks} as the threshold. Based on the results shown in Figure 12, the accuracy rates of both mAP50 and mAP50-95 steadily increase within the first 100 iterations, from 0.5 to 0.8 and from 0.2 to 0.5, respectively. After the 100th iteration, the model's accuracy stabilizes, reaching around 0.8 for mAP50 and around 0.5 for mAP50-95. Precision and Recall stabilize around 0.94 and 0.85, respectively.

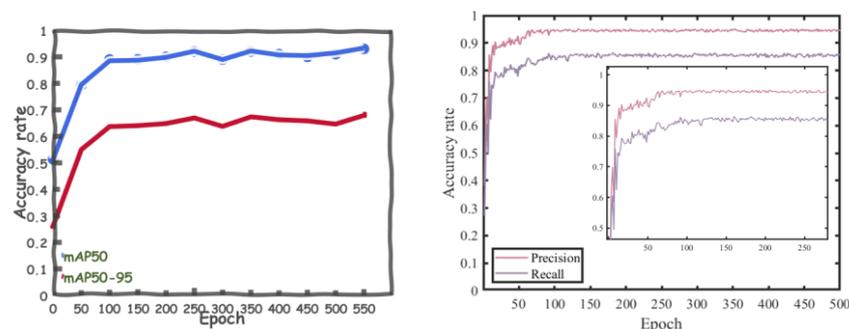


Figure 12. YOLO-Pose model accuracy loss curves.

6.3. Ablation Experiment

In this study, the YOLOv5 network model was utilized, where improvements [34] were made by introducing the GhostNet module in the Backbone section and the ACmix attention mechanism in the Neck section, optimizing the key point prediction in the Head section, and incorporating new loss and confidence functions. To evaluate the impact of these improvements on the overall model performance, ablation experiments were designed. These experiments involved applying different modules to the original network to assess the effects of each component on the model's performance enhancements.

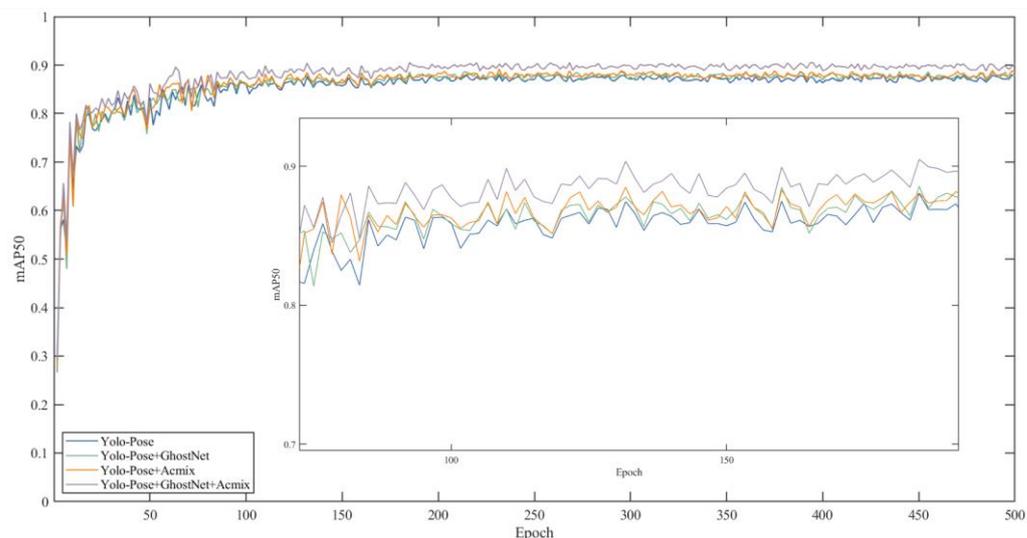
Table 1 presents the improved YOLO-Pose algorithm for human pose estimation, which exhibits enhanced performance compared to the original model across various metrics. In terms of object detection, the improved model achieves an accuracy of 94.58% and a recall rate of 86.54%, representing improvements of 4.87% and 4.11% respectively, as compared to the original model. For key point detection, the improved model achieves mAP50 and mAP50-95 of 93.58% and 69.54% respectively, which demonstrate improvements of 5.24 and 5.05 percentage points over the original model. The improved model has a parameter size of 22.3 M. Furthermore, the detection time for a single image is 19.9 ms, showing respective optimization improvements of 30% and 39.5% compared to the original model, thereby meeting the requirements for real-time detection.

Table 1. Network ablation test.

Module1	Module2	Module3	Module4	mAP50/%	mAP50-95/%	Params/M	Detection Time/ms
—	—	—	—	88.34	64.49	36.9	28.64
✓	—	—	—	88.62	63.45	26.88	23.84
—	✓	—	—	89.81	65.69	28.96	24.5
—	—	✓	—	88.59	66.98	30.11	26.6
—	—	—	✓	90.83	64.32	29.56	25.9
✓	✓	✓	✓	93.58	69.54	22.3	19.9

Notes: Module1 represents the introduction of the GhostNet module, Module2 represents the introduction of the ACmix module, Module3 represents the optimization of the Head part of the key point prediction, and Module4 represents the introduction of the new loss function and confidence function.

According to the graph, the training curves of Module1 GhostNet [35] and Module2 ACmix [36] are shown in Figure 13. It can be observed that the mAP50 performance of all four configurations shows a significant upward trend in the first 50 epochs. This indicates that the model quickly learns the patterns in the dataset during the early learning stages. Subsequently, the growth rate of the four curves begins to slow down and gradually enters a relatively stable state, indicating that the model starts to converge.

**Figure 13.** The training curve graphs for the ablation experiments of GhostNet and ACmix.

Throughout the entire training process, the configuration of YOLO-Pose + GhostNet + ACmix typically demonstrates the best performance (90.29%), followed by YOLO-Pose + ACmix (89.81%), YOLO-Pose + GhostNet (88.62%), and finally the baseline model YOLO-Pose (88.34%). This performance ranking suggests that combining GhostNet and ACmix effectively improves the model's mAP50 performance.

6.4. Model Comparison

In order to further verify the effect of this paper's algorithm after the improvement of light weight, speed, accuracy, and robustness, this paper's algorithm is compared with other algorithms horizontally. As shown in Table 2, it is tabulated with classical target detection algorithms such as Faster R-CNN, SSD, YOLOv4, YOLOv7, etc., and the evaluated metrics are mAP50, mAP50-95, number of parameters, and detection time, respectively.

According to Table 2, the two-stage detection algorithm, Faster R-CNN [37], has the best detection accuracy and outperforms models such as SSD, YOLOv4, and YOLOv7. However, its detection speed lags far behind that of the one-stage detection algorithms. The SSD algorithm falls behind the YOLO algorithms, specifically YOLOv5 and YOLOv7, in both mAP50 and mAP50-95 metrics. YOLOv7 exhibits improvements of 0.92% and

1.54% in mAP50 and mAP50-95 metrics, respectively, compared to YOLOv4. Additionally, it reduces the detection time by 3.29 ms, but experiences an increase of 3.3 M in terms of parameters.

Table 2. Comparison of test results of different algorithms.

	mAP50/%	mAP50-95/%	Params/M	Detection Time/ms
Faster R-CNN	91.49	68.87	27.6	35.56
SSD	87.6	62.60	27.8	28.83
YOLOv4	90.64	64.94	25.9	28.64
YOLOv7	91.56	66.48	29.2	25.35
YOLO-Pose	93.58	69.54	22.3	19.99

The YOLO-Pose algorithm proposed in this study achieves a higher detection accuracy than the Faster R-CNN algorithm by 2.09% in terms of the mAP50 metric. However, it lags behind by 0.67% in the mAP50-95 metric. This is because the two-stage detection algorithm introduces a CNN for feature extraction, enabling end-to-end training and utilizing bounding box regression to fine-tune the positions of human pose key points. Furthermore, our algorithm outperforms in terms of the Params metric by 5.3 M and in the detection time metric by 15.57 m/s. Overall, our algorithm strives to achieve an optimal balance among detection accuracy, parameter count, and detection time, taking into consideration the aforementioned characteristics.

6.5. Detection Effect

The visual results of the improved YOLO-Pose algorithm for human pose estimation proposed in this study are shown in Figure 14. We conducted detection for three different human poses, including standing, sitting, and participating in sports activities. It can be observed that the human key points are largely detected, even in scenarios involving small targets and complex sports poses. The algorithm demonstrates satisfactory performance in these challenging situations.

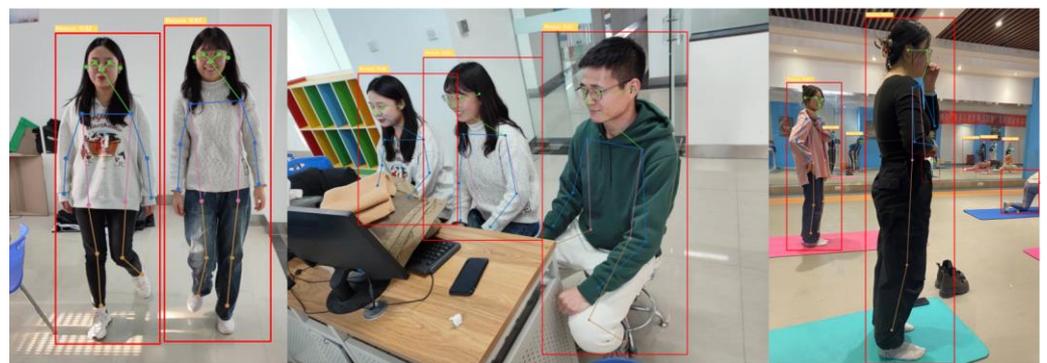


Figure 14. Visualization effect image 1 of YOLO-Pose model.

After deploying the YOLO-Pose model on an unmanned aerial vehicle (UAV), we conducted human pose detection in a different scenario. As shown in Figure 15, despite the challenges posed by the high number of targets and their small areas, the model was able to successfully accomplish the task of estimating human poses. Even when confronted with smaller target areas and more complex detection scenes, the model demonstrated its capability to largely complete the task, as depicted in Figure 16.

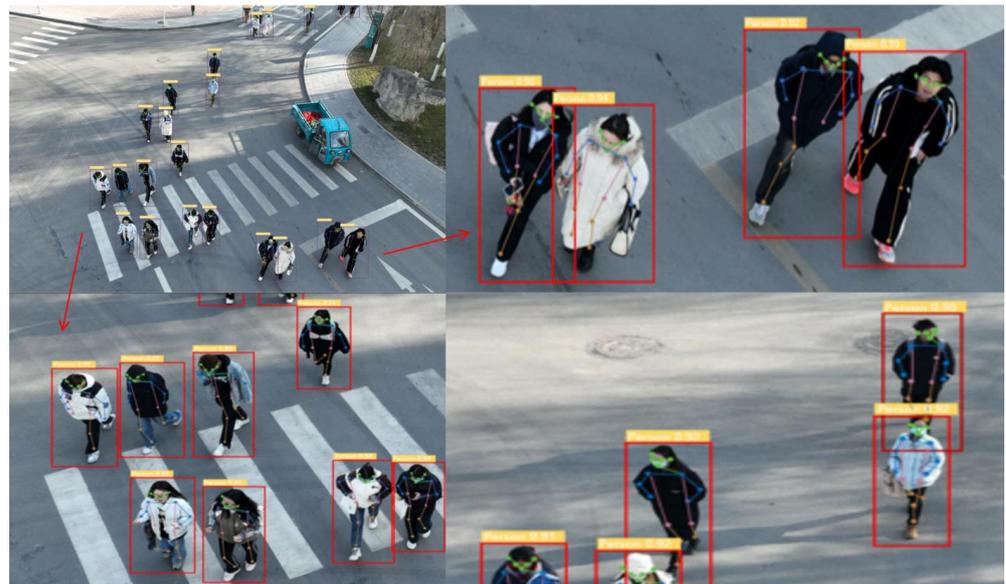


Figure 15. Visualization effect image 2 of YOLO-Pose model.



Figure 16. Visualization effect image 3 of YOLO-Pose model.

7. Conclusions

- (1) Human pose estimation is a significant computer vision task; however, practical applications are often hindered by challenges such as low lighting conditions, dense target presence, severe edge occlusion, limited application scenarios, complex backgrounds, and poor recognition accuracy when targets are occluded. In this paper, we propose a YOLO-Pose model that leverages the lightweight and precision-enhanced features of the YOLOv5 object detection model, enabling its effective deployment on unmanned aerial vehicles (UAVs).
- (2) Additionally, we employ transfer learning techniques by utilizing pre-trained models trained on the ImageNet-1K and COCO datasets to train our local dataset. In the YOLO-Pose model, we integrate lightweight GhostNet modules into the Backbone section to reduce the model's parameter count and computational requirements, making it more suitable for deployment on unmanned aerial vehicles (UAVs) to accomplish specific human pose detection tasks. In the Neck section, we introduce the ACmix attention mechanism to enhance detection speed during object judgment and

- localization. Furthermore, we optimize the Head section's key points by incorporating coordinate attention mechanisms to improve key point localization accuracy. We also enhance the loss function and confidence function to enhance the model's robustness.
- (3) The improved model demonstrates a reduction of 14.6 M parameters, an 8.47 ms decrease in detection time, a 5.24% improvement in mAP50, and a 5.05% improvement in mAP50-95. Notably, the parameter count and detection speed have been optimized by 30% and 39.5%, respectively, resulting in a detection speed of 19.9 ms per image. These enhancements enable the model to possess concise, user-friendly, and efficient features, making it suitable for monitoring students' movement poses and assessing their body posture. The model provides valuable technical support by identifying and evaluating various types and levels of poor posture and offering low-cost and easily implementable intervention strategies for physical activities.

Author Contributions: Conceptualization, S.N.; Methodology, Z.N. and J.D.; Project administration, J.D. and W.Z.; Software, S.N.; Supervision, Z.N.; Validation, J.D.; Visualization, W.Z.; Writing—original draft, S.N. and Z.N.; Writing—review and editing, Z.N. All authors have read and agreed to the published version of the manuscript.

Funding: This project received funding from the Youth Tutor Support Fund (GAU-QDFC-2022-19) of Gansu Agricultural University, the Student Innovation and Entrepreneurship Training Program (202319001) of Gansu Agricultural University, and the Industrial Support Program Project (2022CYZC-41) of the Education Department of Gansu Province.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article. You can send an email to the second author and corresponding author to request the data and code.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Zheng, C.; Zhu, S.; Mendieta, M.; Yang, T.; Chen, C.; Ding, Z. 3d human pose estimation with spatial and temporal transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 11656–11665.
- Liu, Z.; Chen, H.; Feng, R.; Wu, S.; Ji, S.; Yang, B.; Wang, X. Deep dual consecutive network for human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 525–534.
- Li, J.; Bian, S.; Zeng, A.; Wang, C.; Pang, B.; Liu, W.; Lu, C. Human pose regression with residual log-likelihood estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 11025–11034.
- Zhang, Y.; Zhu, T.; Ning, H.; Liu, Z. Classroom student posture recognition based on an improved high-resolution network. *EURASIP J. Wirel. Commun. Netw.* **2021**, *2021*, 140. [[CrossRef](#)]
- Li, W.; Liu, H.; Ding, R.; Liu, M.; Wang, P.; Yang, W. Exploiting temporal contexts with strided transformer for 3d human pose estimation. *IEEE Trans. Multimed.* **2022**, *25*, 1282–1293. [[CrossRef](#)]
- Liu, H.; Liu, T.; Chen, Y.; Zhang, Z.; Li, Y.F. EHPE: Skeleton cues-based gaussian coordinate encoding for efficient human pose estimation. *IEEE Trans. Multimed.* **2022**, 1–12. [[CrossRef](#)]
- Yuan, C.; Zhang, P.; Yang, Q.; Wang, J. Fall detection and direction judgment based on posture estimation. *Discret. Dyn. Nat. Soc.* **2022**, *2022*, 8372291. [[CrossRef](#)]
- Lee, M.F.; Chen, Y.C.; Tsai, C.Y. Deep Learning-Based Human Body Posture Recognition and Tracking for Unmanned Aerial Vehicles. *Processes* **2022**, *10*, 2295. [[CrossRef](#)]
- Su, W.; Feng, J. Research on Methods of Physical Aided Education Based on Deep Learning. *Sci. Program.* **2022**, *2022*, 6447471. [[CrossRef](#)]
- Amadi, L.; Agam, G. PosturePose: Optimized Posture Analysis for Semi-Supervised Monocular 3D Human Pose Estimation. *Sensors* **2023**, *23*, 9749. [[CrossRef](#)]
- Manesco, J.R.R.; Berretti, S.; Marana, A.N. DUA: A Domain-Unified Approach for Cross-Dataset 3D Human Pose Estimation. *Sensors* **2023**, *23*, 7312. [[CrossRef](#)]
- Li, H.; Yao, H.; Hou, Y. HPnet: Hybrid Parallel Network for Human Pose Estimation. *Sensors* **2023**, *23*, 4425. [[CrossRef](#)]
- Mathew, M.P.; Mahesh, T.Y. Leaf-based disease detection in bell pepper plant using YOLO v5. *Signal Image Video Process.* **2022**, *16*, 841–847. [[CrossRef](#)]

14. Niu, S.; Nie, Z.; Li, G.; Zhu, W. Early Drought Detection in Maize Using UAV Images and YOLOv8+. *Drones* **2024**, *8*, 170. [[CrossRef](#)]
15. Yi, P.; Wang, Z.; Jiang, K.; Shao, Z.; Ma, J. Multi-temporal ultra dense memory network for video super-resolution. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 2503–2516. [[CrossRef](#)]
16. Liu, C.; Yi, W.; Liu, M.; Wang, Y.; Hu, S.; Wu, M. A Lightweight Network Based on Improved YOLOv5s for Insulator Defect Detection. *Electronics* **2023**, *12*, 4292. [[CrossRef](#)]
17. Ryu, S.; Yun, S.; Lee, S.; Jeong, I.C. Exploring the Possibility of Photoplethysmography-Based Human Activity Recognition Using Convolutional Neural Networks. *Sensors* **2024**, *24*, 1610. [[CrossRef](#)] [[PubMed](#)]
18. Duan, X.; Lin, Y.; Li, L.; Zhang, F.; Li, S.; Liao, Y. Hierarchical Detection of *Gastrodia elata* Based on Improved YOLOX. *Agronomy* **2023**, *13*, 1477. [[CrossRef](#)]
19. Wang, T.; Zheng, Y.; Ma, H.; Wang, C.; Liu, Y.; Li, X. An improved YOLOv5-based bird detection algorithm for transmission line. In Proceedings of the 2022 International Conference on Computer Engineering and Artificial Intelligence (ICCEAI), Shijiazhuang, China, 22–24 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 79–83.
20. Lloyd, D.T.; Abela, A.; Farrugia, R.A.; Galea, A.; Valentino, G. Optically enhanced super-resolution of sea surface temperature using deep learning. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5000814. [[CrossRef](#)]
21. Hou, J.; Zhou, H.; Hu, J.; Yu, H.; Hu, H. A Multi-Scale Convolution and Multi-Layer Fusion Network for Remote Sensing Forest Tree Species Recognition. *Remote Sens.* **2023**, *15*, 4732. [[CrossRef](#)]
22. Ren, S.; He, K.; Girshick, R.; Zhang, X.; Sun, J. Object detection networks on convolutional feature maps. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1476–1481. [[CrossRef](#)] [[PubMed](#)]
23. Lin, M.; Ji, R.; Wang, Y.; Zhang, Y.; Zhang, B.; Tian, Y.; Shao, L. Hrank: Filter pruning using high-rank feature map. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 1529–1538.
24. Gränzig, T.; Fassnacht, F.E.; Kleinschmit, B.; Förster, M. Mapping the fractional coverage of the invasive shrub *Ulex europaeus* with multi-temporal Sentinel-2 imagery utilizing UAV orthoimages and a new spatial optimization approach. *Int. J. Appl. Earth Obs. Geoinformation* **2021**, *96*, 102281. [[CrossRef](#)]
25. Yin, D.; Wang, L.; Lu, Y.; Shi, C. Mangrove tree height growth monitoring from multi-temporal UAV-LiDAR. *Remote Sens. Environ.* **2024**, *303*, 114002. [[CrossRef](#)]
26. Gunduz, M.Z.; Das, R. Smart Grid Security: An Effective Hybrid CNN-Based Approach for Detecting Energy Theft Using Consumption Patterns. *Sensors* **2024**, *24*, 1148. [[CrossRef](#)] [[PubMed](#)]
27. Diwan, T.; Anirudh, G.; Tembhurne, J.V. Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimed. Tools Appl.* **2023**, *82*, 9243–9275. [[CrossRef](#)]
28. Yin, Y.; Li, H.; Fu, W. Faster-YOLO: An accurate and faster object detection method. *Digit. Signal Process.* **2020**, *102*, 102756. [[CrossRef](#)]
29. Hsu, W.Y.; Lin, W.Y. Ratio-and-scale-aware YOLO for pedestrian detection. *IEEE Trans. Image Process.* **2020**, *30*, 934–947. [[CrossRef](#)] [[PubMed](#)]
30. Terven, J.; Córdova-Esparza, D.M.; Romero-González, J.A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1680–1716. [[CrossRef](#)]
31. Nghonda Tchinda, E.; Panoff, M.K.; Tchuinkou Kwadjo, D.; Bobda, C. Semi-Supervised Image Stitching from Unstructured Camera Arrays. *Sensors* **2023**, *23*, 9481. [[CrossRef](#)] [[PubMed](#)]
32. Summerfield, G.I.; De Freitas, A.; van Marle-Koster, E.; Myburgh, H.C. Automated Cow Body Condition Scoring Using Multiple 3D Cameras and Convolutional Neural Networks. *Sensors* **2023**, *23*, 9051. [[CrossRef](#)] [[PubMed](#)]
33. Souza, B.J.; Stefenon, S.F.; Singh, G.; Freire, R.Z. Hybrid-YOLO for classification of insulators defects in transmission lines based on UAV. *Int. J. Electr. Power Energy Syst.* **2023**, *148*, 108982. [[CrossRef](#)]
34. Xu, F.; Liu, S.; Zhang, W. Research on power equipment troubleshooting based on improved AlexNet neural network. *J. Meas. Eng.* **2024**, *12*, 162–182. [[CrossRef](#)]
35. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020.
36. Pan, X.; Ge, C.; Lu, R.; Song, S.; Chen, G.; Huang, Z.; Huang, G. On the integration of self-attention and convolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
37. Su, P.; Han, H.; Liu, M.; Yang, T.; Liu, S. MOD-YOLO: Rethinking the YOLO architecture at the level of feature information and applying it to crack detection. *Expert Syst. Appl.* **2024**, *237*, 121346. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.