# Towards automatic generation of image recognition models for industrial robot arms

Halldor Arnarson
*Department of Industrial Engineering*
*UiT The Arctic University of Norway*
Narvik, Norway
halldor.arnarson@uit.no

Bernt Arild Bremdal
*Department of Computer Science and Computational Engineering*
*UiT The Arctic University of Norway and Smart Innovation Norway*
Narvik and Halden, Norway
bernt.a.bremdal@uit.no and bernt.bremdal@smartinnovationnorway.com

Magnus Fredheim Hanssen
*Department of Computer Science and Computational Engineering*
*UiT The Arctic University of Norway*
Narvik, Norway
mha430@post.uit.no

***Abstract*—As the world moves towards mass customization, there is a need for a manufacturing system that can quickly adapt to market changes. Reconfigurable manufacturing systems (RMS) have been proposed as a solution. RMS is designed to be modular with a high degree of flexibility. However, such a structure creates a lot of complexity. For instance, if the modules are moved or changed, the robot arms in the system must be re-programmed. Adding 3D cameras and image recognition to the robot arms can solve some of these problems. Nevertheless, creating image recognition models is time-consuming work, requires human labor, and can increase the cost of manufacturing. To manufacture a large variety of products, there is a need to create image recognition models for each product. One method to automate the generation of image recognition models can be to use synthetic data. Synthetic data can be used to generate a large amount of labeled data, which can be used to train image recognition models.**

**In this paper, we propose a method for training image recognition models using synthetic data, which can further automate robots in RMS. Specifically, the system utilizes a 3D model of a part to generate images, which are then processed by a cycle generative adversarial network (GAN) to enhance their realism. These images are subsequently auto-labeled and employed to train an image recognition model compatible with an industrial robot arm.**

***Index Terms*—Industrial robot arm, Image recognition, Reconfigurable Manufacturing System (RMS), Cycle Generative Adversarial Network (GAN)**

## I. Introduction

The manufacturing industry is transitioning from mass production towards mass customization, necessitating more frequent changes in manufacturing systems to accommodate new products and variations in demand [1]. Reconfigurable manufacturing systems (RMS) offer a solution to address these changes by providing modular manufacturing systems that can easily scale up or down and adapt to market fluctuations [2]. However, RMS still faces several challenges. For instance, RMS is designed to be reconfigurable at both the hardware and software levels [1], which requires the entire system and control software to be adjustable and flexible. Furthermore, the modular nature of RMS introduces additional complexity to the system [3], resulting in extended setup and programming times for manufacturing systems.

An example of an RMS featuring a modular structure is presented in [4]. This RMS employs a mobile robot to autonomously reconfigure the system's platforms without human intervention. A demonstration video of the system can be viewed at https://youtu.be/UXUlaawd8Ps. However, the system has a drawback: the mobile robot lacks accuracy when positioning the platforms that form the customized production line. Therefore, the robot arms in the system must be programmed for each reconfiguration. Given that RMS is designed for frequent reconfigurations, the robots and system require constant reprogramming, which is both time-consuming and demands robotics expertise. For future research, it is suggested to investigate how the system can be automatically programmed.

One approach that can automate the programming/control of the robot arms in an RMS is to use Industry 4.0 technologies. Industry 4.0 is the fourth industrial revolution which brings new technologies such as the internet of things (IoT), cyber-physical systems (CPS), big data and analytics, simulation and digital twin, artificial intelligence (AI) and additive manufacturing [5]. Singh et al. [6] note that Industry 4.0 technologies are essential for the future success of RMS.

One Industry 4.0 technology that can be used to automate the programming/control of the robot arms is to use AI with a bin-picking system. For example, using 3D cameras with image recognition to pick objects automatically. For example, robot arms equipped with 3D cameras and image recognition can classify objects and determine the distance [7]. Fujita et al. [8] looked at four state-of-the-art bin-picking solutions and investigated what technologies should be combined for effective bin-picking by robots. They found that in all the systems industrial robot arms were used because of their high accuracy and were combined with suction grippers and RGB-D sensors with CNN-based algorithms.

However, the challenge of using the CNN-based algorithms. In a typical machine learning project can be categorized into four steps, data collection, data labeling, model training, and deployment. One challenge is that the labeling step can consume up to 80% of development time [9]. Moreover, deep neural networks require substantial amounts of labeled data for training [10].

This relates to big data, which comprises four dimensions: volume, velocity, variety, and veracity. Volume relates to the amount of data, variety describes the types of data that are available, velocity is related to the speed at which the data is generated and the speed the data is processed, and veracity refers to the reliability (correctness) of the data [11].

Large, diverse, and accurate labeled datasets can be used to develop effective machine learning models. For image recognition, this entails capturing multiple images of an object from various angles, backgrounds, and lighting conditions. This process can be time-consuming, labor-intensive, and expensive, especially when considering the need to adapt to mass customization in manufacturing. Consequently, new machine learning models must be developed for each new product manufactured.

Therefore, to automate this process there is a need for a method to create data that can be used to train the machine learning model. One method that can be used to create training data, is generating synthetic data. Using synthetic data can give a cost-effective method to get large amounts of labeled training data [10].

One of the most used methods to generate synthetic data is generative adversarial networks (GAN) [10]. The GAN are neural networks that consist of two networks, one generator that generates the data and a discriminator. When the model is trained, the generator generates images, and the discriminator will try to identify which images are real and which are fake. The goal when training is to reach an equilibrium where the generated images follow the same distribution as the real images.

Generative Adversarial Networks (GANs) are a versatile class of neural networks that can be employed for a wide range of applications. For instance, Zou et al. [12] utilized GANs to enhance the calibration process of a welding robot, resulting in improved performance, while Mishra et al. [13] leveraged GANs for effective footstep planning in humanoid robots. However, a significant challenge associated with many GANs is the necessity for large datasets containing paired image-to-image translations, such as Pix2Pix [14]. Acquiring these datasets can be difficult and time-consuming.

To tackle this challenge, Zhu et al. [15] used another approach, namely, cycle GAN. Cycle GAN does not require paired images and is trained in an unsupervised manner. The cycle GAN uses two generators and two discriminators, and when training, the images are translated two times. One to translate the image, and a second time to translate the translated image back to the original image. Rao et al. [16] explored the use of cycle Generative Adversarial Networks (GAN) to make simulations more realistic. By using reinforcement learning, robot arms can be trained to pick objects automatically. However, the challenge lies in ensuring the simulation accurately reflects reality, which is where cycle GAN comes in, transforming simulated images to appear more realistic.

In the manufacturing of new products using CNC machines or additive manufacturing, CAD 3D models of the product are often readily available. These 3D models can be harnessed to create synthetic images for training machine learning algorithms. Building on this concept, Hanssen [17] designed a system that employs 3D models to generate images in various orientations, which were subsequently used to train a VGG16 model for image recognition. However, solely relying on the generated images with the VGG16 model [18] did not result in an effective image recognition system. Furthermore, Jordon et al. [10] highlight that the utilization of synthetic data remains an emerging research area, characterized by a scarcity of established frameworks for implementing the technology.

In this paper, we build upon Hanssen's work [17] by combining 3D models with a cycle GAN to create more realistic images and implementing YOLOv5, a fast and powerful image recognition model. We also propose a system structure detailing the necessary steps for creating an image recognition model from a 3D model.

The main contribution is to propose a novel method for automatically generating image recognition models for industrial robot arms in RMS, eliminating the need for reprogramming robots after system reconfigurations. Additionally, we showcase the practical implementation of this approach.

The rest of the paper is organized as follows: Section II proposes how the image recognition model can be generated from the 3D model and how the system works, and in Section III, experimental testing of the system is conducted. Then the paper discusses the results and concludes in Section IV and V.

## II. A METHOD FOR GENERATING SYNTHETIC TRAINING DATA

This section presents a system for automatically generating image recognition models for 3D-printed parts. These models can then be seamlessly transferred to robot arm platforms, enabling the robot arms to directly utilize the image recognition models for object detection.

### A. Generating synthetic data

The first step is to generate images from the 3D model. A Python program imports a 3D model as an STL file, rotates the model to different orientations and generates images from the model, as can be seen in Fig. 1. However, the resulting images may not resemble realistic 3D-printed parts. Therefore, it is necessary to further process and enhance the images to achieve a more lifelike appearance.

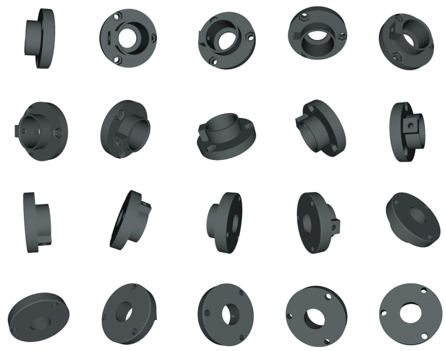

Fig. 1. The generated images of an STL file with different orientations.

## B. Cycle GAN

As mentioned, the generated images do not have realistic features. One method that can be used to make the image look more realistic, is a translation system. The translation system can be used to generate new synthetic images based on real or synthetic images.

Therefore, cycle GAN is trained to translate the synthetic images from the 3D model into real-looking 3D printed parts. When training the cycle GAN, it was noted that if the generated images have white backgrounds, as shown in Fig. 1, the cycle GAN network will end up focusing on the background instead of the parts. Therefore, background images can be inserted into all the generated images for the training of the cycle GAN.

Moreover, filters can also be used. The idea of the filters is to slightly change the images with either a blur filter or by increasing or decreasing the brightness, sharpness, and contrast. If the filters made too big changes to the images, these filters would be added to the cycle GAN. However, small adjustments in the generated images would improve the translated images from the cycle GAN. Fig. 2, shows the images used to train the Cycle GAN.


Fig. 2. The cycle GAN training approach: a) is the generated images, where backgrounds have been inserted, and b) is the real 3D printed parts used to train the cycle GAN.

In this study, we utilized 24 unique 3D models to generate a total of 2,700 synthetic images. The same 3D models were also 3D-printed and photographed, resulting in an additional 2,700 images. This provided us with a combined dataset of 5,400 images, comprising both generated and photographed images. Furthermore, we employed the code from [19] to implement the cycle GAN. The cycle GAN was tested on a 3D model not included in the training dataset, yielding the results illustrated in Fig. 3.

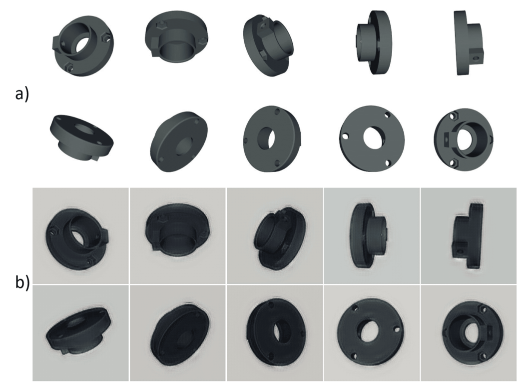


Fig. 3. The image shows the resulting cycle GAN, where a) The input images of the cycle GAN. b) The output from the cycle GAN.

## C. Image recognition model

The You-Only-Look-Once (YOLO) object detection algorithm is known for its high accuracy and rapid processing capabilities, making it suitable for real-time applications [20]. By extracting the x and y coordinates of detected objects, YOLO can be employed to control robots [21]. In the proposed system, YOLOv5 [22] is employed to provide object position information to the robot arm controller.

YOLOv5 primarily consists of four models: YOLOv5x, YOLOv5l, YOLOv5m, and YOLOv5s. The YOLOv5x model is the most comprehensive, generally yielding the best results, while the other three models are simplified versions. The models differ in terms of feature extraction, convolutional kernels, specific network locations, parameter count, and overall size [23].

Given that the generated images contain only one part centrally positioned, an automatic labeler can be used. The "Automatic YOLO Labeler" library on GitHub [24] is capable of identifying the main object within a frame and saving its position. This library leverages the $U^2$-$Net$ [25] for salient object detection, which removes backgrounds in images.

When the images are labeled, a background is added to the pictures and a filter to improve the training of the image recognition model. An illustration of the automatic labeling can be seen in Fig. 6.

## D. The image recognition system

The automatic generation of the image recognition model can be divided into four main steps:

1) Generate images with different orientations.

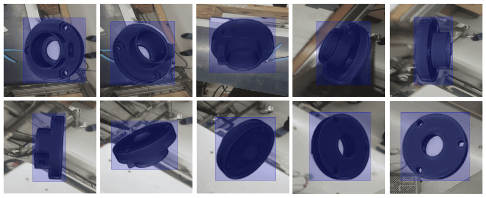

Fig. 4. The images are automatically labeled, and a new background is inserted.

2) Run the images through a cycle GAN to make the images look more realistic.
3) Then label the images, insert background images, and run the images through a filter.
4) Finally, the images are used to train the YOLOv5 model.

All of these steps can be executed automatically, and the image recognition model can be transferred to a robot arm and start picking objects automatically. An illustration of the steps can be seen in Fig. 5.

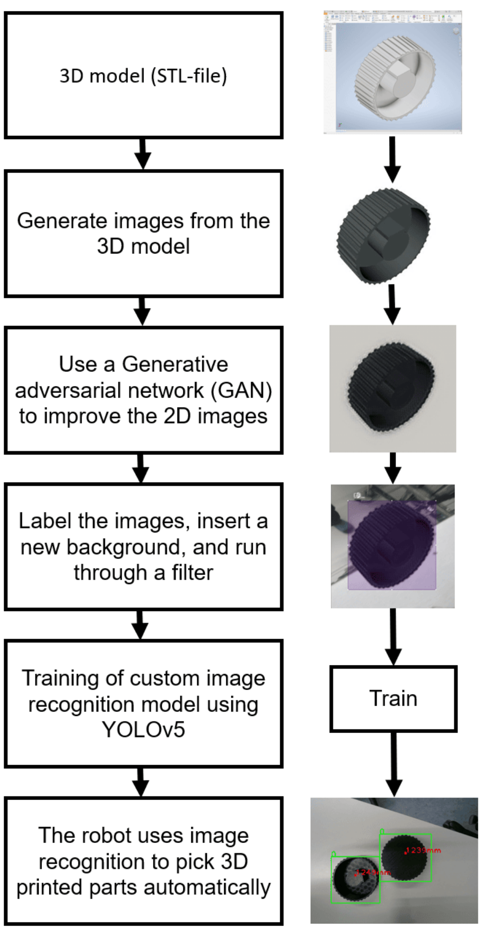


Fig. 5. On the left side, all the steps are used to create the image recognition model, and on the right side, the images are transformed.

## III. System demonstration

A demonstration has been built to showcase how the system works. First, we explain how the image recognition model is created and then show video demonstrations of the system with robot arms.

### *A. Generating the image recognition model*

The system is demonstrated using the three objects. A total of 12,000 images were generated by creating 4,000 images for each of the three 3D models with varying rotations. These images were then processed through the GAN to enhance realism and incorporate background images. As previously mentioned, the YOLOv5 algorithm is employed for the image recognition model, specifically using the largest pre-trained weights model, YOLOv5x [26].

Initial tests revealed that training the model with 100 epochs led to mislabeling and incorrect object identification, whereas training with 200 epochs resulted in overfitting, preventing the model from recognizing the objects. Consequently, training the model for 150 epochs yielded the best outcomes and the loss from the training can be seen in Fig 6. Additionally, an Intel RealSense D405 camera is utilized in the demonstration to obtain depth information from the camera frame.

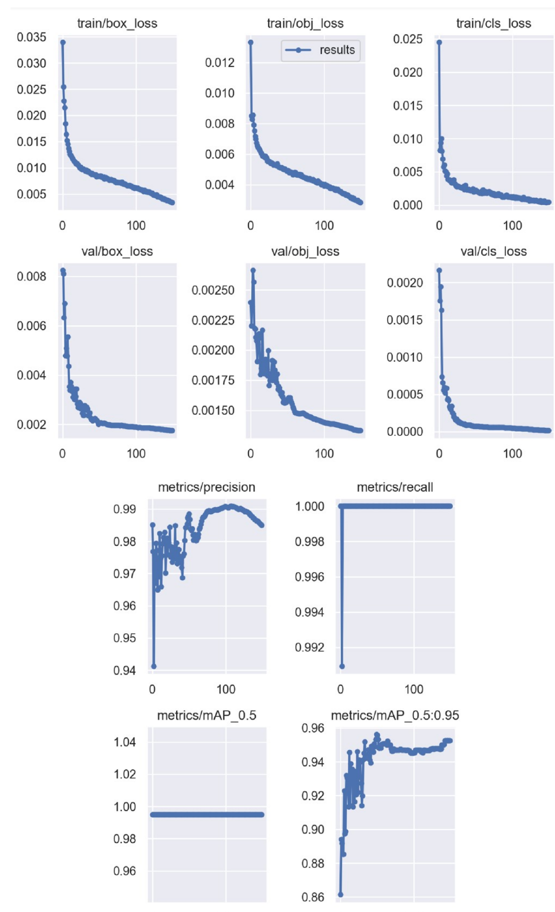


Fig. 6. The loss from training with 150 epochs.

### *B. Demonstrations 1 and 2*

The initial two demonstrations illustrate the performance of the image recognition model in conjunction with different robot arm movements. In the first demonstration, the robot arm moves in a square pattern, increasing its height after each completed pattern. The image recognition model operates simultaneously with the robot arm's movement. A screenshot of this test is provided in Fig. 7, and the video can be viewed at https://youtu.be/6lGjiVP21Dg.

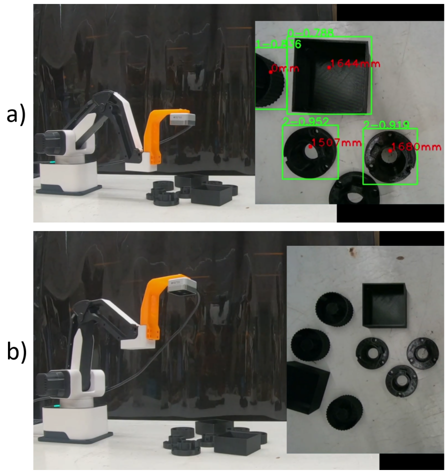


Fig. 7. Screenshot from the first demonstration, with a) depicting the camera approximately 160mm from the table and b) showing the camera 300mm from the table.

The second demonstration, available at https://youtu.be/6TmoyWvbd5Q, features the robot arm moving up and down slowly while the image recognition model runs concurrently.

Both demonstrations reveal that the image recognition model performs well at close distances. However, as the distance increases, the model's ability to recognize the object deteriorates.

### C. Demonstration 3

The objective of the third demonstration is to automatically pick up an object using the image recognition model. A Nachi MZ07 six-axis industrial robot arm equipped with a suction gripper is utilized for this purpose, and the demonstration is limited to a single object. In this demonstration, the robot arm relies on the camera for navigation, adjusting its position based on the object's location within the camera frame. Once the suction cup is aligned with the object, the robot arm descends with a fixed movement to pick it up and then places it in a designated red box. To demonstrate the system's reliability, the robot arm repeats the process three times. The video can be found at https://youtu.be/oD82GAP8Ffs.

## IV. Discussion

Traditionally, RMS needs to set up and program robots for each reconfiguration of the RMS. In this paper, we have proposed a method that can be used to automate the process of creating an image recognition model. This again can allow robot arms in manufacturing systems to become more automated and reduce the need for humans.

Moreover, in Industry 4.0, we have gotten new digital technologies such as digital twins, Big data, and simulation. These technologies can be used to digitalize manufacturing systems, but connecting or using these technologies with physical/real systems can be challenging. Using cycle GAN, can be an effective method to transform digital 3D models and make them look more realistic (real).

We also propose a system to generate the image recognition model automatically. The system takes in a 3D model, which is used to generate synthetic images. These images are then transformed with a cycle GAN, to make them more realistic. Then the images are automatically labeled, a background is added, and a filter is applied to make them ready to be trained. In this system, we use YOLOv5 since it is a fast method that can accurately detect objects but also tell where in the picture the object is. The image recognition model can be directly transferred to the robot arm for the pick and place of parts. It can also allow robot arms to work with objects without any human intervention.

The method achieved good results for close-ups, but several issues were experienced from a distance. To see what the image recognition model is focusing on, EigenCAM [27] is implemented. EigenCAM is a class activation map that can be used to find what pixels of the image the model is focusing on. After implementing EigenCAM, the main problem seems to be that the model is focused on specific parts of the part and not the general shape of the part. Another challenge is the effect of different lighting conditions and the background surface. If the light in the room is too strong or not strong enough can lead to no recognition. In addition, if the object is on a reflective surface and there is a lot of glare, the object will not be recognized.

Furthermore, as seen from the first two demonstration videos, the box is rarely recognized. However, the other objects are very well recognized at a close distance and the image recognition model can label them correctly. The box detection might be worse because it does not contain any clear feature that the image recognition model can focus on.

## V. Conclusion and further work

In this paper, we have developed a method on how an image recognition model can be created automatically without the need for humans. The system takes a 3D model as input and generates images from the 3D model with different orientations. These images are transformed with a cycle GAN, to make them look more realistic. The Images can be automatically labeled, trained, and deployed on a robot arm for pick-and-place operations. This method can therefore be used to automatically create image recognition models, which can reduce the reconfiguration time of RMS.

We have also developed three demonstration videos. The first two videos show the performance of the image recognition model when the robot arm is moving. The third video shows pick and place with an industrial robot arm.

As mentioned in the discussion, there are many challenges with this system that must be solved before this system can be deployed in an RMS. For instance:

- To improve the detection of parts, the image recognition model must be improved. The first part is to find a method that allows for the detection of parts from a distance.
- In this paper, we create a cycle GAN that is used for 3D-printed parts in black. Further work should investigate if the same cycle GAN can be used from parts that come from CNC or turning machines. In addition, create a GAN which can work with all colors, not only black.
- The cycle GAN used in this system can be expanded and improved. This can be done by adding more images of real parts and using more 3D models. In addition, the system can be tested with other methods to create synthetic data, such as variational auto-encoders (VAE).

## Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 825196.

## References

[1] Yoram Koren and Moshe Shpitalni. "Design of reconfigurable manufacturing systems". In: *Journal of Manufacturing Systems* 29.4 (2010), pp. 130–141. ISSN: 0278-6125. DOI: https://doi.org/10.1016/j.jmsy.2011.01.001. URL: https://www.sciencedirect.com/science/article/pii/S0278612511000021.

[2] Yoram Koren, Xi Gu, and Weihong Guo. "Reconfigurable manufacturing systems: Principles, design, and future trends". In: *Frontiers of Mechanical Engineering* 13 (Nov. 2017). DOI: 10.1007/s11465-018-0483-0.

[3] Carin Rösiö and Jessica Bruch. "Exploring the design process of reconfigurable industrial production systems: Activities, challenges, and tactics". In: *Journal of Manufacturing Technology Management* 29.1 (Dec. 2017), pp. 85–103. DOI: 10.1108/JMTM-06-2016-0090.

[4] Halldor Arnarson et al. "Towards automatic configuration and programming of a manufacturing cell". In: *Journal of Manufacturing Systems* 64 (2022), pp. 225–235. ISSN: 0278-6125. DOI: https://doi.org/10.1016/j.jmsy.2022.06.005. URL: https://www.sciencedirect.com/science/article/pii/S0278612522001005.

[5] Ting Zheng et al. "The applications of Industry 4.0 technologies in manufacturing context: a systematic literature review". In: *International Journal of Production Research* 59.6 (2021), pp. 1922–1954. DOI: 10.1080/00207543.2020.1824085.

[6] Ashutosh Singh, Piyush Gupta, and Mohammad Asjad. "Reconfigurable Manufacturing System (RMS): Accelerate Towards Industries 4.0". In: *SSRN Electronic Journal* (Jan. 2019). DOI: 10.2139/ssrn.3354485.

[7] Kai-Tai Song, Yu-Hsien Chang, and Jen-Hao Chen. "3D Vision for Object Grasp and Obstacle Avoidance of a Collaborative Robot". In: *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. 2019, pp. 254–258. DOI: 10.1109/AIM.2019.8868694.

[8] M. Fujita et al. "What are the important technologies for bin picking? Technology analysis of robots in competitions based on a set of performance metrics". In: *Advanced Robotics* 34.7-8 (2020), pp. 560–574. DOI: 10.1080/01691864.2019.1698463.

[9] Sergey Nikolenko. *Synthetic Data for Deep Learning*. Sept. 2019. DOI: https://doi.org/10.1007/978-3-030-75178-4.

[10] James Jordon et al. "Synthetic Data – what, why and how?" In: *arXiv* (May 2022). DOI: 10.48550/arXiv.2205.03257. eprint: 2205.03257.

[11] Alexandra L'Heureux et al. "Machine Learning With Big Data: Challenges and Approaches". In: *IEEE Access* PP (Apr. 2017), pp. 1–1. DOI: 10.1109/ACCESS.2017.2696365.

[12] Yanbiao Zou, Jiaxin Chen, and Xianzhong Wei. "A Calibration Optimization Method for a Welding Robot Laser Vision System Based on Generative Adversarial Network". In: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–13. DOI: 10.1109/TIM.2021.3113950.

[13] Pradumn Mishra et al. "Footstep planning of humanoid robot in ROS environment using Generative Adversarial Networks (GANs) deep learning". In: *Robotics and Autonomous Systems* 158 (2022), p. 104269. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2022.104269. URL: https://www.sciencedirect.com/science/article/pii/S0921889022001580.

[14] Phillip Isola et al. "Image-to-Image Translation with Conditional Adversarial Networks". In: *arXiv* (Nov. 2016). DOI: 10.48550/arXiv.1611.07004. eprint: 1611.07004.

[15] Jun-Yan Zhu et al. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. 2017. DOI: 10.48550/ARXIV.1703.10593. URL: https://arxiv.org/abs/1703.10593.

[16] Kanishka Rao et al. "RL-CycleGAN: Reinforcement Learning Aware Simulation-to-Real". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 11154–11163. DOI: 10.1109/CVPR42600.2020.01117.

[17] Magnus Fredheim Hanssen. "Automatic Generation of Custom Image Recognition Models". UiT Norges arktiske universitet, May 2022. URL: https://munin.uit.no/handle/10037/25768.

[18] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *arXiv 1409.1556* (Sept. 2014).

[19] Aladdin Persson. *Machine-Learning-Collection*. [Online; accessed 8. Dec. 2022]. Dec. 2022. URL: https://github.com/aladdinpersson/Machine-Learning-Collection/tree/master/ML/Pytorch/GANs/CycleGAN.

[20] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91.

[21] Hirohisa Kato et al. "Partial Depth Estimation with Single Image Using YOLO and CNN for Robot Arm Control". In: *2022 IEEE International Conference on Mechatronics and Automation (ICMA)*. 2022, pp. 1727–1731. DOI: 10.1109/ICMA54519.2022.9856055.

[22] ultralytics. *yolov5*. [Online; accessed 8. Dec. 2022]. Dec. 2022. URL: https://github.com/ultralytics/yolov5.

[23] Qisong Song et al. "Object Detection Method for Grasping Robot Based on Improved YOLOv5". In: *Micromachines* 12.11 (2021). ISSN: 2072-666X. DOI: 10.3390/mi12111273. URL: https://www.mdpi.com/2072-666X/12/11/1273.

[24] Abhi Panchal (abpanchal95). *yolo-labeler*. [Online; accessed 8. Dec. 2022]. Dec. 2022. URL: https://github.com/abpanchal95/yolo-labeler.

[25] Xuebin Qin et al. "U2-Net: Going deeper with nested U-structure for salient object detection". In: *Pattern Recognition* 106 (2020), p. 107404. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2020.107404. URL: https://www.sciencedirect.com/science/article/pii/S0031320320302077.

[26] ultralytics. *yolov5*. [Online; accessed 14. Dec. 2022]. Dec. 2022. URL: https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data.

[27] Mohammed Bany Muhammad and Mohammed Yeasin. "Eigen-CAM: Class Activation Map using Principal Components". In: *CoRR* abs/2008.00299 (2020). arXiv: 2008.00299. URL: https://arxiv.org/abs/2008.00299.