Vision-based Excavator Activity Analysis and Safety Monitoring System

Sibo Zhang[1] **and Liangjun Zhang**[1]

[1] Baidu Research, USA

sibozhang1@gmail.com, liangjunzhang@baidu.com

Abstract -

In this paper, we propose an excavator activity analysis and safety monitoring system, leveraging recent advancements in deep learning and computer vision. Our proposed system detects the surrounding environment and the excavators while estimating the poses and actions of the excavators. Compared to previous systems, our method achieves higher accuracy in object detection, pose estimation, and action recognition tasks. In addition, we build an excavator dataset using the Autonomous Excavator System (AES) on the waste disposal recycle scene to demonstrate the effectiveness of our system. We also evaluate our method on a benchmark construction dataset. The experimental results show that the proposed action recognition approach outperforms the state-of-the-art approaches on top-1 accuracy by about 5.18%.

Keywords -

Computer Vision; Deep Learning; Object Detection; Pose Estimation; Action Recognition; Safety Monitor; Activity Analysis

1 Introduction

Operating excavators in a real-world environment can be challenging due to extreme conditions, such as multiple fatalities and injuries occur each year during excavations. Safety is one of the main requirements on construction sites. With the advance of deep learning and computer vision technology, Autonomous Excavator System (AES) has made solid progress [1]. In AES system, the excavator is assigned to load the waste disposal material into a designated area. While the system is capable of operating a whole 24-hour day without any human intervention, in this paper, we mainly address the issue of safety, where the excavator could potentially collide with the environment or other construction machines. We propose a camera-based safety monitoring system that detects the excavator poses, the surrounding environment, and other construction machines, and warns of any potential collisions. In addition, based on action recognition algorithm on human activity, we successfully extend the algorithm to excavator actions and use it to develop an excavator productivity analysis system to analyze activities of the excavator. We note that although developed for AES, this system can also be generally applied to manned excavators.

To build an excavator safety monitor system, we first need to build a perception system for the surrounding environment. The perception system includes detection, pose estimation, and activity recognition of construction machines. Detecting the excavator pose in real-time is a key requirement to inform the workers and to enable autonomous operation. Vision-based (marker-less, marker-based) and sensor-based (IMU, UWB) are two of the main methods for estimating robot pose. The marker-based and sensor-based methods require some additional pre-installed sensors or markers, whereas the marker-less method only requires an on-site camera system, which is common on modern construction sites. Therefore, we adopt a marker-less approach and develop the system solely from camera video input, leveraging state-of-the-art deep learning methods.

In this paper, we propose a deep learning-based excavator activity analysis and safety monitor system which can detect the surrounding environment, estimate poses, and recognize actions of excavators. The main contributions of this paper are summarized as follows:

1) We collect an excavator dataset from our Autonomous Excavator System (AES) in Waste Disposal Recycle scene with ground truth annotations.

2) We develop a deep learning-based perception system for multi-class object detection, pose estimation, and action recognition of construction machinery on construction sites. Then we showed our network get SOTA results on the AES dataset and a benchmark construction dataset.

3) We propose a novel excavator safety monitor and productivity analysis system based on the aforementioned perception system.

2 Related Works

Previous studies related to safety and productivity analysis are reviewed here. We start with some of the most basic tasks in computer vision that are essential to activity analysis and safety monitoring system, including object detection, image segmentation, pose estimation and action recognition. Then, we review vision-based activity analysis and safety monitoring system.

Object Detection. The first category is object detection. More recently, Wang et al. [2] used a region-based CNN framework named Faster R-CNN [3] to detect work-

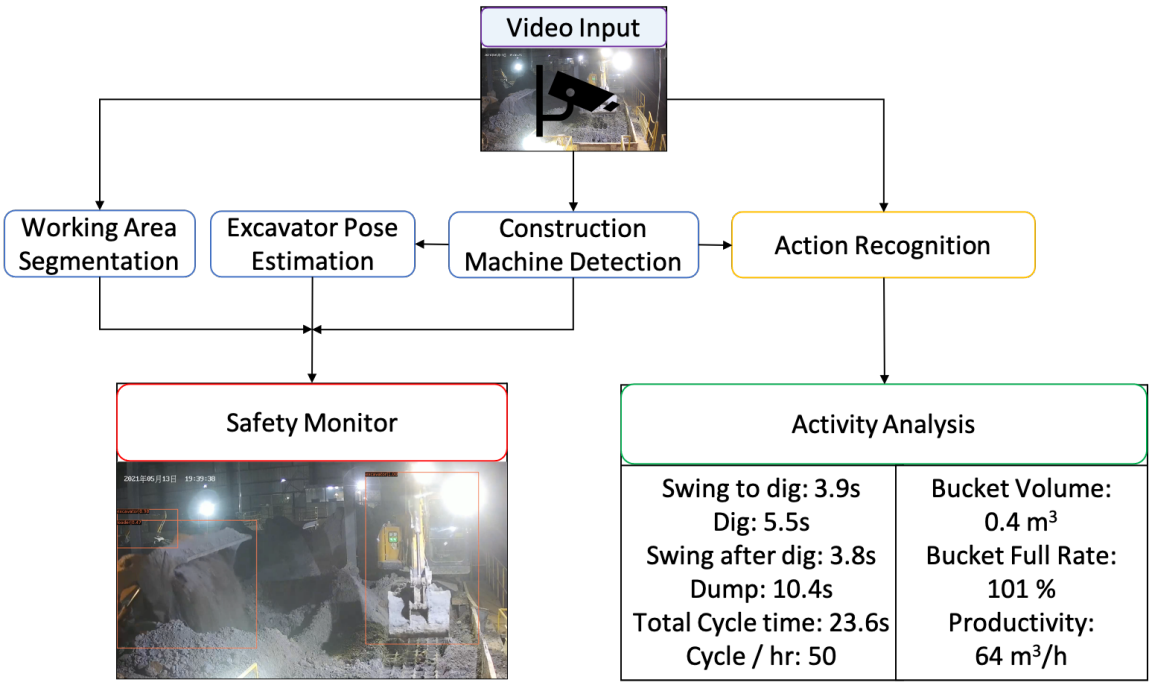

Figure 1. Autonomous Excavator System (AES) activity analysis and safety monitoring system pipeline.

ers standing on scaffolds. A deep CNN then classified whether workers are wearing safety belts. Those without safety belts appropriately harnessed were identified to prevent any fall from height.

Image Segmentation. Raoofi et al. [4] used Mask R-CNN to detect construction machinery on Job sites. More importantly, a segmentation network like Mask R-CNN can be used to decide areas like digging and dumping.

Pose Estimation. The second group of technology is skeleton pose estimation. Pose estimation has been studied [5] based on human pose estimation network like OpenPose. Soltani et al. [6] proposed skeleton parts estimation of excavators.

Action Recognition. Learning-based action recognition methods. Feichtenhofer et al. [7] proposed a SlowFast network for video recognition. The model involves a low pathway that operating at a low frame rate, to capture spatial semantics, and a Fast pathway that operating at a high frame rate, to capture motion at fine temporal resolution. Bertasius et al. [8] presented a convolution-free approach to video classification built exclusively on self-attention over space and time.

Activity Analysis and Safety Monitoring. Here we review recent vision based activity analysis and safety monitoring methods in the construction area. For example, Ding et al. [9] combined CNN with Long-Short-Term-Memory (LSTM) to identify unsafe actions of workers, such as climbing ladders with hand-carry objects, backward-facing, or reaching far. While safety hazards of workers were effectively identified, their method only captured a single worker, and multi-object analysis was not considered. On the other hand, Soltani et al. [6] used background subtraction to estimate the posture of an excavator by individually detecting each of its three skeleton parts including the excavator dipper, boom, and body. Although knowing the operating state of construction equipment would allow safety monitoring nearby, the influence of the equipment on the surrounding objects was not studied. Chen et al. [10] propose a framework to automatically recognize activities and analyze the productivity of multiple excavators. Wang et al. [2] proposed a methodology to monitor and analyze the interaction between workers and equipment by detecting their locations and trajectories and identifying the danger zones using computer vision and deep learning techniques. However, the excavator state is not considered in their model. Roberts et al. [11] proposed a benchmark dataset. However, their action recognition model accuracy is low compared to our deep learning-based model.

Overall, in terms of activity analysis and safety monitoring with computer vision techniques, previous studies focused on different parts separately, such as identifying the working status of construction equipment or pose estimation of the excavator. Our method combine the advantages of SOTA deep learning models from detection, pose estimation, and action recognition tasks.

3 Proposed Framework

The framework for construction machine activity recognition, safety monitor, and productivity analysis is shown in Fig. 1. The framework contains six main modules: construction machine detection, excavator pose estimation, working area segmentation, activity recognition, safety monitor and productivity analysis. The input to our system is surveillance camera video. First, working areas are being segmented into digging and dumping areas. Then, the detection method is used to identify all construction machines in video frames with equipment type. Second, the excavator is identified through pose estimation and detection-based tracking. Then, the action state of the tracked excavators is recognized with pose estimation and working area segmentation. Finally, construction site safety is monitored based on detection and activity recognition results. Besides, the productivity of the excavator is calculated by the activity recognition results. The details about each module in the framework are provided in the following sub-sections.

3.1 Construction Machine Detection

The detection of construction equipment is realized based on Faster R-CNN [3]. The architecture of Faster R-CNN includes (1) backbone network to extract image features; (2) region proposal generate (RPN) network for generating region of interest (ROI), and (3) classification network for producing class scores and bounding boxes for objects. To remove duplicate bounding box, we applied Soft-NMS [12] to limit max bounding box per object to 1.

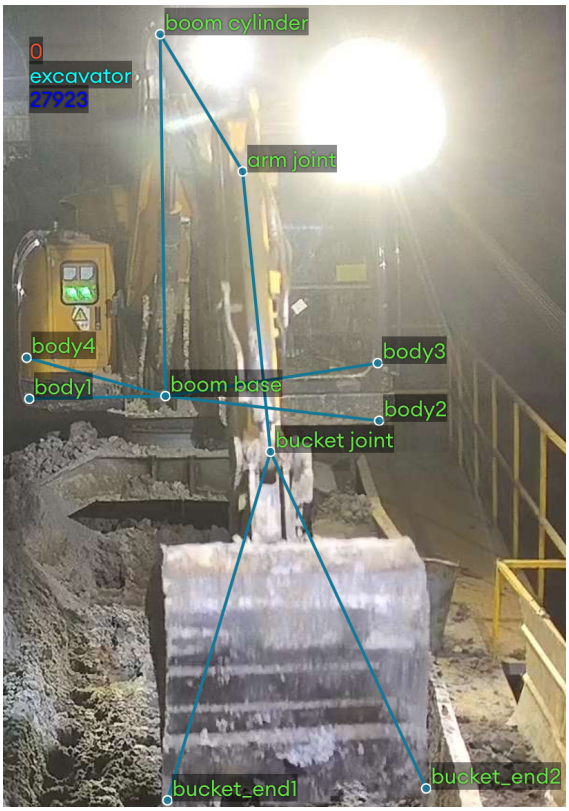

Figure 2. Excavator and corresponding pose labels. We labeled 10 parts of excavators including 2 bucket end keypoints (bucket end1, bucket end2), bucket joint, arm joint, boom cylinder, boom base and 4 body keypoints (body1, body2, body3, body4).

3.2 Excavator Pose Estimation

The pose estimation is based on the output bounding box from detection. We use [13] for pose estimation, which backbone is ResNet. We design a labeling method for the fixed crawler excavator as 10 keypoints. Those keypoints of excavator parts annotated are shown in Fig. 2. These 10 keypoints including 2 bucket end keypoints, bucket joint, arm joint, boom cylinder, boom base, and 4 body keypoints. Unlike other pose label methods [5] to label bucket/ excavator body as the middle point, we label corner point to improve accuracy.

3.3 Working Area Segmentation

We use image segmentation to decide digging and dumping areas as shown in Fig. 3.

The segmentation network is based on ResNet [14]. A digging area is defined as the waste recycling area which including various toxic materials. A dumping area is a designated area to dump waste.

Figure 3. Area segmentation. The pink color area is dumping area and the blue color area is digging area.

3.4 Excavator Action Recognition

We define three actions for excavator: 1. Digging 2. Swinging 3. Dumping. Specifically, we define four states of our autonomous excavator: 1. Digging state 2. Swinging after digging state 3. Dumping state 4. Swinging for digging state. More precisely, Digging indicates loading the excavator bucket with target material; Swinging after digging indicates swinging the excavator bucket to the dumping area; Dumping means unloading the material from the bucket to the dumping area, and Swinging for digging means swinging the bucket to the working area. Besides, there is an optional idle state when the excavator is in manned mode or malfunction status.

To determine the excavator action state, we first determine excavator position based on keypoints from pose estimation and image segmentation results. Then we use continuous frames of pose keypoints of body 1-4 to decide whether the excavator is in the swing state. We set a threshold for keypoints movement: if the mean of pose keypoints of body 1-4 movements is smaller than a set value, then we think the excavator body is still. Otherwise, we think the excavator body is not still. This rule-based module is used in our safety monitor system. Our excavator action states are defined as follows:

1. Digging state: buckets/ arm joint in digging area and body 1-4 is fixed points (excavator body is stilled).

2. Swinging state: buckets/ arm joint in working area and body 1-4 is not fixed points (excavator body is not stilled). Then we can decide whether it is Swing for digging state or Swing after digging state by the previous state. If the previous state is a Dumping state then it will be Swing for digging state. Otherwise, it will be Swing after digging state.

3. Dumping state: buckets/ arm joint in dumping area and body 1-4 is fixed points (excavator body is stilled).

4. Idle state: buckets/ arm joint in dumping area and buckets/ arm joint/ body 1-4 is fixed points (excavator arm and body are both stilled).

Then, we implement a more general deep learning-based action recognition method based on SlowFast [7]. The model involves (i) a Slow pathway, operating at a low frame rate, to capture spatial semantics, and (ii) a Fast pathway, operating at a high frame rate, to capture motion at fine temporal resolution. The Fast pathway can be made very lightweight by reducing its channel capacity, yet can learn useful temporal information for video recognition. This deep learning action recognition model is used in the productivity analysis module.

3.5 Safety Monitor

Detect Potential Construction Machine Collision. The autonomous excavator and the loader may have potential collision as Fig. 4 shows. So it is important to detect potential collision since the loader is hard to know which state excavator is currently at from his view. If more than one machine is detected with in the same region (digging or working area), then an alert may be indicated to the user, and the autonomous vehicles may pause until the issue is cleared.

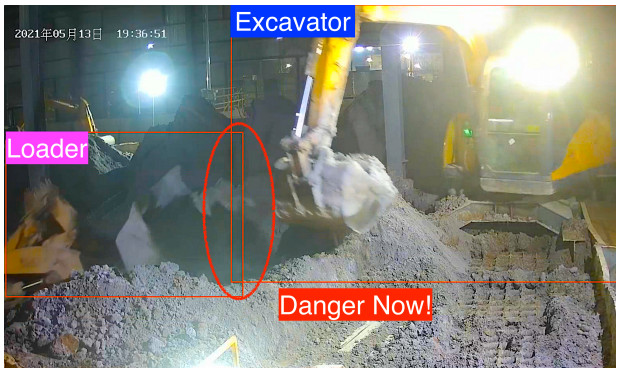

Figure 4. The autonomous excavator and loader potential collision scene when loader tries to load in digging area. The danger signal is sent when the autonomous excavator and the loader machines are both detected in the digging area.

3.6 Productivity Analysis

The productivity of the excavator is based on the activity recognition results. In the solid waste recycle scene, excavators usually work with other equipment, such as loaders. For example, an excavator digs the waste and dumps it into a dumping area. When waste is empty in the digging area, the loader will load and dump waste in the digging area. The excavator's productivity can be calculated with the cycle time, the bucket payload and the average bucket full rate, as shown in Equation 1. Since the bucket payload is given by the manufacturer, the target of the productivity calculation becomes to determine the cycle time of the excavator and the bucket full rate. To simplify the procedure, the two types of swinging (swinging after digging and swinging for digging) are not distinguished in this paper.

$$Productivity(m^3/hr) = \frac{Cycles}{hr} \times BucketVolume(m^3) \times BucketFullRate \quad (1)$$

The time for each cycle is measured by the workflow showing in Fig. 5. Our action recognition module labels each video frame of the excavator with an action label. Next, the action labels of two consecutive frames are com-

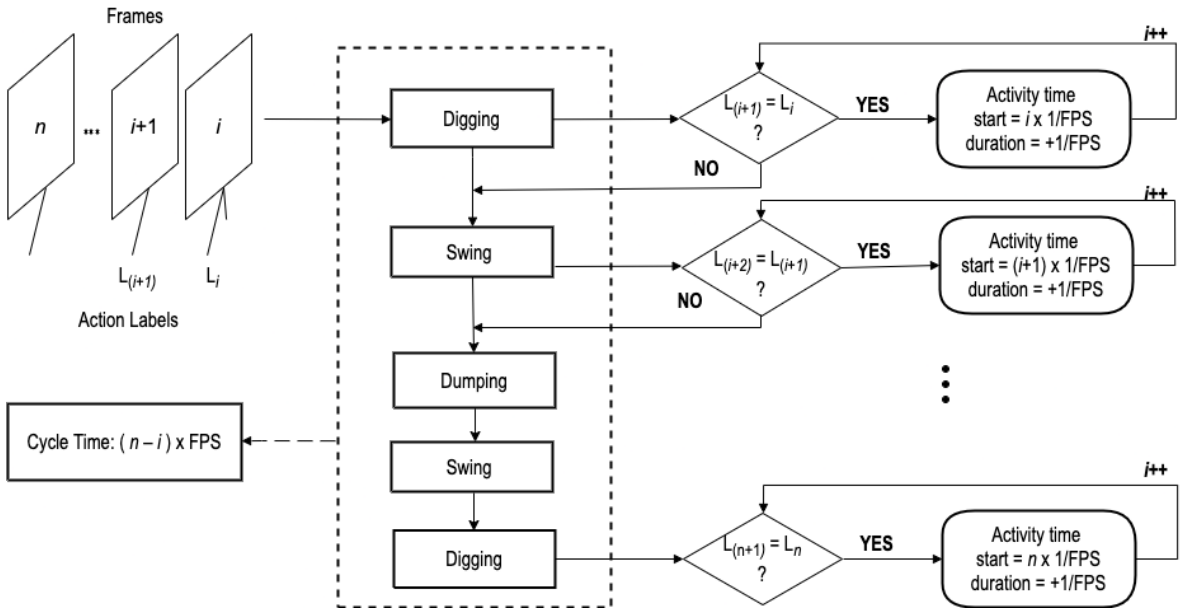

Figure 5. Excavator cycle time calculation method.

pared. If they are the same, it means that the action remains same. Thus, the cumulative time for the current action is increased by 1/FPS (frame per second). If the labels are different, it means that a new action has started, and the time of the newly recognized activity will increase by 1/FPS. We define the total time of one cycle as the difference between the start times of two neighboring digging actions.

4 Experiments

4.1 Dataset

We collect an excavator dataset from our Autonomous Excavator System (AES) from the waste disposal recycle scene [1]. The dataset including 1 hour of videos containing 2 types of construction equipment (i.e. excavators, loaders). To demonstrate the effectiveness of our dataset, we labeled 601 images with object detection bounding boxes, excavator poses, and background segmentation. 80% of the images are used for model training while 20% are for model validation and testing. Besides, we labeled 102 clips of excavator videos with 3 actions (digging, dumping, swinging). The videos were captured at 1920*1080 and filmed at 25 frames per second.

We also test our method based on the benchmark construction dataset [11] which including 479 action videos of interacting pairs of excavators and dump trucks performing earth-moving operations, accompanied with annotations for object detection, object tracking, and actions. The videos were captured at 480*720 and filmed at 25 frames per second.

4.2 Evaluation

4.2.1 Object Detection Evaluation

The detection evaluation metrics are based on the Microsoft COCO dataset [15]. The network's performance is evaluated using Average precision (AP). Precision measures how many of the predictions that the model made were correct and recall measures how well the model finds all the positives. For a specific value of Intersection over Union (IoU), the AP measures the precision/recall curve at recall values (r1, r2, etc.) when the maximum precision value drops. The AP is then computed as the area under the curve by numerical integration. The mean average precision is the average of AP in each object class. More precisely, AP is defined as:

$$AP = \frac{1}{11} \sum_{r \in \{0.0, 0.1, \ldots, 1\}} AP_r, \tag{2}$$

4.2.2 Pose Estimation Evaluation

The pose estimation matrix is based on The COCO evaluation, which defines the object keypoint similarity (OKS). It uses the mean average precision (AP) over the number of classes for OKS thresholds as main competition metric. The OKS is calculated from the distance between predicted points and ground truth points of the construction machine.

Figure 6. Excavators and loader detection result. Our system is capable detecting multi-class construction machines in real-time.

Table 1. Accuracy of construction machine detection

Network	Backbone	mAP (%)
Faster R-CNN	Resnet-50-FPN	90.1
Faster R-CNN	Resnet-152-FPN	92.3
YOLOv3	DarkNet-53	73.2

4.2.3 Action Recognition Evaluation

The performance metric is the mean Average Precision (mAP) over each object class, using a frame-level IoU threshold of 0.5.

4.3 Accuracy

4.3.1 Accuracy of the detection model

We implement experiments on the Faster R-CNN model with a backbone network of Resnet-50-FPN and Resnet-152-FPN. The model achieved high detection accuracy for construction equipment. The Average Precision (AP) values of the excavator achieved 93.0% and the loader achieved 85.2%. With an mAP of 90.1%, the model is demonstrated to be promising for detecting multi-class construction equipment accurately on the construction site.

We also compared the result with Yolo V3 [16]. YOLOv3 is a one-stage state-of-art detector with extremely fast speed. In this study, the image input size is 416x416 and this algorithm can process 20 images in one second. Compared with some two-stage detectors, the performance of YOLOv3 is slightly low, but the speed is much faster and that is important for real-time applications. The construction detection dataset from the previous step is used for training YOLOv3, which takes 12 hours for the training process. The mAP of YOLOv3 on our testing set is 73.2% from an overall view, where the AP is 80.2% in the excavator category and 60.2% in the loader category. The detailed comparison result is shown in Table 1. The result is shown in Fig. 6.

Figure 7. Excavator pose estimation result.

Table 2. Accuracy of the pose estimation model.

Network	Backbone	Input size	AP (%)
SimpleBaseline	Resnet-50	256*192	91.79
SimpleBaseline	Resnet-50	384*288	94.19
SimpleBaseline	Resnet-152	384*288	96.50

4.3.2 Accuracy of the Pose Estimation

We apply SimpleBaseline [13] to our pose estimation model and get the following result. Experiments have been conducted on different Backbone networks including Resnet-50 and Resnet-152. Besides, experiments on different image input sizes have been implemented. The detailed comparison result is shown in Table 2. The result is shown in Fig. 7.

4.3.3 Accuracy of the Action Recognition

We applied Slow-Fast [7] to our action recognition model and get the following result. Experiments have been conducted on the different networks including SlowFast-101 and SlowFast-152. Besides, experiments on different

Figure 8. Excavators long video action detection result.

Figure 9. Long video demos of action recognition result on different scenes of the construction dataset. Prediction with the highest possibility is showing in the first line.

Table 3. Accuracy of the action recognition model on our AES dataset and UIUC dataset from [11].

Dataset	Network	Backbone	Top1 Acc. (%)
AES	SlowFast-50	ResNet3d	89.70
	SlowFast-152	ResNet3d	91.44
UIUC	Roberts[11]	N/A	86.8
	SlowFast-50	ResNet3d	91.9
	SlowFast-152	ResNet3d	93.3

clip lengths have been implemented. The detailed comparison result is shown in Table 3. The result of top 3 action prediction is showing in the Fig. 8. We input a excavator video and the system can predict action result in almost real-time. Prediction with the highest possibility is showing in the first line. Here the system predict the action as digging with 54% confidence.

Comparing our result with Roberts [11] on their UIUC dataset, out proposed action recognition approach outperforms their accuracy by about 5.18%. The action recognition video demo result of the construction dataset is showing in Fig. 9. The result shows the advantage of using deep learning model on action recognition task over their Hidden Markov Model (HMM) + Gaussian Mixture Model (GMM) + Support Vector Machine (SVM) method.

4.4 Productivity Analysis

The proposed framework was tested to estimate the productivity of excavators on a long video sequence, which contains 15 min of excavator's operation. In our video, the XCMG 7.5-ton compact excavator (bucket volume of 0.4 m^3) completed 40 working cycles in 15 minutes and the average bucket full rate is 101%. So the excavation productivity is 64.64 m^3/h according to Equation 1. Our system detects 39 working cycles in the video which the accuracy of productivity calculation is 97.5%. The test results showed the feasibility of using our pipeline to analyze real construction projects and to monitor the operation of excavators.

4.5 Implementation Details

We implement our detection module based on MMDetection, segmentation module based on MMSegmentation, pose estimation module based on MMPose, and action recognition module based on MMAction2 toolbox [17, 18, 19, 20]. We use NVIDIA M40 24GB GPUs to train and test the network.

It takes 6 hours to train detection, pose estimation, and action recognition module. The inference time of detection, pose estimation, and action recognition are 5, 2, and 1 frames per second.

5 Conclusion

In this study, we collect a benchmark dataset from Autonomous Excavator System (AES). Besides, we proposed a safety monitor and productivity system pipeline based on computer vision and deep learning techniques. We integrate detection, pose estimation, activity recognition modules into our system. We also evaluate our method on a general construction dataset and achieve SOTA results. However, our current system may have some limitations. Our dataset is relatively small due to the relatively simple waste disposal recycle scene captured from AES system.

References

[1] Liangjun Zhang, Jinxin Zhao, Pinxin Long, Liyang Wang, Lingfeng Qian, Feixiang Lu, Xibin Song, and Dinesh Manocha. An autonomous excavator system for material loading tasks. *Science Robotics*, 6(55), 2021.

[2] Mingzhu Wang, P Wong, Han Luo, Sudip Kumar, V Delhi, and J Cheng. Predicting safety hazards among construction workers and equipment using computer vision and deep learning techniques. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 36, pages 399–406. IAARC Publications, 2019.

[3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6): 1137–1149, 2016.

[4] H Raoofia and A Motamedib. Mask r-cnn deep learning-based approach to detect construction machinery on jobsites.

[5] Hinako Nakamura, Yumeno Tsukada, Toru Tamaki, Bisser Raytchev, and Kazufumi Kaneda. Pose estimation of excavators. In *International Workshop on Advanced Imaging Technology (IWAIT) 2020*, volume 11515, page 115152J. International Society for Optics and Photonics, 2020.

[6] Mohammad Mostafa Soltani, Zhenhua Zhu, and Amin Hammad. Skeleton estimation of excavator by detecting its parts. *Automation in Construction*, 82:1–15, 2017.

[7] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[8] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021.

[9] Lieyun Ding, Weili Fang, Hanbin Luo, Peter ED Love, Botao Zhong, and Xi Ouyang. A deep hybrid learning model to detect unsafe behavior: Integrating convolution neural networks and long short-term memory. *Automation in construction*, 86:118–124, 2018.

[10] Chen Chen, Zhenhua Zhu, and Amin Hammad. Automated excavators activity recognition and productivity analysis from construction site surveillance videos. *Automation in construction*, 110:103045, 2020.

[11] Dominic Roberts and Mani Golparvar-Fard. End-to-end vision-based detection, tracking and activity analysis of earthmoving equipment filmed at ground level. *Automation in Construction*, 105:102811, 2019.

[12] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017.

[13] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[16] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement, 2018.

[17] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[18] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020.

[19] MMPose Contributors. Openmmlab pose estimation toolbox and benchmark. `https://github.com/open-mmlab/mmpose`, 2020.

[20] MMAction2 Contributors. Openmmlab's next generation video understanding toolbox and benchmark. `https://github.com/open-mmlab/mmaction2`, 2020.