Proceedings of the 2023 IEEE / CVF Computer Vision and Pattern Recognition Conference, 4th Agriculture Vision Workshop, Vancouver Canada, June 2023.

A Realistic Synthetic Mushroom Scenes Dataset

Dafni Anagnostopoulou
dafnianagnostopoulou@mail.ntua.gr

George Retsinas
gretsinas@central.ntua.gr

Niki Efthymiou
nefthymiou@central.ntua.gr

Panagiotis Filntisis
filby@central.ntua.gr

Petros Maragos
maragos@cs.ntua.gr

School of E.C.E., National Technical University of Athens
15773, Athens, Greece

Abstract

In this work, we present the Realistic Synthetic Mushroom Scenes Dataset, which encompasses images depicting mushrooms in various settings in relatively cluttered scenes. The dataset is composed of 15,000 high-quality, realistic images with various useful annotations. The dataset can be leveraged to address problems associated with mushroom detection, instance segmentation, and 3D pose estimation. These tasks are of paramount importance in automating the mushroom harvesting process in mushroom farms, which is a challenging and costly procedure. Also, we proffer a three-step pipeline that can generate annotated and realistic synthetic images, commencing with a singular 3D model that can be easily applied to a range of crops beyond mushrooms (https://github.com/dafniana/Synthetic-Mushroom-Dataset).

1. Introduction

Over the last few years, there has been significant progress in the development of robotic applications for agriculture. The aim of the ongoing advancements in agricultural robotics is to address the challenges posed by factors such as population growth, intense competition for high-quality products, and the need for environmental conservation [9]. By integrating robotics into agriculture, not only can productivity and environmental sustainability be enhanced, but working conditions for laborers can also be improved significantly [12].

Machine vision algorithms are necessary for the successful operation of many tasks performed by robots in agriculture. These algorithms are adapted to suit the specific function of the robot, whether it be plowing fields, planting seeds, handling weeds, monitoring growth, picking fruits and vegetables, sorting, grading, or packaging [4, 8]. Mushroom harvesting is one example of the different agricultural operations where robotic automation can produce significant outcomes and recently there has been a growing interest for such robotic systems. However, one of the key challenges in this field is developing computer vision models that can accurately locate and identify mushroom poses in complex and cluttered environments. This is a particularly challenging task, especially when constrained to single view RGB images. In order to extract the requisite informa-

(a)

(b)

Figure 1. (a) An example of a very simple mushroom scene (b) An image coming from the Realistic Synthetic Mushroom Scenes Dataset

tion from such images, the algorithms have to be trained on annotated data. Unfortunately, annotating sufficient quantities of this data, particularly regarding 3D pose, is almost impossible and involves high costs and time investments.

Thus, a viable solution is to generate a synthetic dataset containing realistic mushroom scenes. One of the key advantages of a synthetic dataset is that it is automatically annotated with ground truth labels that can be used to train and evaluate machine learning models at no additional cost. Another advantage is that it allows for a high degree of control over various parameters such as mushroom numbers, shapes, and relative positions. A synthetic dataset can be easily scaled up to include many images, which can be used to train deep learning models that require large amounts of data to be accurate and robust. We are mainly interested in creating a dataset of images preserving the pose of mushrooms from the synthetic 3D scenes, while at the same time achieving realistic and varying mushroom and background textures.

To this end, we have developed a three-stage automated and randomized pipeline. The first stage of our pipeline generates different 3D mushroom scenes based on a singular 3D mushroom model. Subsequently, the pipeline renders images from various viewpoints of these scenes. Finally, utilizing the state-of-the-art ControlNet network [17] which is based on diffusion models, we deploy the original synthetic images as input to generate realistic-looking images that accurately depict mushrooms in different conditions.

The first two stages of our pipeline, namely the scene generation and the rendering procedure, control numerous features in our images, like the number, the size, the shape and the poses of the mushrooms, while ensuring that all the requisite ground truth information will be available in the final images. The last stage of our pipeline, namely the synthetic-to-realistic procedure, introduces diversity in our dataset and renders our images substantially more realistic.

Using this pipeline we created the Realistic Synthetic Mushroom Scenes Dataset that contains:

- 15,000 images of synthetic realistic mushroom scenes.
- 2D annotations of visible individual mushrooms in each image: 2D bounding boxes, segmentation masks as well as the corresponding 2D annotations in COCO format [2].
- 3D annotations of visible individual mushrooms in each image: 3D bounding boxes, their 2D projections, as well as the corresponding annotations of the 3D mushroom poses in BOP format [1].

Finally, as a proof of concept using our dataset as training data, we fine-tuned a Mask R-CNN model, and we tested it on images of real mushrooms taken of mushroom cultivation on a mushroom farm. We will release the Realistic Synthetic Mushroom Scenes Dataset and the GitHub repository with the dataset generation pipeline. The contributions of our work can be summarized as follows:

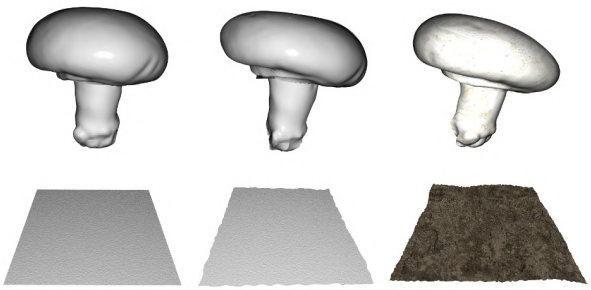


Figure 2. 3D mushroom and ground models used to build the synthetic scenes: initial models (left), deformed models (middle) and textured deformed models (right).

- The release of the Realistic Synthetic Scenes Mushroom Dataset, which is fully annotated and ready to be used for mushroom detection and pose estimation.
- The development of a multistage pipeline that generates 3D scenes with mushrooms, renders 2D images from these scenes and then produces realistic synthetic images.
- This pipeline could be easily adapted to different agricultural and other tasks using a different initial 3D model.

2. Related Work

2.1. Synthetic Datasets

Deep learning networks have grown significantly over the past several years and are now frequently utilized to address various issues. Despite their amazing capabilities, neural networks require a large volume of high-quality data to be trained effectively. Although data are everywhere, annotated data are very expensive and hard to obtain. Many approaches, like semi-supervised and self-supervised learning, have addressed this problem, including the synthetic data generation.

DatasetGAN [18] proposed the following pipeline to generate a synthetic dataset with different segmentation related annotations: At first, StyleGAN generated numerous images, secondly a few of them are manually annotated for the specific targeted task, and finally a small model is trained to produce similar segmentation masks from StyleGAN features. This way, by labeling only a few images, StyleGAN can generate as many labeled images as needed. BigDatasetGAN [7] replaced StyleGAN with BigGAN which is more suitable for generating images with significant variety belonging to numerous categories, and scaled DatasetGAN to ImageNet.

Yang *et al*. [16] proposed a method to tackle the problem of image generation based on a given layout. Their method

compresses RGB images into patch tokens and introduces a Transformer with Focal Attention for exploring dependencies of object-to-object, object-to-patch and patch-to-patch.

In another recent work, Sun *et al*. [13] released SHIFT which is a synthetic driving dataset that presents discrete and continuous shifts in different weather conditions, time of day, and vehicle and pedestrian density. The domain shifts are performed by using four domain adaptation strategies. Furthermore, Yan *et al*. [15] proposed a visual localization system which they train by creating and using a synthetic data generation tool that seamlessly transitions between the real and synthetic world, depending on the geographic camera viewpoint. The synthetic data generation is performed by the CesiumJS engine and the generated data consist of RGB images, scene coordinates, depth, surface normals, and semantics.

2.2. Mushroom Datasets

As far as we are aware, no annotated datasets specific to mushrooms are publicly available. The Open Images collection [6], which contains over 9 million photos and provides image-level labels, object bounding boxes, object segmentation masks, and other annotations, is a well-known image dataset. It contains bounding boxes for 600 different object types, including the mushroom class. There are approximately 1,500 annotated pictures of mushrooms in the dataset. However, because they show a variety of mushrooms -both cultivable and not, edible and not— and because most of the scenes show only a small number of mushrooms spread out apart from one another, these images are not well suited to be used as training data for the automation of mushroom harvesting. Moreover, the absence of 3D pose annotations in this data is most crucial.

3. Towards a Realistic Dataset

3.1. Creating Synthetic Scenes / Scene Generation Pipeline

The first stage towards a realistic synthetic dataset is the creation of complex and interesting synthetic scenes. For that purpose we have been using the Open3D library [19]. Open3D is an open-source library that allows the handling of 3D data and provides rendering possibilities.

The creation of every scene is a randomized process and begins with the 3D model of a simple mushroom (Fig. 2). For our synthetic mushroom dataset we created a total of 100 random scenes. For the creation of each of these scenes the scene generation pipeline described below was followed:

To start with, a ground has to be placed in our scene. We provide the scene generation pipeline with different ground options in order to introduce variety to our scenes. We created 6 different ground models. The base of these models is a texture-less square mesh. To produce this mesh, a 200 by 200 grid of vertices is initialized. Subsequently, a small value of random noise is added to the x, y and z positions of each vertex so that the grid is not exactly Cartesian. Afterwards, the whole grid is filled with triangular faces and the vertices normals are computed. Vertices, vertices normals, and faces along with a simple texture map from image pixels to mesh faces are stored into an object file. Six different images of soil are assigned as textures to the ground object to create 6 different types of ground. In the beginning of each scene's creation, one of the ground objects is randomly chosen. Afterwards, the ground mesh is randomly deformed as follows: some of the mesh's vertices are randomly chosen. For some of the chosen vertices random target new positions are generated. An RBFInterpolator [3] is then used to interpolate between the original and the target positions of the vertices and all ground vertices are shifted according to the interpolation result.

The base of our mushroom scene generation pipeline is, as mentioned above, a 3D mesh representing a simple agaricus mushroom. We added a texture map from an image of a mushroom texture pixels to the faces of the mushroom mesh.

Afterwards, the number of mushrooms in the scene is randomly chosen. Our scenes contain a minimum of 30 and a maximum of 60 mushrooms. Each mushroom's instance is initialized using our textured mushroom 3D model. It is then uniformly scaled with a random scaling factor between 0.5 and 1.5. The mushroom is proportionally to its scaling factor translated along the z axis so that there is no mushroom flying above the ground. Consequently, the mushroom is randomly scaled with a different scaling factor along its axis. This scaling is more gentle with values between 0.85 and 1.15. Moreover, for some of the mushrooms, a deformation similar to that applied on the ground mesh is also applied. Here, vertices are shifted according to the interpolation result weighted by their normal values in order to achieve a more realistic result. Then, the mushroom is randomly rotated up to 22.5°around x and y axis. Finally, the mushroom is placed on a random position on the ground plane.

For the mushroom instance created after all the above transformations, the distance from its center to the furthest of its points is computed and called radius. For all mushrooms placed in our scene until now the distance between their center and the center of the new mushroom must be bigger than the sum of the their radii so that the new mushroom does not collide with an old one. If there is a collision the new mushroom is discarded and another mushroom is generated until the desired mushroom number is reached.

The 3D bounding boxes of all the mushrooms contained in the scene are saved, along with the scene. Open3D provides us with the bounding box of the whole mushroom, but

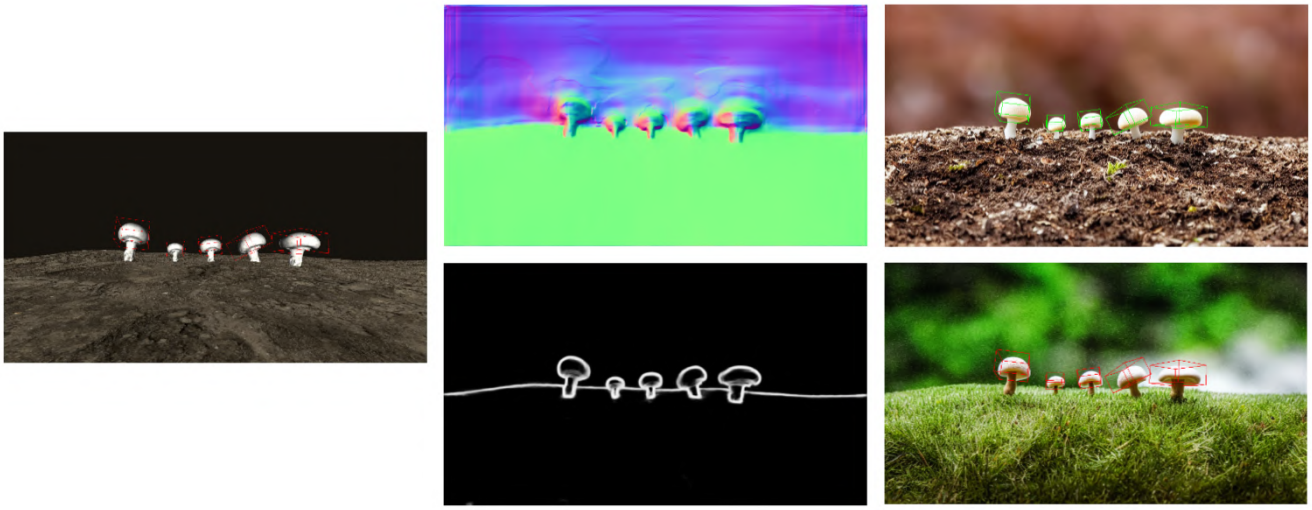


Figure 3. Retaining mushroom shape and pose using as control input to the diffusion model the normals maps (on top) and the HED boundaries (on the bottom).

only the pose of the cup is interesting for us, so the bounding boxes are altered accordingly to include only the cups.

3.2. Rendering 3D Scenes to 2D Images

After the 3D scene generation we proceed with the rendering procedure. From each mushroom scene 30 images are rendered in a random rendering procedure. In the beginning, 30 viewpoints are randomly selected. Viewpoints look at the scene both from different angles around it and from different distances. They can be looking at the scene from directly vertically above it to almost directly horizontally along its ground plane.

Before the rendering of every image, a random background among dark brown, red and gray colors is chosen. Then the Open3D visualizer is used to visualize the scene from the specific chosen viewpoint and the screen image is captured. This procedure can be done in headless mode too.

During the rendering procedure, for every rendered image, masks for all mushrooms existing in the scene are also rendered. This is done in the following way: A texture-less version of the scene is loaded in parallel with the original scene and all meshes are given a black color. Then, using the same viewpoint that was used to render the whole image, the color of one mushroom is set to white and the mask corresponding to it is rendered. Afterwards, the mushroom's color is set back to black so that the mask of the next mushroom can be rendered too.

3.3. From Synthetic to Realistic Images

After 3000 initial 3D annotated synthetic images have been rendered, the goal is to deploy them to obtain realistic 3D annotated images. To achieve this goal a pretrained ControlNet [17] network was employed.

ControlNet is a neural network designed to control large diffusion models with task specific conditions. The Stable Diffusion [11] is used as the diffusion model. It is a large text to image diffusion model with a U-Net structure of an encoder, a middle block and a skip-connected decoder comprising of down sampling and up sampling convolutional layers, resnet layers and Vision Transformers layers, while a CLIP model is used for the text encoding. To control the training of the diffusion model, a special convolutional layer that is called zero-convolution and is a 1×1 convolution with both weights and bias initialized as zeros is used. The desired condition is the input to this zero convolution, while the output is then added to the diffusion network input. ControlNet provides trained networks with various different image based conditions to control the diffusion models, including edges produced by different methods, depth and normal maps, human poses, semantic segmentation, user sketching and others.

Canny edges, Hough lines, HED boundaries, normal maps, depth maps were used as different control conditions possibilities in our experiments. Normal maps and HED boundaries proved to be the best options as they could preserve the mushroom position, shape and pose information in the most efficient way (Figure 3). Hence, we employed these two as our controls. For the normal maps, Midas [10] is used to compute depth maps and then normal maps are estimated by performing normal-from-distance. For the HED boundary the HED Boundary detection [14] method is employed.

Instead of using the normal maps estimated with Midas as control input to the diffusion model, we also experimented with directly using the normal maps that can be rendered from the synthetic 3D scenes. As one can observe in

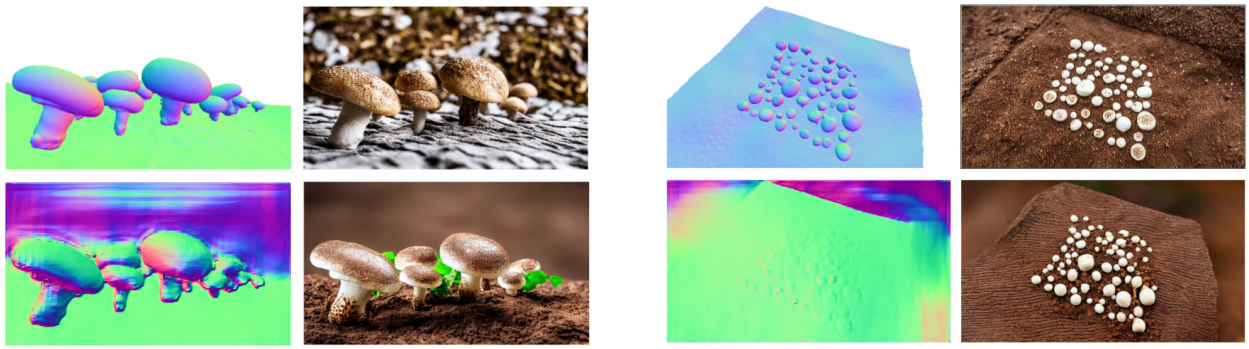


Figure 4. Qualitative comparison between using directly rendered (top row) and estimated (bottom row) normal maps as control input to the diffusion model from a close to the mushroom instances point of view (a) and far from the mushroom instances point of view.

Figure 4 the rendered normal maps are more precise than the estimated ones as expected. Nevertheless, especially from a viewpoint close to the mushroom instances the final result does not show any difference whether the rendered or the estimated normals were used as control input. Mushrooms number and pose are preserved in both cases. In the case of a point of view that is relatively far from the mushroom instances it is observed that the pose of some really small mushrooms is better transferred from the rendered normals input than from the estimated normals input to the realistic RGB image. However, this applies only to mushrooms with so small size that are not considered ready for harvesting. Therefore, since rendering normals multiplies the time required for the creation of the dataset, normal estimation is preferred.

Therefore, from each synthetic image five realistic images were randomly created, using as input either the normals map or the HED boundaries of our synthetic images, different prompts and control parameters. Prompts differentiated in regards to mushroom color and variety description, ground color and material description and background description as shown in Table 1.

3.4. Annotations

Open3D provides 3D bounding boxes for all individual meshes in the scene. Using the cameras intrinsic and extrinsic matrices, the 3D positions of the vertices of the bounding boxes are converted to images pixels and saved as the 2D projections of the 3D bounding boxes

Moreover, the rendered masks of each image are processed so that all non visible mushroom instances in each image are identified. For each image the mean area of the mushroom masks is calculated and a mushroom is considered visible in that image if its mask area is bigger than 15% of the mean mask area.

Afterwards, by processing the instance segmentation masks we create COCO annotations. For each mask corresponding to a visible mushroom we get the 2D bounding box that contains it and we find its contour and calculate the contour's polygon to get the segmentation array.

Finally, to create BOP annotations we need to calculate the model to camera rotation ($m2c\_R$) and translation ($m2c\_t$) transformations. In order to calculate them we calculate the homogeneous model to camera matrix by multiplying cameras extrinsic matrix ($C_{extrinsic}$) with the homogeneous matrix containing the mushrooms rotation ($w2m\_t$) and translation ($w2m\_t$) transformations from the world's origin as in Eq. 1.

$$[m2c\_R|m2c\_t] = C_{extrinsic} * [w2m\_R|w2m\_t] \quad (1)$$

To sum up, our dataset includes for each mushroom instance in each image its 3D bounding box along with its 2D pixel projection, its mask, weather the mushroom is visible in the image, 2D bounding box and segmentation polygon in COCO format and 3D pose BOP annotations.

4. Exploiting the dataset

Our dataset is ready to be used for tasks like object detection, instance segmentation and 3D pose estimation.

For the tasks of detection and instance segmentation of mushrooms, a Mask R-CNN [5] model trained on COCO dataset was employed. The model was fine tuned on our realistic synthetic data using 75% of our dataset as training data for 10 epochs. Final fine tuned model scores 89.5% mean average precision (mAP) and 93.3% mean average recall (mAR) calculated among predicted and ground truth bounding boxes with an Intersection over Union (IoU) of 0.50. We also experimented with fine tuning the Mask R-CNN model training it only with the initial synthetic data. As can be seen in Table 2 the model achieves significantly better mushroom detection and instance segmentation ability when trained with the realistic training data.

The fine tuned model was firstly tested on a small dataset containing simple hand annotated images with a few real mushrooms (Figure 5a). This small dataset contains rela-

Prompts						
white agaricus			dark brown soil ground		sky	
brown			dark soil ground		forest	
white stained agaricus			brown soil ground		dark sky	as a
white dirty agaricus			black soil ground		green forest	background
white agaricus with dirt	mushrooms	on	brown soil	with	trees	
white portabello		in	dark soil	and	garden	in the
portabello			black soil		farm	back
hedgehog			green ground		grains	
shiitake			green field		sky and clouds	
porcini			grass		green trees	
chanterelle			green grass		deep forest	

Table 1. Prompts used to generate the realistic images.

Training Data	mAP	mAR
Synthetic	81.7%	86.4%
Realistic Synthetic	**89.5%**	**93.3%**

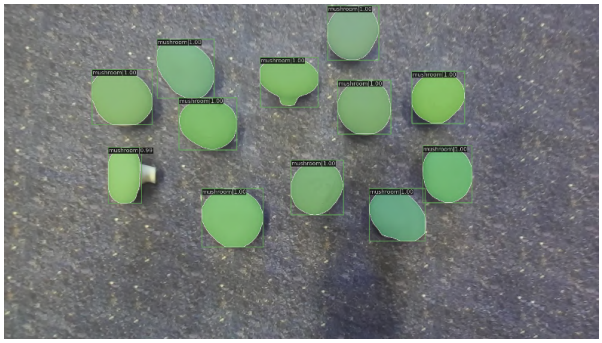
Table 2. Image segmentation results with simple synthetic training data and with the final realistic data.

tively simple scenes without mushroom clusters and occlusions. The model achieves almost 100% accuracy on these data. Subsequently, the model was also tested on images of white cultivated mushrooms, Agaricus bisporus, on industrial mushroom farms [1]. These images depict intensely cluttered mushroom scenes with numerous mushrooms that lie very close to each other, touch each other and show significant occlusions. Instance segmentation results on these scenes are quite satisfying as can be seen in Figure 5b.

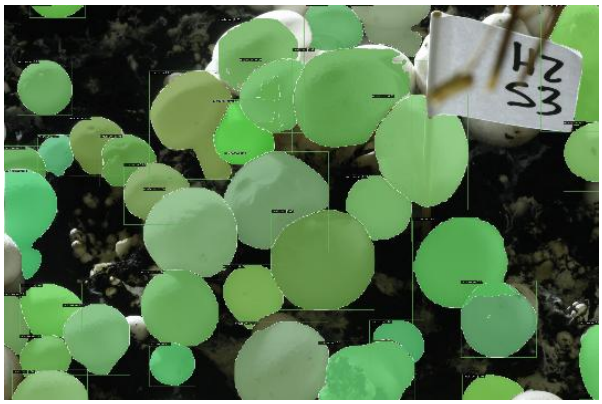
5. Future Work and Conclusions

The next step is to employ our Realistic Synthetic Mushroom Scenes Dataset to deal with the problem of estimating 3D pose of mushrooms from single view RGB images. This is a particularly difficult problem especially for mushrooms taking into consideration that they have a symmetrical ellipsoid shape and that they usually grow in cluttered environments. An annotated dataset, we believe, can significantly boost efforts toward this goal. In conclusion, a synthetic mushroom dataset can be a powerful tool for developing computer vision models that can locate and identify mushroom poses for robotic harvesting. By providing a large and diverse set of training data, the dataset can help improve the accuracy and robustness of the computer vision models, and facilitate the automation of mushroom harvesting.

[1]We thank our collaborators Teagasc, Ireland for providing images of commercial mushroom production.



(a)



(b)

Figure 5. Examples of instance segmentation (a) on real mushrooms positioned on a simple scene (b) on image taken from real mushroom cultivation bed.

6. Acknowledgements

This research has received funding from the European Union's Horizon 2020 research and 628 innovation programme under grant agreement no. 101017054 (project: SoftGrip).

Figure 6. Images from the Realistic Mushroom Scenes Dataset

References

[1] https://bop.felk.cvut.cz/home/. 2

[2] https://cocodataset.org/#home. 2

[3] https://scipy.org/. 3

[4] https://www.rsipvision.com/robots-using-machine-vision-agriculture/. 1

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017. 5

[6] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020. 3

[7] D. Li, H. Ling, S. Kim, K. Kreis, A. Barriuso, S. Fidler, and A. Torralba. Bigdatasetgan: Synthesizing imagenet with pixel-wise annotations, 2022. 2

[8] E. Mavridou, E. Vrochidou, G. Papakostas, T. Pachidis, and V. Kaburlasos. Machine vision systems in precision agriculture for crop farming. *Journal of Imaging*, 5, 2019. 1

[9] L. F. P. Oliveira, A. P. Moreira, and M. F. Silva. Advances in agriculture robotics: A state-of-the-art review and challenges ahead. *Robotics*, 10, 2021. 1

[10] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44, 2022. 4

[11] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 4

[12] R. Shamshiri, C. Weltzien, I. Hameed, I. Yule, T. Grift, S. Balasundram, L. Pitonakova, D. Ahmad, and G. Chowdhary. Research and development in agricultural robotics: A perspective of digital farming. *International Journal of Agricultural and Biological Engineering*, 11, 2018. 1

[13] T. Sun, M. Segu, J. Postels, Y. Wang, L. Van Gool, B. Schiele, F. Tombari, and F. Yu. Shift: A synthetic driving dataset for continuous multi-task domain adaptation. In *CVPR*, 2022. 3

[14] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, 2015. 4

[15] Q. Yan, J. Zheng, S. Reding, S. Li, and I. Doytchinov. Crossloc: Scalable aerial localization assisted by multimodal synthetic data. *arXiv preprint arXiv:2112.09081*, 2021. 3

[16] Z. Yang, D. Liu, C. Wang, J. Yang, and D. Tao. Modeling image composition for complex scene generation. In *CVPR*, 2022. 2

[17] L. Zhang and M. Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 2, 4

[18] Y. Zhang, H. Ling, K. Gao, J.and Yin, J.F. Lafleche, A. Barriuso, A. Torralba, and S. Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *CVPR*, 2021. 2

[19] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 3