Contents

Vision-based excavator pose estimation using synthetically generated datasets with domain randomization

Abstract

The ability to monitor and track the interactions between construction equipment and workers can lead to creating a safer and more productive work environment. In recent years, many efforts have been made to utilize on-site cameras as a non-intrusive method for intelligent monitoring through equipment pose estimation. However, there are limited studies that estimate the pose of articulated equipment using monocular cameras. Most recent studies employ computer vision and deep learning techniques, which rely on the size and quality of the training datasets for optimal performance. However, preparation of large datasets with high quality annotations is a manual process, and due to the limited data availability in specialized fields such as construction, most previous studies have focused on taking a model-centric approaches to improving the performance of the proposed solution.

To overcome this challenge, this study takes a data-centric approach and presents a framework for synthetically generating large and accurately annotated images, required for training deep convolutional neural networks. The study offers several contributions to the current literature. First, a method is developed using a game engine, which employs domain randomization (DR) to produce large labelled datasets for excavator pose estimation. Second, a state-of-the-art deep learning architecture based on high representation network (HRNet) is adapted and modified for excavator pose estimation. This model is trained on synthetically generated datasets and its performance is evaluated on two datasets, containing images of real excavators in the field. Finally, the quality of the synthetically generated datasets is evaluated, two training strategies are compared, and the impacts of various randomization parameters in the simulator are examined. The results reveal that the model trained solely on synthetic data can yield comparable results to the model trained on real images of excavators. This demonstrates the effectiveness of utilizing synthetic datasets for complex vision tasks such as equipment pose estimation. The study concludes by highlighting promising directions for further work in synthetic data studies in construction.

1. Introduction

Construction sites are one of the most hazardous work environments worldwide, with high rates of injuries and fatalities [1]. According to the U.S. Bureau of Labor, on average around one in every five fatal occupational injuries are attributed to the construction industry [2]. Struck-by accidents, in particular, have accounted for around 20% of all construction related accidents in the United States in 2017 [2]. Operations involving heavy construction equipment such as excavators are a critical component of most construction projects. Heavy construction equipment, vehicles and workers are often required to work simultaneously in limited workspaces due to spatial limitations and tight schedules, which often leads to suboptimal performance both in terms of safety and productivity [3]. Equipment and workspace blind spots are one of the primary causes of collision accidents [4]. Workers' exposure to noise, and fatigue also contribute to the risk by impairing workers and operators ability to recognize proximity hazards [5, 6]. To reduce the risk of such accidents, safety trainings are provided, and safety observations and inspections are carried out on site [7]. However, due to the dynamic, complex, and unique nature of construction workplaces, continuous monitoring of entire jobsites by safety personnel is impractical, costly, and ineffective [8]. Moreover, prevention of accidents through manual inspections relies on the reaction of inspectors which does not allow for prompt and accurate response to emerging hazardous situations. It is anticipated that automation of the safety monitoring process can reduce the risk of struck-by accidents by enabling continuous monitoring, and timely response to hazards.

Previous studies have proposed to employ real-time location systems (RTLS) and GPS sensors to track construction equipment and alert workers and operators in the case of a proximity hazard [9-11]. However, Radio frequency based RTLS methods suffer from signal attenuation, and GPS sensors are not suitable for indoor or dense urban environments [12]. Furthermore, these monitoring techniques require pre-installation of sensors on objects of interest, which is a cumbersome task in the busy construction environment, where multiple sub-contractors are involved. In contrast with sensor-based monitoring techniques, computer vision-based methods that utilize surveillance videos to identify and locate workers and equipment have the major advantage of being non-intrusive as they do not require sensor installations [13]. Previous research has studied identification and detection of construction equipment and workers using computer vision techniques [14-16]. While detection, localization and tracking of construction equipment can be beneficial for proximity hazard monitoring [17], heavy construction equipment such as excavators often do not change location during operations, but due to their high articulation, pose a high risk to the safety of the personnel that are working in their vicinity. Therefore, the ability to estimate the location of the individual parts of the equipment with a higher level of granularity is crucial to enable the development of safety monitoring applications. This problem is commonly known as pose estimation, in which the objective is to accurately determine the location of heavy equipment’s individual joints. Pose estimation serves as a pre-processing step for many applications that require an understanding of the relative position of the equipment parts or the interaction between various objects such as proximity and blind spot monitoring, activity recognition, and abnormal behaviour detection.

Soltani et al. [18] proposed to estimate the excavator pose by fusing RTLS data and the data obtained from two cameras. However, the method requires sensor installation, camera calibration, and only performs well when the excavator parts are visible from both cameras. Methods that rely only on visual data from a single or multiple cameras have also been proposed [19]. In the earlier works, fiducial markers are mounted on the equipment’s points of interest [20-24]. This enables the ability to recognise the position and orientation of markers, thus estimating the pose of the articulated machinery. While these methods have shown to be capable of pose estimation with centimetre level accuracy, they require installation of markers on the equipment, and suffer from a number of other major drawbacks that prevent their practical deployment on construction sites, such as marker damage, occlusion and range limitations [23].

Pose estimation using monocular cameras without relying on fiducial markers has also been explored. Direct visual pose estimation approaches can be categorized into methods that use traditional computer vision techniques [25-27], and more recently, methods that rely on machine learning models [7, 28, 29]. The former category typically requires manually designing feature extraction and decision-making criteria to achieve a desired output. For instance, Rezazadeh Azar and McCabe [25] employed a part-based object recognition model based on histogram of oriented gradients (HOG) and proposed to impose spatial-temporal constraints to improve detection results. Yuan et al. [27] proposed a template matching approach based on geometrical shapes and kinematic constraints for detecting individual excavator components and estimating the equipment pose. These methods, however, are not robust to major viewpoint, shape, and illumination variations, and perform poorly in cluttered environments such as construction sites [26, 27].

On the other hand, machine learning methods, particularly, convolutional neural networks have been shown to outperform traditional computer vision techniques, while not relying on hand-crafted feature extraction steps [7, 29]. For instance, Luo et al. [29] prepared a dataset of manually annotated excavators, and compared the performance of Stacked Hourglass Network [30], Cascaded Pyramid

Network [31], and an ensemble model for excavator pose estimation. While convolutional neural networks have dramatically improved the state-of-the-art in visual object detection and pose estimation, they require a large number of labelled data for optimal performance [32]. One of the most prominent contributing factors to the substantial performance improvement of deep learning is the availability of large, accurately labelled, and diverse datasets [33]. For instance, datasets for human pose estimation such as MS COCO [34], and MPII [35] contain more than 250 thousand, and 40 thousand annotated instances, respectively. However, limited availability of data in the context of construction, and substantial time and effort required for preparing datasets with high quality annotations has been repeatedly mentioned in the literature [29, 36, 37].

To overcome the challenge of data availability in construction, a number of solutions have been proposed. For instance, Luo et al. [29] proposed to use data augmentation to increase the dataset size for excavator pose estimation. However, data augmentation can only be applied on an existing dataset, and is unable to overcome the data availability challenges. Liu and Golparvar-Fard [38] proposed a crowdsourcing solution to prepare annotated datasets for activity recognition. Similarly, Wang et al. [39] examined a crowdsourcing approach to prepare a dataset for identification of images with safety-rule violations. In another study, Han and Golparvar-Fard [40] proposed to facilitate the preparation of a dataset of construction materials by guiding the annotation process with BIM overlays. Most recently, Kim et al. [41] proposed to use active learning, an approach that evaluates unlabelled data and identifies the most meaningful to learn instances to be annotated for object detection based on the uncertainty of detections. Although the methods discussed above are proposed to facilitate the manual and laborious process of dataset preparation, their process still involves manual annotation, they may suffer from poor annotation quality due to the manual and subjective nature of the labelling process, and they rely on availability of unlabelled data.

In an attempt to address the data availability constraint, Liang et al. [7] proposed a method for automatically generating a labelled dataset for excavator pose estimation. The process involves an industrial robotic arm equipped with a bucket to resemble an excavator, and the grand-truth annotations are acquired by the robot’s built-in encoders. However, this method is unable to produce a dataset with sufficient variety, consequently, the network trained on the dataset has not been able to achieve high performance when applied to real excavators in the field [7].

Synthetic data generation is another method employed for various computer vision problems to overcome the labelled data sparsity for specialised tasks [42, 43]. For instance, Dwibedi et al. [44] proposed a simple approach that adds segmented object instances on random backgrounds to generate large labelled training datasets. Johnson-Roberson et al. [45] developed a realistic data generation framework using a simulation engine that provides bounding box annotations for vehicle detection. Similarly, in the field of autonomous vehicles, Tsirikoglou et al. [46] proposed a method to generate realistic synthetic data with pixel-level accurate labels for object segmentation. In another study, Tobin et al. [47] proposed to use domain randomization (DR) to identify the location of objects in 3 dimensions, given an RGB image for a robotic task. Domain randomization is a process that randomizes various aspects of the domain in creating the training dataset with sufficient variability, such that the model is able to generalize to real world with no further training [47]. Furthermore, Tremblay et al. [48] proposed a method based on synthetic data generation with DR to detect bounding boxes around cars for autonomous driving applications, and demonstrated the ability of DR-generated synthetic datasets to bridge the gap between synthetic and real datasets in training a neural network.

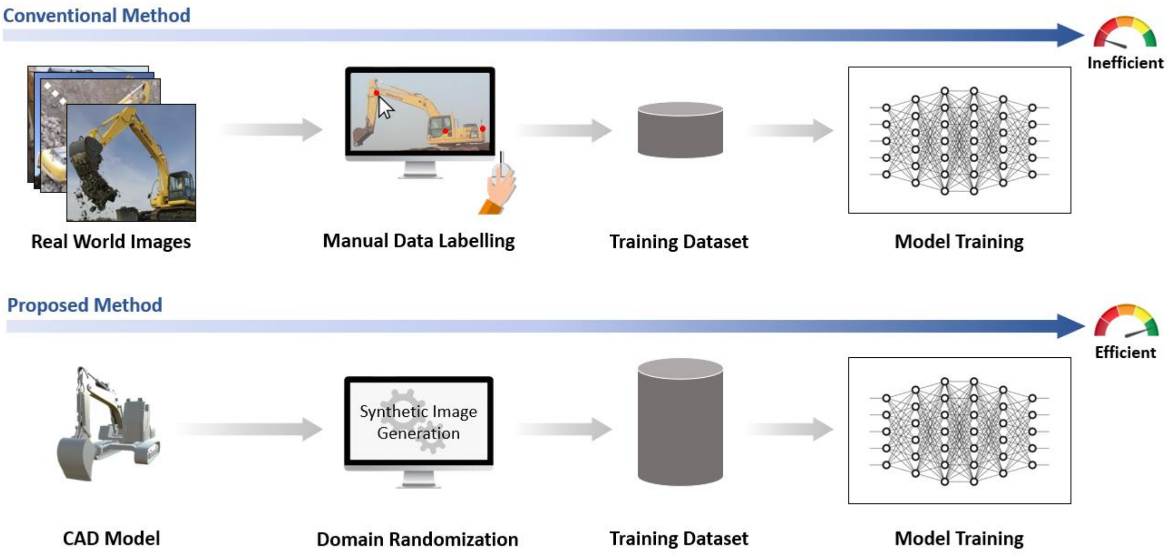


Fig. 1. Comparison of training dataset preparation methods; conventional vs. the proposed method.

This study extends the work of Tremblay et al. [48], which is limited to bounding box detection around vehicles being observed from the perspective of the vehicle driving at the same level as the objects of interest, to pose estimation of highly articulated machinery such as excavators in the context of construction. This study makes several contributions to the literature. First, a method for synthetic dataset generation, developed using a game engine is proposed. The method utilizes DR to automatically produce large labelled datasets required for training deep neural networks. As depicted in Figure 1, the method removes the laborious and error-prone manual annotation process and enables large dataset generation. Second, a state-of-the-art deep learning architecture known as high representation network (HRNet) [49] is adapted for the problem of excavator pose estimation, the model is trained on synthetically generated datasets and its performance is evaluated on two datasets of real excavators in the field. Finally, an evaluation of the proposed method is conducted by answering the following questions:

- To what extent can increasing the size of synthetically generated datasets improve model performance on real images?
- Given the ability of the proposed method in producing pixel-level accurate annotations for occluded keypoints, does inclusion of keypoints that are not directly visible during model training improve performance?
- How does training the model solely on synthetic datasets perform as compared to training on real images?
- What DR parameters in the developed simulation have the highest impact on model performance when the model performance is evaluated on real images?

The remainder of the study is organized as follows. Section 2 presents the proposed method of synthetic dataset generation, provides details of the adapted deep learning architecture and training details, and presents the two datasets of real excavators in the field that are used for evaluation, as well as the employed evaluation metrics. Section 3 presents the results of the experiments. Section 4 provides a discussion of the results, future directions, and limitations of the study. The conclusion is presented in Section 5 of the article.

2. Proposed method

In this section, a framework for generating synthetic data to train a deep learning model for construction equipment pose estimation is presented. Section 3.1 describes the developed synthetic data generator. The details of the pose estimation network are discussed in Section 3.2, and the datasets of real excavators in the field, as well as the evaluation metrics employed are discussed in Section 3.3.

2.1 Synthetic data generation with DR

This section describes the proposed framework for automatic generation of labelled datasets for training deep neural networks. The synthetic dataset generator is developed using Unity, a cross-platform game engine capable of creating 3D games and simulations. The proposed synthetic dataset generator employs DR to produce datasets with pixel-level accurate annotations, and a large amount of variety. Figure 2 illustrates the various aspects of the DR, which include variable parameters of the scene and the equipment model.

The randomized scene parameters include the viewpoint, field of view (FoV) of the virtual camera, lighting, floor textures, background textures, 3D occluding objects and dust simulator as depicted in Figure 2. The camera height and its horizontal offset from the location of the equipment ranges from 2 to 20 meters, and 8 to 35 meters, respectively. The FoV of the camera is selected randomly from the range 20 to 60 degrees for standard lenses, and 60 to 90 degrees for wide angle lenses. The randomized parameters of the light source include its intensity, colour and location.

The floor and background textures are randomly selected from a pool of images that contain both real landscapes and materials, as well as abstract patterns. Similar to Tremblay et al. [48], it is anticipated that by creating datasets that also include non-realistic images, the network will be forced to distinguish the most prominent features related to the shape of the equipment, and therefore, would be able to better generalize to real images. Furthermore, to include instances with occlusion and simulate the presence of dust, which are common occurrences on construction sites, random 3D occluding objects and simulated dusts are added to the scene.

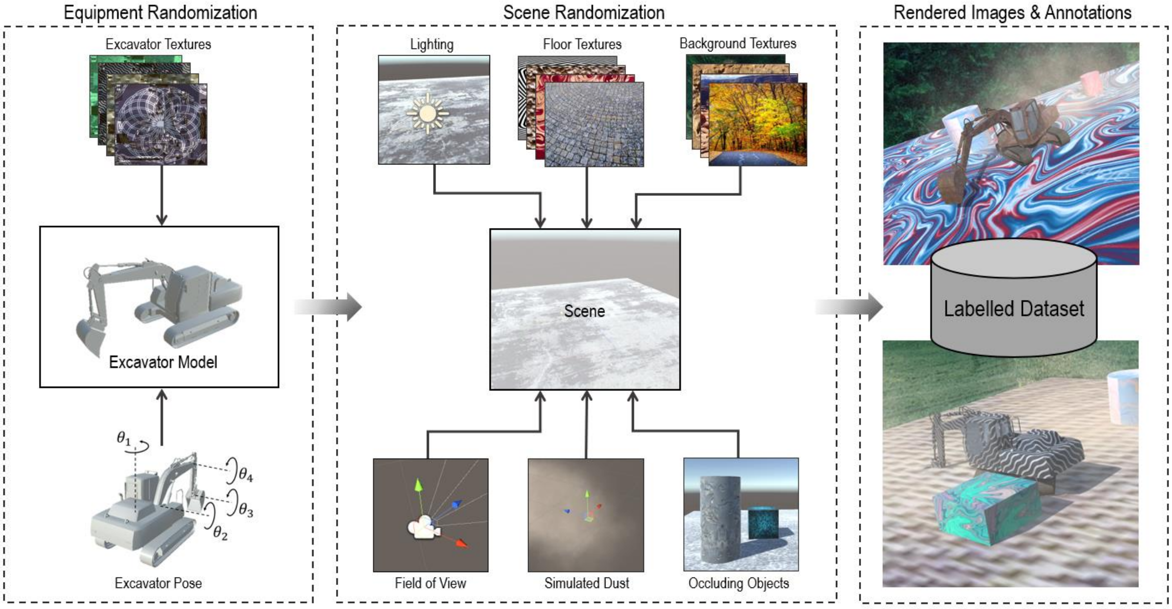


Fig. 2. The process of synthetic dataset generation with domain randomization. Various elements of the simulation are randomized and the rendered synthetic images along with the ground-truth annotations are used for training the pose estimation model.

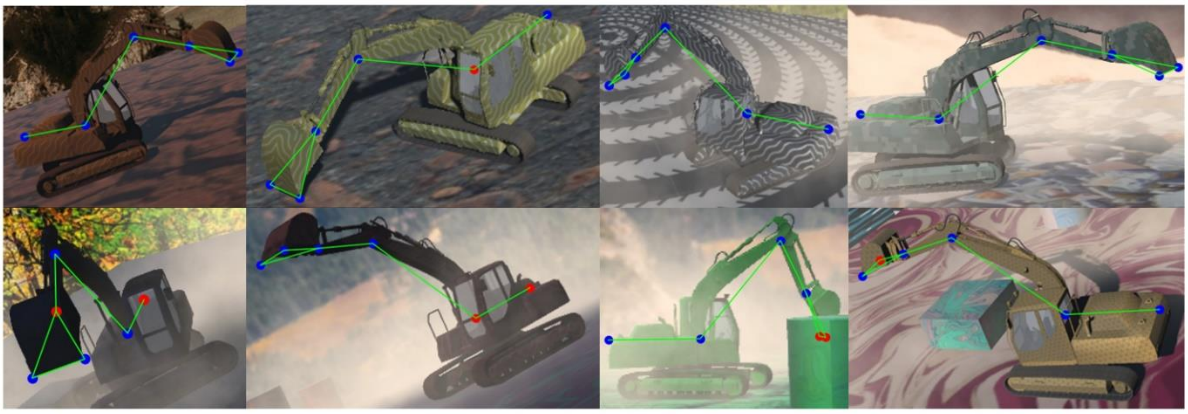


Fig. 3. Representative examples from synthetically generated datasets, showing various lighting conditions, texture patterns, simulated dust, occluding objects, and self-occluded keypoints. The blue dots represent visible keypoints and the occluded keypoints are represented by red dots.

The equipment randomized parameters include its pose with four degrees of freedom as illustrated in Figure 2, equipment texture, and its location on the scene. Similar to the textures used for the floor and background of the scene, the majority of excavator textures are purposefully created with patterns that do not necessarily resemble realistic equipment textures to avoid overfitting during model training.

The developed synthetic data generator employs the Raycast functionality of Unity to determine whether a keypoint is visible or occluded. The visibility check also includes self-occlusion, which refers to the instances where a keypoint is occluded by other parts of the same equipment. Figure 3 illustrates some representative examples of synthetically generated images along with the keypoints, and their visibility status. Moreover, various lighting conditions, textures, presence of dust and occluding objects can be observed in Figure 3. To determine the visibility of each keypoint, an imaginary sphere around each keypoint is created and the keypoint is considered visible if a portion of this sphere is visible from the camera's point of view. The visible and occluded keypoints can be viewed in Figure 3, marked by blue and red dots, respectively.

To enable the comparability of this study with those in the existing literature, this study adopts the excavator keypoint definitions proposed by Luo et al. [29]. This definition considers 6 keypoints, consisting of Body_end, Cab_boom, Boom_arm, Bucket_end_left and Bucket_end_right as depicted in Figure 4. However, as the annotations are generated automatically by the synthetic data generator, the number of keypoints and their definition can be modified to suit specific applications if required.

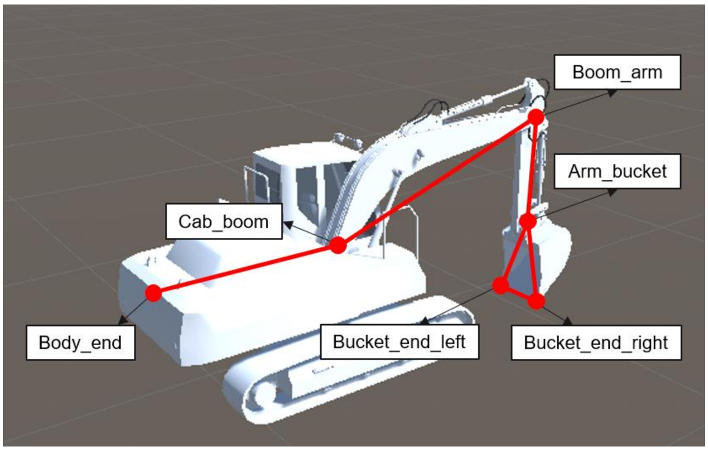


Fig. 4. The definition of keypoints used to describe the full body pose of an excavator in this study.

2.2 Pose estimation model

This sub-section describes the architecture of the pose estimation model in this study, and it provides a summary of the training details adopted in the experiments.

2.2.1 Model architecture

This study adopts HRNet, a state-of-the-art deep learning architecture developed by Wang et al. [49]. HRNet is designed for computer vision problems that require spatially precise representations such as semantic segmentation, object detection and human pose estimation. HRNet connects multi-resolution convolutions in parallel, while repeatedly exchanging information across various resolutions using a number of proposed exchange blocks.

The aim of pose estimation is to detect the location of k keypoints from an input image of size $W \times H \times 3$, where W and H indicate the width and height of the input image, and 3 represents the three colour channels (RGB). The method transforms the problem to estimation of k heatmaps, $\{H_1, H_2, \ldots, H_k\}$, where each heatmap, H_k, indicates the location of the k^{th} keypoint.

The model consists of three main components, the stem, main body and regressor. The stem consists of two convolutional layers with kernel size of 3 and stride of 2, each followed by a batch normalization and ReLU activation function. This reduces the resolution of the input image and feeds the feature maps into the main body of the network which adopts the HRNet [49] architecture. The resolution of the feature maps is maintained in the main body. Therefore, the output feature maps have the same resolution as the input feature maps is produced by the stem. The HRNet adopted in this study consists of four parallel subnetworks as detailed by Sun et al. [50]. The number of channels in the high-resolution subnetworks is 48, and the width of the other parallel subnetworks are 96, 192, and 384, respectively. The output features are then fed to a regressor, which is a convolutional layer with 48 input channels, and K output channels for each of the keypoint heatmaps. Figure 5 illustrates the architecture of the network employed in this study.

2.2.2 Training details

The model is implemented using PyTorch, and the network training is carried out using an NVIDIA Tesla K80. Given the equipment bounding boxes, the synthetic images containing the excavators are cropped and rescaled to a fixed size of 384×288 to be fed to the model. The network outputs the

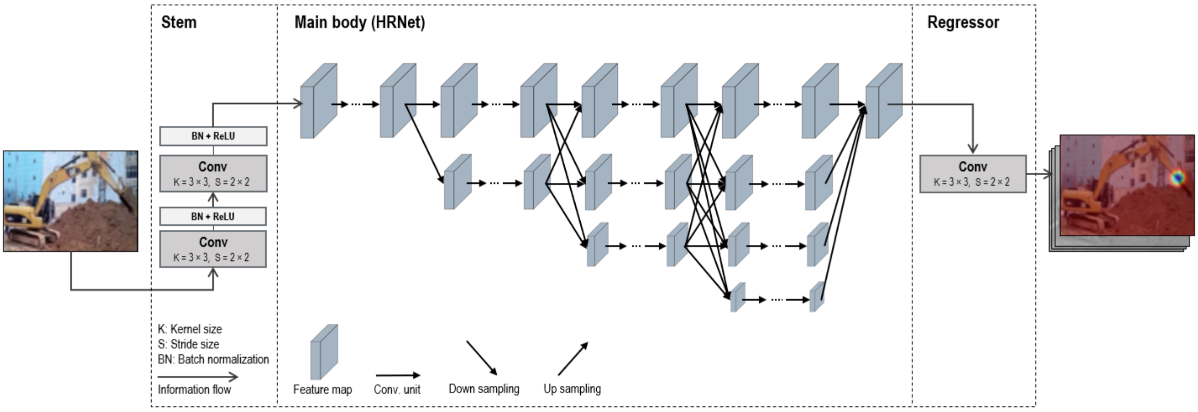


Fig. 5. The architecture of excavator pose estimation model.

predictions in the form of heatmaps of size 96×72. The loss function used for comparing the ground-truth heatmaps with the predictions is defined as the mean squared error (MSE), as shown in Eq. (1):

$$MSE = \sum_{i=1}^{n} \frac{\sum_{1}^{k} (y_{i,k} - y_{i,k}^{p})^2}{k} \tag{1}$$

where $y_{i,k}$ is the ground-truth coordinates of the k^{th} visible keypoint in the i^{th} sample, and $y_{i,k}^{p}$ is the predicted keypoint coordinates. The ground-truth heatmaps are generated by applying a 2D Gaussian centred on the keypoint coordinates with a 1-pixel standard deviation. Adam [51] optimizer is used with a learning rate of 0.001, determined empirically. As the general features learned by deep convolutional networks trained on large datasets are transferable for various computer vision tasks, in this study, the network is initialized using a model pre-trained on the COCO dataset [34] to reduce the training time and number of data required. To avoid overfitting to the synthetic training datasets, early stopping is employed. Training is stopped where the validation loss stops improving for two consecutive epochs. The training is terminated within 20 epochs in all experiments. Figure 6 shows the learning curve of the model trained on a synthetic dataset with 9k training images as an example.

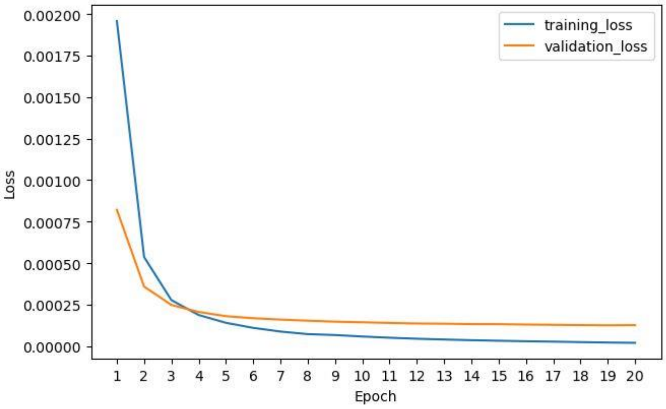
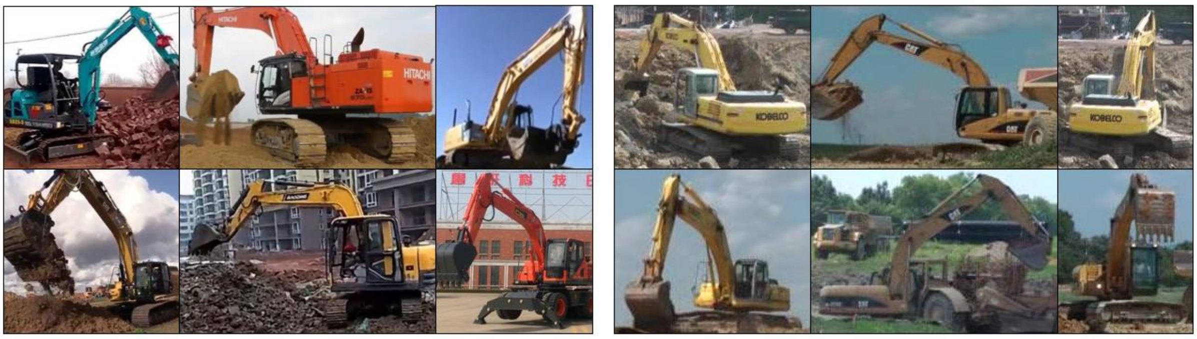


Fig. 6. The learning curve of the model trained on a synthetic dataset with 9k training images generated with full domain randomization.

2.3 Model evaluation

2.3.1 Evaluation datasets

The performance of the proposed pose estimation framework is evaluated on two datasets containing real images of excavators in the field. The first dataset used for evaluation (EvalSet1) is a subset of the dataset that has been made available by Luo et al. [29]. The dataset contains 1281 images in total. However, not all of the images contain annotations for all the keypoints, and thus only the fully annotated portion of the dataset, which contains 384 images with a variety of excavator types, is used in this study. The second evaluation dataset (EvalSet2) is based on a dataset, which contains video footage of a number of earthmoving operations and is made available by Roberts and Golparvar-Fard [52] for activity analysis of earthmoving equipment. As this dataset does not contain any pose information, 440 frames are extracted and annotated using an image annotation tool.



(a) EvalSet1

(b) EvalSet2

Fig. 7. A set of example images from the two datasets (EvalSet1 and EvalSet2) used for evaluation, both containing real images of excavators in the field.

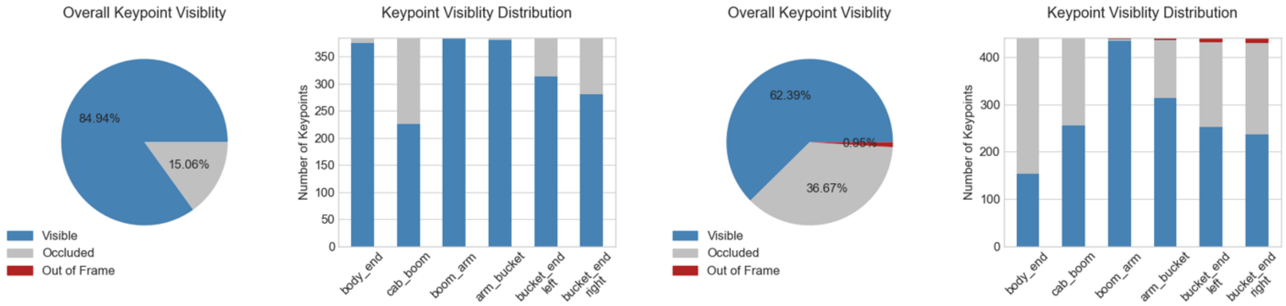
Both datasets contain the keypoints' position in the image frame, as well as the visibility status of the keypoints. Therefore, each keypoint annotation is represented by (x, y, v), where x and y represent the cartesian coordinates of the keypoints in the image frame, and v represents the visibility status. The visibility status value can take 1, 0 and -1, indicating "visible", "occluded", and "out of frame" keypoints, respectively. Figure 7 illustrates some representative examples from the two datasets. EvalSet1 contains a variety of excavator types including mini-excavators, wheel excavators, long-reach, standard and heavy excavators, and close-up images. EvalSet2 only contains standard excavators, and the images are captured from a distance, containing a variety of poses. Figure 8 provides some statistics regarding the visibility status of the keypoints in the two datasets.

2.3.2 Evaluation metrics

Normalized error (NE), and percentage of correct keypoints (PCK) are two commonly used evaluation metrics for keypoint estimation problems [29, 53]. These metrics are adopted to evaluate the performance of the equipment pose estimation models in this study.

Here, NE is defined as the average normalized Euclidean distance between the ground-truth keypoints and the predictions in pixels, normalized by the diagonal of the input image as shown in Eq. (2).

$$NE = \frac{1}{n}\sum_{i=1}^{n}\frac{\left\|y_{i,k} - y_{i,k}^{p}\right\|}{d_i} \tag{2}$$



(a) EvalSet1

(b) EvalSet2

Fig. 8. Statistics of the two evaluation datasets in terms of keypoint visibility status.

where $\|y_{i,k} - y_{i,k}^{p}\|$ represents the Euclidean distance of the prediction to the ground-truth annotation, k represents the k^{th} keypoint in the i^{th} sample, d_i denotes the diagonal of the i^{th} image before being rescaled to be fed to the network, and n is the total number of visible keypoints. Only the visible keypoints are considered in the calculation of the NE.

PCK measures the accuracy of keypoint localization by calculating the percentage of correctly identified keypoints for a given threshold, α. If the normalized distance between the prediction and the ground-truth keypoint is less than α, the keypoint is considered to be correctly located.

3. Experimental results

The following sub-sections describe the results of a number of pose estimation experiments. In all experiments, the model is trained solely on synthetically generated images of excavators and the evaluation is performed on the two datasets introduced in Section 3.3.1, both of which contain images of real excavators in construction fields. First, two training strategies are compared, in which the effect of inclusion or exclusion of occluded keypoints during training is investigated. This is followed by the evaluation of the influence of dataset size on model performance. Moreover, the performance of the model trained on the synthetic data is compared with that trained on real images. And finally, the impact of each DR parameter on model performance is explored.

3.1 Training on synthetic data

This sub-section presents the results of a number of experiments, conducted to evaluate the performance of the proposed framework for excavator pose estimation. In these experiments, the model is solely trained on synthetically generated datasets with no further fine-tuning on real annotated images in order to assess the extent of the proposed framework's capability in generalizing on real images. Six synthetic datasets of various sizes are generated with full DR. It means that all parameters of the simulator, such as pose, camera height and distance, lighting, field of view, textures, occluding objects, and dust simulation are randomized. The smallest and largest generated datasets contain one and 15 thousand training images, respectively. The synthetically generated datasets are split into training and validation sets with an 80/20 ratio. All training parameters are kept constant as described in section 3.2.2. In addition, the performances of the models trained on each of the generated datasets are evaluated on two real datasets (EvalSet1 and EvalSet2) using NE values and PCK curves.

3.1.1 Comparison of two training strategies

The proposed framework allows for generating pixel-level accurate annotations for all keypoints whether they are visible or occluded (including self-occluded keypoints). Due to the difficulty of locating keypoints that are not visible, it is not feasible to produce accurate annotations for occluded keypoints in manual annotation processes. Therefore, previous studies have only included visible keypoints during model training [7, 29].

As the synthetically generated datasets contain accurate annotations for both visible and occluded keypoints, the models are trained on each of the generated datasets twice to enable a comparison of two training strategies. The two training strategies are distinguished by inclusion or exclusion of occluded keypoints during training. Figure 9 shows the results of this experiment, where the dashed lines represent the results of the network trained only on visible keypoints, while the solid lines demonstrate the model performance when all keypoints are included during training. The results indicate that inclusion of occluded keypoints during training improves performance in terms of NE for both evaluation datasets (EvalSet1 and EvalSet2). As illustrated in Figure 9, the results point to a reduction of approximately 23.3% on average when the model performance is evaluated on EvalSet1, and a reduction of about 15.3% is observed when it is evaluated on EvalSet2.

3.1.2 The effect of dataset size

The main advantage of synthetic dataset generation with DR is the ability to produce large datasets. Therefore, the ability of the simulation to produce datasets with high variety is of utmost importance. A high data variety makes it less likely for the models to overfit on the synthetically generated training datasets, which helps the models to learn the essential features of the equipment, and view the real images as just another variation of the synthetic data observed during training.

Therefore, to evaluate the extent of data variability that the DR-based method can produce, the performance of the models trained on datasets of various sizes are compared. Figure 9 shows the result of the experiments on datasets of various sizes from 1k to 15k, for both training strategies. The results of the experiments when occluded keypoints were included during training, as shown in Figure 9, reveal that increasing the training dataset size from 1k to 3k, decreases the NE by 23.4% on average when the model performance is evaluated on the two evaluation datasets. Increasing the training dataset size from 3k to 9k only improved the performance on evaluation datasets marginally. A further reduction in NE values was observed with the datasets of size 12k; however, further increasing the dataset size beyond

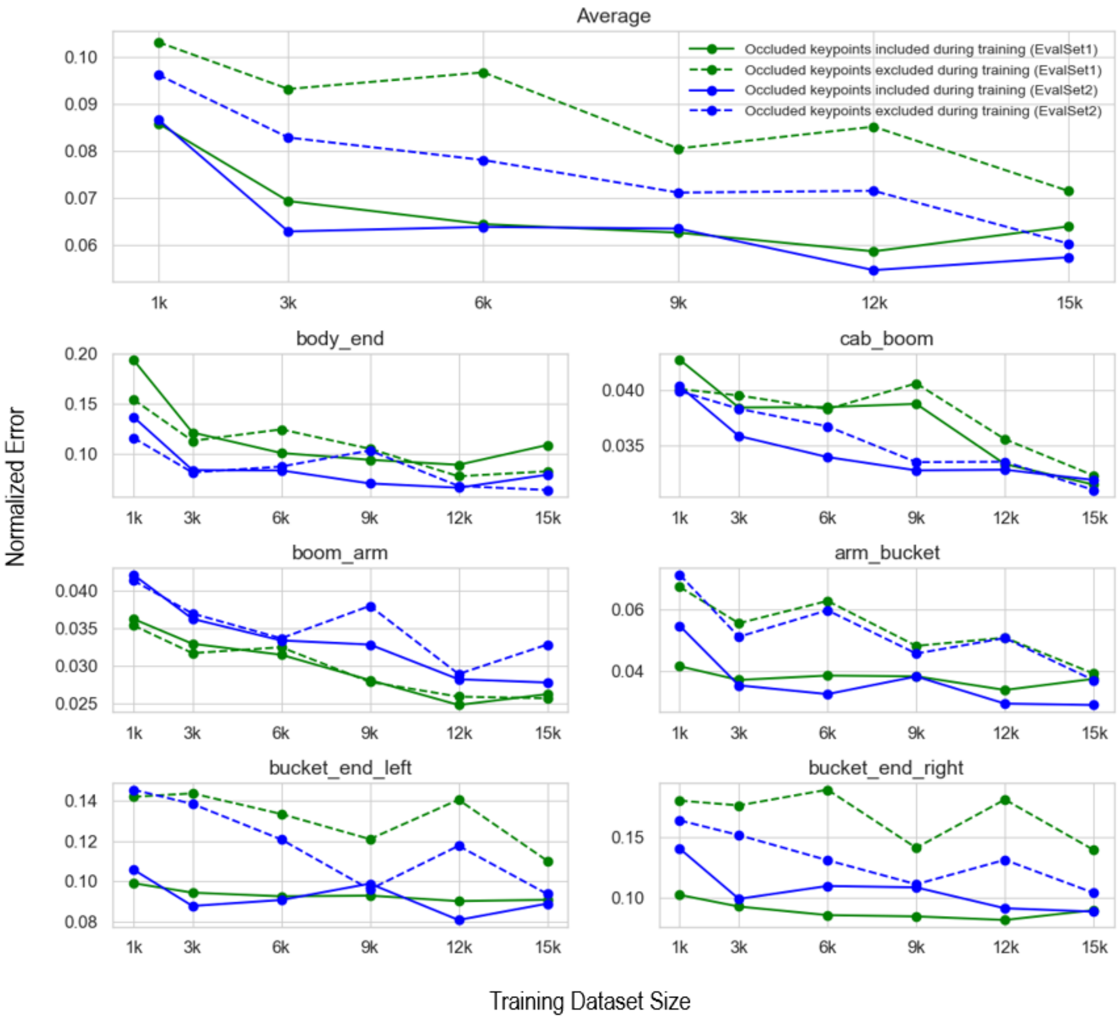


Fig. 9. Comparison of NE for models trained on various dataset sizes with full domain randomization, as well as direct comparison between two training strategies.

this point led to a reduction in the accuracy, which may be an indication that the model has started to overfit to the training dataset.

Table 1. The NE values for the model trained only on synthetic data and the model trained on real excavator images as evaluated on EvalSet2.

Training dataset	Body_end	Cab_boom	Boom_arm	Arm_bucket	Bucket_end_right	Bucket_end_left	Average
Synthetic	0.0669	0.0317	0.0273	**0.0282**	**0.0823**	**0.0916**	0.0498
Real	**0.0314**	**0.0287**	**0.0208**	0.0311	0.0992	0.0969	**0.0478**

3.2 Comparison with training on real images

To quantify the performance of the proposed framework, the results of training the HRNet-based pose estimation model on synthetically generated images are compared with the results of the same model trained on manually annotated images of real excavators.

The dataset of real excavators, which contains 1281 annotated images of excavators in the field, is prepared using the annotated images made available by Luo et al. [29]. However, as the dataset is relatively small in size, four data augmentation techniques, i.e. horizontal flip, random rotation, random translation, and colour inversion are applied to enlarge the dataset, increasing the number of images to 6405 in total. The dataset is split to 5,000, and 1,405 images for the training and validation sets, respectively.

As opposed to preparing datasets of real excavators, which requires manual annotation, creating large synthetic datasets using the proposed method is performed automatically. The model trained on the synthetic dataset with 12,000 images in the training set, and 3,000 images in the validation set is selected for comparison with the model trained on real images. Both models are trained using the same hyperparameters as described in Section 3.2.2, and early stopping is employed.

The evaluation is performed on EvalSet2, the details of which are described in Section 3.3.1. Table 1 provides a comparison of the two models in terms of NE values. The results indicate that the model which is solely trained on synthetic data has achieved comparable performance to the model trained on real images. The NE values for the synthetic dataset are only slightly lower that the real dataset on average. On three keypoints, namely the arm_bucket, bucket_end_right and bucket_end_left, it outperformed the model trained on real images, despite the fact that no real images are used during training for this model.

Furthermore, Figure 10 illustrates the PCK curves for the two models. The model trained only on the synthetic dataset is able to generalize considerably well to real images, even though no real images were seen by the network during training. However, accurate localization of the body_end keypoint, in

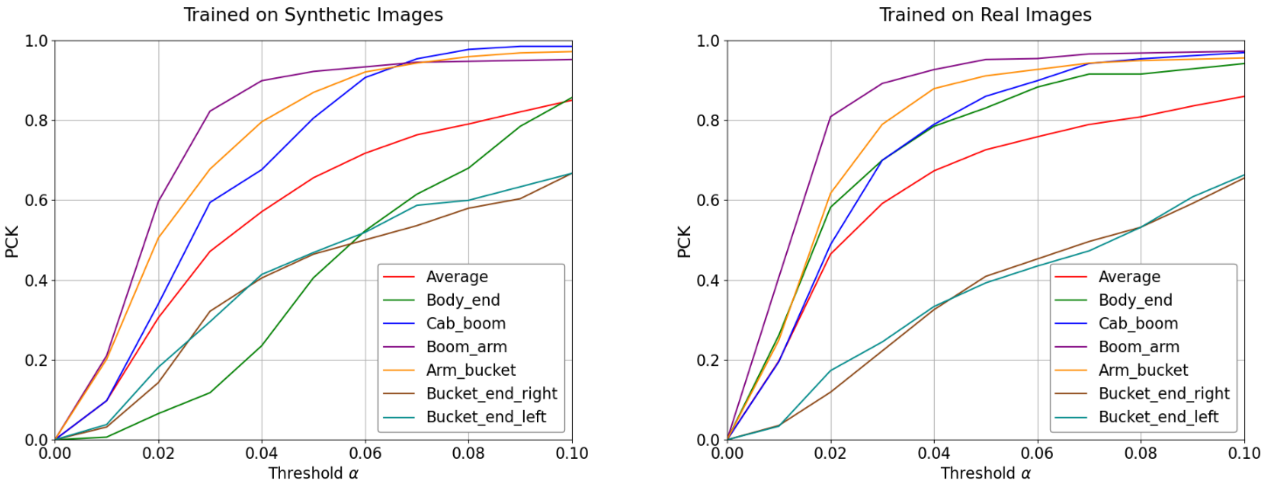


Fig. 10. The PCK curves for the model trained only on synthetic data (left) and the model trained on real images (right). Evaluation is performed on EvalSet2 dataset.

particular, appears to be challenging for the model trained on synthetic data as compared to the model trained on real images.

Figure 11 shows a number of sample predictions for the two models. The examples include various viewpoints, including front and rear views of the excavator, which are more challenging compared to side views. The examples illustrate better performance of the model trained on real images in detecting the location of body_end keypoint. However, as for the estimation of the bucket keypoints, which is the most challenging for both models to localise, the model trained on synthetic data is able to estimate the location of bucket keypoints with a higher accuracy.

3.3 Impact of DR parameters

To study the impact of the individual DR parameters, a systematic experiment is carried out, in which one randomization parameter in synthetic dataset generation is enabled at a time. For this study, the same model as described in Section 3.2.1 is used, and the hyperparameters are set as described in Section 3.2.2. For each experiment, 7,500 synthetic images are generated, 6,000 images are used for training, and the remaining 1,500 are used for validation. The performance of the models trained on each dataset are reported in terms of NE.

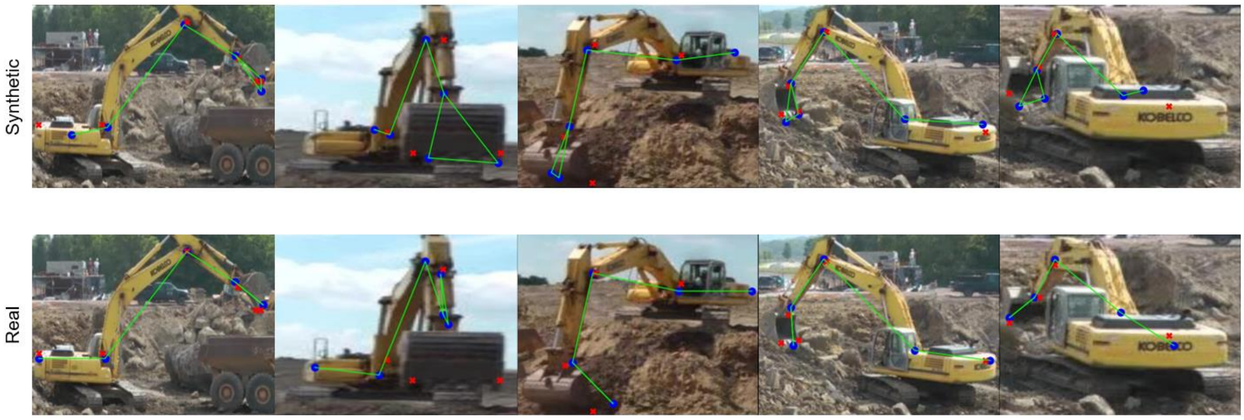


Fig. 11. Sample predictions for the model trained on synthetic images (top) and the model trained on real images (bottom). The predicted keypoint locations are shown with blue dots, and the ground-truth location of visible keypoints are depicted with red crosses.

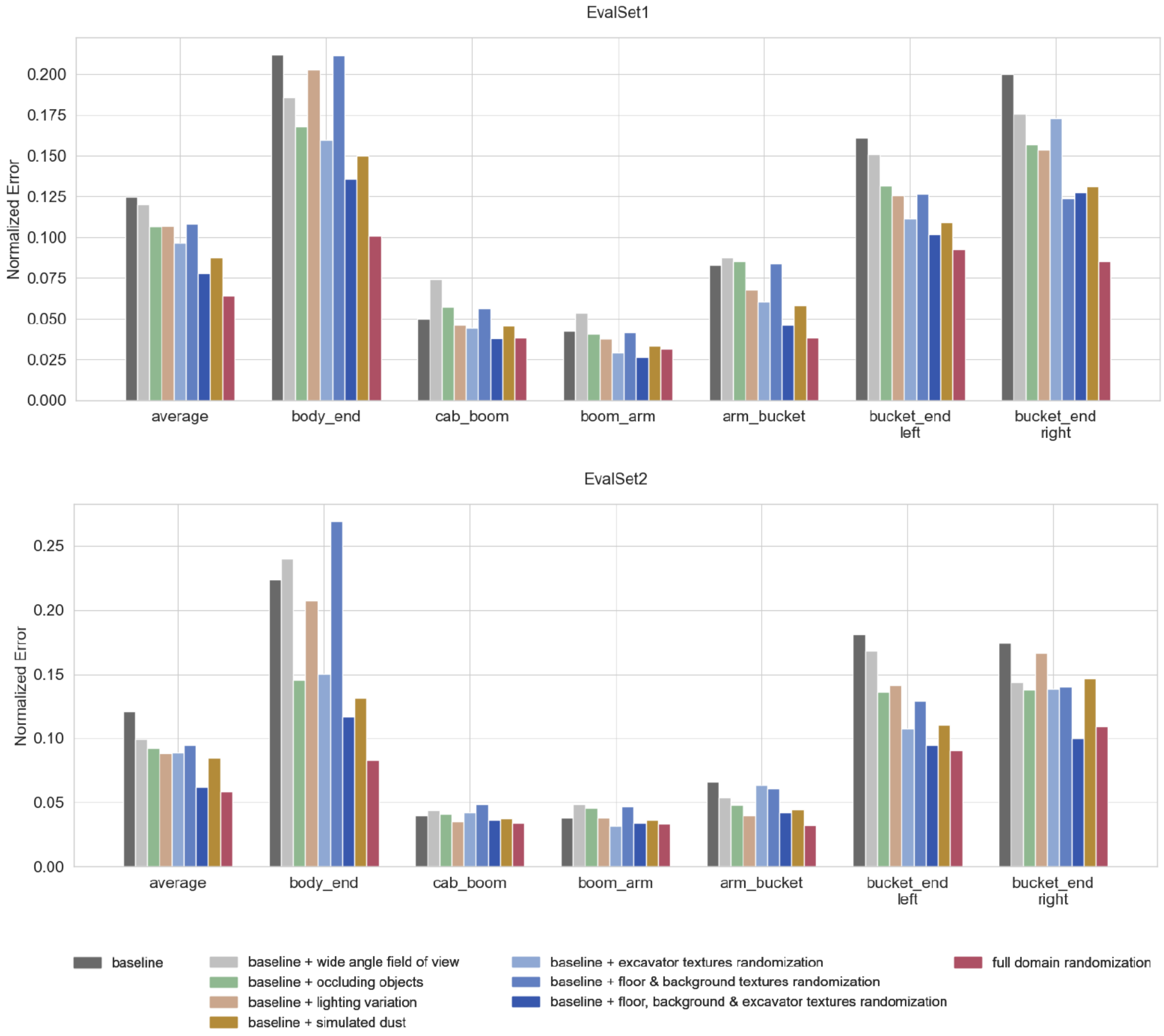


Fig. 12. Impact of enabling individual DR components in the process of synthetic data generation on NE as evaluated on two datasets of real excavators.

Figure 12 shows the result of enabling each DR parameter in generating the training dataset, showing the effect of each parameter on pose estimation performance on the two evaluation datasets. The effect of each parameter is compared with the baseline and full DR. The baseline synthetic data generation uses a single texture, for the excavator, floor and background of the scene, with no lighting variation, occluding objects or simulated dust. Only the excavator's pose and the camera's point of view is randomized. The field of view of the camera is only varied within a 20 to 60 degrees range, which is characteristic of standard cameras. On the other hand, the dataset using full DR was generated by enabling all DR parameters in the simulator, which includes lighting variation, addition of simulated dust and occluding objects. Furthermore, in the dataset generated with FDR, all textures are randomized, and the rendered images also include wide angle photos.

The individual parameters studied in this section are: The background and the floor textures, which indicate the importance of context; the excavator textures, which are purposefully created unrealistically to induce the variety in the dataset and force the network to learn features related to the shape of the equipment rather than their colour or texture; randomly generated occluding objects that are added to make the networks more robust to occlusions, which is a challenging problem for vision-based systems in construction; and simulated dust is also added to create robustness to visual noise. Furthermore, the effect of randomizing the lighting conditions, and inclusion of images with a wide-angle field of view are also studied. In all of the generated datasets, the pose of the excavator, its location, and the location of the camera and its orientation are randomized.

The results indicate that the inclusion of wide-angle field of view in data generation reduces the NE on EvalSet1 only by about 4%, whereas this reduction on EvalSet2 was more significant at about 18%. As compared to varying focal length of the camera in the simulation, the addition of lighting variations had a higher impact on model performance, where NE values on EvalSet1 and EvalSet2 were reduced by 14.3% and 26.7%, respectively. Similarly, the addition of occluding objects to images reduced the NE on EvalSet1 and EvalSet2 by 14.5% and 23.4%, respectively, as compared to the performance of the model on the baseline synthetic dataset. As suggested by the results, the addition of simulated dust was the most impactful non-texture related DR parameter in the simulation. By creating visual noise through simulated dust, the NE values on EvalSet1 were reduced by 29.5%, and similar improvement was observed on EvalSet2 with a reduction of about 30% on average NE.

To study the influence of textures on model performance, three scenarios where studied. The scenarios include 1) randomization of excavator textures while not varying the floor and background textures; 2) using a single excavator texture while randomizing the floor and background textures; and 3) randomizing all the textures applied to excavators, floor and background in the simulation. The results revealed that varying the excavator textures alone had a similar impact on the model performance as adding lighting variations and occluding objects, with 22.7% reduction in NE for EvalSet1, 26.1% for EvalSet2. The effect of randomizing floor and background textures with the use of a single excavator texture was slightly less significant, where the NE was reduced by 13.1% and 21.6% on EvalSet1 and EvalSet2, respectively. However, when all textures where randomized, the impact on model performance was the most significant compared to all DR-parameters on both of the evaluation datasets. Randomizing all textures reduced the NE on EvalSet1 by about 37%, while the improvement on EvalSet2 was more significant, where NE values dropped to almost half with a 48.8% decrease. Lastly, the best performance was achieved when all DR-parameters were enabled during synthetic data generation. However, the amount of improvement did not exceed 52% on both datasets, suggesting diminishing returns as more DR-parameters are enabled.

4. Discussion

The findings of this study suggest that by using only synthetically generated datasets, deep convolutional neural networks can be successfully trained to estimate the pose of construction equipment in real-world images. The major values obtained by utilizing the proposed framework are trifold. First, as compared to traditional data preparation pipelines, the proposed method does not require any manual annotation, which is a labour-intensive and time-consuming process. Second, synthetically generated datasets are advantageous to manually labelled datasets as they can produce pixel-level accurate annotations for the keypoints of interest. In contrast, manually annotated datasets are susceptible to human error, and in many cases different annotators may annotate the same image differently which leads to noisy datasets. Indicated by the results, presented in Section 4.1, inclusion of accurately labelled occluded keypoints in the training process had a considerable impact on model performance. This is due to the ability of the proposed method in producing accurate annotations for occluded keypoints, which is not feasible in most cases for human annotators. Third, by utilizing the proposed method, it not necessary to gather and annotate large training datasets for a specific object, and large datasets can be prepared for various construction equipment by taking advantage of the CAD model of the object of interest. This is a major advantage, in particular, in specialized fields such as construction, where limited data are available.

Using synthetic data for deep learning requires a precise understanding of the critical features that need to be present in the training dataset such that the model can generalize to real-world images. In the case of equipment pose estimation, the model must learn the features related to the shape of the equipment. Therefore, in the course of designing the simulator, rather than relying only on realistic textures, unrealistic textures are also purposefully included in order to force the network to learn shape related features, and therefore enable the model to view real images as just another variation of the synthetic data seen during training. In line with this presumption, the results of the experiment presented in Section 4.3 revealed the major impact of randomizing all textures in generating the synthesized images. Moreover, the addition of elements such as occluding objects, simulated dust and light variations during synthetic image generation also resulted in considerable performance gain when the model performance is evaluated on real images.

Overall, the efficacy of the proposed DR approach to training a deep learning model for pose estimation was shown. However, as indicated by the results, localization of certain keypoints of the excavators remain a challenge. For instance, the bucket_end keypoints were the most difficult to localize accurately both for the model trained on synthetic datasets and the model trained on real images. It is important to note that the proposed synthetic data generator employed in this study has a number of limitations. For instance, only a single excavator model was used in the simulator. Including a variety of excavator types can create more variations in the generated dataset, and therefore, improve the generalizability of the models trained on the synthetic datasets. Moreover, the simulator does not account for the equipment interaction with various materials, as is the case in real construction operations. Simulating such equipment-material interactions may further reduce the gap between synthetic data and real-world images. Synthetic datasets can also be used to initialize a network when an insufficient number of annotated real images is available. Although this study only provides a direct comparison between the model performances of training only on synthetic data and that of training on real images, further studies can explore the benefits of initializing the networks with synthetic data, and further tuning the model with real images.

5. Conclusion

The construction industry has continuously experienced lower than average safety and productivity enhancement as compared to other industries worldwide. Intelligent monitoring and data-driven decision making, powered by the recent advancements in computer vision and deep learning, offer promising non-intrusive solutions for process optimization and ensuring safety of operations. However, one of the key components for optimal performance of deep learning-based applications is the availability of large labelled training datasets, which is limited in specialized fields such as construction.

To overcome the laborious and costly process of manual data collection and annotation, this study presents a synthetic image generator developed using a game engine, and it employs DR to produce large and accurately annotated datasets for excavator pose estimation. The proposed method randomizes various critical features of the scene, such as excavator pose and texture, scene texture and lighting, camera location and field of view, and adds other elements, such as simulated dust and occluding objects, to the scene. A state-of-the-art deep convolutional neural network known as HRNet is adapted in this study. The quality of the synthetically generated datasets is assessed by training the model solely on the synthetic datasets, and evaluating their performance on images of real excavators in the field. Furthermore, the effect of various randomized parameters of the proposed synthetic data generator on model performance are evaluated. The results demonstrate the effectiveness of synthetic data for training convolutional neural networks for complex vision tasks such as pose estimation. The proposed method is a promising approach to overcome the limited data availability in construction, and can be applied to other construction resources.

While this study focused on the problem of pose estimation for excavators, the proposed framework is not limited to this equipment. Using the proposed synthetic data generation method, annotated datasets for a variety of construction equipment can be generated. Furthermore, while only six keypoints were used to define the full body pose of excavators, the automated annotation process allows easy expansion to dense pose estimation as no manual annotation is required in the preparation of the training datasets. The ability to generate complex and rich annotations enabled by synthetic data also paves the way to the design and implementation of simulators that can produce datasets, required for other computer vision tasks, such as semantic segmentation, depth estimation, and 3D pose estimation, in the context of construction.

6. References

[1] A. J. P. Tixier, M. R. Hallowell, B. Rajagopalan, D. Bowman, Automated content analysis for construction safety: A natural language processing system to extract precursors and outcomes from unstructured injury reports, Automation in Construction 62 (2016) 45-56, https://doi.org/10.1016/j.autcon.2015.11.001.

[2] National Census of Fatal Occupational Injuries in 2017, in U. S. B. o. Labor (Ed.), Washington, D.C., 2018.

[3] C. Igwe, F. Nasiri, A. Hammad, Construction workspace management: critical review and roadmap, International Journal of Construction Management (2020) 1-14, https://doi.org/10.1080/15623599.2020.1756028.

[4] J. Teizer, B. S. Allread, U. Mantripragada, Automating the blind spot measurement of construction equipment, Automation in Construction 19 (4) (2010) 491-501.

[5] S. G. Pratt, D. E. Fosbroke, S. M. Marsh, Building safer highway work zones; measures to prevent worker injuries from vehicles and equipment, (2001).

[6] S. Bang, Y. Hong, H. Kim, Proactive proximity monitoring with instance segmentation and unmanned aerial vehicle-acquired video-frame prediction, Computer-Aided Civil and Infrastructure Engineering (2021).

[7] C. J. Liang, K. M. Lundeen, W. McGee, C. C. Menassa, S. Lee, V. R. Kamat, A vision-based marker-less pose estimation system for articulated construction robots, Automation in Construction 104 (2019) 80-94, https://doi.org/10.1016/j.autcon.2019.04.004.

[8] A. Asadzadeh, M. Arashpour, H. Li, T. Ngo, A. Bab-Hadiashar, A. Rashidi, Sensor-based safety management, Automation in Construction 113 (2020) 103128, https://doi.org/10.1016/j.autcon.2020.103128.

[9] O. Golovina, J. Teizer, N. Pradhananga, Heat map generation for predictive safety planning: Preventing struck-by and near miss interactions between workers-on-foot and construction equipment, Automation in Construction 71 (2016) 99-115, https://doi.org/10.1016/j.autcon.2016.03.008.

[10] O. Golovina, M. Perschewski, J. Teizer, M. König, Algorithm for quantitative analysis of close call events and personalized feedback in construction safety, Automation in Construction 99 (2019) 206-222, https://doi.org/10.1016/j.autcon.2018.11.014.

[11] F. Vahdatikhaki, A. Hammad, H. Siddiqui, Optimization-based excavator pose estimation using real-time location systems, Automation in Construction 56 (2015) 76-92, https://doi.org/10.1016/j.autcon.2015.03.006.

[12] H. Luo, J. Liu, W. Fang, P. E. Love, Q. Yu, Z. Lu, Real-time smart video surveillance to manage safety: A case study of a transport mega-project, Advanced Engineering Informatics 45 (2020) 101100, https://doi.org/10.1016/j.aei.2020.101100.

[13] A. Assadzadeh, M. Arashpour, A. Bab-Hadiashar, T. Ngo, H. Li, Automatic far-field camera calibration for construction scene analysis, Computer-Aided Civil and Infrastructure Engineering (2021).

[14] S. Tang, M. Golparvar-Fard, M. Naphade, M. M. Gopalakrishna, Video-Based Motion Trajectory Forecasting Method for Proactive Construction Safety Monitoring Systems, Journal of Computing in Civil Engineering 34 (6) (2020) 04020041, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000923.

[15] X. Yan, H. Zhang, H. Li, Computer vision-based recognition of 3D relationship between construction entities for monitoring struck-by accidents, Computer-Aided Civil and Infrastructure Engineering (2020), https://doi.org/10.1111/mice.12536.

[16] E. Konstantinou, I. Brilakis, Matching Construction Workers across Views for Automated 3D Vision Tracking On-Site, Journal of Construction Engineering and Management 144 (7) (2018) 04018061, https://doi.org/10.1061/(asce)co.1943-7862.0001508.

[17] E. Konstantinou, J. Lasenby, I. Brilakis, Adaptive computer vision-based 2D tracking of workers in complex environments, Automation in Construction 103 (2019) 168-184, https://doi.org/10.1016/j.autcon.2019.01.018.

[18] M. M. Soltani, Z. Zhu, A. Hammad, Framework for Location Data Fusion and Pose Estimation of Excavators Using Stereo Vision, Journal of Computing in Civil Engineering 32 (6) (2018) 04018045, https://doi.org/10.1061/(asce)cp.1943-5487.0000783.

[19] J. Seo, S. Han, S. Lee, H. Kim, Computer vision techniques for construction safety and health monitoring, Advanced Engineering Informatics 29 (2) (2015) 239-251, https://doi.org/10.1016/j.aei.2015.02.001.

[20] E. R. Azar, C. Feng, V. R. Kamat, Feasibility of in-plane articulation monitoring of excavator arm using planar marker tracking, Journal of Information Technology in Construction (ITcon) 20 (15) (2015) 213-229.

[21] C. Feng, S. Dong, K. Lundeen, Y. Xiao, V. Kamat, Vision-based articulated machine pose estimation for excavation monitoring and guidance, ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction, Vol. 32, IAARC Publications, 2015, p. 1, https://doi.org/10.22260/ISARC2015/0029.

[22] K. M. Lundeen, S. Dong, N. Fredricks, M. Akula, V. R. Kamat, Electromechanical development of a low cost end effector pose estimation system for articulated excavators, ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction, Vol. 32, Citeseer, 2015, p. 1, https://doi.org/10.22260/ISARC2015/0081.

[23] C. Feng, V. R. Kamat, H. Cai, Camera marker networks for articulated machine pose estimation, Automation in Construction 96 (2018) 148-160, https://doi.org/10.1016/j.autcon.2018.09.004.

[24] K. M. Lundeen, S. Dong, N. Fredricks, M. Akula, J. Seo, V. R. Kamat, Optical marker-based end effector pose estimation for articulated excavators, Automation in Construction 65 (2016) 51-64, https://doi.org/10.1016/j.autcon.2016.02.003.

[25] E. Rezazadeh Azar, B. McCabe, Part based model and spatial-temporal reasoning to recognize hydraulic excavators in construction images and videos, Automation in Construction 24 (2012) 194-202, https://doi.org/10.1016/j.autcon.2012.03.003.

[26] M. M. Soltani, Z. Zhu, A. Hammad, Skeleton estimation of excavator by detecting its parts, Automation in Construction 82 (2017) 1-15, https://doi.org/10.1016/j.autcon.2017.06.023.

[27] C. Yuan, S. Li, H. Cai, Vision-based excavator detection and tracking using hybrid kinematic shapes and key nodes, Journal of Computing in Civil Engineering 31 (1) (2017) 04016038, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000602.

[28] J. Xu, H.-S. Yoon, Vision-based estimation of excavator manipulator pose for automated grading control, Automation in Construction 98 (2019) 122-131, https://doi.org/10.1016/j.autcon.2018.11.022.

[29] H. Luo, M. Wang, P. K.-Y. Wong, J. C. P. Cheng, Full body pose estimation of construction equipment using computer vision and deep learning techniques, Automation in Construction 110 (2020) 103016, https://doi.org/10.1016/j.autcon.2019.103016.

[30] A. Newell, K. Yang, J. Deng, Stacked hourglass networks for human pose estimation, European conference on computer vision, Springer, 2016, pp. 483-499, https://doi.org/10.1007/978-3-319-46484-8_29.

[31] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, J. Sun, Cascaded pyramid network for multi-person pose estimation, Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7103-7112, https://arxiv.org/abs/1711.07319.

[32] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436-444, https://doi.org/10.1038/nature14539.

[33] A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis, Deep Learning for Computer Vision: A Brief Review, Computational intelligence and neuroscience 2018 (2018), https://doi.org/10.1155/2018/7068349.

[34] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common Objects in Context, in: D. Fleet, T. Pajdla, B. Schiele and T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, Springer International Publishing, Cham, 2014, pp. 740-755, https://doi.org/10.1007/978-3-319-10602-1_48.

[35] M. Andriluka, L. Pishchulin, P. Gehler, B. Schiele, 2D Human Pose Estimation: New Benchmark and State of the Art Analysis, 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3686-3693, https://doi.org/10.1109/CVPR.2014.471.

[36] K. Mostafa, T. Hegazy, Review of image-based analysis and applications in construction, Automation in Construction 122 (2021) 103516, https://https://doi.org/10.1016/j.autcon.2020.103516.

[37] M. Zhang, R. Shi, Z. Yang, A critical review of vision-based occupational health and safety monitoring of construction site workers, Safety Science 126 (2020) 104658.

[38] K. Liu, M. Golparvar-Fard, Crowdsourcing construction activity analysis from jobsite video streams, Journal of Construction Engineering and Management 141 (11) (2015) 04015035, https://doi.org/10.1061/(ASCE)CO.1943-7862.0001010.

[39] Y. Wang, P.-C. Liao, C. Zhang, Y. Ren, X. Sun, P. Tang, Crowdsourced reliable labeling of safety-rule violations on images of complex construction scenes for advanced vision-based workplace safety, Advanced Engineering Informatics 42 (2019) 101001, https://doi.org/10.1016/j.aei.2019.101001.

[40] K. Han, M. Golparvar-Fard, Crowdsourcing BIM-guided collection of construction material library from site photologs, Visualization in Engineering 5 (1) (2017), https://doi.org/10.1186/s40327-017-0052-3.

[41] J. Kim, J. Hwang, S. Chi, J. Seo, Towards database-free vision-based monitoring on construction sites: A deep active learning approach, Automation in Construction 120 (2020) 103376, https://doi.org/10.1016/j.autcon.2020.103376.

[42] S. I. Nikolenko, Synthetic Data for Deep Learning, arXiv preprint arXiv:1909.11512 (2019), https://doi.org/arxiv.org/abs/1909.11512.

[43] Y. Toda, F. Okura, J. Ito, S. Okada, T. Kinoshita, H. Tsuji, D. Saisho, Training instance segmentation neural network with synthetic datasets for crop seed phenotyping, Communications biology 3 (1) (2020) 1-12, https://doi.org/10.1038/s42003-020-0905-5.

[44] D. Dwibedi, I. Misra, M. Hebert, Cut, paste and learn: Surprisingly easy synthesis for instance detection, Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1301-1310, https://arxiv.org/abs/1708.01642v1.

[45] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, R. Vasudevan, Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?, arXiv preprint arXiv:1610.01983 (2016), https://arxiv.org./abs/1610.01983.

[46] A. Tsirikoglou, J. Kronander, M. Wrenninge, J. Unger, Procedural modeling and physically based rendering for synthetic data generation in automotive applications, arXiv preprint arXiv:1710.06270 (2017), https://arxiv.org/abs/1710.06270.

[47] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, P. Abbeel, Domain randomization for transferring deep neural networks from simulation to the real world, 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE, 2017, pp. 23-30, https://doi.org/10.1109/IROS.2017.8202133.

[48] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, S. Birchfield, Training deep networks with synthetic data: Bridging the reality gap by domain randomization, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 969-977, https://doi.org/10.1109/CVPRW.2018.00143.

[49] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, Deep high-resolution representation learning for visual recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence (2020), https://doi.org/10.1109/TPAMI.2020.2983686.

[50] K. Sun, B. Xiao, D. Liu, J. Wang, Deep high-resolution representation learning for human pose estimation, Proceedings of the IEEE conference on computer vision and pattern recognition, 2019, pp. 5693-5703, https://doi.org/10.1109/CVPR.2019.00584.

[51] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014), https://arxiv.org/abs/1412.6980.

[52] D. Roberts, M. Golparvar-Fard, End-to-end vision-based detection, tracking and activity analysis of earthmoving equipment filmed at ground level, Automation in Construction 105 (2019), https://doi.org/10.1016/j.autcon.2019.04.006.

[53] Y. Chen, Y. Tian, M. He, Monocular human pose estimation: A survey of deep learning-based methods, Computer Vision and Image Understanding 192 (2020) 102897, https://doi.org/10.1016/j.cviu.2019.102897.

Appendix

Algorithm: Synthetic Dataset Generation for Excavator Pose Estimation

Inputs:

Excavator model

// Texture sets

$\mathcal{T}_e$: Excavator texture set, $\mathcal{T}_f$: Floor texture set

$\mathcal{T}_b$: Background texture set, $\mathcal{T}_o$: Occluding objects texture set

Outputs:

Set of synthetically generated images

Labels for each image, which includes 2D keypoint coordinates in the image frame, visibility status of the keypoints (visible vs occluded), excavator bounding box coordinates, and camera parameters

```
Initialize resources // load excavator model, backgrounds, and textures
for i = 1 to synthetic_dataset_size do

    procedure RandomizeFloorTexture(enable = True or False)
        if enable = True
            Randomly select floor texture, $T_f$, from set of floor textures, $\mathcal{T}_f$
        else
            Set default floor texture, $T_{f\_default}$
        end
        Apply floor texture
    end
    procedure RandomizeBackgroundTexture(enable = True or False)
        if enable = True
            Randomly select background texture, $T_b$, from set of background textures, $\mathcal{T}_b$
        else
            Set default floor texture, $T_{b\_default}$
        end
        Apply background texture
    end
    procedure RandomizeCamera(lens_type = “standard” or “wide”)
        if lens_type = “standard”
            Randomly set camera field of view from the range [20, 60] // units in degrees
            Randomly set camera height (Z) from the range [2, 20]      // units in meters
            Randomly set camera offset (X, Y) from the range [15, 35] // units in meters
            Apply camera rotation to have the camera facing the center of the scene
        elseif lens_type = “wide”
            Randomly set camera field of view from the range [60, 90] // units in degrees
            Randomly set camera height (Z) from the range [2, 10]      // units in meters
            Randomly set camera offset (X, Y) from the range [8, 14]   // units in meters
            Apply camera rotation to have the camera facing the center of the scene
        end
        Insert camera object
    end

    procedure LightingRandomization(enable = True or False)
        if enable = True
            Randomly set light source position and orientation
            Randomly set light intensity
            Randomly set vary light source color with a probability, $P_1$ = 0.2
        else
            Set default lighting
        end
        Insert light source object
    end

    procedure OccludingObjectRandomization(enable = True or False)
        if enable = True
            Randomly select occluding object texture, $T_o$, from set of textures, $\mathcal{T}_o$
            Randomly set occluding object scale
            Randomly set occluding object location
            Insert occluding objects
            else
                Pass
            end
    end

    procedure SmokeGeneratorObjectRandomization(enable = True or False)
        if enable = True
            Let $P_2$ = 0.5 be the probability of creating a smoke generator object
            Let x be a random number in the range [0, 1]
            if  $x <= P_2$
                Let $P_3$ = 0.1 be the probability of varying the smoke color
                Let $y$ be a random number in the range [0, 1]
                if  $y <= P_3$
                    Randomly set smoke object’s hue
                else
                    Set default hue for the smoke generator object
                end
                Randomly set smoke intensity
                Randomly set smoke object location
            end
            Insert smoke generator object
        else
            Pass
        end
    end

    procedure ExcavatorGenerator(texture_randomization = True or False)
        Let $\theta_1, \theta_2, \theta_3$ and $\theta_4$ be the four angles defining the pose of the excavator
        Randomly set $\theta_1, \theta_2, \theta_3$ and $\theta_4$ from the operable range
        if  texture_randomization = True
            Randomly select excavator texture, $T_e$, from set of textures, $\mathcal{T}_e$
        else
            Set default excavator texture, $T_{e\_default}$
        end
        Randomly set excavator location and orientation
        Insert excavator object
    end

    procedure DatasetGenerator()
        Get 2D keypoints' coordinates in the image frame, *keypoints_coords*
        Get the visibility status of the keypoints ("visible" or "occluded"), *keypoints_vis*
        Get the bounding box coordinates around the excavator in the image, *bbox_coords*
        Get camera parameters, *camera_params*
        Render image, $I_i$
        Write the image, $I_i$ to file
        Append $keypoints\_coords_i$ to CSV file
        Append $keypoints\_vis_i$ to CSV file
        Append $bbox\_coords_i$ to CSV file
        Append $camera\_params_i$ to CSV file
    end
end
```