Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2021.013795
Article

Tech Science Press

Generation of Synthetic Images of Randomly Stacked Object Scenes for Network Training Applications

Yajun Zhang[1,*], Jianjun Yi[1], Jiahao Zhang[1], Yuanhao Chen[1] and Liang He[2]

[1]East China University of Science and Technology, Shanghai, 200237, China
[2]Shanghai Aerospace Control Technology Institute, Shanghai, 201109, China
*Corresponding Author: Yajun Zhang. Email: yajunzhang1993@163.com
Received: 30 August 2020; Accepted: 15 November 2020

Abstract: Image recognition algorithms based on deep learning have been widely developed in recent years owing to their capability of automatically capturing recognition features from image datasets and constantly improving the accuracy and efficiency of the image recognition process. However, the task of training deep learning networks is time-consuming and expensive because large training datasets are generally required, and extensive manpower is needed to annotate each of the images in the training dataset to support the supervised learning process. This task is particularly arduous when the image scenes involve randomly stacked objects. The present work addresses this issue by developing a synthetic training dataset generation method based on OpenGL and the Bullet physics engine which can automatically generate annotated synthetic datasets by simulating the freefall of a collection of objects under the force of gravity. Rigorous statistical comparison of a real image dataset of staked scenes with a synthetic image dataset generated by the proposed approach demonstrates that the two datasets exhibit no significant differences. Moreover, the object detection performances obtained by three popular network architectures trained using the synthetic dataset generated by the proposed approach are demonstrated to be much better than the results of training conducted using a synthetic dataset generated by a conventional cut and paste approach, and these performances are also competitive with the results of training conducted using a dataset composed of real images.

Keywords: Synthetic dataset; stacked object scenes; OpenGL; Bullet physics engine; image recognition; parts position

1 Introduction

Image recognition has been increasingly and widely applied in fields such as transportation, industrial detection, and animal and plant identification. Conventional image recognition methods are limited by their reliance on distinct features and characteristics of images to detect objects [1]. However, the development of deep learning technology has greatly enhanced the accuracy and efficiency of image recognition methods for identifying comprehensive information regarding object categories and positions within images. Image recognition methods based on deep learning generally apply large training datasets for automatically

learning the effective feature representations of objects that enable the classification of individual objects within images. However, the task of training deep learning networks is time-consuming and expensive because large training datasets are generally required, and extensive manpower is needed for annotating each of the images in the training dataset to provide ground truth data supporting the supervised learning process. This task is particularly arduous when the image recognition features involve pixel-level segmentation, or when the image scenes involve randomly stacked objects.

This issue can be potentially addressed by replacing the large collection of annotated images in the training dataset with synthetic images that are generated and automatically annotated using computational methods, and then training the model using the synthetic training dataset for the recognition of objects in actual image scenes. Two approaches are commonly used to construct synthetic datasets. The first approach simply cuts and pastes an instance of a target object onto a random background [2]. The second approach obtains synthetic images by the programmed modeling of virtual scenes meeting given specifications [3]. Each of these approaches have advantages and disadvantages.

The cut and paste method has been widely applied. For example, semi-synthetic image data were generated by combining dry pulp sheet backgrounds with typical dirt particles patches to train a detection and classification system for realizing the automated quality evaluation of dry pulp sheets [4]. Gupta et al. [5] employed this method to generate image datasets of text, and thereby extended text recognition from a single letter to entire words. Su et al. [6] explored two-dimensional (2D) image target viewpoint estimation by training a convolutional neural network (CNN) using synthetically rendered objects cut and pasted into the images of the training dataset. Although this method is convenient and fast, pixel artifacts associated with the pasted edges in the generated images can cause the trained model to perform poorly. Dwibedi et al. [7] addressed this issue by blurring the pasted edges to ensure that the model was not trained with these features, and the generated synthetic data were demonstrated to be competitive with actual images. Other work addressed this issue by replacing the extracted object instances with three-dimensional (3D) models of objects [6,8–10].

Most techniques based on the modeling approach employ a game engine for building virtual scenes [11–13]. For example, virtual scenes were built using the Unreal Engine4 game engine, and the generated data were applied to a robot environment perception experiment [14]. Handa et al. [15] applied depth data synthesized through rendering to address the problem of indoor scene understanding, which is a prerequisite for many advanced tasks of automated intelligent machines. A game simulation engine was used to simulate the driving conditions of automobiles on roadways [16,17]. Data collected for training a deep learning network was applied to an automatic driving application. Applications in these conditions can benefit greatly from the use of virtual scenes because the scene parameters, such as camera angle, light conditions, and weather conditions, can be arbitrarily changed to provide datasets that include a wide range of scenarios. The virtual scene rendering method has far-reaching significance in a wide variety of fields, such as autonomous driving [16,18], scene perception [15,19], unmanned aerial vehicle tracking [20], and object recognition [4,21–23].

While both of the above-discussed image synthesis approaches have provided significant benefit for generating annotated image datasets with greatly reduced workload, these approaches have been mainly applied for generating sparse-object scenes, such as those involving individual vehicle, text, and human objects, and very few studies have focused on the synthesis of image data for scenes involving randomly stacked objects. Moreover, the cut and paste approach is not suitable for generating scenes of randomly stacked objects because the pixel artifacts associated with the pasted edges in the generated images cannot be effectively addressed under stacked object conditions. Furthermore, the modeling approach suffers from serious complications because the poses of the various objects in a synthetic dataset typically depend on the pose of the original model, and the random collection of the objects will not conform to

the distribution observed in a natural state. As such, the synthetic stacked object images will deviate from natural images, and thereby limit the object recognition accuracy of the model trained using the synthetic dataset. No image synthesis approach has yet been proposed that can be effectively applied to complex stacked object scenes, and thereby random fetching operations of stacked objects remains a challenging task for robot automation in industry. As a result, industrial robot applications typically require that items be in a stack-free state. While this can be addressed by the use of additional equipment, such as a vibration disc added between the part bin and grabbing platform, to separate the individual parts, this can make the system inefficient and expensive. Clearly, image recognition based on deep learning represents a promising and powerful solution to this problem. However, as discussed, the annotation of stacked objects in images is particularly tedious and expensive. Therefore, the use of training datasets composed of synthetic images would greatly improve the efficiency of the learning process.

The above-discussed issue is addressed in the present work by proposing a new framework that automatically generates an annotated synthetic dataset of images comprising complex stacking scenes based on OpenGL and the Bullet physics engine by simulating the freefall of a collection of objects under the force of gravity. The proposed approach can quickly output a large number of well-annotated images that faithfully represent the distributions observed in a natural stacking state. Rigorous statistical comparison of a real image dataset of staked scenes with a synthetic image dataset generated by the proposed approach demonstrates that the two datasets exhibit no significant differences. Moreover, the object detection performances obtained by three popular network architectures trained using the synthetic dataset generated by the proposed approach are demonstrated to be much better than the results of training conducted using a synthetic dataset generated by a conventional cut and paste approach, and these performances are also competitive with the results of training conducted using a dataset composed of real images.

The remainder of this paper is structured as follows. Section 2 introduces the proposed synthetic image dataset generation approach using OpenGL and the Bullet physics engine. Section 3 explores the differences between a synthetic dataset generated by the proposed approach and a dataset composed of actual images. The training performance of the proposed synthetic dataset generation approach is analyzed for three popular network architectures from various perspectives in Section 4. Finally, we draw the conclusions of the paper in Section 5.

2 Synthetic Dataset Generation Approach

The framework of the proposed synthetic dataset generation approach is illustrated in Fig. 1. Briefly, a 3D computer aided design (CAD) model of the principle object is obtained, and 12 copies of these objects are input into the OpenGL simulation platform. Here, common 6060 aluminum corner connectors are employed as an example. Then, the Bullet physics engine, which is a cross-platform, open-source physics engine that supports 3D collision detection and flexible, rigid-body dynamics, is applied to build the collective structure of randomly stacked objects. We then capture red-green-blue (RGB) channel images by a virtual camera placed within the simulation platform with a specific position and posture under predetermined lighting conditions. The detailed process is given as follows.

Establish object models. The 3D CAD model of the object is built using the SOLIDWORKS platform. Models of more complex parts can be obtained through 3D reconstruction. The model is textured and rendered by UV mapping in 3Ds Max software to enable it to be rendered in OpenGL with a realistic appearance and illumination state.

Import the model. The Assimp library is applied to load model files with a consistent data structure to an OpenGL 3D simulation scene. The object model is loaded into the scene in a preset fixed position with 6 degrees of freedom.

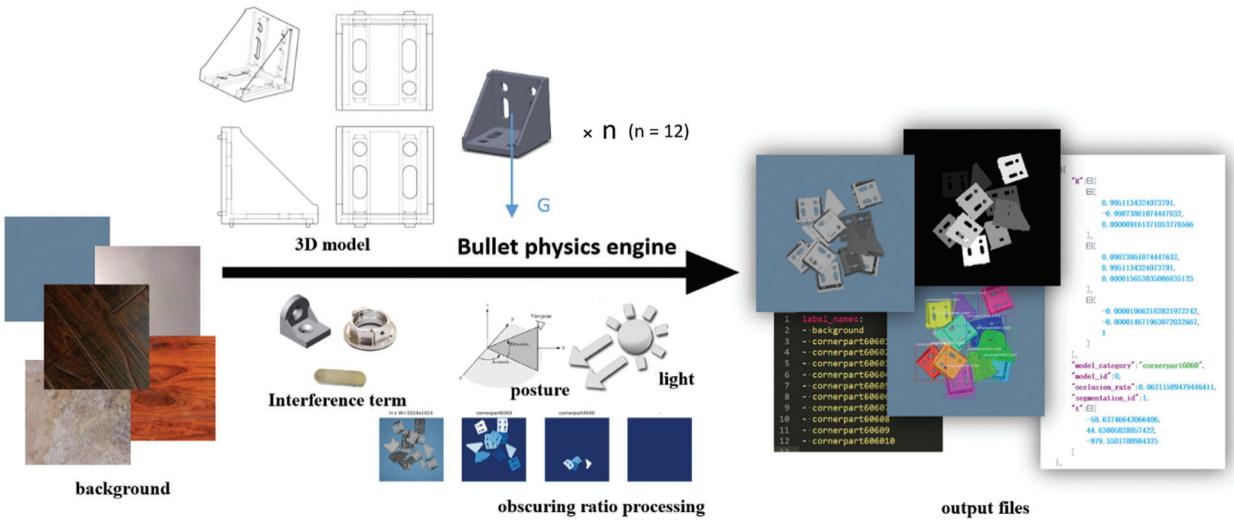


Figure 1: Framework for generating synthetic datasets of randomly stacked object scenes

Construct the stacking scene. Stacking scenes caused by gravity, collision, and other factors are simulated by constructing a container in OpenGL (similar to the bins used in industry). The Bullet engine is applied to simulate the freefall of models from a fixed point and their collisions with each other to form an object stacking scene.

Load the virtual camera. Like actual cameras used to capture real scenes, virtual cameras must be configured in a specific position and posture within virtual 3D simulation scenes, and the lighting conditions must also be specified in OpenGL.

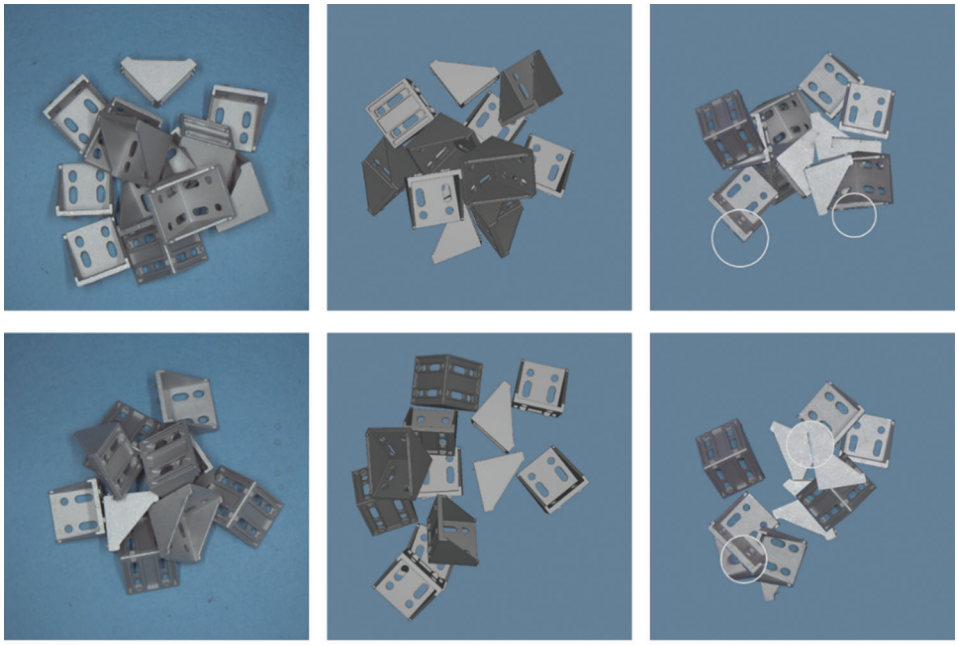
Capture synthetic image. The RGB image corresponding to the scene rendered from the viewpoint of the virtual camera is captured, and segmentation annotation, depth, and all other corresponding information are recorded according to the position and orientation of each object model in the scene.

In addition, different background images are applied to the virtual scenes to ensure that the deep learning model trained by our synthetic dataset ignores features associated with background patterns. Moreover, we increase the robustness of the deep learning model by randomly adjusting the parameters of the synthesis process to vary the lighting and camera parameters.

3 Comparison of Datasets

Figs. 2(a)–2(c) respectively present example images of randomly stacked 6060 corner connectors based on actual images, synthetic images generated by the proposed approach, and synthetic images generated by the cut and paste (C&P) approach [5,6]. Here, the cut and pasted image scenes were assembled from images of 3D models of a single connector in various postures without considering the mutual relationships of the parts. We note that the synthetic images generated by the proposed method cannot be effectively distinguished from the real images, while the synthetic images generated by the C&P approach appear to be considerably more artificial. In addition, artifacts are clearly observable in the images, as indicated by the circles in Fig. 2(c).

In terms of annotation, the real dataset was manually annotated using *labelme*, which is an image annotation tool that can annotate images at the pixel level. However, because of errors in manual labeling process and the inherent limitations of *labelme* (e.g., the holes on the corner connectors cannot be removed), the automatic annotation obtained by the proposed approach has advantages over the real dataset with respect to representing the ground truth information.



(a) (b) (c)

Figure 2: Example images of randomly stacked corner connectors: (a) Actual images; (b) Synthetic images generated by the proposed approach; (c) Synthetic images generated by a cut and paste (C&P) approach

The level of congruence between the synthetic dataset generated by the proposed approach and the real dataset was evaluated statistically by comparing the 3D and 2D posture distributions of the two datasets, as shown in Fig. 3. Both datasets included more than 200 images composed of 12 randomly stacked corner connectors. As illustrated in Fig. 3(a), the posture of each corner connector in each image of both datasets was defined according to the angles α, β, and γ that are respectively measured with respect to the Z, Y, and X axes with the origin located at the center of the bottom connector plate. The posture distributions of the synthetic and real datasets are respectively presented in Fig. 3, where the plots at left are 3D (α, β, γ) scatter diagrams, while the plots at right are 2D (β, α) and (β, γ) projections and their corresponding frequency cumulative diagrams.

The posture distributions of the two datasets are obviously similar. Several densely distributed postures are observed in the diagrams because the probability of the connectors landing on one of their flat surfaces is much greater than those of any other postures. This is particularly evident in the frequency cumulative distribution diagrams of the 2D (β, α) projections, where peaks can be observed at three α angles and three β angles. The intersecting lines formed by these faces represent the most obvious lines in the 3D distributions. We also note that the frequency cumulative distribution diagrams of the 2D (β, γ) projections represent normal distributions with respect to the angle γ.

To analyze the similarity of the two datasets quantitatively, we note that the two sets of data are inevitably biased. Many factors can lead to deviations between the statistical results, and the most important factors is expected to be errors in the measurement results of the acceleration sensor employed for determining the connector postures in the real scenes. Therefore, we must eliminate the influence of this factor when seeking to quantify the similarity of the distributions. This is addressed by adopting normalized and standardized methods for evaluating the consistency of the distributions of α, β, and γ angles. Accordingly, we applied a Kolmogorov–Smirnov test with two sample sizes n of n_1 = 2,400 and

n_2 = 2,000, and a significance level of 0.05. This test begins by processing the (α, β, γ) coordinates of each connector as follows.

$$(\alpha', \beta', \gamma') = \frac{(\frac{(\alpha,\beta,\gamma) - (\alpha,\beta,\gamma)_{\min}}{(\alpha,\beta,\gamma)_{\max} - (\alpha,\beta,\gamma)_{\min}}) - \frac{1}{n}\sum_{i=1}^{n}(\alpha,\beta,\gamma)_i}{\sqrt{\frac{\sum((\alpha,\beta,\gamma) - \frac{1}{n}\sum_{i=1}^{n}(\alpha,\beta,\gamma)_i)^2}{n-1}}} \tag{1}$$

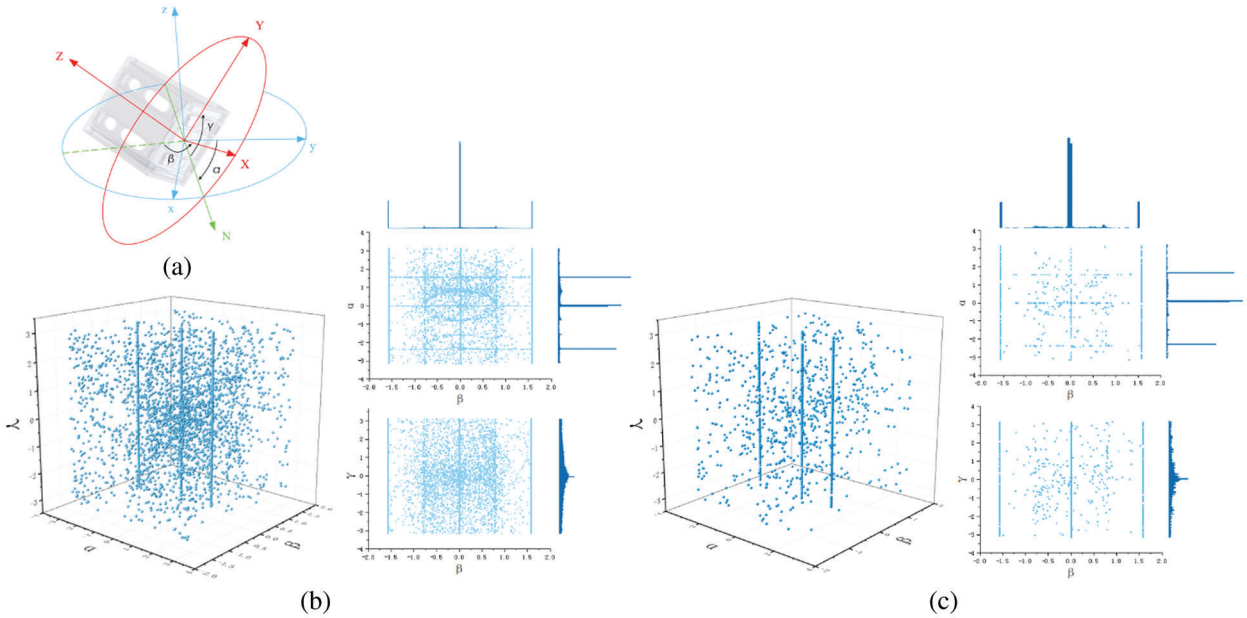


Figure 3: Corner connector posture distributions: (a) Object coordinate system; (b) Posture distribution plots of the synthetic dataset generated by the proposed approach; (c) Posture distribution plots of the real dataset. Here, the plots at left are 3D (α, β, γ) scatter diagrams of connector posture, while the plots at right are 2D (β, α) and (β, γ) projections and their corresponding frequency cumulative diagrams

After preprocessing the data, the following hypotheses H_0 and H_1 are proposed.

H_0: The two sets of data conform to the same distribution.

H_1: The two sets of data correspond to different distributions.

We then define the difference of the cumulative frequency at data x_j as follows:

$$D_j = F_1(x_j) - F_2(x_j) \tag{2}$$

where $F_1(x_j)$ and $F_2(x_j)$ are cumulative frequency functions of two groups of data. The statistic Z is then calculated according to the maximum value of D_j:

$$Z = \max_j |D_j| \sqrt{\frac{n_1 n_2}{n_1 + n_2}} \tag{3}$$

$$p_{value} = e^{-2z^2 - \frac{2z(n_1+2n_2)}{\sqrt[3]{n_1 n_2 (n_1+n_2)}}} \tag{4}$$

Then, the values of P_{Value} are computed for the angles α, β, and γ based on the Z value. This yields the following results.

$$P_{Value}, \alpha = 6.4 \times 10^{-11} \tag{5}$$

$$P_{Value}, \beta = 2.7 \times 10^{-10} \tag{6}$$

$$P_{Value}, \gamma = 0.0082 \tag{7}$$

Accordingly, the values of P_{Value} for angles α, β, and γ are all less than the required significance level of 0.05, which demonstrates that the posture distributions of the two datasets exhibit no significant differences. As such, hypothesis H_0 is accepted, and the synthetic data effectively simulates the stacking postures of objects obtained under actual conditions. Moreover, the convenience of the proposed synthetic dataset generation approach enables the incorporation of a much great number of annotated object postures.

4 Evaluation

We evaluated the performance of the proposed synthetic dataset generation approach from various perspectives. First, we investigated the object detection performance of deep learning models trained using a real dataset and synthetic datasets. Second, we conducted ablation studies reflecting the effect of introducing variations in environmental conditions when generating the synthetic dataset using the proposed approach on the object detection performance of a deep learning model. Finally, we investigated the effects of different training strategies on the object detection performance of a deep learning model trained using synthetic datasets generated by the proposed approach.

Unless otherwise specified, we adopted training datasets composed of 2,000 RGB images of stacked scenes comprising twelve 6060 corner connectors. As discussed, random variations in image backgrounds, lighting conditions, and added noise provides synthetic images with greater diversity for focusing deep learning networks on the characteristics pertaining to the objects themselves rather than on meaningless artifacts in the images. Therefore, these random variations were uniformly applied to all synthetic datasets, unless otherwise specified. In addition, we applied some data-enhancement methods, such as flip, crop, zoom, affine transformation, and random erasure, for each dataset. The testing dataset was composed of 50 real images of stacked scenes comprising twelve 6060 corner connectors, all of which were manually annotated using *labelme*.

The mean intersection over union (MIoU) was adopted as the evaluation standard during segmentation. Here, IoU = TP/(TP + FP + FN), where TP, FP, and FN represent true positive, false positive, and false negative results, respectively. Correct object detection was established when the IoU between a predicted bounding box (mask) and a ground-truth bounding box (mask) was greater than 0.6. We also prevented model overfitting by terminating the training process when the object detection performance became saturated.

4.1 Object Detection

Three popular neural networks were trained using a real dataset and synthetic datasets generated by the proposed approach and by the C & P approach. The datasets were converted to the standard COCO format to facilitate training for the open-source implementations of the three networks. The network architectures are briefly described as follows.

YOLOv3 is the third version of the YOLO (you only look once) series of target detection algorithms that have demonstrated significant improvements in accuracy and speed [24]. The network divides an image into $S \times S$ grids, and the grid at the object center is responsible for object detection. YOLOv3 consists of 75 convolutional layers and applies no fully connected layer. Due to the absence of a fully connected layer, the network can accommodate input images of any size. During training, the learning rate and momentum were set to 0.002 and 0.9, respectively.

Mobilenet-SSD is a lightweight network designed for mobile terminals. Accordingly, it uses the Mobilenet [25] feature extraction network rather than vgg16, as used in the original single shot detector (SSD) [26]. Mobilenet-SSD has the advantages of high speed, few network parameters, and robust detection accuracy, and is suitable for deployment in embedded and edge computing devices in industrial scenarios. We trained the Mobilenet-SSD network with a learning rate of 0.015 and a momentum of 0.9.

Mask R-CNN adds an additional branch to the original framework of Fast region-based CNN (R-CNN), and uses a small fully convolutional network to obtain the mask of a predicted object while simultaneously predicting classifications and bounding boxes [27]. The instance segmentation of an object can be obtained through a Mask R-CNN model. This pixel-level segmentation requires annotation information of the training dataset at the pixel level. Clearly, the application of synthetic datasets can substantially facilitate the training of such neural networks. We trained the Mask R-CNN model with an initial learning rate of 0.001, a weight decay factor of 0.0001, and a momentum of 0.9.

Two datasets were respectively generated by the proposed approach and the C&P approach. In addition, another training dataset composed of 200 real images captured using a Basler scA1600-20gc area scan camera from actual stacked scenes comprising 12 corner connectors was adopted during object detection performance evaluations, and the images were manually annotated by *labelme*. Although the scale of the synthetic data was greater than that of the real data, these synthetic images came essentially for free, as they were generated automatically. In fact, only 30 min were required to synthesize the 2000 images in the training dataset using the proposed approach, which was only one-eleventh of the approximately 5.5 h required for manually labeling the real images. While the datasets were smaller than those generally employed in similar studies, they were sufficient to provide meaningful comparisons.

The MIoU values obtained by the three network architectures when training using each of the three training datasets are listed in Tab. 1. The model trained using the real dataset performed best because the images in this training dataset were obtained in the same manner as the images in the testing dataset. Nonetheless, the MIoU values obtained by the networks trained using the synthetic dataset generated by the proposed approach, while slightly less, were competitive. However, taking the greatly reduced time cost into account, the slightly diminished detection accuracy can be regarded as an acceptable tradeoff. In contrast, the object detection performances of these three frameworks trained using the synthetic dataset generated by the C&P approach were uniformly poor, particularly for the Mobilenet-SSD framework.

Table 1: Object detection performances (MIoU values) of the three network architectures trained using each of the three image datasets

Architecture	Real dataset	Synthetic dataset (Proposed)	Synthetic dataset (C & P)
YOLOv3	0.905	0.821	0.677
Mobilenet-SSD	0.871	0.762	0.539
Mask R-CNN	0.886	0.832	0.744

The object detection results obtained by the Mask R-CNN model trained using the three datasets are illustrated in Fig. 4. While the Mask R-CNN model trained using the real dataset indentified all connectors in the images, the model trained using the synthetic dataset generated by the proposed approach failed to identify some of the corner connectors at the bottoms of the stacks, which are circled in yellow in the figures. However, objects would not be grabbed from the bottom first by robots in automated industrial applications. Therefore, these unrecognized examples are of relatively little importance in practical usage. In contrast, the model trained using the synthetic dataset generated by the C&P approach tended to identify gaps between the connectors as actual connectors, as indicated by the regions within the red circles, leading to a high false recognition rate, which represents an unacceptable condition.

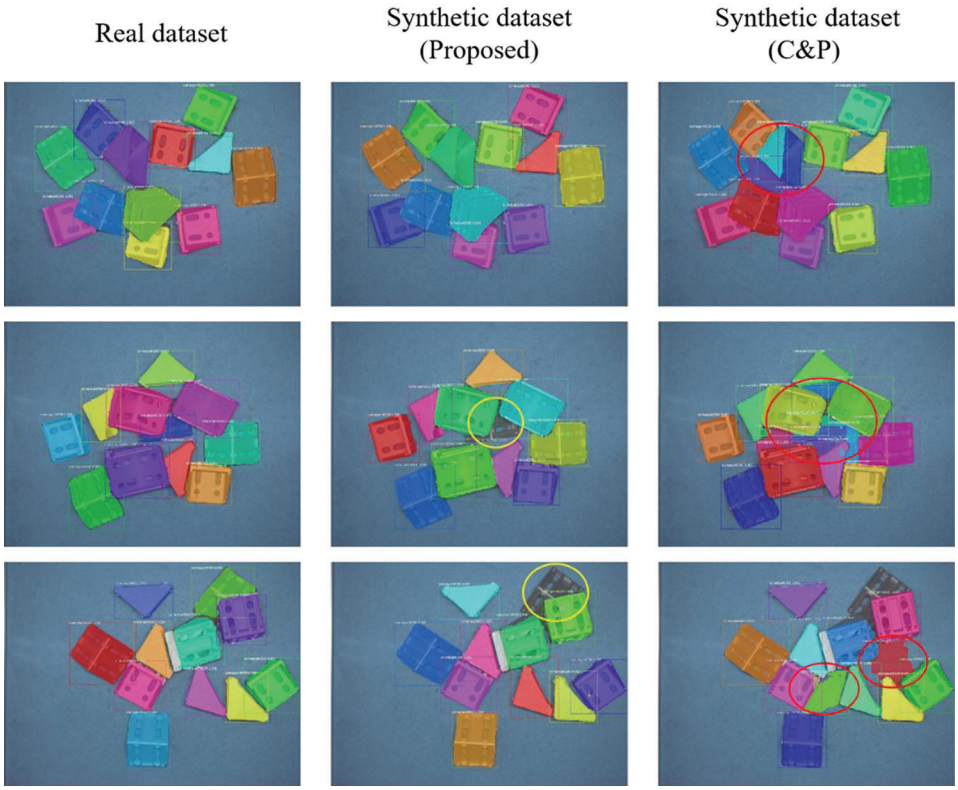


Figure 4: Object detection results obtained by the Mask R-CNN model trained on the real image and synthetic image datasets. Here, unidentified corner connectors at the bottoms of the stacks are circled in yellow, while gaps between the connectors identified as actual connectors are circled in red

These experiments further demonstrate that the proposed approach for generating synthetic datasets can effectively simulate the stacked relationship between objects, and therefore replace real image datasets requiring tedious annotation to train deep learning models. Moreover, the proposed synthetic training datasets not only ensure good detection accuracy, but also greatly improve the efficiency and expandability of the training process owing to the very low time cost involved.

4.2 Ablation Studies

Variations in lighting conditions were considered in OpenGL by varying the property values of the variables GL AMBIENT, GL DIFFUSE, and GL SPECULAR. Here, GL AMBIENT represents the light

intensity (color) of various ambient light rays that reflect many times from the object surfaces until they form an ambient light source in the environment. The variable GL DIFFUSE represents the light intensity (color) in the object on which light is diffused. Finally, GL SPECULAR is the intensity of light (color) formed by the reflection of light rays from the object. We also considered the effect of data-enhancement algorithms, such as flip, rotation, shift, brightness, or contrast adjustment, on the object detection performance. In our experiments, some objects were also added as an interference to diversify datasets. Meanwhile, we note that occlusions are very common in randomly stacked object scenes, and some objects may be represented by so few pixels that they are difficult to be detected. Therefore, we also considered the removal of annotation referring to objects with high shielding rates. The shielding rate is defined as

$$\varsigma = \frac{N_p}{N_A}, \tag{8}$$

where N_p is the number of pixels representing the object in the image, and N_A is the number of pixels required to represent the object in the absence of occlusion. For this purpose, we removed the annotation for objects whose shielding rate was greater than 70% to ensure that they were not recognized as real objects.

The MIoU values obtained by the Mask R-CNN model trained using synthetic datasets generated by the proposed approach while omitting single randomized parameters during the dataset generation process are presented in Fig. 5. The MIoU value obtained for the control group where none of the parameters were changed was 0.758, while the MIoU value of the other control group where all of the parameters were randomly varied was 0.832. As we can see, full variation yields an MIoU value that was 8.9% greater than that obtained by the statically-derived dataset. This is reasonable because these parameters are volatile in reality. The effects of variations in the background, added noise, lighting, and shielding rate operation on the synthetic dataset are illustrated in Fig. 6.

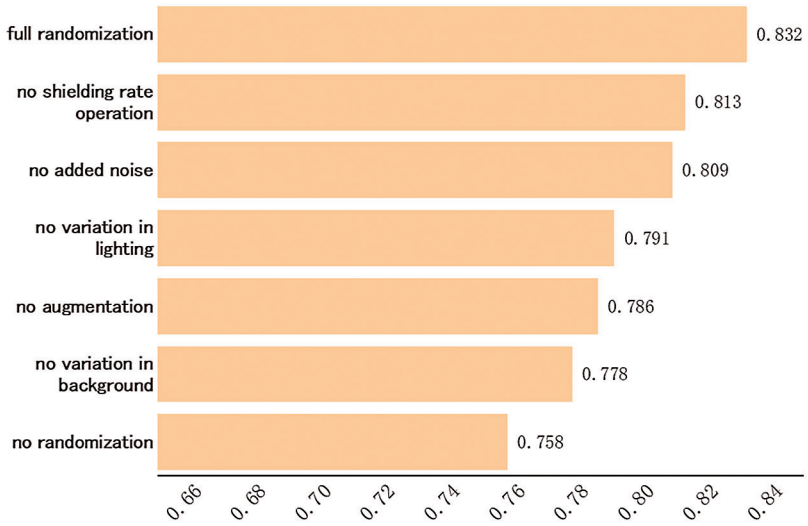


Figure 5: MIoU values obtained by the Mask R-CNN model trained using synthetic datasets generated by the proposed approach while omitting single randomized parameters during the dataset generation process

The effects of the full set of parameter variations on the MIoU values obtained by the Mask R-CNN model trained using synthetic datasets generated by the proposed approach are described as follows based on the MIoU values presented in Fig. 5.

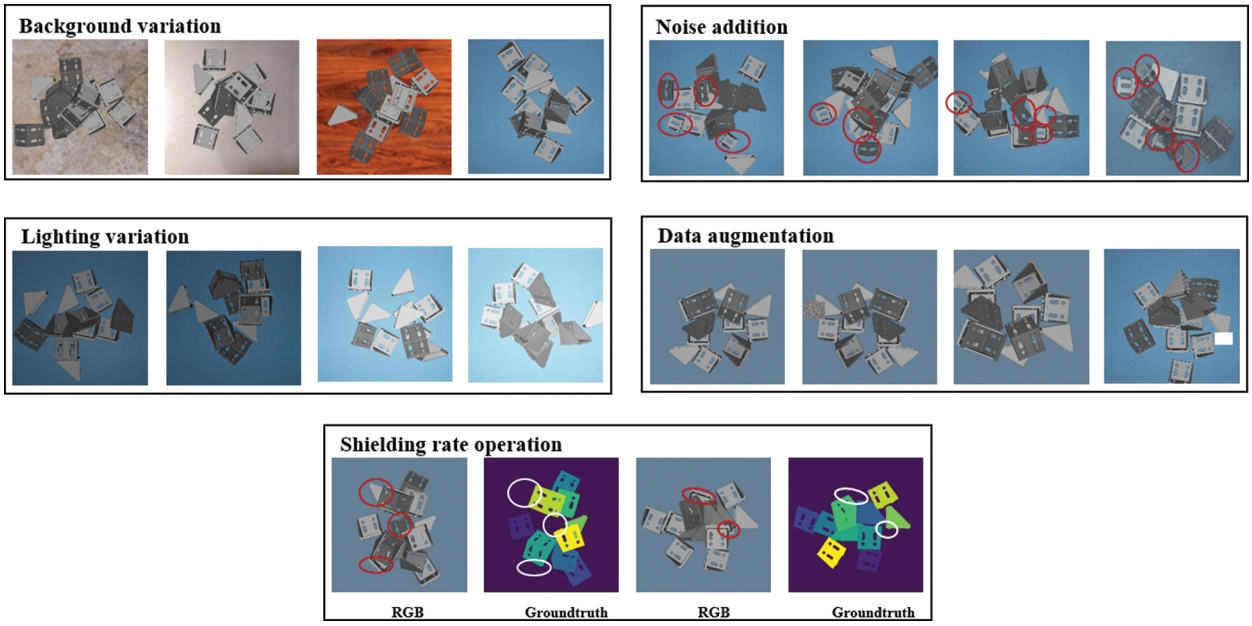


Figure 6: Effects of variations in the parameters of the synthetic dataset on object detection

Background. The MIoU decreased by 6.4% from 0.832 to 0.778 when the background in the synthetic dataset was not varied by using other monochrome colors or complex patterns. Moreover, the object detection accuracy was improved if the test image included complex backgrounds.

Data augmentation. In the absence of applying data augmentation operations in the synthetic dataset, the MIoU decreased by 5.5% from 0.832 to 0.786.

Lighting variation. In the absence of lighting variations in the synthetic dataset, the MIoU value decreased by 4.9% from 0.832 to 0.791.

Noise addition. Without the addition of noise, the MIoU value decreased by 2.7% from 0.832 to 0.809.

Shielding rate operation. Without applying the shielding rate operation, the MIoU value decreased by 2.3% from 0.832 to 0.813. In addition, we note that including the shielding rate operation increased the convergence rate of model training.

4.3 Effect of Training Strategy

The effects of different training strategies on the object detection performance of the Mask R-CNN model trained using the dataset generated by the proposed image synthesis approach were investigated. These different strategies included training with grayscale images, with different numbers of RGB images in the training dataset, and with mixed datasets composed of both real and synthetic RGB images.

Grayscale image training. The corner connectors employed in our experiments were monochrome objects. Therefore, the objects could be reasonably represented by grayscale images. The MIoU value obtained by the Mask R-CNN model trained using the grayscale image dataset was 0.831, indicating that the conversion from RGB to grayscale images had no significant effect on the object detection performance of the model. The recognition effects of the models trained using the two different synthetic datasets are illustrated in Fig. 7. We also present a comparison of the model convergence observed during the training process using RGB and grayscale images in Fig. 8. We note that model convergence was significantly faster when training was conducted with grayscale images rather than with RGB images,

where the model converged at step 80 when trained with grayscale images, while the model converged at step 122 when trained with RGB images.

Figure 7: Object recognition effects of the Mask R-CNN model trained with RGB images and grayscale images generated by the proposed synthesis approach

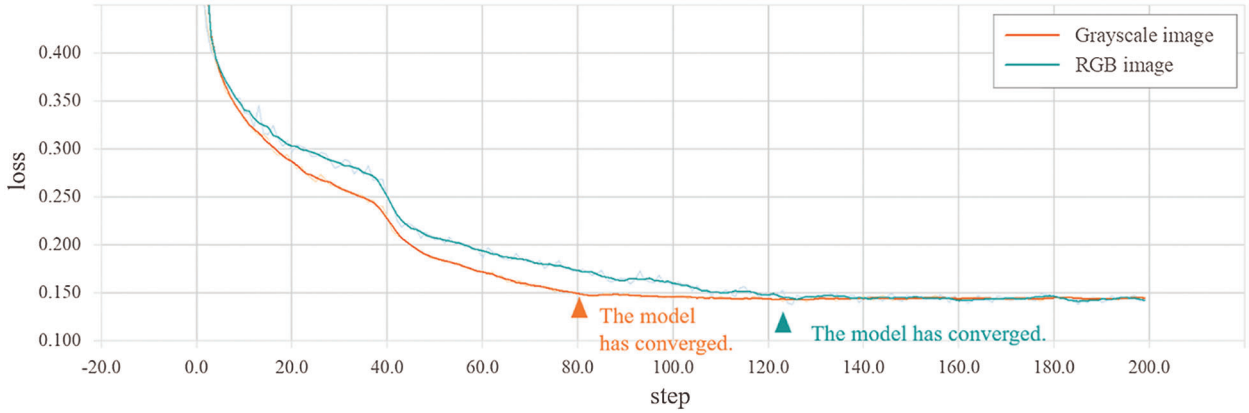


Figure 8: Loss curve of the Mask R-CNN model trained using synthetic grayscale image and RGB image datasets

Number of images in training dataset. The MIoU values obtained by the three network architectures when trained using the proposed synthetic datasets are presented in Fig. 9 with respect to the number of images included in the training dataset. Except for the number of images, all of the other parameters were equivalently randomized, as discussed above, for the different datasets. As shown in the figure, the object detection performances of the three network architectures increased substantially with increasing training dataset size from 200 to 1000. However, further changes in the detection performances with increasing training dataset size were either minor increases, or, occasionally, minor decreases, indicating relatively

stable performance for dataset sizes of around 4,000 images. That is to say, the recognition accuracy of the model can be further improved by increasing the size of the dataset, which can be easily realized in our synthetic dataset.

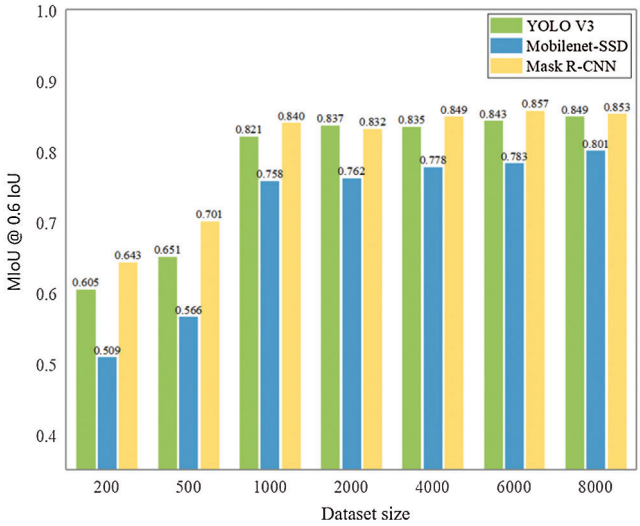


Figure 9: Influence of training dataset size on the object detection performances of the three network architectures

Training with mixed datasets. The MIoU values obtained by the Mask R-CNN model when trained using mixed datasets composed of both real and synthetic RGB images are listed in Tab. 2. Surprisingly, the training dataset composed of just 200 real images facilitates a high object detection performance that is not surpassed until 1,800 synthetic images are added to it. However, a very high MIoU value of 0.913 was obtained when mixing 200 real images with 3,800 synthetic images. These results indicate that training datasets based on real images of stacked object scenes offer advantages regarding the object detection performance of models trained with these datasets. However, these results also demonstrate that the addition of only a small number of real images to a large synthetic training dataset can have marked benefits for ensuring high identification accuracy, while also ensuring excellent training efficiency and low cost.

Table 2: Object detection performance of the Mask R-CNN model trained using mixed datasets composed of both real and synthetic RGB images

Dataset size	Real images	Synthetic images	MIoU
200	200 (100%)	0	0.886
2,000	0 (0%)	2,000	0.832
2,000	50 (2.5%)	1,950	0.860
2,000	100 (5%)	1,900	0.868
2,000	200 (10%)	1,800	0.891
4,000	200 (5%)	3,800	0.913

5 Conclusion

The present work addressed the highly time-consuming and tedious nature of assembling annotated image datasets for training deep learning networks to detect objects within image scenes involving randomly stacked objects by proposing a synthetic training dataset generation method based on OpenGL and the Bullet physics engine, which automatically generates annotated synthetic datasets by simulating the freefall of a collection of objects under the force of gravity. Rigorous statistical comparison of a real image dataset of staked scenes with a synthetic image dataset generated by the proposed approach demonstrated that the two datasets exhibit no significant differences. Moreover, the object detection performances obtained by three popular network architectures trained using the synthetic dataset generated by the proposed approach were much better than the results of training conducted using a synthetic dataset generated by the conventional C&P approach, and these performances were also competitive with the results of training conducted using a dataset composed of real images. Therefore, a synthetic dataset comprising images of stacked object scenes generated by the proposed approach can replace a dataset composed of real images for training deep learning models both effectively and efficiently. The results further demonstrated that the addition of only a small number of real images to a large synthetic training dataset can have marked benefits for ensuring high identification accuracy, while also ensuring excellent training efficiency and low cost. Future work could focus on the machine recognition of textured objects and groups of multiple objects in stacked object scenes based on synthetic datasets generated by the proposed approach, and improve the approach to further narrow the gap between synthetic images and real images.

Acknowledgement: We thank the faculty of the Mechatronics Institute of East China University of Science and Technology and our colleagues for their insightful comments and constructive suggestions for improving the quality of this research work.

Funding Statement: Financial support for this work was provided by the National Natural Science Foundation of China [Grant No. 51575186, Jianjun Yi, www.nsfc.gov.cn] and the Shanghai Science and Technology Action Plan [Grant Nos. 18DZ1204000, 18510745500, and 18510730600, Jianjun Yi, www.sh.gov.cn].

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] V. Bolón-Canedo, N. Sánchez-Maroño and A. Alonso-Betanzos, "A review of feature selection methods on synthetic data," *Knowledge and Information Systems*, vol. 34, no. 3, pp. 483–519, 2013.

[2] G. Georgakis, A. Mousavian, A. C. Berg and J. Kosecka, *Synthesizing Training Data for Object Detection in Indoor Scenes*. Robotics, Cambridge, Massachusetts, USA: Science and Systems, pp. 43–51, 2017.

[3] Q. W. Chao, H. K. Bi, W. Z. Li, T. L. Mao, Z. Q. Wang *et al.,* "A survey on visual traffic simulation: Models, evaluations, and applications in autonomous driving," *Computer Graphics Forum*, vol. 39, pp. 287–308, 2020.

[4] N. Strokina, A. Mankki, T. Eerola, L. Lensu, J. Käyhkö *et al.,* "Framework for developing image-based dirt particle classifiers for dry pulp sheets," *Machine Vision and Applications*, vol. 24, no. 4, pp. 869–881, 2013.

[5] A. Gupta, A. Vedaldi and A. Zisserman, "Synthetic data for text localisation in natural images," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, USA, pp. 2315–2324, 2016.

[6] H. Su, C. R. Qi, Y. Li and L. J. Guibas, "Render for CNN: Viewpoint estimation in images using cnns trained with rendered 3D model views," in *Int. Conf. on Computer Vision*, Santiago, Chile, pp. 2686–2694, 2015.

[7] D. Dwibedi, I. Misra and M. Hebert, "Cut, paste and learn: Surprisingly easy synthesis for instance detection," in *Int. Conf. on Computer Vision*, Venice, Italy, pp. 1301–1310, 2017.

[8] S. Hinterstoisser, V. Lepetit, P. Wohlhart and K. Konolige, "On pre-trained image features and synthetic images for deep learning," in *European Conf. on Computer Vision*, Munich, Germany, pp. 682–697, 2018.

[9] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani *et al.,* "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, USA, pp. 969–977, 2018.

[10] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black *et al.,* "Learning from synthetic humans," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, Hawaii, USA, pp. 109–117, 2017.

[11] W. Qiu and A. Yuille, "UnrealCV: Connecting computer vision to unreal engine," in *European Conf. on Computer Vision*, Amsterdam, The Netherlands, pp. 909–916, 2016.

[12] S. R. Richter, V. Vineet, S. Roth and V. Koltun, "Playing for data: Ground truth from computer games," in *European Conf. on Computer Vision*, Amsterdam, The Netherlands, pp. 102–118, 2016.

[13] C. R. de Souza, A. Gaidon, Y. Cabon, N. Murray and A. M. López, "Generating human action videos by coupling 3D game engines and probabilistic graphical models," *International Journal of Computer Vision*, vol. 128, no. 5, pp. 1505–1536, 2020.

[14] Y. Zhang, W. Qiu, Q. Chen, X. Hu and A. Yuille, "UnrealStereo: Controlling hazardous factors to analyze stereo vision," in *Int. Conf. on 3D Vision*, Verona, Italy, pp. 228–237, 2018.

[15] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent and R. Cipolla, "SceneNet: Understanding real world indoor scenes with synthetic data," *Computer Vision and Pattern Recognitio*, 2015.

[16] A. Gaidon, Q. Wang, Y. Cabon and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, USA, pp. 4340–4349, 2016.

[17] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen *et al.,* "Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?," in *IEEE Int. Conf. on Robotics and Automation*, Marina Bay Sands, Singapore, pp. 746–753, 2017.

[18] G. Ros, L. Sellart, J. Materzynska, D. Vazquez and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, USA, pp. 3234–3243, 2016.

[19] P. Martinez-Gonzalez, S. Oprea, A. Garcia-Garcia, A. Jover-Alvarez, S. Orts-Escolano *et al.,* "UnrealROX: An extremely photorealistic virtual reality environment for robotics simulations and synthetic data generation," *Virtual Reality*, vol. 24, no. 2, pp. 271–288, 2020.

[20] M. Müller, V. Casser, J. Lahoud, N. Smith and B. Ghanem, "Sim4cv: A photo-realistic simulator for computer vision applications," *International Journal of Computer Vision*, vol. 126, no. 9, pp. 902–919, 2018.

[21] H. Zhang, Y. Tian, K. Wang, H. He and F. Y. Wang, "Synthetic-to-real domain adaptation for object instance segmentation," in *2019 Int. Joint Conf. on Neural Networks*, Budapest, Hungary, pp. 1–7, 2019.

[22] V. Satish, J. Mahler and K. Goldberg, "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1357–1364, 2019.

[23] R. Barth, J. IJsselmuiden, J. Hemming and E. J. Van Henten, "Synthetic bootstrapping of convolutional neural networks for semantic plant part segmentation," *Computers and Electronics in Agriculture*, vol. 161, pp. 291–304, 2019.

[24] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv: 1804.02767, 2018.

[25] A. Howard, M. Sandler, G. Chu, L. C. Chen, B. Chen *et al.,* "Searching for mobilenetv3," in *IEEE Int. Conf. on Computer Vision*, Seoul, Korea, pp. 1314–1324, 2019.

[26] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed *et al.,* "SSD: Single shot multibox detector," in *European Conf. on Computer Vision*, Amsterdam, The Netherlands, pp. 21–37, 2016.

[27] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," in *Int. Conf. on Computer Vision*, Venice, Italy, pp. 2961–2969, 2017.