Study Protocol

Research on Real-Time Detection of Safety Harness Wearing of Workshop Personnel Based on YOLOv5 and OpenPose

Chengle Fang, Huiyu Xiang *, Chongjie Leng, Jiayue Chen and Qian Yu

School of Artificial Intelligence, Beijing Technology and Business University, Beijing 102401, China; 15655228850@163.com (C.F.); lcj@th.btbu.edu.cn (C.L.); lidongyang1225@163.com (J.C.); fang15655228850@126.com (Q.Y.)
* Correspondence: xianghy@th.btbu.edu.cn

Abstract: Wearing safety harness is essential for workers when carrying out work. When posture of the workers in the workshop is complex, using real-time detection program to detect workers wearing safety harness is challenging, with a high false alarm rate. In order to solve this problem, we use object detection network YOLOv5 and human body posture estimation network OpenPose for the detection of safety harnesses. We collected video streams of workers wearing safety harnesses to create a dataset, and trained the YOLOv5 model for safety harness detection. The OpenPose algorithm was used to estimate human body posture. Firstly, the images containing different postures of workers were processed to obtain 18 skeletal key points of the human torso. Then, we analyzed the key point information and designed the judgment criterion for different postures. Finally, the real-time detection program combined the results of object detection and human body posture estimation to judge the safety harness wearing situation within the current screen and output the final detection results. The experimental results prove that the accuracy rate of the YOLOv5 model in recognizing the safety harness reaches 89%, and the detection method of this study can ensure that the detection program accurately recognizes safety harnesses, and at the same time reduces the false alarm rate of the output results, which has high application value.

Keywords: safety harness detection; YOLOv5; OpenPose; detection program

1. Introduction

At present, artificial intelligence technology continues to develop and detection programs use deep learning methods and image vision technology to achieve automatic detection of whether workers wear a safety harness, and automatically output detection results. This is especially important for enterprise safety production and construction personnel life safety. Scholars have conducted considerable research on seat belt detection, and the detection methods and practical application scenarios vary greatly. Guo et al. [1] proposed an image processing-based seat belt detection method for car driving, which was applied to the vehicle motion full-scene monitoring image. First, the image is pre-processed by vertical boundary detection and horizontal boundary detection to obtain the driver area in the image. Then, the seat belt is detected by the edge detection method, and finally further verified using the judgment rule. The edge detection method has since improved based on the directional information measure in the HSV (hue, saturation, value) color space. However, the high demands of the detection method on the image quality and the camera recording angle make it difficult to promote this method. Feng et al. [2] proposed a new algorithm, based on the Mask R-CNN (region-based convolutional neural networks) algorithm, to detect incorrectly applied construction workers’ safety harnesses. First, based on the human localization detection template, the algorithm locates important skeletal key points in the specific regions of the knee. Then, the algorithm combines with the safety harness detection module to determine whether there is a safety harness at the location...
of the important skeletal key points. Traditional target detection focuses more on feature extraction, with general features and relatively strong interpretability. The core of deep learning is feature learning, which aims to obtain hierarchical feature information through hierarchical networks [3]. It can learn features by itself and does not need external users to design features for it artificially. Fu [4] proposed a deep learning-based seat belt detection method, for the application scenario of motor vehicles on the road. The details of this method are as follows. The detection model uses frame difference, edge detection and overall projection to locate the driver’s window area in the pre-processed image. Then, the processed image samples are used to train a convolutional neural network model, and the trained model is used to detect the seat belt. Jin et al. [5] proposed a helmet wearing detection algorithm based on improved YOLOv4 (You Only Look Once version 4). The algorithm optimizes the feature map output and feature fusion module. The algorithm adds $128 \times 128$ feature map outputs to the three feature map outputs of the YOLOv4 algorithm, providing smaller targets for feature fusion. Secondly, it improves the feature fusion module, which enables the YOLO Head classifier to combine different levels of features to achieve better object detection and classification. In order to solve the problem that existing helmet wearing detection cannot easily detect the helmet when the posture of the construction personnel is complex, Wang et al. [6] proposed a helmet wearing detection method based on posture estimation. The method provides ideas to solve the problem of difficult to determine relative positions of the helmet and human body in the complex posture of construction personnel.

Our study was conducted for the real-time detection of safety harnesses in the workshop. The complex posture of the worker (e.g., bending, squatting) increases the difficulty of the program to detect the safety harness and makes the program output have a high false alarm rate. The contributions of this study are as follows.

The scenario of the application of the detection program is the workshop; we obtain video stream files through workshop surveillance to create the dataset. We train the YOLOv5 (You Only Look Once version 5) model using the dataset.

The posture of the worker in the image is estimated by the OpenPose algorithm, and 18 skeletal key points are obtained. We design human body posture judgment criterion for the program based on the key point information.

Combining the YOLOv5 model detection results and the human body posture estimation results, we design the detection workflow for the program.

The experiments show that the detection program can accurately identify the safety harness on the worker. Compared with the program that only uses the YOLOv5 model for detection, the output of the improved program has a lower false alarm rate and reflects the worker’s posture in real-time. The rest of the article is organized as follows: Section 2 briefly summarizes the work related to our study. Section 3 describes in detail what we have done. Section 4 presents the experiments, results, and the discussion. Section 5 presents the conclusion and shortcoming of the proposed method, and ideas for improvement to our research in the next stage.

2. Related Work

In recent years, with the rapid development of deep learning technology, algorithms based on deep learning have been widely used in various fields. Tan et al. [7] proposed a novel phantom machine gesture interaction system improved for lightweight OpenPose. They used lightweight OpenPose to simplify the human hand into 21 key points, used MobileNetV1 as the base model, applied part affinity fields to detect the key points of the human hand, drew a simplified skeleton map, and used Ghost Module to downscale the convolutional layers to further improve the real-time performance of the human-computer interaction system. Wang et al. [8] proposed a detection algorithm for camouflage objects based on the YOLOv5 algorithm. The algorithm combines the attention mechanism to design a new feature extraction network that highlights the feature information of a camouflage object, and improves the original aggregation network.
Safety harnesses play a protective role for the body of personnel in the construction and production environment. At present, deep learning is gradually being applied to the detection of seat belts [9]. Less attention is paid to the detection of the workers wearing safety harnesses in workshops. Wu [10] conducted a study on visual inspection for safety protection of construction site personnel. In his research, he used the improved YOLOv3-tiny algorithm to detect workers’ safety harness and helmets. The algorithm did not take into account the temporal and spatial correlation in the actual site surveillance video. Cai et al. [11] designed a novel one-stage detection framework by incorporating several promising modules into a YOLO network, which is end-to-end trained. In addition, to improve the convergence of the proposed framework, a novel loss function was designed by adding a penalty term into the loss function. Fang et al. [12] developed an automated computer vision-based method. This method used two convolutional neural network (CNN) models to determine if workers were wearing their harness when performing tasks while working at heights. The algorithms developed were: (1) a Faster-R-CNN to detect the presence of a worker; and (2) a deep CNN model to identify the harness.

YOLO algorithm is a type of the one-stage algorithm, which not only has excellent performance in detection accuracy, but also has high detection efficiency [13]. The YOLO algorithm has been updated to the YOLOv5. Compared with YOLOv4, the network model of YOLOv5 has higher detection speed, which can reach 140 frames per second. Additionally, the network model size of YOLOv5 is nearly 90% smaller than of YOLOv4 [14,15]. YOLOv5 uses the Pytorch framework, which makes it easy to train datasets. Compared to the Darknet framework used by YOLOv4, the Pytorch framework is easier to put into production. OpenPose is one of the most popular open-source posture estimation technologies [16], which is a bottom-up detection method that can estimate human movements, recognize facial expressions, and capture finger movements. We can observe the movement of human skeleton key points for posture estimation. OpenPose mainly detects 18 key points of the human skeleton, such as knee and shoulder. Xu et al. [17] used OpenPose to get the data set of a human skeleton map and trained to get a new model that can predict the fall. Chen et al. [18] extracted the skeleton information of the human body by OpenPose and identified the fall through three critical parameters. In summary, we consider the application scenario of the detection program. Therefore, safety harness detection by the YOLOv5 algorithm and human posture estimation by OpenPose are the two main parts of the detection program.

3. Methods

3.1. Image Collection of Safety Harness

Firstly, the images in the dataset we used were obtained by web crawlers. The images included the safety harness in different directions, and workers in the safety harness. Figure 1 shows a selection of the images.
We found that these images did not reflect the real environment in the workshop, so we collected images from the workshop. Some of the images were taken with an iPhone, with a resolution of 480 × 800. Others were obtained from the workshop monitoring video, with a resolution of 1920 × 1080. The brightness of the environment, the distance between the worker and the lens, the posture of the human body, and the angle of the lens were considered in the process of acquiring images. Figure 2 shows a selection of this dataset.

![Figure 2. A selection of images from the dataset.](image)

We randomly divided the 2500 images in the dataset into two groups in the ratio of 4:1, one group as the training dataset and the other as the test dataset (the training set had 2000 images and the test set had 500 images). We made the images in both the training set and the test set come from the same distribution. Next, the dataset labeling website “Make Sense” was used to draw rectangular boxes to achieve labeling of a “person”, “safety belts”, and “safety helmet” in the image. After the marker was completed, a VOC format file was generated, then we converted the dataset from VOC format to txt format, as required by YOLOv5. The txt format file contains the annotation information of the images used for training or testing.

3.2. The Network Structure of YOLOv5

The YOLO algorithm uses a separate CNN model for end-to-end object detection; the CNN network of YOLO splits the input image into S × S grids [19]. Each cell is responsible for detecting the target whose center point falls within the cell, and each cell will predict B bounding boxes and the confidence level of the bounding box. The formula for calculating the confidence of the bounding box level is as follows [20]:

\[
Conf(obj) = Pr(object) \times IOU_{pred}^{truth}
\]  

(1)
$Pr(\text{object})$ indicates the size of the probability that this bounding box contains the object; when the bounding box is the background, the value is 0. On the contrary, when the bounding box contains the object, the value is 1. $\text{IOU}$ (intersection over union) indicates the accuracy of the bounding box, expressed as the intersection ratio of the predicted box to the actual box. The size and position of the bounding box is characterized by four values of $(x, y, w, h)$, $(x, y)$ is the center coordinate of the bounding box, and $w$ and $h$ are the width and height of the bounding box. In addition, each cell also needs to give the predicted $C$ categories probability value; each cell needs to predict $(B \times 5 + C)$ values, so the final prediction is a tensor of size $S \times S \times (B \times 5 + C)$ [21]. We choose YOLOv5s, one of the four versions of YOLOv5. Figure 3 shows the structure of the YOLOv5s network model [22]. Its structure consists of four main parts, namely input, backbone, neck, and prediction [23]. The backbone network is a convolutional neural network that extracts image features, and the neck network performs feature fusion using the FPN (feature pyramid network) algorithm and the PAN (pixel aggregation network) algorithm [24].

![YOLOv5s Structure](image)

Figure 3. YOLOv5s network model structure.

Compared to the YOLOv4 algorithm, YOLOv5 adds a focus structure to the backbone network and adaptive image scaling to the input side. In common object detection algorithms, different images have different lengths and widths, and the common approach is to scale the original image to a standard size uniformly, and then the image is fed into the network. In practice, the size of the black edges at both ends of the scaled image are different, which affect the inference speed once more information is filled. Therefore, YOLOv5 adaptively adds the least amount of black borders to the original image, increasing the reasoning speed. In addition, the backbone network uses the CSP (cross stage partial) structure in YOLOv4, and two CSP structures are used in YOLOv5. CSP1_X structure is applied to the backbone network, and CSP2_X structure is applied in the neck network to enhance network feature fusion. YOLOv5 uses CSPDarknet (cross stage partial Darknet) as the backbone network for feature extraction. CSPDarknet is the combination of the CSP structure [25] and the Darknet network. In the object detection problem, the use of CSPNet brings a larger boost to the backbone, effectively enhancing the learning capability of CNN, while reducing the computational effort. The CSP structure is shown in Figure 4.
Focus structure can be a further feature extraction; the key step is to perform a slicing operation on the image. The 640 × 640 × 3 image is divided into four slices after the slicing operation, and the size of each slice is 320 × 320 × 3. The connection layer combines four pieces together, resulting in a feature map of size 320 × 320 × 12. Then, after one convolution operation with 64 convolution kernels, the feature map of 320 × 320 × 32 is formed [26]. The focus structure is shown in Figure 5.

The loss function can reflect the degree of difference between the predicted value and the true value of the model. In the process of object detection, the relationship between the prediction frame and the real frame needs to be judged, and the model parameters are adjusted according to this relationship to correct the position of the prediction frame. The specific measure is denoted as IOU (intersection over union) [27]. However, IOU has some shortcomings. When there is no intersection between the prediction frame and the real frame, the IOU is 0, and the model cannot calculate the gradient and optimize the parameters. When the prediction frame is the same size as the real frame, IOU also cannot make a judgment. To solve this problem, Rezatofighi et al. [28] proposed a new metric, GIOU (generalized intersection over union). The GIOU calculation formula is as shown in Equations (2) and (3). C represents the area of the smallest box that can frame both the real
box and the predicted box; \( b \) and \( b_{gt} \), respectively, represent the centroids of the predicted box and the real box.

\[
GIOU (b, b_{gt}) = IOU (b, b_{gt}) - \frac{|c - (b \cup b_{gt})|}{|c|}
\]

(2)

\[
L_{GIOU} (b, b_{gt}) = 1 - GIOU (b, b_{gt})
\]

(3)

YOLOv5 has a total of three loss functions, which are the classification loss function, the localization loss function, and the confidence loss function. Classification loss and confidence loss are calculated using a binary cross-entropy damage function. YOLOv5 replaces the SoftMax function with multiple independent logistic classifiers. When calculating the classification loss for training, YOLOv5 uses a binary cross-entropy impairment for each label, which also avoids the use of SoftMax function and reduces the computational complexity. The confidence of the bounding box actually indicates whether there is a center point at this grid, i.e., whether there is an object. Therefore, YOLO treats it as a dichotomous problem. When the prediction value is closer to 1, it means that the place is more likely to have a target; on the contrary, the place is less likely to have a target. GIOU is usually used for the calculation of the localization loss function.

3.3. Acquisition of Information on Key Points of the Human Skeleton

OpenPose human body posture recognition project is an open-source library written by Mellon (CMU) based on convolutional neural networks and supervised learning with Caffe (Convolutional Architecture for Fast Feature Embedding) as the framework [29]. OpenPose network structure is two-branch, using the “two-branch multi-stage CNN” scheme. Figure 6 shows the OpenPose network structure. The upper branch is responsible for predicting the confidence map of the key points and generating the heatmap of the key points. The lower branch is responsible for predicting the part affinity fields between key points and generating the vector graph of key points [30]. Part affinity fields is a two-dimensional vector of each limb of the body, which also holds the position and orientation information between limb regions [31]. First, the VGG-19 (visual geometry neural network with 19 layers) deep neural network performs feature extraction on the input image, resulting in the generation of feature map \( F \). In stage 1, the input is \( F \), the output is the key point heat map \( S^1 \), and the partial affinity domain \( L^1 \) after the convolution operation. Next, from the stage 2, the inputs are the two prediction results of the previous stage and the image features \( F \). Accordingly, the inputs for each stage are as follows [32]:

\[
S^1 = \rho^1 (F)
\]

(4)

\[
L^1 = \phi^1 (F)
\]

(5)

\[
S^t = \rho^t \big( F, S^{t-1}, L^{t-1} \big), \forall t \geq 2
\]

(6)

\[
L^t = \phi^t \big( F, S^{t-1}, L^{t-1} \big), \forall t \geq 2
\]

(7)

Through multi-stage iterations, the model predicts key points more accurately. Finally, the model obtains the confidence of all the key points and the direction vectors of the connected key points. For any two key points, the model pairs the two key points by calculating the linear integral of the part affinity fields, based on the confidence level of the key points [33]. With the Hungarian algorithm, high quality pairs can be generated. In the end, the human body key point skeleton map is obtained.
Figure 6. The network structure of OpenPose.

3.4. Criterion for Judging Human Posture

The OpenPose network can realize the detection of 18 key points of human body, and the change of human body posture can be expressed by the information of 18 key points. The skeleton diagram of the human body is shown in Figure 7.

When outputting images, we can clearly determine the posture of the human body by looking at the human skeleton diagram. In order for the program to automatically determine the detected human posture so that the program can proceed to the next step, it is necessary to set the judgement criteria of human body posture for the program to make the determination. According to the survey, workers may appear in the posture of standing, squatting, bending in workshop, and we set the rules for the determination of these postures. First, the distance in the vertical direction (y-direction) of the skeletal joint points is used as the determination feature. When the worker is far away from the camera, they will shrink on the screen, and the distance between two points will become shorter. When worker is close to the camera, they will become larger on the screen, and the distance between two points will become larger. Therefore, the angle is selected as an auxiliary determination parameter. In this study, we used the vertical distance and specific angle to

Figure 7. Skeleton diagram of the human body.
determine the posture of the human body; we have selected some skeletal joints that can determine the posture of the human body, as shown in Table 1.

### Table 1. Selection of feature distance and feature angle.

| Human Posture | Feature Distance of Y-Direction | Feature Angle |
|---------------|---------------------------------|--------------|
| Standing      | 1-8, 8-10                       | 1-8-9        |
| Bending       | 1-8, 8-10                       | 1-8-9        |
| Squatting     | 1-8, 8-10                       | 8-9-10       |

OpenPose outputs key point information as \([x_i, y_i, \text{score}, i]\), where \(x_i\) and \(y_i\) denote the horizontal and vertical coordinates of the \(i\)th key point in the pixel coordinate system, respectively, and \(\text{score}\) denotes the confidence level of the \(i\)th key point. In two-dimensional space, the formula for the distance between two points is as follows:

\[
\rho = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}
\]  

\(y_{1-8}\) indicates the distance between key point 1 and key point 8 in the vertical direction. 
\(y_{8-10}\) indicates the distance between key point 8 and key point 10 in the vertical direction. 
\(\theta_{1-8-9}\) represents the angle between line segments 1-8 and line segments 9-8; similarly, \(\theta_{8-9-10}\) and \(\theta_{8-1-11}\). Taking \(\theta_{1-8-9}\) as an example, the formula for calculating the angle is as follows:

\[
\theta = \arccos\left(\frac{\rho^2_{1-8} + \rho^2_{9-8} - \rho^2_{9-1}}{2\rho_{1-8}\rho_{9-8}}\right)
\]  

The experiment was conducted to detect the possible standing posture of workers in the workshop environment, and a total of 60 images were detected, including 20 each of close, medium and far views. Calculating \(y_{1-8}, y_{8-10}, \theta_{1-8-9}\) of each image from the detected key point coordinates, each feature had a total of 60 data points. To enable better evaluation of the overall data, we calculated the harmonic mean of data for feature distance in the y-direction. The formula for the harmonic mean is as follows, where \(n\) is the number of features, and \(x_j\) represents the feature data.

\[
H = \frac{n}{\sum_{j=1}^{n} \frac{1}{x_j}}
\]  

In this case of standing, the calculation of the feature distance was denoted as \(H^{\text{stand}}_{1-8}, H^{\text{stand}}_{8-10}\). For the feature angle, we choose the angle that is just right for the detected human posture in the experiment. The maximum angle was noted as \(\theta^{\text{stand-max}}_{1-8-9}\), the minimum angle was noted as \(\theta^{\text{stand-min}}_{1-8-9}\). The above value was used as the threshold for determination. The same method is used to obtain the threshold values for the determination of different postures, as shown in Table 2.

### Table 2. Threshold values for judgments.

| Posture   | Threshold Value | Threshold Value |
|-----------|-----------------|-----------------|
| Standing  | \(H^{\text{stand}}_{1-8}, H^{\text{stand}}_{8-10}\) | \(\theta^{\text{stand-max}}_{1-8-9}, \theta^{\text{stand-min}}_{1-8-9}\) |
| Bending   | \(H^{\text{bend}}_{1-8}, H^{\text{bend}}_{8-10}\)  | \(\theta^{\text{bend-max}}_{1-8-9}, \theta^{\text{bend-min}}_{1-8-9}\) |
| Squatting | \(H^{\text{down}}_{1-8}, H^{\text{down}}_{8-10}\)   | \(\theta^{\text{down-max}}_{8-9-10}, \theta^{\text{down-min}}_{8-9-10}\) |

The human posture determination guidelines are as follows.
(1) If \( y_{1-8} \leq H_{\text{down}}^{1-8} \) or \( H_{\text{down}}^{1-8} < y_{1-8} < H_{\text{bend}}^{1-8} \) or \( y_{8-10} < H_{\text{down}}^{8-10} \) or \( H_{\text{down}}^{8-10} < y_{8-10} < H_{\text{bend}}^{8-10} \), \( \theta_{\text{down-min}}^{8-9-10} \ll \theta_{8-9-10} \ll \theta_{\text{down-max}}^{8-9-10} \), the program determines that the worker’s posture is squatting.

(2) If \( H_{\text{bend}}^{1-8} \ll y_{1-8} < H_{\text{stand}}^{1-8} \) or \( H_{\text{bend}}^{8-10} \ll y_{8-10} < H_{\text{stand}}^{8-10} \), \( \theta_{\text{bend-min}}^{1-8-9} \ll \theta_{1-8-9} \ll \theta_{\text{bend-max}}^{1-8-9} \), the program determines that the worker’s posture is bending.

(3) If \( H_{\text{stand}}^{1-8} \ll y_{1-8} \), \( H_{\text{stand}}^{8-10} \ll y_{8-10} \), \( \theta_{\text{stand-min}}^{1-8-9} \ll \theta_{1-8-9} \ll \theta_{\text{stand-max}}^{1-8-9} \), the program determines that that the worker’s posture is standing.

### 3.5. Program Detection Flow

In previous studies, scholars improved the accuracy of the algorithm to detect object features by improving the object detection algorithm, which had significant implications. However, in some practical applications, because the features of the object are easily obscured by other objects, the results of the algorithm detection may not match the real situation. In this study, at the beginning, the program detection flow is shown in Figure 8. When the worker’s posture is bending down, squatting, the feature of safety harness is obscured by the limb, and it is difficult for the program to detect the safety harness and the output result does not match with the real situation. Therefore, we have redesigned the program detection flow. The flow chart is shown in Figure 9.
4. Results and Discussion

4.1. Experimental Environment

The computer configuration used for this experiment was as follows. CPU: Gen Intel® Core™ i7-11700F@ 2.50 GHz, RAM: 32 GB, GPU: NVIDIA GeForce RTX 3070Ti, Operating System: Win10 operating system. The deep learning framework used in this study was Pytorch. We utilized the Python language to write the code for program. Nvidia CUDA and Nvidia CUDNN were used to accelerate GPU operations. According to the actual needs of the experiment, the model file we used was YOLOv5s.yaml, and the initial weight file was YOLOv5s.pt. Initializing model parameters, we set the initial learning rate to 0.01, the number of iterations of the algorithm was set to 299, the batch size was set to 16, and the attenuation coefficient was set to 0.0005.

4.2. Results and Analysis of Safety Harness Detection

For YOLO algorithm object detection, the metrics to evaluate the detection performance are loss function (GIoU), recall (Recall), precision (Precision), validation set loss function (val GIoU), average accuracy mean (mAP), etc. The model was trained 299 times; the test results are shown in Figure 10.

Figure 9. Flow chart of the improved program.

Figure 10. Result graphs.
The precision, recall, AP, and mAP can be calculated by the following equations:

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (11)
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (12)
\]

\[
AP = \frac{\sum \text{Precision}}{N} \quad (13)
\]

\[
mAP = \frac{\sum AP}{NC} \quad (14)
\]

TP is the positive sample correctly classified in the algorithm, FP is the misclassified positive sample, FN is a negative sample of misclassification, N is the number of images, and NC is the object type.

The GIOU value finally stabilized at about 0.02, which indicated that the difference between the predicted and actual values of the model for the object is small. The precision value was around 88%, and the recall value was around 86%. Precision represents the ability of the model to find relevant targets, i.e., the ability of the model to hit the true target among all predictions given, and recall represents the ability of the model to find relevant targets among all targets. We observed the value of the precision rate with the value of the recall rate and found that the model was more accurate in predicting the safety harness. In order to directly show the detection results of the algorithm, the detection images were selected for illustration, as shown in Figure 11.

Figure 11. Object detection results of the YOLOv5 network.

As we can be seen in Figure 11, when the features of safety harness were obvious in the image, the model had very good recognition of the safety harness, accurate positioning, and the accuracy could reach up to 90%. When worker was bending or squatting, we found
that the features of the safety harness were obscured by the worker’s limbs, and the model was not able to detect the safety harness. In summary, the YOLOv5 algorithm had excellent effect in this aspect of the safety harness detection in the workshop.

In order to verify the detection accuracy and detection speed of the YOLOv5 algorithm for the safety harness, we compare the accuracy and detection speed with the data in reference No.2. The comparison results are shown in Table 3. In reference No.2, the authors used the Mask R-CNN algorithm for the detection of aerial work harnesses. The algorithm identified the aerial work harness with an accuracy of 98%, and the algorithm recognized each image in about 4 s on average. In our experiments, the accuracy of the YOLOv5 algorithm to identify the safety harness was about 89%. The YOLOv5 algorithm processed an image at a speed of 0.018 s, which meant that the YOLOv5 algorithm was able to process about 56 images per second.

Table 3. Comparison of test results.

|                | Accuracy | Speed of Processing an Image |
|----------------|----------|------------------------------|
| Reference No.2 | 98%      | 4 s                          |
| YOLOv5         | 89%      | 0.018 s                      |

The detection accuracy of Mask R-CNN is slightly higher than that of YOLOv5, but the detection speed of Mask R-CNN is lower than that of YOLOv5. This is because YOLOv5 is one of the one-stage algorithms and Mask R-CNN is one of the two-stage algorithms. Currently, target detection algorithms can be divided into two categories, which are one-stage and two-stage. The fundamental difference between the two methods is the difference in the candidate region boxes. The Mask R-CNN algorithm first generates candidate regions and then performs convolutional neural network classification on each candidate box [34]. The detection speed of these algorithms is relatively slow, as it requires multiple runs of the detection and classification process. The one-stage detection method, on the other hand, can predict all the bounding boxes by simply feeding the images into the network once, which makes it faster. So, one-stage has slightly lower detection accuracy and faster detection than two-stage. In general, the YOLOv5 algorithm meets the requirements for real-time detection of the safety harness.

4.3. Human Body Posture Estimation

The OpenPose algorithm detected images containing a human body in a squatting or bending posture. In Figure 12, we can see the human skeleton diagram in different postures. We can easily understand the posture of the human body by the human skeleton diagram.

Figure 12. OpenPose detection results.

Next, we stopped the program to output the post-detection image, and used Python to write the designed criterion for judging human posture into the program. We ran the
program and let the program output the judgment result. As shown in Table 4, a total of 130 images were detected, including 30 images with standing posture, 50 images with bending posture, and 50 images with squatting posture.

Table 4. Classification of detect results.

| Posture | Program Output Results          | Number of Detected Images |
|---------|---------------------------------|---------------------------|
| Standing| The human body posture is standing | 30                        |
| Bending | The human body posture is bending  | 50                        |
| Squatting| The human body posture is squatting  | 50                        |

In the posture test, there were three outputs possible. In the first case, the program outputted the correct result; in the second case, the program did not output results, this was because the confidence level of feature points detected by the algorithm was too low; in the third case, the program outputted a result of posture detection that did not match the real situation. For example, a worker was standing, but the output of program was that the worker’s posture was bending. Using the statistics of 130 experimental results, the specific situation is shown in the Figure 13. Case 1 accounted for the majority of all test results, demonstrating the feasibility and accuracy of the designed criterion for judging human body posture.

Figure 13. Result statistics.

4.4. Detect Safety Harness According to the Program

According to the designed program flow chart (Figure 9), we wrote the YOLOv5 model and the OpenPose model, and added some judgment rules into the program code using Python language. During the test, the program reads a video of three minutes in length. Because the detection speed of YOLOv5 algorithm was very fast, in order to enable the program to detect safety harness in real time, we controlled the speed of outputting the result to once per second. Therefore, the program outputted a total of 180 results. We designed five categories for the output results, which were “Not wearing safety harness”, “Wearing safety harness”, “Be careful, posture is standing”, “Be careful, posture is bending”, and “Be careful, posture is squatting”. The five categories were represented by “A”, “B”, “C”, “D” and “E”, respectively. After the program detection was completed, we verified the 180 output results with the contents of 180 frames of images. Finally, statistical analyses of the data were performed. In addition, we compared the results of the improved program detection with the results of no improved program. For the not improved program, only the YOLOv5 model was used in the test. The experimental results are shown in Figure 14.
verified the 180 output results with the contents of 180 frames of images. Finally, statistical analyses of the data were performed. In addition, we compared the results of the improved program detection with the results of the not improved program. For the not improved program, only the YOLOv5 model was used in the test. The experimental results are shown in Figure 14.

In this experiment, \( TP \), \( FP \), \( TN \) and \( FN \) were defined as follows:

- **True positive (\( TP \))**: workers wear safety harness, posture is standing, bending, or squatting, and the program detects it.
- **False positive (\( FP \))**: workers wear safety harness, posture is standing, bending, or squatting, but the program does not detect it, and the program output result is different from the real situation.
- **True negative (\( TN \))**: workers do not wear safety harness, the posture is not squatting, standing, or bending, and the program detects it.
- **False negative (\( FN \))**: workers do not wear safety harness, the posture is not standing, bending, or squatting, but the program does not detect it, and the program output result is different from the real situation.

The false alarm rate is the ability of the program to correctly predict the purity of a positive sample.

\[
\text{False alarm rate} = \frac{FP}{FP + TN} \tag{15}
\]

Specificity is the ability of the program to correctly predict negative sample fullness.

\[
\text{Specificity} = \frac{TN}{FP + TN} \tag{16}
\]

Accuracy is the ability of the program to judge the overall sample correctly.

\[
\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \tag{17}
\]

According to the calculation formula, the results of false alarm rate, accuracy and specificity are shown in Table 5. The not improved program used the trained YOLOv5 model to detect the safety harness with a high recognition rate. When the features of safety harness were evident in the image, the program detected them properly and was able to recognize them. However, when features of the safety harness were not obvious due to the change of human body posture and the program only used YOLOv5 to detect the safety harness, the program was not able to recognize the features of the safety harness or recognize fewer features at this point, and it was easy to output the wrong result. From the test results, the accuracy of the program that was not improved was 56.7%, and it had a high false alarm rate, up to 56.5%. The improved program added the OpenPose model and criterion for judging human posture with an accuracy of 92.2%, which reduced the false alarm rate and indirectly improved the accuracy of the program. When the
improved program was confronted with the problem that the features of safety harness in the image were not obvious, the improved program would not output the results immediately. According to the flow chart of the improved program (Figure 9), the following results can be recorded: firstly, the improved program determined the current posture of the human body based on OpenPose and criterion for judging human posture. Secondly, when the improved program detected that the human posture was standing, it outputted the human posture and the result of YOLOv5 algorithm detection. Finally, if improved program detected that the human pose was squatting or standing, the program decided the output based on the YOLOv5 algorithm’s detection result of the next frame. After a series of operations, the improved program was much better than the not improved program.

Table 5. Comparison of program detection results.

|                  | Accuracy | False Alarm Rate | Specificity |
|------------------|----------|------------------|-------------|
| Not improved program | 56.7%    | 56.5%            | 43.5%       |
| Improved program  | 92.2%    | 18.9%            | 81.1%       |

We choose the YOLOv5s network and the lightweight OpenPose network as the basic models of program in this study, and YOLOv5s is the smallest network in YOLOv5. Therefore, the program has considerable portability. However, the program still has also some shortcomings. First, in fact, the posture of workers are more complex than the postures studied in this study. Our designed criterion for judging human posture cannot adequately address the judgment of actual complex posture. Second, when workers are far away from the camera, we find that the YOLOv5 algorithm has difficulty in detecting the safety harness, because features of safety harness are not obvious in those images. Third, YOLOv5 and OpenPose algorithms take time to process the images, and there are some judgment rules in the program. So, in our tests, we found a delay of two to three seconds in the program output. These problems need to be further addressed. On the whole, the improved program has the following advantages: it outputs results with high accuracy and a low false alarm rate; and it is relatively light and has a lot of portability, so it can be easily installed on embedded equipment and mobile phones.

5. Conclusions

In this paper, we conduct a study for the real-time detection of safety harnesses in workshops, and propose a detection scheme based on YOLOv5 object detection and OpenPose posture estimation. In our proposed method, first, we collect representative images from the workshop, make datasets, and then train the YOLOv5s model. Second, based on the information of the key points of human skeleton detected by the OpenPose algorithm, we design a human body posture judgment criterion for the program. Finally, we redesign the detection process of the program by combining YOLOv5s and OpenPose. The improved program was compared with the program without improvement for experiments. According to the experimental results, the improved program has a high accuracy rate and a low false alarm rate, which meets the needs of real scenarios. In the meantime, we implement the deployment of the program on embedded equipment in the workshop and remotely control the program to detect the safety harness.

In addition, we analyze improved program’s output errors. Results of the analysis are as follows. Firstly, the confidence of feature points detected by the OpenPose algorithm is too low, making the program unable to determine human body posture. Secondly, when human body posture changes from standing to squatting or bending, in this process, the improved program is not able to correctly handle the distance and angle between the relevant feature points due to the change of feature point positions. In the future, we will improve the OpenPose algorithm to enhance its ability to identify feature points. For the second analysis result above, we will optimize criterion for judging human posture, and we will also consider training OpenPose models with specific datasets, so that it can
automatically recognize and classify human postures. Furthermore, we will also further improve the ability of the YOLOv5 algorithm to identify safety harnesses by changing the network structure.

**Author Contributions:** Conceptualization, H.X. and C.F.; methodology, C.F.; software, H.X.; formal analysis, H.X. and C.F.; investigation, C.L.; resources, J.C. and Q.Y.; writing—original draft preparation, C.F.; writing—review and editing, C.F.; funding acquisition, H.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The dataset used in the experiment was constructed by collecting images from web crawlers and inspecting the company’s production workshop. The dataset contains 2500 images. The dataset includes images of safety harness in different situations, such as different ambient light, different postures of the human body, and different distances between the safety harness and the lens.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

1. Guo, H.; Lin, H.; Zhang, S.; Li, S. Imaged-based seat belt detection. In Proceedings of the 2011 IEEE International Conference on Vehicular Electronics and Safety, Beijing, China, 10–12 July 2011; pp. 161–164. Available online: https://www.researchgate.net/publication/252029744 (accessed on 17 March 2022).

2. Feng, Z.; Zhang, W.; Zhen, Z. Mask R-CNN based aerial work harness detection. *Comput. Syst. Appl.* 2021, 30, 202–207. [CrossRef]

3. Ghods, A.; Cook, D.J. A survey of deep network techniques all classifiers can adopt. *Data Min. Knowl. Discov.* 2020, 35, 46–87. [CrossRef] [PubMed]

4. Fu, C. Research on Seat Belt Detection Method Based on Deep Learning. Master’s Thesis, Huazhong University of Science and Technology, Wuhan, China, 2015.

5. Jin, Y.; Wu, X.; Dong, H.; Yu, L.; Zhang, L. Helmet wearing detection algorithm based on improved YOLOv4. *Comput. Sci.* 2021, 48, 268–275.

6. Wang, Y.; Gu, Y.; Feng, X.; Fu, X.; Zhuang, L.; Xu, S. Research on helmet wearing detection method based on pose estimation. *Comput. Appl. Res.* 2021, 38, 937–940. [CrossRef]

7. Tan, L.; Lu, J.; Zhang, X.; Liu, Y.; Zhang, R. Improved gesture interaction system for phantom machines based on lightweight OpenPose. *Comput. Eng. Appl.* 2021, 57, 159–166.

8. Wang, Y.; Cao, T.; Cao, T.; Yang, J.; Zheng, Y.; Fang, Z.; Deng, X.; Wu, J.; Lin, J. Research on camouflage target detection technology based on YOLOv5 algorithm. *Comput. Sci.* 2021, 48, 226–232.

9. Hao, X.; Meng, X.; Zhang, Y.; Xue, J.; Xia, J. Conveyor Belt Detection Based on Deep Convolution GANs. *Intell. Autom. Soft Comput.* 2021, 29, 601–613. Available online: http://www.techscience.com/iasc/v30n2/44027 (accessed on 15 March 2022). [CrossRef]

10. Wu, J. Research on Visual Inspection for Safety Protection of Construction Site Personnel. Master’s Thesis, Guangdong University of Technology, Guangzhou, China, 2020. [CrossRef]

11. Wu, L.; Cai, N.; Liu, Z.; Yuan, A.; Wang, H. A one-stage deep learning framework for automatic detection of safety harnesses in high-altitude operations. *Signal Image Video Process.* 2022, 4, 15. [CrossRef]

12. Fang, W.; Ding, L.; Luo, H.; Love, P.E.D. Falls from heights: A computer vision-based approach for safety harness detection. *Autom. Constr.* 2018, 91, 53–61. Available online: https://www.sciencedirect.com/science/article/pii/S0926580517308403 (accessed on 15 March 2022). [CrossRef]

13. Liu, C.; Wu, Y.; Liu, J.; Sun, Z.; Xu, H. Insulator Faults Detection in Aerial Images from High-Voltage Transmission Lines Based on Deep Learning Model. *Appl. Sci.* 2021, 11, 4647. Available online: https://www.mdpi.com/2076-3417/11/10/4647 (accessed on 16 March 2022). [CrossRef]

14. Adibhatla, V.A.; Chih, H.-C.; Hsu, C.-C.; Cheng, J.; Abbod, M.F.; Shieh, J.-S. Applying deep learning to defect detection in printed circuit boards via a newest model of you-only-look-once. *Math. Biosci. Eng.* 2021, 18, 4411–4428. [CrossRef] [PubMed]

15. Ren, P.; Wang, L.; Fang, W.; Song, S.; Djahel, S. A novel squeeze YOLO-based real-time people counting approach. *Int. J. Bio-Inspired Comput.* 2020, 16, 94. [CrossRef]

16. Zago, M.; Luzzago, M.; Marangoni, T.; De Cecco, M.; Tarabini, M.; Galli, M. 3D Tracking of Human Motion Using Visual Skeletonization and Stereoscopic Vision. *Front. Bioeng. Biotechnol.* 2020, 8, 181. [CrossRef] [PubMed]
17. Xu, Q.; Huang, G.; Yu, M.; Guo, Y. Fall prediction based on key points of human bones. *Phys. A Stat. Mech. Its Appl.* **2020**, *540*, 123205. Available online: https://www.sciencedirect.com/science/article/pii/S0378437119318011 (accessed on 5 March 2022). [CrossRef]

18. Chen, W.; Jiang, Z.; Guo, H.; Ni, X. Fall Detection Based on Key Points of Human-Skeleton Using OpenPose. *Symmetry* **2020**, *12*, 744. [CrossRef]

19. Liu, C.; Wu, Y.; Liu, J.; Han, J. MTI-YOLO: A Light-Weight and Real-Time Deep Neural Network for Insulator Detection in Complex Aerial Images. *Energies* **2021**, *14*, 1426. [CrossRef]

20. Li, N.; Wang, X.; Fu, Y.; Zheng, F.; He, D.; Yuan, S. A traffic police target detection method with optimized YOLO model. *J. Graph.* **2022**, *1*, 11.

21. Lu, J.; Lu, Z.; Zhan, T.; Dai, Y.; Wang, P. Dog face detection algorithm based on YOLO and deep residual hybrid network. *Comput. Appl. Softw.* **2021**, *38*, 140–145. [CrossRef]

22. He, Y.; Li, H. Mask wearing recognition in complex scenes based on improved YOLOv5 model. *Microprocessor* **2022**, *43*, 42–46.

23. Wu, W.; Liu, H.; Li, L.; Long, Y.; Wang, X.; Wang, Z.; Li, J.; Chang, Y. Application of local fully Convolutional Neural Network combined with YOLOv5 algorithm in small target detection of remote sensing image. *PLoS ONE* **2021**, *16*, e0259283. [CrossRef]

24. Palucci Vieira, L.H.; Santiago, P.R.P.; Pinto, A.; Aquino, R.; Torres, R.d.S.; Barbieri, F.A. Automatic Markerless Motion Detector Method against Traditional Digitisation for 3-Dimensional Movement Kinematic Analysis of Ball Kicking in Soccer Field Context. *Int. J. Environ. Res. Public Health* **2022**, *19*, 1179. [CrossRef] [PubMed]

25. Wang, C.-Y.; Liao, H.-Y.M.; Yeh, I.-H.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops(CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 1571–1580.

26. Guo, K.; He, C.; Yang, M.; Wang, S. A pavement distresses identification method optimized for YOLOv5s. *Sci. Rep.* **2022**, *12*, 3542. [CrossRef]

27. Wu, S.; Yang, J.; Wang, X.; Li, X. IoU-Balanced Loss Functions for Single-stage Object Detection. *arXiv* 2019, arXiv:abs/1908.05641. [CrossRef]

28. Rezatofighi, H.; Tsoi, N.; Gwak, J.Y.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection over Union: A Metric and a Loss for Bounding Box Regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 658–666. [CrossRef]

29. Cao, Z.; Simon, T.; Wei, S.-E.; Sheikh, Y. Realtime multi-person 2d pose estimation using part affinity fields. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7291–7299.

30. Lin, C.-B.; Dong, Z.; Kuan, W.-K.; Huang, Y.-F. A Framework for Fall Detection Based on OpenPose Skeleton and LSTM/GRU Models. *Appl. Sci.* **2020**, *11*, 329. [CrossRef]

31. Su, C.; Wang, G. A study of student behavior recognition based on improved OpenPose. *Comput. Appl. Res.* **2021**, *38*, 3183–3188. [CrossRef]

32. Fu, N.; Liu, D.; Cheng, X.; Jing, Y.; Zhang, X. Fall detection algorithm based on lightweight OpenPose model. *Sens. Microsyst.* **2021**, *40*, 131–134. [CrossRef]

33. Lin, C.-B.; Dong, Z.; Kuan, W.-K.; Huang, Y.-F. A Framework for Fall Detection Based on OpenPose Skeleton and LSTM/GRU Models. *Appl. Sci.* **2020**, *11*, 329. [CrossRef]