sensors

MDPI

Article

Evaluating the Work Productivity of Assembling Reinforcement through the Objects Detected by Deep Learning

Jiaqi Li [1], Xuefeng Zhao [1,2,*], Guangyi Zhou [1,3], Mingyuan Zhang [1], Dongfang Li [1,3] and Yaochen Zhou [3]

[1] Faculty of Infrastructure Engineering, Dalian University of Technology, Dalian 116024, China; lijq@mail.dlut.edu.cn (J.L.); zgy0829@163.com (G.Z.); myzhang@dlut.edu.cn (M.Z.); lidongfang0303@163.com (D.L.)

[2] State Key Laboratory of Coastal and Offshore Engineering, Dalian University of Technology, Dalian 116024, China

[3] Northeast Branch China Construction Eighth Engineering Division Corp., Ltd., Dalian 116019, China; zhouyaochen1949@163.com

* Correspondence: zhaoxf@dlut.edu.cn

Abstract: With the rapid development of deep learning, computer vision has assisted in solving a variety of problems in engineering construction. However, very few computer vision-based approaches have been proposed on work productivity's evaluation. Therefore, taking a super high-rise project as a research case, using the detected object information obtained by a deep learning algorithm, a computer vision-based method for evaluating the productivity of assembling reinforcement is proposed. Firstly, a detector that can accurately distinguish various entities related to assembling reinforcement based on CenterNet is established. DLA34 is selected as the backbone. The mAP reaches 0.9682, and the speed of detecting a single image can be as low as 0.076 s. Secondly, the trained detector is used to detect the video frames, and images with detected boxes and documents with coordinates can be obtained. The position relationship between the detected work objects and detected workers is used to determine how many workers (***N***) have participated in the task. The time (***T***) to perform the process can be obtained from the change of coordinates of the work object. Finally, the productivity is evaluated according to ***N*** and ***T***. The authors use four actual construction videos for validation, and the results show that the productivity evaluation is generally consistent with the actual conditions. The contribution of this research to construction management is twofold: On the one hand, without affecting the normal behavior of workers, a connection between construction individuals and work object is established, and the work productivity evaluation is realized. On the other hand, the proposed method has a positive effect on improving the efficiency of construction management.

Keywords: construction engineering; construction management; work productivity; computer vision; deep learning

check for updates

Citation: Li, J.; Zhao, X.; Zhou, G.; Zhang, M.; Li, D.; Zhou, Y. Evaluating the Work Productivity of Assembling Reinforcement through the Objects Detected by Deep Learning. *Sensors* **2021**, *21*, 5598. https://doi.org/10.3390/s21165598

Academic Editors: Hoon Sohn and Amir H. Alavi

Received: 21 June 2021
Accepted: 17 August 2021
Published: 19 August 2021

1. Introduction

Construction sites usually consist of various elements such as personnel, machinery, materials, method, and environment. At the same time, construction sites are also characterized by multiple work types, simultaneous operation of various large-scale pieces of machinery and complex construction environments. Any problem in any links mentioned above may lead to hidden construction quality problems and threaten structural safety and personal safety. Therefore, it is necessary to comprehensively supervise construction sites' condition in real-time [1]. Work productivity is an essential factor in engineering construction, which directly affects each process' completion time and quality. However, limited by human resources and time costs, most construction parties only control the project's progress at a macro level. They do not supervise and record the work productivity of each process in detail.

In the field of engineering construction, most traditional monitoring methods are based on manual monitoring. However, manual supervision often fails to cover every aspect, preventing complete and timely notification of hazardous conditions, irresponsible and irregular worker behavior, sub-standard work quality, and inefficient task execution at construction sites. Therefore, more supervisors are needed, which increases management costs. As a result, many advanced technologies have been introduced into engineering construction, such as the application of acceleration sensors for structural health monitoring [2–4] and workers' activity recognition [5–8]. However, contact sensors may cause some inconvenience to construction when performing practical applications in engineering. Therefore, an efficient way is to use a non-contact monitoring sensor with high accuracy using video and image signals. In recent years, artificial intelligence and computer vision have accelerated, showing new features such as deep learning, cross-border integration, human–machine collaboration, the openness of group intelligence, and autonomous control [9]. Related deep learning-based monitoring methods have been applied to medical diagnosis [10–12], food inspection [13–15], vehicle identification [16–18], and structural health monitoring [19–22], providing the possibility to solve problems related to engineering construction.

The convolutional neural network (CNN) is the most common deep learning network, which originated from the handwritten numbers recognition problem proposed by Lecun in the 1990s [23]. By optimizing the configuration of convolution and pooling layers, many CNNs have been developed, including AlexNet [24], ZFNet [25], VGGNet [26], GoogleNet [27], and ResNet [28]. An effective way to improve CNNs is to increase the number of layers. In this way, the approximate structure of the objective function can be obtained using the increased nonlinearity, and better features can be obtained. Figure 1 illustrates the operation principle of convolution and pooling layers in a CNN. Several object detection methods based on CNNs that can be applied to different scenes have been developed. Currently, there are various object detection algorithms, including CenterNet [29] and Faster R-CNN [30], such as Mask R-CNN [31] and FCN [32] for object segmentation, and YOLO [33–36], SSD [37] and MobileNet [38,39] for fast detection of mobile devices.

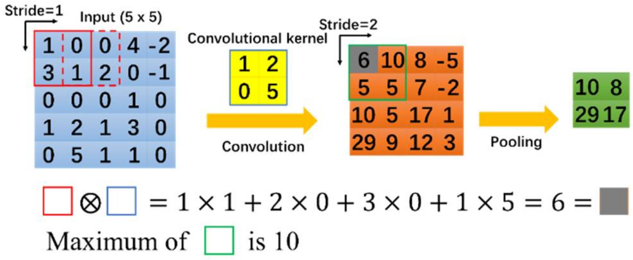


Figure 1. Convolution and pooling operations.

The urgent need to solve practical problems in engineering construction and the growing maturity of deep learning technology have contributed to the rapid development of computer vision technology in engineering construction in recent years. A large amount of literature has focused on this issue.

The first is construction individual-related issues, which receive more attention by recognizing workers' activities and usage of personal protective equipment (PPE). In activities recognition: Luo and Li et al. [40–42] used various computer vision algorithms for construction worker activity recognition; Cai et al. [43] and Liu et al. [44] also carried out computer vision-based approaches for construction activities' recognition. Cai used a two-step LSTM (long short-term memory network), while Liu combined computer vision and natural language processing methods; Han et al. [45], Yu et al. [46] and Yang et al. [47] extracted workers' joint coordinate to recognize the activity and judge the safety status. In the aspect of PPE's usage: Park et al. [48], Fang et al. [49], and Wu et al. [50] introduced different computer vision-based methods for hardhat detection; Fang et al. [51] and Tang et al. [52] achieved PPE usage detection not limited to hardhats.

The second is material-related issues, Zhang and Zhao et al. [53,54] presented a bolt looseness detection method based on MobileNet. Cha et al. [55–58] used deep learning technology based on convolutional neural networks to complete the identification and location of surface cracks in concrete structures, the volume measurement of surface corrosion on steel structures, and the volume measurement of concrete spalling damage. Concrete surface defect identification is also an issue that has been studied frequently in recent years, and representative ones are Xu et al. [59], G Li et al. [60], S Li et al. [61], Miao et al. [62] and others. However, most objects covered in the literature mentioned above are materials of existing built structures, which are not strictly speaking construction materials. These studies are more focused on damage detection of building structures. In the aspect of construction materials, Li et al. [63] proposed a YOLOv3-based method for counting rebars. He et al. [64] introduced an object detection framework called Classification Priority Network for defect detection of hot-rolled steels. Zhou et al. [65] described an approach to analyze concrete pore structure based on deep learning.

The third is machinery-related issues, Kim et al. [66] used unmanned aerial vehicles and monitored mobile machinery devices on a construction site remotely based on the YOLOv3 algorithm. Roberts et al. [67] combined unmanned aerial vehicles and image recognition technology to track a crane on a construction site and estimated the three-dimensional posture. Slaton et al. [68] introduced a method to recognize activities of roller compactor and activator by using a convolutional recurrent network. Yang et al. [69] proposed a video monitoring method to evaluate the working state of a tower crane. Yang et al. [70] successfully identified the safe distance between the hook and the worker using a monitoring camera installed on a tower crane.

These studies have contributed to taking a significant step forward in introducing computer vision technologies to construction engineering. However, there are still some limitations:

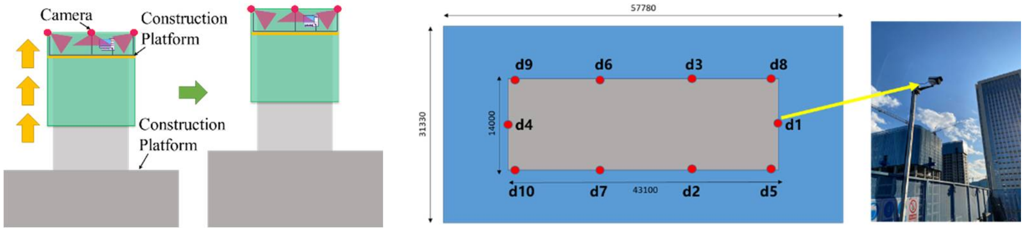
- In the case of individuals and machinery, most extant computer vision-based approaches focus only on safety monitoring and activity recognition. No literature has been found to use computer vision to analyze their work productivity in some dynamic processes.
- In terms of material, no studies have focused on the changes in materials during dynamic construction, and no studies have connected them to individual work and work productivity.
- Although we have not found studies addressing work productivity evaluation, the large number of successful applications of deep learning and computer vision in engineering construction illustrate their potential to assist in filling this research gap in work productivity evaluation.

When evaluating work productivity, it is necessary to know the time consumed and the number of workers involved in the task, and it will take much time to use manual methods to make statistics. To address the above issues, the authors select construction processes of assembling column reinforcement (ACR) and assembling beam reinforcement (ABR) as research cases and propose a new computer vision-based method for work productivity evaluation. Firstly, we train a detector that can accurately distinguish various assembling reinforcement-related entities using the video images collected from on-site surveillance cameras. An anchor-free center point estimation network (CenterNet) is adopted, which can have a good detection speed without loss of accuracy. Secondly, we determine the number of workers who participated in the ACR\ABR task in every moment according to the position relationship between the detected work object and the detected workers to establish a connection between the workers and construction materials. Finally, we record the change of coordinates of the work object in the video and evaluate the work productivity by combining the time consumed and the number of participants. In this article, computer vision technology is used to realize construction activity recognition of ACR\ABR work, and the number of workers participating in the task can be judged. Additionally, using the results output by CenterNet, the productivity evaluation of the ACR\ABR process is realized. Final inspection documents, tables, and work productivity images can be used for project managers to view the work details of this process more intuitively and quickly. The rest of this paper is organized as follows. Section 2 describes the proposed method in detail. Section 3 describes the establishment of the CenterNet model. Section 4 reports the evaluation tests based on construction video clips. Section 5 is the comparison. Sections 6 and 7 outline the discussion of the results and conclusions, respectively.

2. Methods

2.1. Preparation of Dataset

In this paper, images used to train the CenterNet model are from the construction site of a super high-rise project in Donggang Business District, Dalian City, Liaoning Province, China. The structural type of the project is frame–core wall structure, and the slip-form construction technology, as shown in Figure 2a, is applied. With the increase in the storey, the fixed surveillance cameras installed on the construction site will not be able to record the conditions of the construction platform. Therefore, according to Figure 2b, in this research, several cameras that can continuously record construction videos without being disturbed by the increase in the storey were arranged at the edge guardrail.



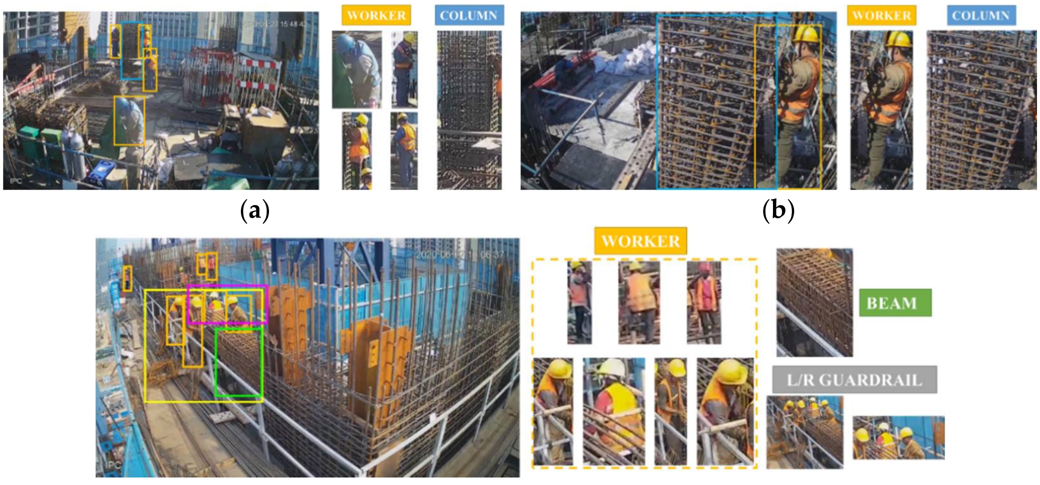
(**a**) (**b**)

Figure 2. Detail of the construction project: (**a**) Front elevation diagram of slip-form construction; (**b**) Camera layout.

The cameras selected are all Dahua brand, with 4.0 megapixels and a focal length of 4 mm. All cameras were connected to a switch through network cables, and then the switch transmitted video signals to a PC. The whole video recording system was equipped with a special power supply module to ensure continuous video recording. Four construction video clips from cameras No. 2, No. 6, and No. 9 were extracted. In the videos, workers assemble beam reinforcements (ABR) and assemble column reinforcements (ACR). The behavior studied in this paper is the installation and binding of stirrups after the longitudinal reinforcement is installed. From the videos, 1051 static images were intercepted. The images were not randomly intercepted to better show the changes in construction materials throughout the process; the beginning, proceeding, and ending phases of the task account for 20%, 60%, and 20% of the dataset, respectively. Table 1 shows the source of images in detail. The dataset contains five categories: workers performing construction tasks (“worker”), parts indicating the assembled reinforcement in beams and columns (“beam” and “column”), and guardrails to assist in determining the location of workers (“Lguardrail” and “Rguardrail”). Figure 3 shows some examples of the dataset.

Table 1. Details of the dataset.

Camera Number	D02	D06	D09	D09
data	23 April 2020	23 April 2020	20 June 2020	27 June 2020
task	ACR	ACR	ABR	ABR
number of images	227	286	265	273



(**a**) (**b**)

(**c**)

Figure 3. Detail of the dataset: (**a**) Image from D02, (**b**) Image from D06, (**c**) Image from D09.

2.2. CenterNet

In the initial years of object detection, anchor-based algorithms [30–32,34–39] dominated. Anchor boxes are essentially a kind of candidate box, and after designing anchor boxes of different scales, CNNs are trained to have the ability to classify the candidate boxes. Eventually, it can distinguish whether the candidate box contains objects and what objects are contained in it. Among various anchor-based algorithms, the two-stage-based have higher accuracy, but it takes more time and computing power to generate candidate boxes in the prediction stage. A one-stage-based algorithm has a faster speed than a two-stage-based algorithm, but accuracy is often lower. In recent years, some scholars have gradually started to study the anchor-free method, which directly eliminates the step of anchor boxes.

CornerNet [71] is the first to introduce the method of predicting detected boxes through key points, which first predicts the two corner points of the rectangular box and then regresses the rectangular detected box. CenterNet takes this idea, but the difference is that it regresses detected boxes through the center point. Figure 4 shows the structure of CenterNet. In the prediction stage, input image is resized into 512 × 512 × 3, and the backbone network is used for feature extraction (DLA34 is chosen for the backbone network in this paper). Then, the 128 × 128 × 256 feature map obtained by down-sampling is predicted, and the heatmap, size (w and h), and offset are obtained. The specific way to extract the detected box is to use 3 × 3 max pooling for the heatmap, check whether the value of the current point is larger than the value of the surrounding eight neighboring points, and then filter from the eligible points. Combined with the offset, the center point can be obtained. Finally, coordinates of four corner points of the detected box are calculated from Equation (1) [29]:

$$[X_{min}, X_{max}, Y_{min}, Y_{max}] = [(X_c - w/2), (X_c + w/2), (Y_c - h/2), (Y_c + h/2)] \tag{1}$$

in which w is the width of the detected box, and h is the height of the detected box. X_c and Y_c are coordinate values of the center point.

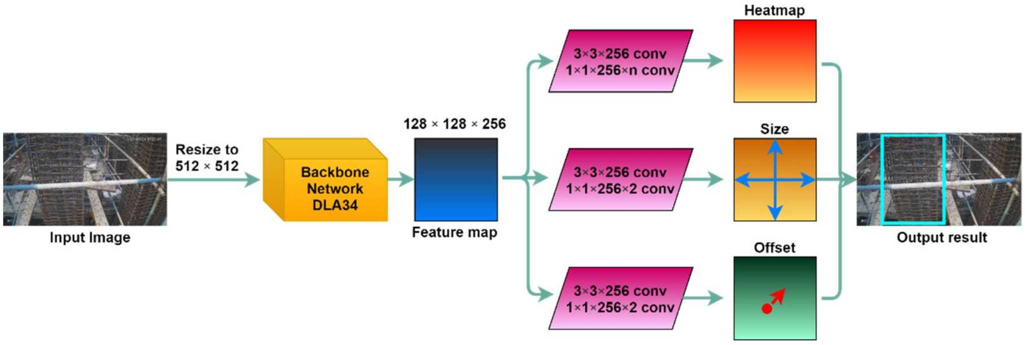


Figure 4. Flowchart of CenterNet.

In the last two years, some scholars have used this algorithm to solve practical problems, such as pedestrian detection [72], phoneme recognition [73], foreign object detection [74], and ship detection [75], and achieved satisfactory recognition accuracy and speed. Although deep learning is still evolving rapidly, most current research based on object detection still only implements the output of detected boxes, i.e., an account of what class of object is detected in the image. Information embedded in detected boxes is not exploited in-depth, and the three limiting points proposed in the first section are not addressed. Combined with the excellent balance of detection speed and accuracy shown by CenterNet in [72–75], it is chosen to detect objects related to the ABR\ACR process.

2.3. Evaluation for the Productivity

Before evaluation of ACR\ABR productivity, the performance of the model must be guaranteed. We trained several CenterNet-based models and selected the one with the highest mAP.

After identifying the various types of ACR\ABR-related objects, it is necessary to find out the relationship between them according to the position of detected boxes. Figure 5 shows the positional relationship between "worker" and "column" at a specific ACR process moment. It can be seen that three of the four workers in the figure above are performing the ACR process. In the figure below, two of the three workers are performing the ACR process. After verifying 100 images, it was found that the coordinates of workers' detected boxes performing this process all satisfied the three conditions from Equations (2)–(4).

$$Y_{worker\_min} \geq Y_{column\_min} \tag{2}$$

$$|X_{worker\_max} - X_{worker\_min}| \geq 0.3|X_{column\_max} - X_{column\_min}| \tag{3}$$

$$|(X_{worker\_max} + X_{worker\_min})/2 - (X_{column\_max} + X_{column\_min})/2| \leq 2.5|X_{worker\_max} - X_{worker\_min}| \tag{4}$$

in which Y_{worker_min} and Y_{column_min} represent the minimum and maximum values of the coordinates of "worker's" detected box and "column's" detected box in the Y-axis direction, respectively. X_{worker_max} and X_{worker_min} are the maximum and minimum values of the "worker's" detected box, respectively. X_{column_max} and X_{column_min} are the maximum and minimum values of "column's" detected box, respectively.

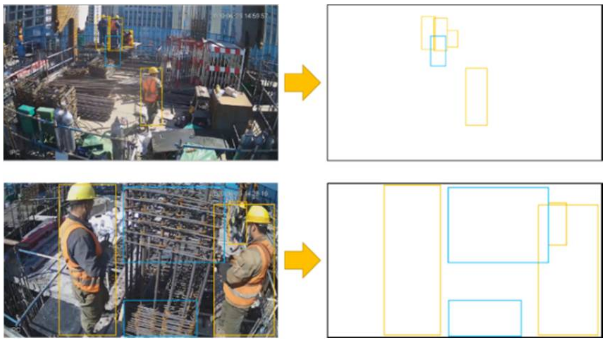


Figure 5. Detected boxes during ACR process.

Equation (2) indicates that the height of the worker in Y-axis direction is not lower than the column when he performs this work, Equation (3) limits the distance of the worker and the column from the camera, and Equation (4) indicates that the worker should surround the processed object.

Figure 6 shows the position of each detection box at a given moment during the ABR process, which contains four categories, namely "Lguardrail", "Rguardrail", "beam", and "worker". From the coordinates of "Lguardrail" and "Rguardrail", a quadrilateral ABCD can be obtained. When the worker performs the ABR process, his position is inside the guardrail. The detection image shows the phenomenon presented in Equation (5) that the worker's detected box intersects the quadrilateral:

$$A_{ABCD} \cap A_{worker} \neq 0 \tag{5}$$

in which A_{ABCD} refers to the area of quadrilateral ABCD. A_{worker} refers to the area of worker detected box.

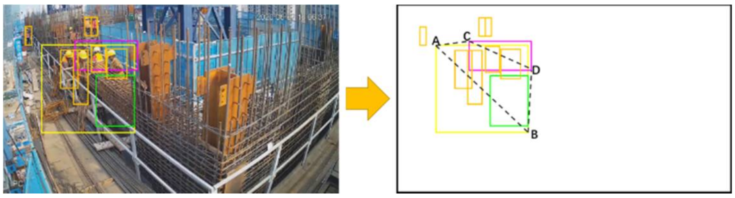


Figure 6. Detected boxes during ABR process.

This article uses the above Equations (2)–(5) as the basis for identifying the ACR\ABR process. After identifying the ACR\ABR process, the work productivity evaluation can be carried out using the obtained detected results. When a worker performs the ACR\ABR process, the number of stirrups will increase in one direction, and the change of coordinates is shown in the detection image. In this paper, the height $\boldsymbol{H}$ of the "column's" detected box in the ACR process and the diagonal length $\boldsymbol{L}$ of the "beam's" detected box in the ABR process are selected as indicators. When these two values do not change significantly within ten minutes, and no more workers perform the ABR/ACR process, the duration of the process is recorded. The work productivity can be calculated from Equation (6):

$$\boldsymbol{P} = \boldsymbol{A}/\boldsymbol{T};\ \boldsymbol{P_n} = \boldsymbol{P}/\boldsymbol{N} \tag{6}$$

in which $\boldsymbol{P}$ represents the work productivity of the ACR\ABR process. $\boldsymbol{P_n}$ is workers' average work productivity. $\boldsymbol{A}$ is an amplification factor, which is taken as 10,000 in this paper. $\boldsymbol{T}$ refers to the duration to perform the task. $\boldsymbol{N}$ refers to the average number of workers performing the ACR\ABR process. The usual method used to calculate work productivity is to divide workload by work duration. In this study, the workload is the same when performing the same kind of process, so the inverse of $\boldsymbol{T}$ can be directly considered as the work productivity. However, as a variable representing time, the value of $\boldsymbol{T}$ may be tremendous. For example, when $\boldsymbol{T}$ = 5000 s, $\boldsymbol{P}$ is only 0.0002. Too many digits after the decimal point may be unfavorable for analysis and comparison. Therefore, to better present the evaluation results, the method used in this paper is to multiply by an amplification factor $\boldsymbol{A}$. By dividing the calculated $\boldsymbol{P}$ by $\boldsymbol{N}$, we can obtain the $\boldsymbol{P_n}$. Figure 7 shows the flowchart of the proposed method.

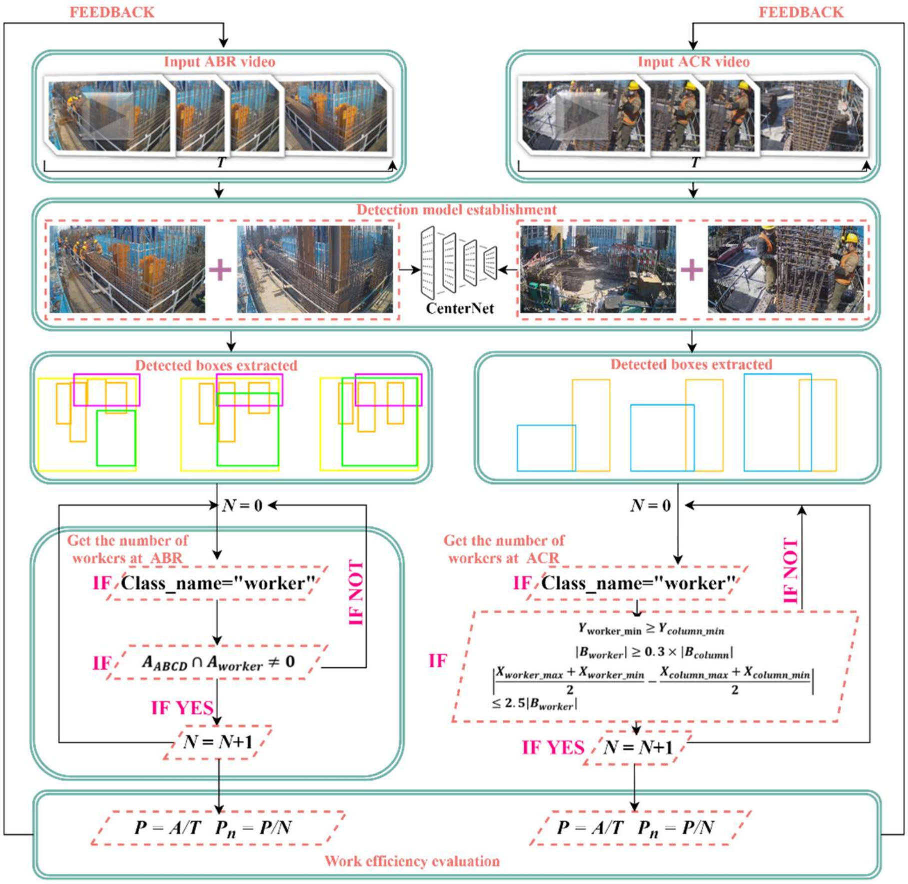

FEEDBACK
FEEDBACK
Input ABR video
T
Input ACR video
T
Detection model establishment
CenterNet
Detected boxes extracted
Detected boxes extracted
N = 0
N = 0
Get the number of workers at ABR
IF Class_name="worker"
IF NOT
IF $A_{ABCD} \cap A_{worker} \neq 0$
IF YES
N = N+1
Get the number of workers at ACR
IF Class_name="worker"
IF NOT
IF
$Y_{worker\_min} \geq Y_{column\_min}$
$|B_{worker}| \geq 0.3 \times |B_{column}|$
$\left|\frac{X_{worker\_max}+X_{worker\_min}}{2} - \frac{X_{column\_max}+X_{column\_min}}{2}\right| \leq 2.5|B_{worker}|$
IF YES
N = N+1
$P = A/T \quad P_n = P/N$
$P = A/T \quad P_n = P/N$
Work efficiency evaluation

Figure 7. Flowchart of the proposed method.

3. Establishment of CenterNet Model

3.1. Operating Environment and Parameter Settings

In this article, a PC with NVIDIA 1080ti GPU, Core i7-8700 CPU, 16GB RAM, and Pytorch framework is applied for training. In the dataset, the ratio of the number of images in the training set to the test set is 8:2. Eighty percent of the images are used to train the model in the training set, and the remaining twenty are used for validation. Different learning rates (0.0001, 0.0005) and different batch sizes (1, 2, 4, 8, and 16) are combined. Ten detection models are trained, from which the best is selected. Epoch is set to 200 uniformly, and evaluation is performed every five epochs.

3.2. Training Results

Table 2 lists the AP (Average Precision) and mAP (mean Average Precision) of the trained models. Among them, AP represents the recognition accuracy of each category, and mAP is the average value of AP, which represents the overall classification performance of the model. It can be seen from Table 2 that the model with a learning rate of 0.0001 and a batch size of 2 has the best recognition ability when it is trained to the 110th epoch, and its mAP is 0.9682. Figure 8a,b show the training loss and validation loss, respectively. Loss value indicates the deviation of the predicted value from the actual value, and the smaller the value, the lower the deviation. As we can see from the curves, the loss value decreases continuously as the training continues, and the validation loss reaches the minimum value at the 110th epoch. Then, it smooths out at the end, indicating that the model has reached convergence. Eventually, this 110th epoch model is retained for subsequent studies. To verify the feasibility of the model trained in this paper, we also applied three classical anchor-based object detection algorithms, the one-stage SSD and YOLO v3, and the two-stage Faster R-CNN. The mAP values and the time consumed (*t*) to detect a single image are listed in Table 3. It can be seen that the model trained in this paper is 3.125 times faster than the Faster R-CNN, although it is slightly inferior to the Faster R-CNN in terms of accuracy. Compared with YOLO v3, the recognition speed is slower, but the accuracy is higher. In other words, the CenterNet-based detection model has a good balance of speed and accuracy. From the partial detection results listed in Figure 9, each object can be well recognized in the images. Therefore, it is feasible to select this object detection model for the subsequent work productivity evaluation.

Table 2. AP and mAP corresponding to each category.

Learning Rate	Batch Size	AP					Best Epoch	mAP
		Worker	Column	Beam	Lguardrail	Rguardrail		
0.0001	1	0.933	0.894	0.869	0.946	0.959	125	0.9202
0.0005	1	0.766	0.936	0.773	0.965	0.923	115	0.8726
0.0001	2	0.935	0.973	0.943	0.996	0.980	110	0.9682
0.0005	2	0.952	0.957	0.869	0.996	0.991	135	0.953
0.0001	4	0.947	0.979	0.942	0.996	0.955	95	0.9638
0.0005	4	0.933	0.975	0.922	0.998	0.959	125	0.9574
0.0001	8	0.943	0.965	0.951	0.997	0.965	100	0.9642
0.0005	8	0.943	0.969	0.929	0.996	0.941	125	0.9556
0.0001	16	0.936	0.966	0.910	0.995	0.976	100	0.9566
0.0005	16	0.948	0.984	0.927	0.996	0.965	100	0.9551

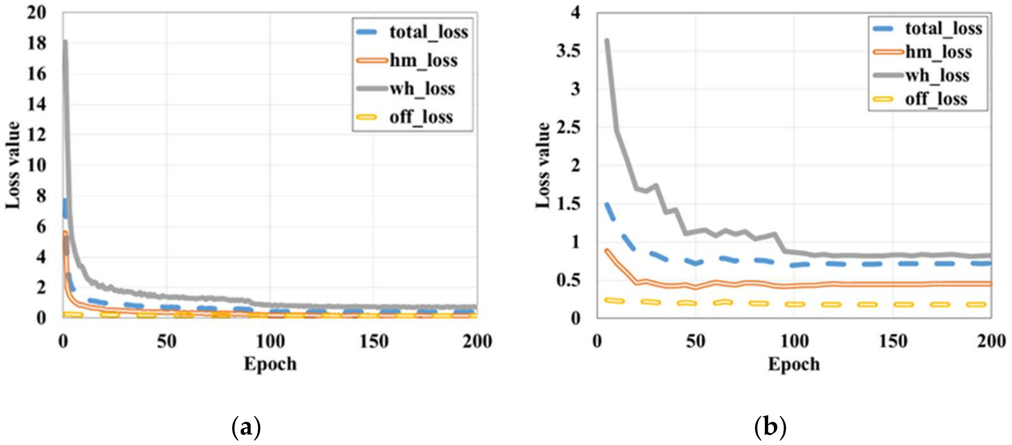


Figure 8. Training and validation losses versus epochs: (**a**) Training loss, (**b**) Validation loss.

Table 3. Comparison between different object detection algorithms.

	mAP	t (s)
Centernet	0.9682	0.076
Faster R-CNN	0.9751	0.240
SSD	0.9282	0.085
YOLO v3	0.9530	0.060

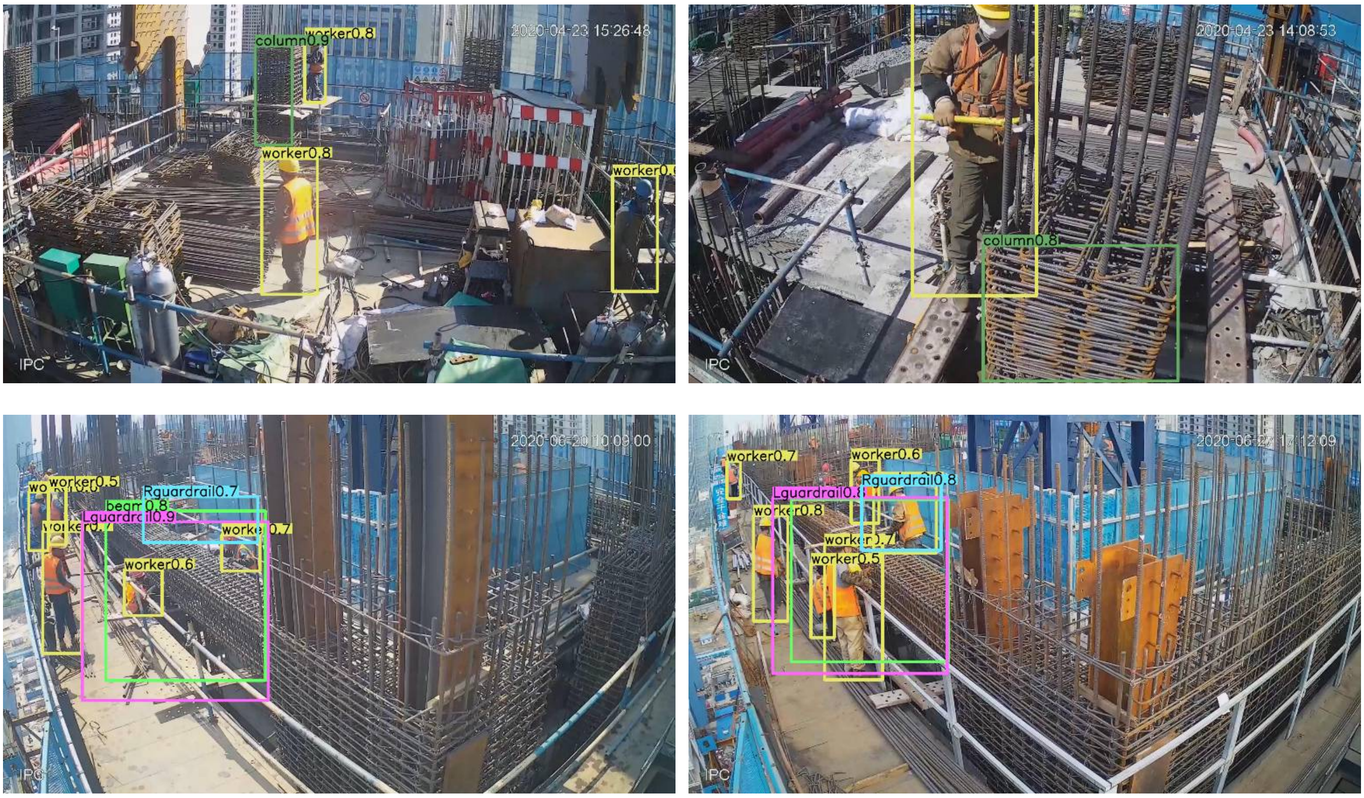


Figure 9. Detection results based on CenterNet.

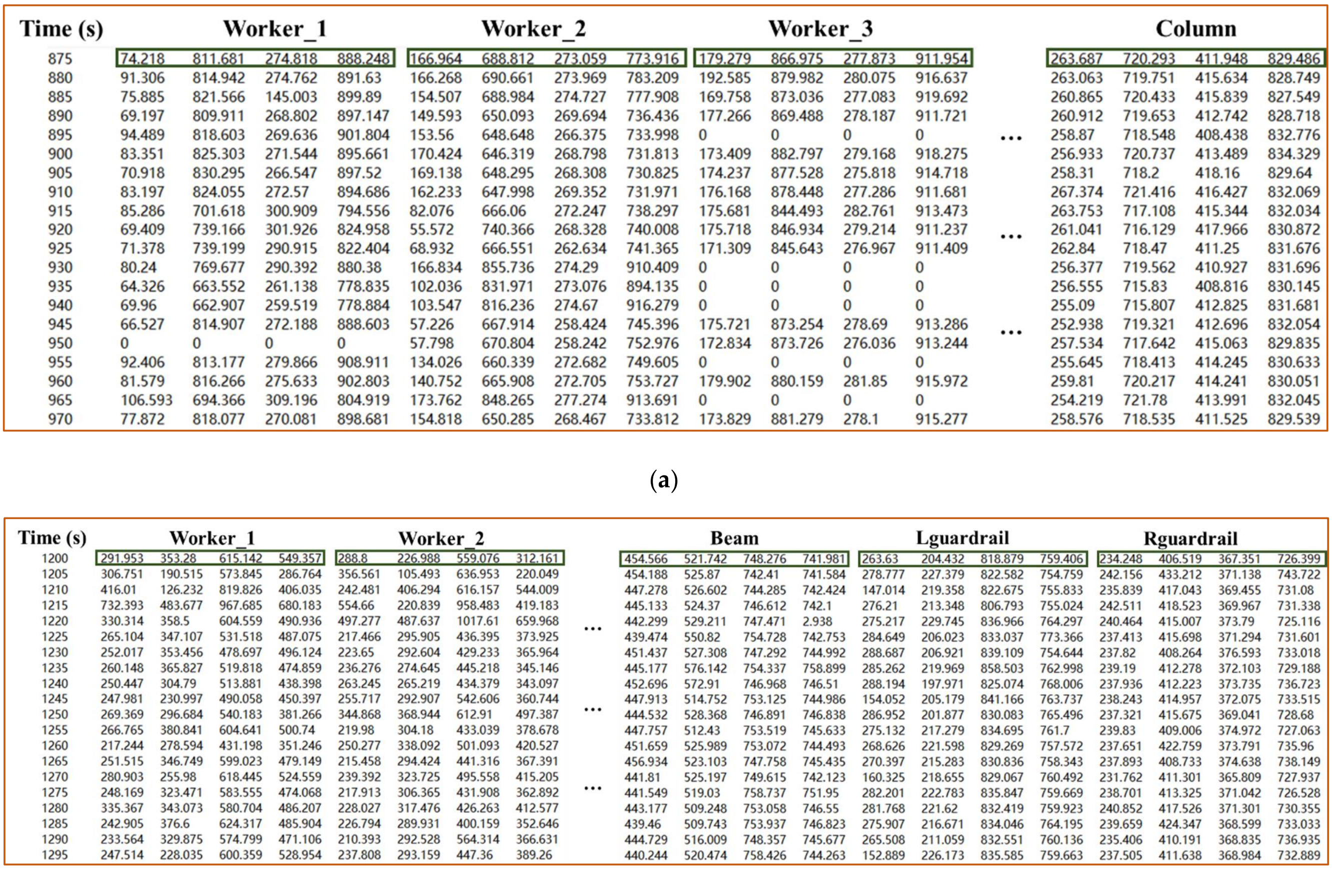
4. Experiment

4.1. ACR\ABR Activity Recognition

After detection results are obtained, the next step is to use the detected box of each object to identify ABR and ACR's activity through Equations (2)–(5) and then determine how many workers have performed these two tasks. To test the method's practical effectiveness, videos described in Section 2.1 are input into the trained CenterNet model, and Table 4 lists the information such as the duration of these videos. The frame rate is 25, which means that there are 25 images per second. If the detection model is used to detect all 25 images per second, it will increase the computer memory consumption. Therefore, the video is detected every five seconds (125 frames) after input into the CenterNet model. Finally, after processing, the detection document is generated, as shown in Figure 10. Inside the document, the category detected at the current moment and the coordinates of detected boxes are recorded in detail, and the four values represent the $\boldsymbol{Y_{min}}$, $\boldsymbol{X_{min}}$, $\boldsymbol{Y_{max}}$, $\boldsymbol{X_{max}}$, respectively.

Table 4. Details of the videos.

Video Number	V01	V02	V03	V04
duration	1 h 46 min 30 s	1 h 58 min 50 s	3 h 04 min 30 s	1 h 39 min 05 s
task	ACR	ACR	ABR	ABR
camera	D02	D06	D09	D09
format	MP4			
frame rate	25 Fps			



Time (s)	Worker_1				Worker_2				Worker_3					Column			
875	74.218	811.681	274.818	888.248	166.964	688.812	273.059	773.916	179.279	866.975	277.873	911.954		263.687	720.293	411.948	829.486
880	91.306	814.942	274.762	891.63	166.268	690.661	273.969	783.209	192.585	879.982	280.075	916.637		263.063	719.751	415.634	828.749
885	75.885	821.566	145.003	899.89	154.507	688.984	274.727	777.908	169.758	873.036	277.083	919.692		260.865	720.433	415.839	827.549
890	69.197	809.911	268.802	897.147	149.593	650.093	269.694	736.436	177.266	869.488	278.187	911.721		260.912	719.653	412.742	828.718
895	94.489	818.603	269.636	901.804	153.56	648.648	266.375	733.998	0	0	0	0	…	258.87	718.548	408.438	832.776
900	83.351	825.303	271.544	895.661	170.424	646.319	268.798	731.813	173.409	882.797	279.168	918.275		256.933	720.737	413.489	834.329
905	70.918	830.295	266.547	897.52	169.138	648.295	268.308	730.825	174.237	877.528	275.818	914.718		258.31	718.2	418.16	829.64
910	83.197	824.055	272.57	894.686	162.233	647.998	269.352	731.971	176.168	878.448	277.286	911.681		267.374	721.416	416.427	832.069
915	85.286	701.618	300.909	794.556	82.076	666.06	272.247	738.297	175.681	844.493	282.761	913.473		263.753	717.108	415.344	832.034
920	69.409	739.166	301.926	824.958	55.572	740.366	268.328	740.008	175.718	846.934	279.214	911.237	…	261.041	716.129	417.966	830.872
925	71.378	739.199	290.915	822.404	68.932	666.551	262.634	741.365	171.309	845.643	276.967	911.409		262.84	718.47	411.25	831.676
930	80.24	769.677	290.392	880.38	166.834	855.736	274.29	910.409	0	0	0	0		256.377	719.562	410.927	831.696
935	64.326	663.552	261.138	778.835	102.036	831.971	273.076	894.135	0	0	0	0		256.555	715.83	408.816	830.145
940	69.96	662.907	259.519	778.884	103.547	816.236	274.67	916.279	0	0	0	0		255.09	715.807	412.825	831.681
945	66.527	814.907	272.188	888.603	57.226	667.914	258.424	745.396	175.721	873.254	278.69	913.286	…	252.938	719.321	412.696	832.054
950	0	0	0	0	57.798	670.804	258.242	752.976	172.834	873.726	276.036	913.244		257.534	717.642	415.063	829.835
955	92.406	813.177	279.866	908.911	134.026	660.339	272.682	749.605	0	0	0	0		255.645	718.413	414.245	830.633
960	81.579	816.266	275.633	902.803	140.752	665.908	272.705	753.727	179.902	880.159	281.85	915.972		259.81	720.217	414.241	830.051
965	106.593	694.366	309.196	804.919	173.762	848.265	277.274	913.691	0	0	0	0		254.219	721.78	413.991	832.045
970	77.872	818.077	270.081	898.681	154.818	650.285	268.467	733.812	173.829	881.279	278.1	915.277		258.576	718.535	411.525	829.539

(**a**)

Time (s)	Worker_1				Worker_2					Beam				Lguardrail				Rguardrail			
1200	291.953	353.28	615.142	549.357	288.8	226.988	559.076	312.161		454.566	521.742	748.276	741.981	263.63	204.432	818.879	759.406	234.248	406.519	367.351	726.399
1205	306.751	190.515	573.845	286.764	356.561	105.493	636.953	220.049		454.188	525.87	742.41	741.584	278.777	227.379	822.582	754.759	242.156	433.212	371.138	743.722
1210	416.01	126.232	819.826	406.035	242.481	406.294	616.157	544.009		447.278	526.602	744.285	742.424	147.014	219.358	822.675	755.833	235.839	417.043	369.455	731.08
1215	732.393	483.677	967.685	680.183	554.66	220.839	958.483	419.183		445.133	524.37	746.612	742.1	276.21	213.348	806.793	755.024	242.511	418.523	369.967	731.338
1220	330.314	358.5	604.559	490.936	497.277	487.637	1017.61	659.968	…	442.299	529.211	747.471	2.938	275.217	229.745	836.966	764.297	240.464	415.007	373.79	725.116
1225	265.104	347.107	531.518	487.075	217.466	295.905	436.395	373.925		439.474	550.82	754.728	742.753	284.649	206.023	833.037	773.366	237.413	415.698	371.294	731.601
1230	252.017	353.456	478.697	496.124	223.65	292.604	429.233	365.964		451.437	527.308	747.292	744.992	288.687	206.921	839.109	754.644	237.82	408.264	376.593	733.018
1235	260.148	365.827	519.818	474.859	236.276	274.645	445.218	345.146		445.177	576.142	754.337	758.899	285.262	219.969	858.503	762.998	239.19	412.278	372.103	729.188
1240	250.447	304.79	513.881	438.398	263.245	265.219	434.379	343.097		452.696	572.91	746.968	746.51	288.194	197.971	825.074	768.006	237.936	412.223	373.735	736.723
1245	247.981	230.997	490.058	450.397	255.717	292.907	542.606	360.744		447.913	514.752	753.125	744.986	154.052	205.179	841.166	763.737	238.243	414.957	372.075	733.515
1250	269.369	296.684	540.183	381.266	344.868	368.944	612.91	497.387	…	444.532	528.368	746.891	746.838	286.952	201.877	830.083	765.496	237.321	415.675	369.041	728.68
1255	266.765	380.841	604.641	500.74	219.98	304.18	433.039	378.678		447.757	512.43	753.519	745.633	275.132	217.279	834.695	761.7	239.83	409.006	374.972	727.063
1260	217.244	278.594	431.198	351.246	250.277	338.092	501.093	420.527		451.659	525.989	753.072	744.493	268.626	221.598	829.269	757.572	237.651	422.759	373.791	735.96
1265	251.515	346.749	599.023	479.149	215.458	294.424	441.316	367.391		456.934	523.103	747.758	745.435	270.397	215.283	830.836	758.343	237.893	408.733	374.638	738.149
1270	280.903	255.98	618.445	524.559	239.392	323.725	495.558	415.205		441.81	525.197	749.615	742.123	160.325	218.655	829.067	760.492	231.762	411.301	365.809	727.937
1275	248.169	323.471	583.555	474.068	217.913	306.365	431.908	362.892	…	441.549	519.03	758.737	751.95	282.201	222.783	835.847	759.669	238.701	413.325	371.042	726.528
1280	335.367	343.073	580.704	486.207	228.027	317.476	426.263	412.577		443.177	509.248	753.058	746.55	281.768	221.62	832.419	759.923	240.852	417.526	371.301	730.355
1285	242.905	376.6	624.317	485.904	226.794	289.931	400.159	352.646		439.46	509.743	753.937	746.823	275.907	216.671	834.046	764.195	239.659	424.347	368.599	733.033
1290	233.564	329.875	574.799	471.106	210.393	292.528	564.314	366.631		444.729	516.009	748.357	745.677	265.508	211.059	832.551	760.136	235.406	410.191	368.835	736.935
1295	247.514	228.035	600.359	528.954	237.808	293.159	447.36	389.26		440.244	520.474	758.426	744.263	152.889	226.173	835.585	759.663	237.505	411.638	368.984	732.889

(**b**)

Figure 10. Coordinates of boxes at each moment: (**a**) ACR, (**b**) ABR.

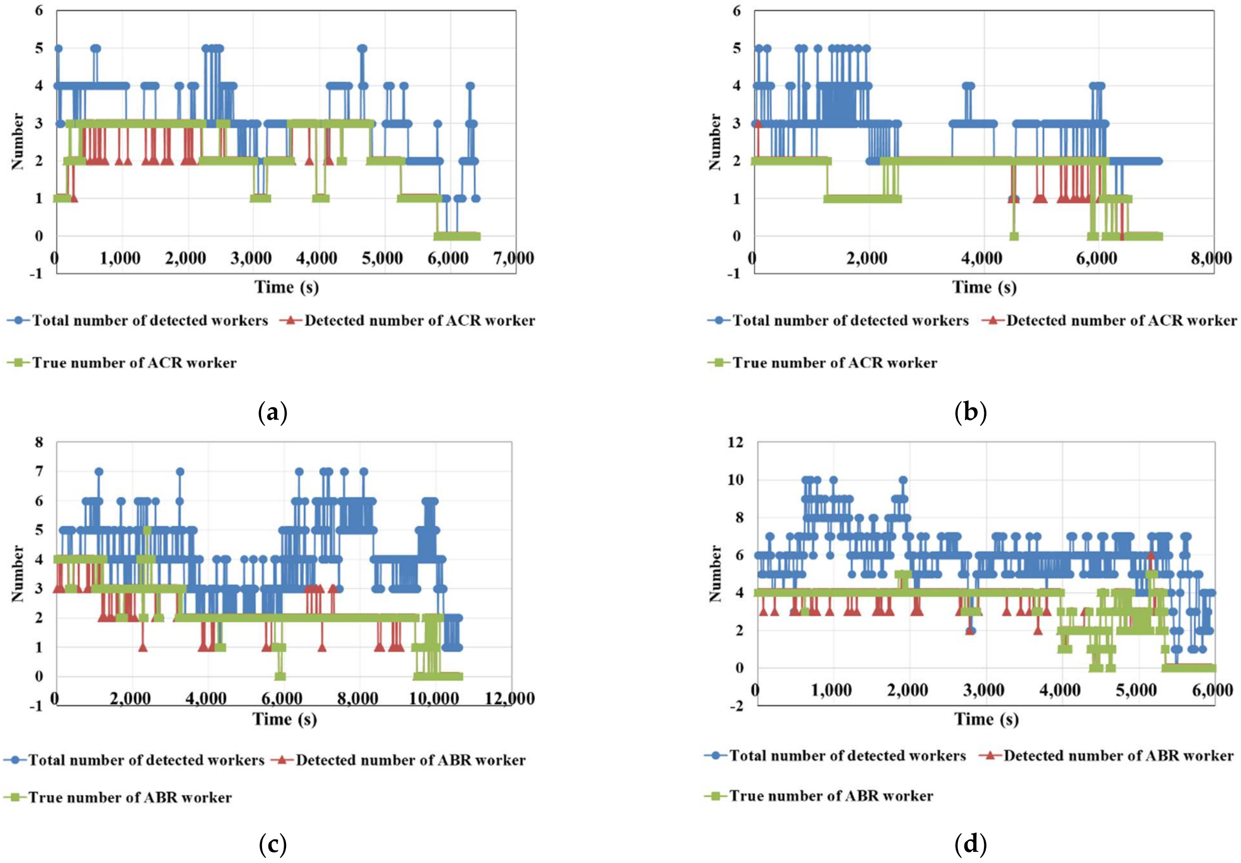
After obtaining the coordinate values shown in Figure 10, the first step is to distinguish which process this video belongs to. When $\Delta \boldsymbol{L} > \Delta \boldsymbol{H}$, an ABR process is executed in the video. When $\Delta \boldsymbol{L} < \Delta \boldsymbol{H}$, that is an ACR process. $\Delta \boldsymbol{L}$ and $\Delta \boldsymbol{H}$ are calculated as shown in Equation (7):

$$\Delta \boldsymbol{L} = \boldsymbol{L}_{\boldsymbol{end}} - \boldsymbol{L}_{\boldsymbol{start}};\ \Delta \boldsymbol{H} = \boldsymbol{H}_{\boldsymbol{end}} - \boldsymbol{H}_{\boldsymbol{start}} \quad (7)$$

in which $\boldsymbol{L}_{\boldsymbol{end}}$ and $\boldsymbol{L}_{\boldsymbol{start}}$ refer to the length of the diagonal line of the "beam's" detected box at the end of the task and at the beginning of the task, respectively. $\boldsymbol{H}_{\boldsymbol{end}}$ and $\boldsymbol{H}_{\boldsymbol{start}}$ represent the height of the "column's" detected box at the end of the process and the height at the beginning, respectively. Since ABR and ACR work will not be performed simultaneously during the construction of the building structure in this paper, $\Delta \boldsymbol{H}$ is usually 0 when performing ABR tasks, and $\Delta \boldsymbol{L}$ is usually 0 when performing ACR tasks.

For "worker's" coordinates at each moment in the document, ACR\ABR activity recognition is performed according to the flow in Figure 7. For the number of workers $\boldsymbol{N}$ participating in the ABR/ACR task at each moment, an initial value $\boldsymbol{N} = 0$ is assigned. Equation (5) is executed for the coordinates of each "worker" in the ABR task, and $\boldsymbol{N}$ is added by 1 when a worker is detected satisfying the condition. For each "worker's" coordinate in the ACR task, we carried out the Equations (2)–(4). $\boldsymbol{N}$ is added to 1 for each time when the condition is satisfied at this moment. By iterating through each line of the detection document in this way, the number of workers performing the ACR\ABR process at each moment is obtained.

The authors counted the total number of detected workers at each moment, the number $\boldsymbol{N}_{\boldsymbol{detected}}$ of workers involved in ABR\ACR work detected by the method we proposed in Section 2.3, and the true number $\boldsymbol{N}_{\boldsymbol{true}}$ of workers involved in ABR\ACR work. Figure 11 presents the three values above, and it can be seen that $\boldsymbol{N}_{\boldsymbol{true}}$ and $\boldsymbol{N}_{\boldsymbol{detected}}$ have a good overlap. Table 5 lists the number of moments when the count is correct ($N_{true} = N_{detected}$) and the number of moments when the count is incorrect ($N_{true} \neq N_{detected}$). The errors in V01 and V02 are mainly due to workers entering the blind area of the camera, while the errors in V03 and V04 are mainly due to some workers being blocked by other workers and unrelated individuals entering the working area of the ABR.



(**a**) (**b**)

(**c**) (**d**)

Figure 11. Curves of the number of detected workers: (**a**) V01, (**b**) V02, (**c**) V03, (**d**) V04.

Table 5. Correct moments and incorrect moments.

Video Number	V01	V02	V03	V04
correct	1203	1426	2095	1098
incorrect	75	81	147	94
accuracy	0.941	0.946	0.934	0.921

To test the accuracy of ABR\ACR activity recognition, we calculated N_{true_total} and $N_{detected\_total}$ according to Equation (8), and the calculated results are listed in Table 6:

$$N_{true\_total} = \sum_{t=0}^{t=end} N_{true};\ N_{detected\_total} = \sum_{t=0}^{t=end} N_{detected} \tag{8}$$

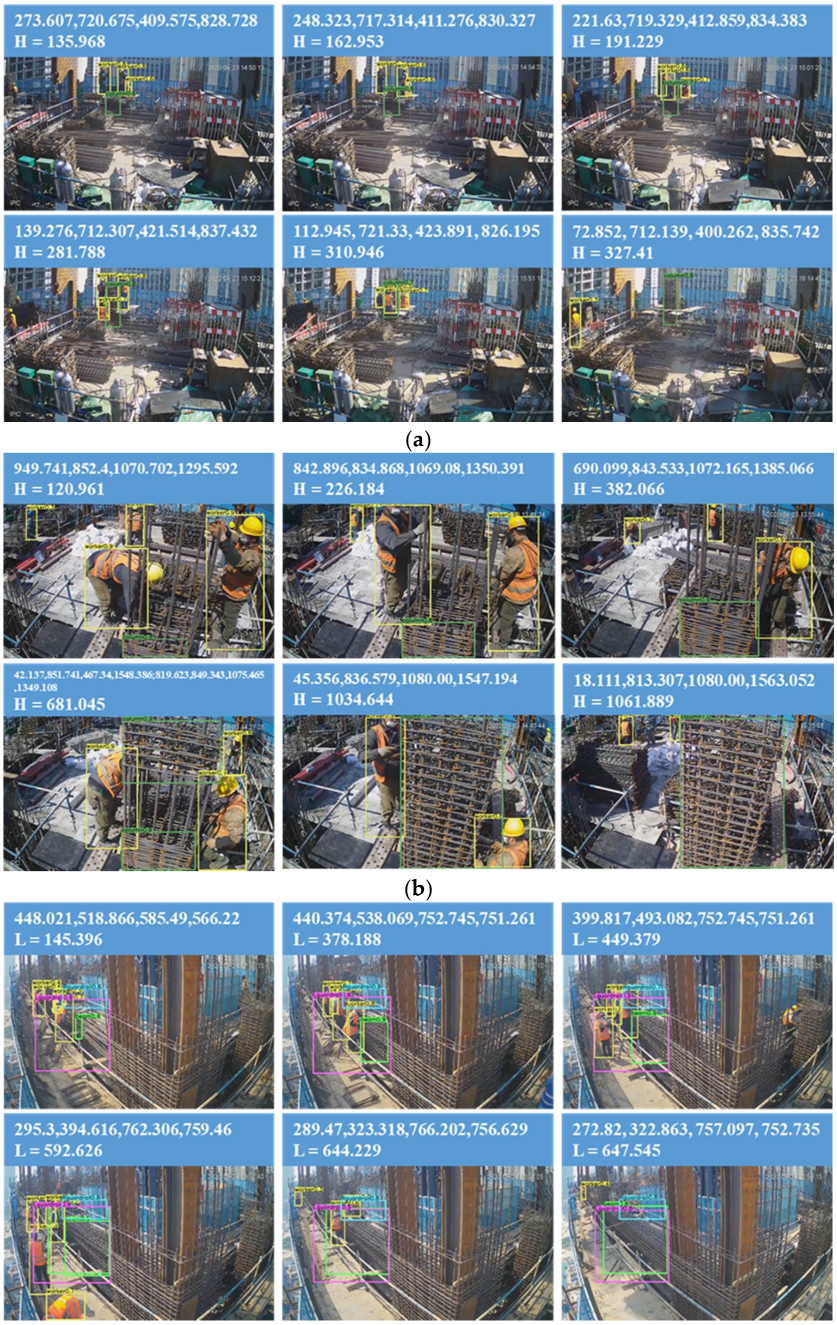
Table 6. Number of detected workers who participate in ABR\ACR process.

Video Number	V01	V02	V03	V04
N_{true_total}	2721	2257	4671	3885
$N_{detected\_total}$	2629	2176	4560	3799
accuracy	0.966	0.964	0.976	0.977

The N_{true_total} and $N_{detected\_total}$ in Equation (8) refer to the sum of N_{true} and $N_{detected}$ for each moment from the start to the end of the process, respectively. From the results in Tables 5 and 6, it can be seen that the moments in which the number of workers involved in the ABR\ACR process is correctly accounted for in 93.4% of all moments, and the average accuracy of ABR\ACR's activity recognition reached 0.970. The errors are mainly caused by workers being completely obscured and irrelevant workers staying in the work area.

4.2. Evaluation for Work Productivity

Figure 12 shows some of the images when the CenterNet-based model is used to detect these videos. The trained detection model accurately reflects the change in the coordinates of the working object during the ACR\ABR process, and both the ***H*** and ***L*** values gradually increase with the work.

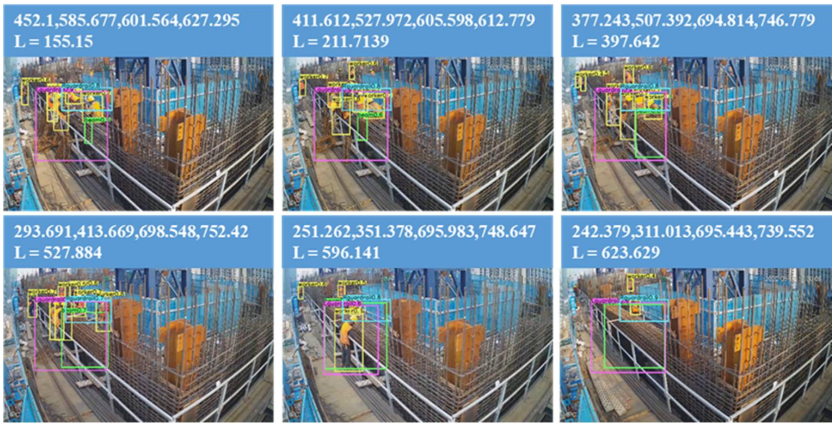


(**a**)

(**b**)

(**c**)

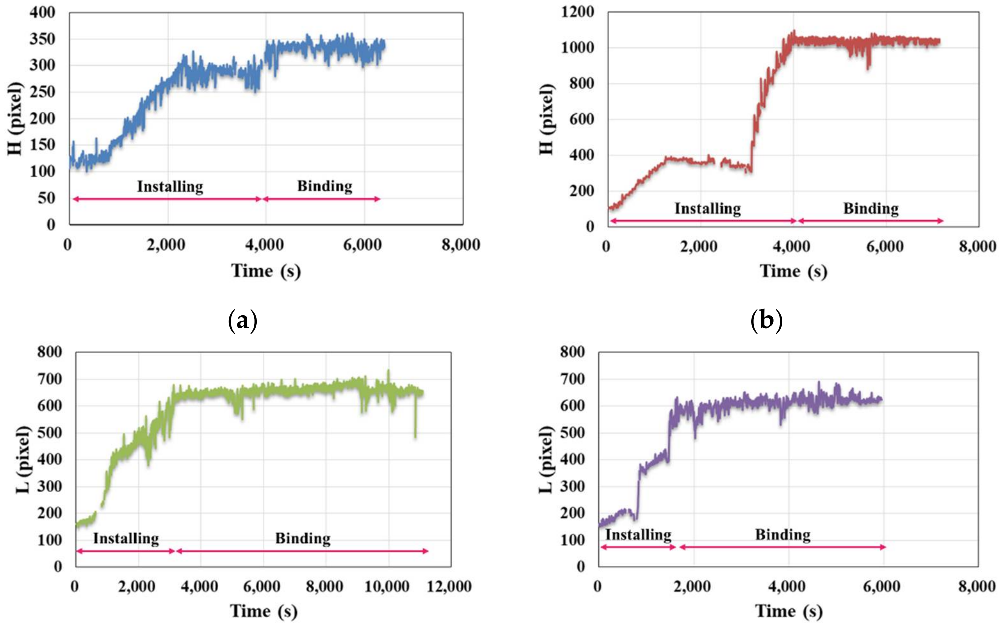
Figure 12. *Cont.*



(**d**)

Figure 12. Test images of the four videos: (**a**) V01, (**b**) V02, (**c**) V03, (**d**) V04.

To further illustrate this phenomenon, the coordinates of "column" and "beam" are extracted from the detection file shown in Figure 10, and the curves of these two values versus time shown in Figure 13 are plotted. Although there are fluctuations in the curve due to errors, the trend accurately reflects the actual condition of the ACR\ABR construction process. The information contained is sufficient to assist in evaluating work productivity. It also shows, from the side, that the model trained by applying the dataset prepared in Section 2.1 can accurately reflect the changing trend of construction materials. As shown in Figure 13, the process can be divided into two stages: first is the Installing stage, in which the workers assemble all the stirrups into the final finished shape. Then, in the Binding stage, the workers use thin iron wires to fix the stirrups to the longitudinal bars. In Figure 13a,b, there is a phenomenon that the ***H*** value first rises and then falls in the first stage. It is caused by the fact that workers push them down after the stirrups are assembled to a certain height, and the spacing between them becomes smaller. After the longitudinal force reinforcement is connected, the assembly work is resumed until it is formed.



(**a**) (**b**)

(**c**) (**d**)

Figure 13. Curves of the H\L versus time: (**a**) V01, (**b**) V02, (**c**) V03, (**d**) V04.

To determine $\boldsymbol{T}$ for each stage, the method shown in Figure 14 is adopted, i.e., the point with the smallest value among the last five points is selected, and then a line with slope 0 is made to the left. The point to the right of the last intersection is taken as the dividing point between Installing and Binding.

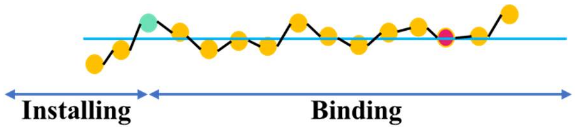


Figure 14. Determination of the dividing point between "installing" and "binding".

From the data in Figure 11, the average number of workers involved in each stage of the ABR\ACR process can be calculated, and from Figures 13 and 14, $\boldsymbol{T}$ for each stage can be obtained, and from these conditions, $\boldsymbol{P}$ and $\boldsymbol{P_n}$ can then be calculated from Equation (6). The final results are shown in Table 7.

Table 7. Evaluation of work productivity.

Video Number	V01	V02	V03	V04
$\boldsymbol{T}$ (s)	6410	7135	11,075	5960
$\boldsymbol{T_{installing}}$ (s)	3870	4050	3210	1565
$\boldsymbol{T_{binding}}$ (s)	2540	3085	7865	4395
$\boldsymbol{N}$	2.06	1.55	2.13	3.19
$\boldsymbol{N_{installing}}$	2.41	1.70	3.19	3.95
$\boldsymbol{N_{binding}}$	1.53	1.34	1.68	2.93
$\boldsymbol{P}$	1.560	1.402	0.903	1.672
$\boldsymbol{P_{installing}}$	2.584	2.470	3.115	6.389
$\boldsymbol{P_{binding}}$	3.937	3.241	1.271	2.275
$\boldsymbol{P_n}$	0.757	0.904	0.424	0.524
$\boldsymbol{P_{n\_installing}}$	1.072	1.453	0.976	1.617
$\boldsymbol{P_{n\_binding}}$	2.57	2.419	0.757	0.776

When calculating $\boldsymbol{P}$ and $\boldsymbol{P_n}$ in Table 7, a uniform $\boldsymbol{A}$-value is used as the $\boldsymbol{T_{installing}}$ is different from $\boldsymbol{T_{binding}}$, which leads to a significant difference in $\boldsymbol{P}$ and $\boldsymbol{P_n}$ between different stages of the same ABR\ACR process. The workload between different stages is different, so the $\boldsymbol{P}$ and $\boldsymbol{P_n}$ in Table 7 are not applicable for vertical comparison but can only be used for horizontal comparison. It can be seen that the $\boldsymbol{P}$ of V01 is 11.3% higher than that of V02. It is because its $\boldsymbol{N}$ is also greater than V02, which is 32.9%. The advantage of V01's $\boldsymbol{P}$ compared to V02 is not as big as the difference between $\boldsymbol{N}$. This is also reflected in $\boldsymbol{P_n}$'s value, which is 16.3% lower for V01 than V02. When viewed in stages, in the installing stage, the $\boldsymbol{P_{installing}}$ of V01 is slightly higher than that of V02. However, the difference in $\boldsymbol{N_{installing}}$ between the two is as high as 0.71, so the $\boldsymbol{P_{n\_installing}}$ of V01 is lower than V02. In the binding stage, the $\boldsymbol{P_{binding}}$ of V01 is 21.5% higher than that of V02. However, the $\boldsymbol{N_{binding}}$ is only 14.1% higher than V02, which indicates that the $\boldsymbol{P_{n\_binding}}$ of V01 should be higher than that of V02. The data in Table 7 confirm this. In comparing V03 and V04, the ABR process of the beam in the same position, V04's productivity in all stages is higher than V03's, and with a slight difference in $\boldsymbol{P_n}$, it can be seen that $\boldsymbol{N}$ plays a key factor.

Combined with Table 7 and Figure 13, it can be seen that the productivity evaluation index calculated by the coordinate detected by CenterNet conforms to the actual situation and satisfies the objective law. It is worth mentioning that the dataset of this paper has 1051 images. As many as 6239 images are involved in CenterNet detection in this section, indicating that the proposed computer vision method has relatively good stability. The results obtained in this paper validate that using a computer vision-based method to assist in productivity evaluation is a feasible approach, and the generated images help assist construction managers to analyze better the reasons for the excessively fast or slow construction speed, which in turn helps them to allocate on-site human resources and improve construction productivity rationally.

5. Comparison

To further verify the robustness of the CenterNet-based deep learning model used in this paper, Table 8 explores the evaluation of the productivity of the ACR process after replacing the object detection algorithm with other algorithms, using V02 as a case study. The actual values are listed for comparison.

Table 8. Comparison when evaluating the work productivity of V02.

	Real	**CenterNet**	**Faster R-CNN**	**SSD**	**YOLO v3**
$\boldsymbol{T}$ (s)	7135	7135	7135	7135	7135
$\boldsymbol{T_{installing}}$ (s)	4060	4050	4045	4035	4050
$\boldsymbol{T_{binding}}$ (s)	3075	3085	3095	3100	3085
$\boldsymbol{N}$	1.59	1.55	1.55	1.52	1.54
$\boldsymbol{N_{installing}}$	1.71	1.70	1.70	1.68	1.69
$\boldsymbol{N_{binding}}$	1.44	1.34	1.34	1.31	1.33
$\boldsymbol{P}$	1.402	1.402	1.402	1.402	1.402
$\boldsymbol{P_{installing}}$	2.463	2.470	2.472	2.478	2.470
$\boldsymbol{P_{binding}}$	3.252	3.241	3.231	3.225	3.241
$\boldsymbol{P_n}$	0.882	0.904	0.904	0.922	0.910
$\boldsymbol{P_{n\_installing}}$	1.440	1.453	1.459	1.475	1.461
$\boldsymbol{P_{n\_binding}}$	2.268	2.389	2.399	2.461	2.436

The data listed in Table 8 show that applying the work productivity evaluation method we proposed in Section 2.3 in combination with different object detection algorithms can achieve good results. Even the SSD algorithm with lower precision has a maximum error of only 8% compared with the actual value. It shows that the work productivity evaluation with the method proposed in Section 2.3 is feasible and further verifies the effectiveness of the object detection algorithm in solving the productivity evaluation problem in engineering construction. Comparing the four object detection algorithms together, CenterNet and Faster R-CNN have better agreement with the actual results, mainly due to "worker's" lower missed detection rate. Combined with the advantages of CenterNet in detection speed, the CenterNet-based object detection model can be considered comparable and applicable to the problem presented in this paper.

6. Discussion

In the field of engineering construction, construction productivity has not received much attention for a long time. Generally, as long as the tasks that should be carried out are completed by the specified deadline, most construction companies do not add the cost of monitoring each process' speed. Noting the potential of computer vision technology for applications in engineering construction, the authors propose a computer vision-based approach for the productivity evaluation of assembling reinforcement processes. This paper contributes to engineering construction in the following aspects:

Firstly, advanced deep learning technology is employed to detect the frequency observed five classes of objects in ABR\ACR images. To achieve this plan, the authors collect and annotate the dataset to train the CenterNet model and evaluate the performance through the test set. It is found that the CenterNet-based model presents satisfactory mAP and detection speed.

Secondly, based on the detected ABR\ACR-related objects, a connection between the worker and construction object is established. The ABR\ACR task can be recognized through the position relationship between the worker and the construction object, so as to obtain the number of workers who participate in the process ($\boldsymbol{N}$). The time ($\boldsymbol{T}$) to perform the task can also be obtained through the recognized materials changing.

Thirdly, with $\boldsymbol{N}$ and $\boldsymbol{T}$, productivity can be evaluated. The results of this paper validate the feasibility of computer vision-based methods in evaluating work productivity. With this paper's results, managers can check the work productivity in detail, determine which workers are performing inefficient work, and then allocate labor resources more reasonably to promote and improve the complete quality of the whole project, forming a virtuous cycle. With the refinement of the proposed method and the expansion of its application, the computer vision-based approach will make it possible to perform rapid productivity evaluation for each process in construction.

The study in this paper contains some limitations that need to be further improved in future work. First of all, error analysis: the results of this paper have some errors, and most of these errors are caused by workers or construction objects being obscured. It is a common problem faced by most current computer vision-based methods. When a worker is not completely obscured, it is still possible for him/her to be recognized, for example, at some moments when the worker only shows half of his/her upper body, but it can still be detected. However, if the worker is completely obscured, it will lead to an error in the judgment of $\boldsymbol{N}$. When the work object is obscured, it will affect the results of $\boldsymbol{L}$ and $\boldsymbol{H}$, and thus fluctuate in the curve shown in Figure 13. In the future, we can consider adding several cameras in a process scene to expand the dataset, to try to reduce errors from multi-view monitoring. Second is the applicability issue: the results of this paper can be extended to other processes of civil engineering construction. Processes such as earthwork, concrete pouring projects, masonry structure projects can apply the ideas of this paper for productivity evaluation, which is the direction in which future work needs to be improved.

7. Conclusions

This paper introduces a new method to evaluate the productivity of assembling reinforcement through the position relationship of objects detected by CenterNet. Firstly, a dataset of 1051 images with five categories is created based on entities related to assembling reinforcement. Eighty percent of the dataset is used to train and evaluate the CenterNet model, and the remaining twenty is used to test the detector's performance. The results showed that the mAP reached 0.9682. Compared with the other three object detection models, the detector trained in this paper is comparable. Then, by inputting the videos into the model, the coordinate of the detected boxes at each moment can be obtained, and the number of workers engaged in the task can be judged through the boxes' position relationship. Finally, evaluation of work productivity is realized by the change of coordinates of the work object in the video, the time consumed to perform the task, and the number of workers involved in the process.

The work productivity evaluation value obtained matches the construction site's actual condition and satisfies the objective law, indicating that the application of computer vision to evaluate engineering work productivity is a feasible approach. Applying the dataset proposed in this paper, the trend of construction material changes can be accurately reflected. With this method, project managers can quickly visualize the productivity of assembling reinforcement without a significant cost increase. The information obtained can be used to allocate human resources to construction sites more rationally. As the method is improved, it will potentially to be used for productivity evaluation of various processes in engineering construction.

Author Contributions: Conceptualization, J.L. and X.Z.; methodology, J.L. and X.Z.; software, J.L. and G.Z.; validation, J.L., G.Z. and M.Z.; formal analysis, J.L., D.L. and Y.Z.; investigation, J.L., M.Z. and Y.Z.; resources, G.Z., D.L. and Y.Z.; data curation, J.L. and X.Z.; writing—original draft preparation, J.L.; writing—review and editing, J.L. and X.Z.; visualization, M.Z.; supervision, G.Z.; project administration, D.L. and Y.Z.; funding acquisition, X.Z. and G.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Science and Technology Research and Development Program of China Construction Eighth Engineering Division Co., Ltd. under Grant 2019-3-17, and in part by the Key Research and Development Program of Liaoning Province under Grant 2019010237-JH8/103.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

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

References

1. Zhong, B.; Wu, H.; Ding, L.; Love, P.E.D.; Li, H.; Luo, H.; Jiao, L. Mapping computer vision research in construction: Developments, knowledge gaps and implications for research. *Autom. Constr.* **2019**, *107*, 102919. [CrossRef]
2. Zhao, X.; Han, R.; Yu, Y.; Hu, W.; Jiao, D.; Mao, X.; Li, M.; Ou, J. Smartphone-Based Mobile Testing Technique for Quick Bridge Cable–Force Measurement. *J. Bridg. Eng.* **2017**, *22*, 06016012. [CrossRef]
3. Nayeri, R.D.; Masri, S.F.; Ghanem, R.G.; Nigbor, R.L. A novel approach for the structural identification and monitoring of a full-scale 17-story building based on ambient vibration measurements. *Smart Mater. Struct.* **2008**, *17*, 2. [CrossRef]
4. Zhao, X.F.; Han, R.C.; Loh, K.J.; Xie, B.T.; Lie, J.K.; Ou, J.P. Shaking table tests for evaluating the damage features under earthquake excitations using smartphones. *Health Monit. Struct. Biol. Syst. XII* **2018**, *10600*, 106000L. [CrossRef]
5. Zhang, M.; Chen, S.; Zhao, X.; Yang, Z. Research on Construction Workers' Activity Recognition Based on Smartphone. *Sensors* **2018**, *18*, 2667. [CrossRef]
6. Tao, W.J.; Lai, Z.H.; Leu, M.C.; Yin, Z.Z. Worker Activity Recognition in Smart Manufacturing Using IMU and sEMG Signals with Convolutional Neural Networks. In Proceedings of the 46th Sme North American Manufacturing Research Conference, Namrc 46, College Station, TX, USA, 18–22 June 2018; Volume 26, pp. 1159–1166.
7. Akhavian, R.; Behzadan, A. Wearable Sensor-Based Activity Recognition for Data-Driven Simulation of Construction Workers' Activities. In Proceedings of the 2015 Winter Simulation Conference (Wsc), Huntington Beach, CA, USA, 6–9 December 2015; pp. 3333–3344.
8. Ryu, J.; Seo, J.; Jebelli, H.; Lee, S. Automated Action Recognition Using an Accelerometer-Embedded Wristband-Type Activity Tracker. *J. Constr. Eng. Manag.* **2019**, *145*, 04018114. [CrossRef]
9. Development Plan for New Generation Artificial Intelligence. Available online: http://www.gov.cn/zhengce/content/2017-07/20/content_5211996.htm (accessed on 1 May 2021).
10. Gulshan, V.; Peng, L.; Coram, M.; Stumpe, M.C.; Wu, D.; Narayanaswamy, A.; Venugopalan, S.; Widner, K.; Madams, T.; Cuadros, J.; et al. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. *JAMA* **2016**, *316*, 2402–2410. [CrossRef]
11. Setio, A.A.; Ciompi, F.; Litjens, G.; Gerke, P.; Jacobs, C.; van Riel, S.J.; Wille, M.M.; Naqibullah, M.; Sanchez, C.I.; van Ginneken, B. Pulmonary Nodule Detection in CT Images: False Positive Reduction Using Multi-View Convolutional Networks. *IEEE Trans. Med. Imaging* **2016**, *35*, 1160–1169. [CrossRef]
12. Esteva, A.; Kuprel, B.; Novoa, R.A.; Ko, J.; Swetter, S.M.; Blau, H.M.; Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **2017**, *542*, 115–118. [CrossRef]
13. Yanai, K.; Kawano, Y. Food Image Recognition Using Deep Convolutional Network with Pre-Training and Fine-Tuning. In Proceedings of the 2015 IEEE International Conference on Multimedia & Expo Workshops (Icmew), Turin, Italy, 29 June–3 July 2015.
14. Dwivedi, R.; Dey, S.; Chakraborty, C.; Tiwari, S. Grape Disease Detection Network based on Multi-task Learning and Attention Features. *IEEE Sens. J.* **2021**, *21*, 1. [CrossRef]
15. Liu, C.; Cao, Y.; Luo, Y.; Chen, G.L.; Vokkarane, V.; Ma, Y.S. DeepFood: Deep Learning-Based Food Image Recognition for Computer-Aided Dietary Assessment. *Incl. Smart Cities Digit. Health* **2016**, *9677*, 37–48. [CrossRef]
16. Al-Qizwini, M.; Barjasteh, I.; Al-Qassab, H.; Radha, H. Deep learning algorithm for autonomous driving using googlenet. In Proceedings of the 2017 28th IEEE Intelligent Vehicles Symposium (Iv 2017), Los Angeles, CA, USA, 11–14 June 2017; pp. 89–96.
17. Ravindran, R.; Santora, M.J.; Jamali, M.M. Multi-Object Detection and Tracking, Based on DNN, for Autonomous Vehicles: A Review. *IEEE Sens. J.* **2021**, *21*, 5668–5677. [CrossRef]
18. Ammour, N.; Alhichri, H.; Bazi, Y.; Ben Jdira, B.; Alajlan, N.; Zuair, M. Deep Learning Approach for Car Detection in UAV Imagery. *Remote Sens.* **2017**, *9*, 312. [CrossRef]
19. Zou, Z.; Zhao, X.; Zhao, P.; Qi, F.; Wang, N. CNN-based statistics and location estimation of missing components in routine inspection of historic buildings. *J. Cult. Herit.* **2019**, *38*, 221–230. [CrossRef]

20. Zhang, A.; Wang, K.C.P.; Li, B.; Yang, E.; Dai, X.; Peng, Y.; Fei, Y.; Liu, Y.; Li, J.Q.; Chen, C. Automated Pixel-Level Pavement Crack Detection on 3D Asphalt Surfaces Using a Deep-Learning Network. *Comput.-Aided Civil. Infrastruct. Eng.* **2017**, *32*, 805–819. [CrossRef]
21. Abdeljaber, O.; Avci, O.; Kiranyaz, S.; Gabbouj, M.; Inman, D.J. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *J. Sound Vib.* **2017**, *388*, 154–170. [CrossRef]
22. Li, J.Q.; Zhao, X.F.; Li, H.W. Method for detecting road pavement damage based on deep learning. *Health Monit. Struct. Biol. Syst. XIII* **2019**, *10972*, 109722D. [CrossRef]
23. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
24. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
25. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2014; pp. 818–833.
26. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
27. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
29. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as Points. *arXiv* **2019**, arXiv:1904.07850.
30. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
31. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 386–397. [CrossRef]
32. Shelhamer, E.; Long, J.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 640–651. [CrossRef]
33. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
34. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
35. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
36. Bochkovskiy, A.; Wang, C.; Liao, H. YOLOv4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
37. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, D.; Reed, S.; Fu, C.; Berg, A.C. SSD: Single shot multiBox detector. In *Proceedings of the Proceedings of European Conference on Computer Vision*; Amsterdam, The Netherlands, 11–14 October 2016, pp. 21–37.
38. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
39. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L. MobileNetV2: Inverted residuals and linear bottlenecks. *arXiv* **2018**, arXiv:1801.04381.
40. Luo, X.; Li, H.; Cao, D.; Dai, F.; Seo, J.; Lee, S. Recognizing Diverse Construction Activities in Site Images via Relevance Networks of Construction-Related Objects Detected by Convolutional Neural Networks. *J. Comput. Civ. Eng.* **2018**, *32*, 04018012. [CrossRef]
41. Luo, X.; Li, H.; Cao, D.; Yu, Y.; Yang, X.; Huang, T. Towards efficient and objective work sampling: Recognizing workers' activities in site surveillance videos with two-stream convolutional networks. *Autom. Constr.* **2018**, *94*, 360–370. [CrossRef]
42. Luo, X.; Li, H.; Yang, X.; Yu, Y.; Cao, D. Capturing and Understanding Workers' Activities in Far-Field Surveillance Videos with Deep Action Recognition and Bayesian Nonparametric Learning. *Comput. Civ. Infrastruct. Eng.* **2018**, *34*, 333–351. [CrossRef]
43. Cai, J.; Zhang, Y.; Cai, H. Two-step long short-term memory method for identifying construction activities through positional and attentional cues. *Autom. Constr.* **2019**, *106*, 102886. [CrossRef]
44. Liu, H.; Wang, G.; Huang, T.; He, P.; Skitmore, M.; Luo, X. Manifesting construction activity scenes via image captioning. *Autom. Constr.* **2020**, *119*, 103334. [CrossRef]
45. Han, S.; Lee, S. A vision-based motion capture and recognition framework for behavior-based safety management. *Autom. Constr.* **2013**, *35*, 131–141. [CrossRef]
46. Yu, Y.; Guo, H.; Ding, Q.; Li, H.; Skitmore, M. An experimental study of real-time identification of construction workers' unsafe behaviors. *Autom. Constr.* **2017**, *82*, 193–206. [CrossRef]
47. Wang, F.; Luo, X.; Li, H.; Yu, Y.; Yang, X. Motion-based analysis for construction workers using biomechanical methods. *Front. Eng. Manag.* **2017**, *4*, 84–91. [CrossRef]
48. Park, M.-W.; Elsafty, N.; Zhu, Z. Hardhat-Wearing Detection for Enhancing On-Site Safety of Construction Workers. *J. Constr. Eng. Manag.* **2015**, *141*, 04015024. [CrossRef]
49. Fang, Q.; Li, H.; Luo, X.; Ding, L.; Luo, H.; Rose, T.M.; An, W. Detecting non-hardhat-use by a deep learning method from far-field surveillance videos. *Autom. Constr.* **2018**, *85*, 1–9. [CrossRef]
50. Wu, J.; Cai, N.; Chen, W.; Wang, H.; Wang, G. Automatic detection of hardhats worn by construction personnel: A deep learning approach and benchmark dataset. *Autom. Constr.* **2019**, *106*, 102894. [CrossRef]

51. 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. [CrossRef]
52. Tang, S.; Roberts, D.; Golparvar-Fard, M. Human-object interaction recognition for automatic construction site safety inspection. *Autom. Constr.* **2020**, *120*, 103356. [CrossRef]
53. Zhao, X.; Zhang, Y.; Wang, N. Bolt loosening angle detection technology using deep learning. *Struct. Control Health Monit.* **2019**, *26*, e2292. [CrossRef]
54. Zhang, Y.; Sun, X.; Loh, K.J.; Su, W.; Xue, Z.; Zhao, X. Autonomous bolt loosening detection using deep learning. *Struct. Health Monit.* **2020**, *19*, 105–122. [CrossRef]
55. Choi, W.; Cha, Y.-J. SDDNet: Real-Time Crack Segmentation. *IEEE Trans. Ind. Electron.* **2020**, *67*, 8016–8025. [CrossRef]
56. Beckman, G.H.; Polyzois, D.; Cha, Y.-J. Deep learning-based automatic volumetric damage quantification using depth camera. *Autom. Constr.* **2019**, *99*, 114–124. [CrossRef]
57. Kang, D.; Cha, Y.-J. Autonomous UAVs for Structural Health Monitoring Using Deep Learning and an Ultrasonic Beacon System with Geo-Tagging. *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *33*, 885–902. [CrossRef]
58. Cha, Y.-J.; Choi, W.; Suh, G.; Mahmoudkhani, S.; Büyüköztürk, O. Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types. *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *33*, 731–747. [CrossRef]
59. Xu, Y.; Wei, S.; Bao, Y.; Li, H. Automatic seismic damage identification of reinforced concrete columns from images by a region-based deep convolutional neural network. *Struct. Control Health Monit.* **2019**, *26*, e2313. [CrossRef]
60. Li, G.; Ren, X.; Qiao, W.; Ma, B.; Li, Y. Automatic bridge crack identification from concrete surface using ResNeXt with postprocessing. *Struct. Control Health Monit.* **2020**, *27*, e2620. [CrossRef]
61. Li, S.; Zhao, X. Automatic Crack Detection and Measurement of Concrete Structure Using Convolutional Encoder-Decoder Network. *IEEE Access* **2020**, *8*, 134602–134618. [CrossRef]
62. Miao, X.; Wang, J.; Wang, Z.; Sui, Q.; Gao, Y.; Jiang, P. Automatic Recognition of Highway Tunnel Defects Based on an Improved U-net Model. *IEEE Sens. J.* **2019**, *19*, 11413–11423. [CrossRef]
63. Li, Y.; Lu, Y.; Chen, J. A deep learning approach for real-time rebar counting on the construction site based on YOLOv3 detector. *Autom. Constr.* **2021**, *124*, 103602. [CrossRef]
64. He, D.; Xu, K.; Zhou, P. Defect detection of hot rolled steels with a new object detection framework called classification priority network. *Comput. Ind. Eng.* **2019**, *128*, 290–297. [CrossRef]
65. Zhou, S.X.; Sheng, W.; Wang, Z.P.; Yao, W.; Huang, H.W.; Wei, Y.Q.; Li, R.G. Quick image analysis of concrete pore structure based on deep learning. *Constr. Build. Mater.* **2019**, *208*, 144–157. [CrossRef]
66. Kim, D.; Liu, M.; Lee, S.; Kamat, V.R. Remote proximity monitoring between mobile construction resources using camera-mounted UAVs. *Autom. Constr.* **2019**, *99*, 168–182. [CrossRef]
67. Roberts, D.; Bretl, T.; Golparvar-Fard, M. Detecting and Classifying Cranes Using Camera-Equipped UAVs for Monitoring Crane-Related Safety Hazards. In Proceedings of the Computing in Civil Engineering 2017, Seattle, WA, USA, 25–27 June 2017; pp. 442–449.
68. Slaton, T.; Hernandez, C.; Akhavian, R. Construction activity recognition with convolutional recurrent networks. *Autom. Constr.* **2020**, *113*, 103138. [CrossRef]
69. Yang, J.; Vela, P.; Teizer, J.; Shi, Z. Vision-Based Tower Crane Tracking for Understanding Construction Activity. *J. Comput. Civ. Eng.* **2014**, *28*, 103–112. [CrossRef]
70. Yang, Z.; Yuan, Y.; Zhang, M.; Zhao, X.; Zhang, Y.; Tian, B. Safety Distance Identification for Crane Drivers Based on Mask R-CNN. *Sensors* **2019**, *19*, 2789. [CrossRef] [PubMed]
71. Law, H.; Deng, J. CornerNet: Detecting Objects as Paired Keypoints. *arXiv* **2018**, arXiv:1808.01244.
72. Ahmed, I.; Ahmad, M.; Rodrigues, J.J.P.C.; Jeon, G. Edge computing-based person detection system for top view surveillance: Using CenterNet with transfer learning. *Appl. Soft Comput.* **2021**, *107*, 107489. [CrossRef]
73. Algabri, M.; Mathkour, H.; Bencherif, M.A.; Alsulaiman, M.; Mekhtiche, M.A. Towards Deep Object Detection Techniques for Phoneme Recognition. *IEEE Access* **2020**, *8*, 54663–54680. [CrossRef]
74. Dai, Y.; Liu, W.; Li, H.; Liu, L. Efficient Foreign Object Detection Between PSDs and Metro Doors via Deep Neural Networks. *IEEE Access* **2020**, *8*, 46723–46734. [CrossRef]
75. Cui, Z.; Wang, X.; Liu, N.; Cao, Z.; Yang, J. Ship Detection in Large-Scale SAR Images Via Spatial Shuffle-Group Enhance Attention. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 379–391. [CrossRef]