Generation of Synthetic Training Data for Object Detection in Piles

Elvijs Buls, Roberts Kadiķis, Ričards Cacurs, Jānis Ārents
Institute of Electronics and Computer Science (EDI), Riga, Latvia

ABSTRACT

Current state-of-the-art object detectors are based on supervised deep learning approaches. These methods require a large amount of annotated training data, which hinders a wider use of these methods in industry. We propose a method for generating synthetic training data for the task of detecting which objects in a pile can be picked up by a robot arm. The method requires few input images, which are used to create annotated images of piles. After training a state-of-the-art detector on the synthetic data, we test it on real images. The results show that the model trained in such a way is not a rival to the best object detectors trained on large datasets of real images, but it is good for the specific task of detecting pickable objects in the piles. The main advantage of the proposed training approach is that the existing models can be easily re-trained to work with piles of different objects by personnel who do not specialize in machine learning.

Keywords: Industrial robot arm, object detection in a pile, training data generation, deep learning

1. INTRODUCTION

According to statistics gathered by the International Federation of Robotics [1], the sales of industrial robots have increased for the fourth year in a row. The organization also estimates that 1.7 million new industrial robots will be installed in factories around the world between 2017 and 2020. Majority of these robots will be used to automatize processes where motions of the robot can be preprogrammed to manipulate objects with known position and orientation. However, there are many tasks in the industry where the information about object positions and orientation cannot be predefined.

Current developments in machine vision could automate such ill-defined tasks as picking up objects from a pile. The first step in such a task is to detect which of the objects are unobstructed by other objects and so can be picked up by the robot. In machine vision, the detection algorithms often produce a list of objects present in the input image along with bounding boxes that indicate the position and scale of these objects.

Classical machine vision algorithms, such as edge detection and blob analysis, can be used to pick up separate objects on a uniform background, as was done in [2]. However, objects in a pile are occluding each other, casting shadows, and creating difficult backgrounds. When the pile consists of similar or identical objects, it is hard to distinguish between the background and the pickable objects. Although this task has been studied for a long time [3], [4], it still remains a challenge to use existing approaches in the actual industry.

Some approaches try to detect shapes in a pile and compare them with known shapes or models of the objects [5]. Along with cameras, some additional hardware is tried to get more robust shapes. For example, [5] and [6] use a depth sensor Microsoft Kinect, while [7] uses a multi-flash camera for depth edge detection. Article [8] suggests adding a step of interaction between the robot and objects (grasp attempts, perturbation pushes), which helps to separate the object from the pile. Alternatively, as shown in [9], the use of machine learning allows training vision systems that do not use or build models of the objects but instead learn to directly identify points where a robot arm should grasp.

Furthermore, for several years, the object detection competitions such as ImageNet [10] and Microsoft COCO [11] are won by the supervised deep learning-based approaches, which motivates us to apply these methods in the industry robot vision. The state-of-the-art detection methods are usually based on deep convolutional neural networks (CNN). Examples include SSD (Single Shot MultiBox Detector) [12], several variations of R-CNN (Regions with CNN features) [13], [14], [15] and YOLO (You Only Look Once) [16], [17]. These are supervised machine learning approaches, so to train a model that works with new kinds of objects, one needs to have a large dataset of annotated images containing these objects. Manual annotation of such images means that a human has to determine and annotate the coordinates of at least two points of a bounding box for every object in the image (four numbers per object). It is time-consuming and

repetitive labor, which impedes the implementation and use of modern CNN-based computer vision in a real industry environment.

This paper presents an approach that greatly accelerates the acquisition of labeled data by synthesizing a large dataset from a small number of object's photos. This approach is customized for the complex task of detecting which object can be picked up from a pile by a robot arm.

2. RELATED WORK

2.1 Generation by rendering

One of the solutions for data generation is the use of computer-generated imagery, which is getting more and more lifelike. In a self-made 3D CAD scene, all objects are known and precisely defined. One can freely change the camera position, location and pose of the objects, and then render a realistic synthetic image while simultaneously acquiring the corresponding instance level annotations. Therefore, papers such as [18], [19] try to train object detection models on images rendered from 3D CAD scenes and then use the trained models on real images.

This approach also becomes more available to non-specialists. The tools to create realistic and even physics-based scenes are becoming cheaper or even free (e.g., 3D game engines Unreal Engine and Unity or even commercial computer games [20]). In addition, more and more open datasets of virtual 3D scenes, objects, and synthetic images are becoming available, for example, SUNCG [21] - room and furniture layouts, Virtual KITTI [22] - street videos viewed from a virtual car, ShapeNet [23] - extensively annotated 3D models, A Large Dataset of Object Scans [24].

The combination of physics-based virtual 3D environments and reinforcement learning is a prospective solution for the end-to-end training of robot arms. In this setup, a virtual arm uses visual information and makes many attempts at picking objects. Successful tries are rewarded. Article [25] presents some promising initial results of transferring the virtual learning to the manipulation of a real robot arm. However, the scope of the current paper deals only with the vision part of the robot arm-based pick-up system and tries to make the retraining of the object detector as human-skill independent as possible. With such considerations, the current virtual modeling of a whole scene is not considered to be easy to use in the production industry by personnel unrelated to computer science.

2.2 Generation by rendering + composing

To reduce the manual modeling when generating data from 3D scenes, instead of creating objects and their surroundings, one can model only the object and then insert the rendered image of the object into an image of a real scene. Article [26], which uses some human annotation before inserting the model, shows that the result can be so realistic as to fool a human. Thus, it is reasonable to try to develop fully automatic methods for realistic insertion of synthetic objects into real backgrounds.

A method in [27] automatically inserts existing 3D models as well as their modifications into real background images to augment training data for a viewpoint estimation system. In [28], this approach is used to model and analyze the capabilities of CNNs to learn texture, color, background, 3D pose, and 3D shape invariant object features. A similar approach can also be used with two-dimensional objects. For example, in [29] the annotated data for a text localization task is generated by automatically placing differently sized and stylized texts on different background images.

Generating realistically looking images from 3D or 2D models of new objects in new environments still requires specific human skills in modeling and rendering. Authors of [30] propose an algorithm that estimates rendering parameters automatically from a small set of real images. The drawback is that the required set of real images has to consist of image pairs, where one image includes the object and the background, and the other has the same background without the object. The 3D model of the object is rendered and inserted into the empty background image, which then is compared to the image with the real object at the same location. Different rendering parameters are tried in order to optimize the similarity of real and rendered images.

Another possible solution is explored in [31] where images created from a simple 3D model are augmented with realistic blur, lighting, background, and image detail learned by a Generative Adversarial Network (GAN). The proposed RenderGAN learns in an unsupervised fashion. It consists of generator and discriminator networks. The generator learns such augmentation functions for blur, lighting, etc. that would fool a discriminator, which compares the generated examples with real unlabeled images. The proposed approach is very promising; however, currently, the training of

GANs in new domains is hard because of their unstable training behavior [32]. This behavior limits the current applicability of GANs in the industry since training can't be completely automated and requires specific expertise.

2.3 Generation by composing

Recent literature also explores a simpler approach for generating realistic synthetic data without using models of the objects. In these methods (for example [33], [34], [35]), training images are composed of photos of real objects and real backgrounds. Approaches such as [34] include acquiring a limited number of object's images, cutting out the object, modifying the object, and pasting it into the many new background images. The method in [34] augments the data by randomly changing the viewpoint of the object (2D and 3D rotation of the cropped object images before the composition step). In addition, sometimes the objects are truncated by putting them at the boundaries of the image, and sometimes objects are occluded by placing them in the image, so they partially overlap. Augmentation also includes placing distractor objects in the scene. An important step is the blending of the objects to the scene because a simple pasting creates an unrealistic boundary between the object and the background. Detection model can latch on to this boundary, so authors propose blending techniques that smoothe the boundary.

A similar approach in [35] adds analysis of background images to place the objects in the scene more realistically. The method determines any support surfaces in the scene, where the objects of interest are likely to be placed. Additional analysis of the background depth at the location of the object determines realistic scales of the object.

Some manual work to train systems in [34] and [35] consists of finding the appropriate background images - those images do not have to be labeled, but some environments might still require the acquisition of new images before detection models can be trained. In addition, both methods perform best when they are used as data augmentation methods - a large amount of synthetic data is mixed with a smaller dataset of real labeled data. The method proposed in the current paper tries to avoid the labeling of real data completely.

3. PROPOSED METHOD

Based on the review of the related literature, we propose that the composition approach is well suited for the task of picking objects from piles. Furthermore, for the most efficient and labeling-free training of the detection models, we develop a method that needs only a few named real images as input. It also does not need the dataset of background images because the background is composed of the piled objects. The main steps of the proposed image generation process are shown in Fig.1.

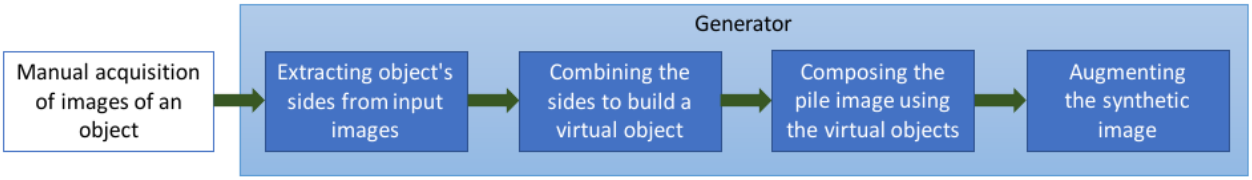
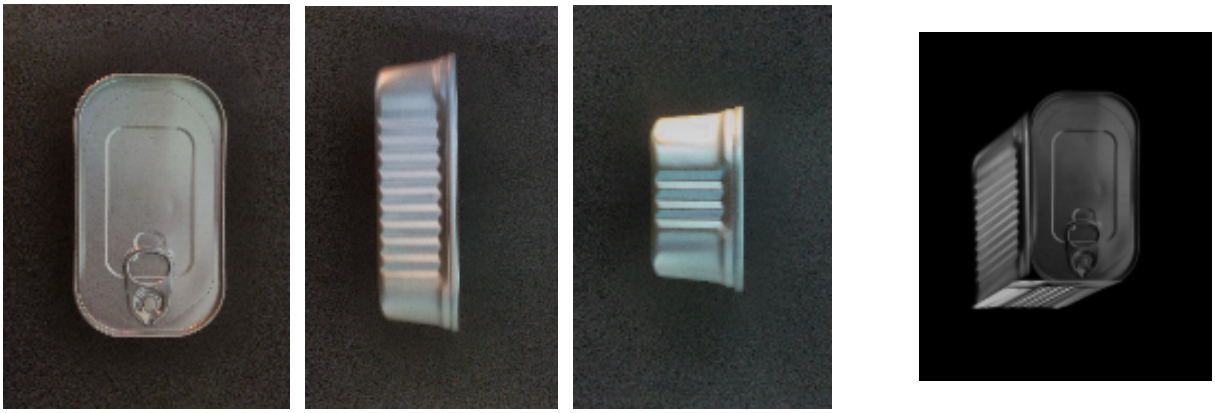


Figure 1. Steps of generating training data

The first step of the method is the only manual part. A human needs to acquire several photos of the object from different sides. The background must be uniform. Resulting images should be named according to the visible side of the object (front, top, left side etc.), and real-life sizes of each side must be given. For example, a metallic can needs at least four images to fully simulate its possible look in a pile. Example images are shown in Fig.2.a. If applicable, the human also needs to specify which sides of the object are suitable for picking up by a robot. For example, if the robot uses a suction cup, then grooves on some sides of the can in Fig.2.a might make these sides unfit for picking.



a) b)

Figure 2. Input images of the generator and the composed object

The generator resizes the input images to 400x400 pixels and automatically cuts out the objects. To do this, the algorithm blurs the resized image with Gaussian filter and applies automatic Canny edge detector. Canny detector uses two thresholds that are found by the following equations:

$$T_1 = max(0, (1 - \sigma) \cdot median(\mathbf{I})) \quad (1)$$

$$T_2 = max(255, (1 + \sigma) \cdot median(\mathbf{I})) \quad (2)$$

Where **I** is the input image (a 2D matrix) and sigma is an empirically chosen value of 0.13. In the resulting edge image, the generator determines all existing contours. The contour with the biggest area is used to create a binary mask image where this contour is filled with positive pixels. The filled contour is further processed by morphological dilation and erosion. The final mask of the contour is used to cut out the object from the input image.

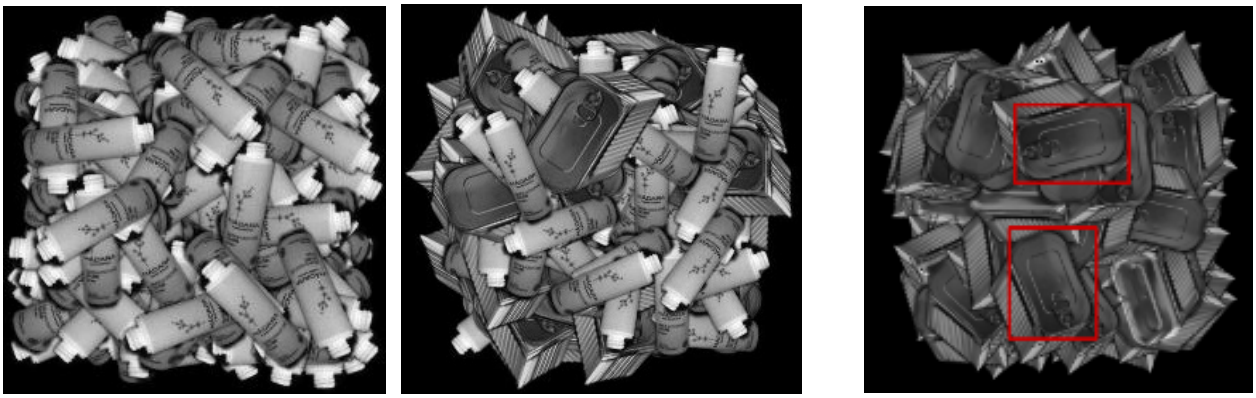
Each extracted side of the object is scaled according to the sizes given by the user. For example, if the name of the image includes the word "Front", then the provided LENGTH and WIDTH parameters are used for the scaling of the cut-out image of this side. In this way, the requirements of the initial image acquisition are relaxed since the different sides of the object can be photographed from different distances and even with different cameras.

The next stage of the generator has different modes depending on the shape of the object. When in a pile, for some objects, such as the box-like cans, several sides can be visible simultaneously. Whereas in the case of flat objects, the detector can simultaneously see only one of the object's faces at a time. So, the extracted side images of flat objects can be directly used by the composing module of the generator. Some other shapes, such as long cylinders can also be simulated similarly to flat objects.

In the case of box-shaped objects, when generating an instance of a virtual object, one of the sides marked as fit for picking is chosen as the main side. The adjacent sides that can be visible alongside the main side are translated and skewed so that when put into the image of the main side, they create an image of a single virtual object (Fig.2.b). In the example, the top side of the metallic can is chosen as the main side. The sides with grooves are never chosen. In the case of flat objects, the only visible side is also the main side.

The last module of the generator copies and places the created virtual objects onto the same image at a random position and with random rotation creating a synthetic pile (Fig.3). The main side of each object has a corresponding bounding box. Then all bounding boxes beginning with the oldest are compared with the later added boxes. If they overlap, then the bounding box that was placed first is deleted. The result of this comparison is that only bounding boxes of the unobstructed objects remain in the image (Fig.3.b). The coordinates of bounding boxes (i.e. annotations) are saved alongside the synthetic image of a pile.

The final step of the generator augments the created image with random changes in brightness and contrast.



a) b)

Figure 3. Examples of a generated synthetic pile images and annotation bounding boxes

After generating thousands of labeled images, one can use them to train deep detection models, that learn to return a bounding box of pickable objects. Such a detector can acquire the center point of the object, and give this coordinate to the robot arm. However, a 2D coordinate is not enough for a robot to pick an object in a 3D environment. For a fully functioning system, we propose to use some distance or depth sensor, for example, Microsoft Kinect. When an object is localized, the depth must be measured at the picking coordinate, so the arm knows how deep it must go to pick the object. Alternatively, a touch sensor (accelerometer or similar) can be placed on the arm, where it would measure when the arm reaches and touches some object at the given coordinate.

4. TESTS

The data generator was implemented in Python language and used OpenCV library. For a detector, we used pretrained YOLO model. Its implementation and training code in TensorFlow can be found at [36]. We trained three models with three different synthetic datasets. First set consisted of generated piles of metallic cans, second – piles of plastic bottles, third - mixed piles with cans and bottles. For each of these tasks, we generated 6000 training and 1000 validation images and trained the model for 25 epochs.

In order to test how well the synthetic learning transfers onto real data, we created three test sets each consisting of 500 images (examples are shown in Fig.4.a). These sets had to be labeled manually, so we created an annotation GUI shown in Fig.4.b. The GUI shows an unlabeled image of the pile to the user. The user has to use a cursor to pick two corner points of the bounding boxes of all objects that can be picked by a robot arm. For faster labeling, while the user picks the location of the top-left corner of the bounding box, the cursor is shaped as a corner. After the user has marked the top-left corner, the GUI draws a possible bounding box from that point to the current cursor location, so the user interactively sees the final bounding box before clicking to approve it.

Table 1 shows the test results. In the considered picking task, the robot has to receive the coordinates of a single object to carry out the picking task. However, the object detector is able to find several objects in the frame. Each of the detected objects has a confidence score; therefore, we send the robot the coordinates of the object with the greatest confidence. If this single object corresponds to one of the manually drawn bounding boxes, then the robot will work correctly with such an input. The bounding boxes are considered to be corresponding if the distance between their corner points is less than 10 pixels. The final accuracy score shows what part of the real test images as an input would result in the robot receiving coordinates corresponding to actually pickable object (last column of the table).

a) b)

Figure 4. Real images from test data and a proposed labeling GUI

Objects	Number of manually labeled objects	Number of detected objects	Number of correctly detected objects	Number of valid first choice picks of an unobstructed object	Accuracy at picking task
Cans	5474	2632	1125	429	85.8%
Bottles	6285	5967	680	356	71.2 %
Mix	9250	7419	2637	494	98.8 %

Table 1. Test results with different object piles

5. CONCLUSIONS

Unsurprisingly, the test results show that one does not achieve state-of-the-art results in object detection by training deep models on synthetic data only. In our tests, many of the manually labeled objects are not detected, and not all returned bounding boxes correspond to actually pickable objects. However, the proposed training approach was developed for a specific task of picking one object per image. The results for this task are promising although dependent on the object of interest.

The proposed system works with simply shaped objects, and such objects are commonly processed in the industry. The examined box-shaped, flat, and cylindrical cases might be combined and augmented to approximate more complex objects - this assumption will be tested in the future work.

Meanwhile, we have shown a practically usable way to deal with the data labeling problem for training deep detection models. This might be an especially valuable result for those factories that often change the objects that their robot arms need to manipulate. The alternative of not using deep models seems less promising since the pile scenario is hard for the classical machine vision approaches. Objects that make up the piles may be almost identical, so the partial detection of the pickable objects using some feature point-based detection is not sufficient to determine which objects are actually fully unobstructed. Model-based detection approaches might be suitable for such task; however, our proposed method seems to offer an easier process of readjusting robots to work with new objects.

Acknowledgment: The research leading to these results has received funding from the research project "Competency Centre of Latvian Electric and Optical Equipment Productive Industry" of EU Structural funds, contract No. 1.2.1.1/16/A/002 signed between LEO Competence Centre and Central Finance and Contracting Agency, Research No.11 " The research on the development of computer vision techniques for the automation of industrial processes".

REFERENCES

[1] International Federation of Robotics, "Executive Summary World Robotics 2017 Industrial Robots", <https://ifr.org/downloads/press/Executive_Summary_WR_2017_Industrial_Robots.pdf> (accessed: 08.07.2018)

[2] R. Kumar, S. Lal et. al. "Object detection and recognition for a pick and place robot," IEEE Asia-Pacific World Congress on Computer Science and Engineering, IEEE. 1-7, (2014)

[3] K. Ikeuchi, B. Horn et. al. "Picking up an Object from a Pile of Objects," Massachusetts Inst of Tech Cambridge Artificial Intelligence Lab. **AI-M-726**, (1983)

[4] A. Gupta, G. Funka-Lea, and K. Wohn, "Segmentation, modeling and classification of the compact objects in a pile," Intelligent Robots and Computer Vision VIII: Algorithms and Techniques. **1192**, 98-110, (1990)

[5] D. Holz, M. Nieuwenhuisen et. al. "Active recognition and manipulation for mobile robot bin picking," Gearing Up and Accelerating Cross-fertilization between Academic and Industrial Robotics Research in Europe, Springer, Cham, 133-153, (2014)

[6] D. Katz, A. Venkatraman et. al. "Perceiving, learning, and exploiting object affordances for autonomous pile manipulation," Autonomous Robots. **37(4)**, 369-382, (2014)

[7] M. Y. Liu, O. Tuzel et. Al. "Fast object localization and pose estimation in heavy clutter for robotic bin picking," The International Journal of Robotics Research. **31(8)**, 951-973, (2012)

[8] L. Chang, J. R. Smith, and D. Fox, "Interactive singulation of objects from a pile," IEEE International Conference on Robotics and Automation, 3875-3882, (2012)

[9] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," The International Journal of Robotics Research. **27(2)**, 157-173, (2008)

[10] O. Russakovsky, J. Deng et. al. "Imagenet large scale visual recognition challenge," International Journal of Computer Vision. **115(3)**, 211-252, (2015)

[11] T.-Y. Lin, M. Maire et. al. "Microsoft coco: Common objects in context," European Conference on Computer Vision, Springer. 740–755, (2014)

[12] W. Liu, D. Anguelov et. al. "Ssd: Single shot multibox detector," European conference on computer vision, Springer, Cham. 21-37, (2016)

[13] R. Girshick, J. Donahue et. al. "Rich feature hierarchies for accurate object detection and semantic segmentation," Proceedings of the IEEE conference on computer vision and pattern recognition. 580-587, (2014)

[14] R. Girshick, "Fast r-cnn," Proceedings of the IEEE international conference on computer vision. 1440-1448, (2015)

[15] S. Ren, K. He et. al. "Faster R-CNN: towards real-time object detection with region proposal networks," IEEE Transactions on Pattern Analysis & Machine Intelligence. **(6)**, 1137-1149, (2017)

[16] J. Redmon, S. Divvala et. al. "You only look once: Unified, real-time object detection," Proceedings of the IEEE conference on computer vision and pattern recognition. 779-788, (2016)

[17] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," arXiv:1612.08242. (2016)

[18] Y. Movshovitz-Attias, T. Kanade, and Y. Sheikh, "How useful is photo-realistic rendering for visual learning?" European Conference on Computer Vision, Springer, Cham. 202-217, (2016)

[19] J. Tobin, R. Fong et. al. "Domain randomization for transferring deep neural networks from simulation to the real world," IEEE/RSJ International Conference on Intelligent Robots and Systems. 23-30, (2017)

[20] S. R. Richter, V. Vineet et. al. "Playing for data: Ground truth from computer games," European Conference on Computer Vision, Springer, Cham. 102-118, (2016)

[21] S. Song, F. Yu et. al. "Semantic scene completion from a single depth image," IEEE Conference on Computer Vision and Pattern Recognition. 190-198, (2017)

[22] A. Gaidon, Q. Wang et. al. "Virtual worlds as proxy for multi-object tracking analysis," Proceedings of the IEEE conference on computer vision and pattern recognition. 4340-4349, (2016)

[23] A. X. Chang, T. Funkhouser et. al. "Shapenet: An information-rich 3d model repository," arXiv:1512.03012. (2015)

[24] S. Choi, Q. Y. Zhou et. al. "A large dataset of object scans," arXiv:1602.02481. (2016)

[25] A. A. Rusu, M. Veceri et. al. "Sim-to-real robot learning from pixels with progressive nets," *arXiv:1610.04286*.(2016)

[26] K. Karsch, V. Hedau et. al. "Rendering synthetic objects into legacy photographs," ACM Transactions on Graphics. **30(6)**, (2011)

[27] H. Su, C. Qi et. al. "Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views," Proceedings of the IEEE International Conference on Computer Vision. 2686-2694, (2015)

[28] X. Peng, B. Sun et. al. "Exploring invariances in deep convolutional neural networks using synthetic images," CoRR, abs/1412.7122. **2(4)**, (2014)

[29] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2315-2324, (2016)

[30] A. Rozantsev, V. Lepetit, and P. Fua, "On rendering synthetic images for training an object detector," Computer Vision and Image Understanding. **137**, 24-37, (2015)

[31] L. Sixt, B. Wild, and T. Landgraf, "Rendergan: Generating realistic labeled data," Frontiers in Robotics and AI. **5(66)**, (2018)

[32] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," arXiv:1701.04862. (2017)

[33] O. Khalil, M. Fathy et. al. "Synthetic training in object detection," 20th IEEE International Conference on Image Processing. 3113-3117, (2013)

[34] D. Dwibedi, I. Misra, and M. Hebert, "Cut, paste and learn: Surprisingly easy synthesis for instance detection," The IEEE international conference on computer vision. (2017)

[35] G. Georgakis, A. Mousavian et. al. "Synthesizing training data for object detection in indoor scenes," arXiv:1702.07836, (2017)

[36] Trieu, "Darkflow" <https://github.com/thtrieu/darkflow> (accessed: 11.07.2018)

AUTHORS' BACKGROUND

Your Name	Title*	Research Field	Personal website
Elvijs Buls		Machine Vision	
Roberts Kadiķis	Phd	Computer Vision	
Ričards Cacurs	Phd candidate	Computer Vision, Sensors	
Jānis Ārents	Phd candidate	Robotics	

*This form helps us to understand your paper better, the form itself will not be published.

*Title can be chosen from: master student, Phd candidate, assistant professor, lecture, senior lecture, associate professor, full professor