Realistic Image Synthesis of Imperfect Specimens using Generative Networks

Deniz Neufeld[1,2], Tobias Würfl[1], Roland Gruber[2], Tobias Schön[2], Andreas Maier[1]

[1]Pattern Recogniton Lab, Friedrich-Alexander-University Erlangen-Nuremberg, Erlangen, Germany,
deniz.neufeld@fau.de, tobias.wuerfl@fau.de, andreas.maier@fau.de

[2]Fraunhofer Entwicklungszentrum Röntgentechnik EZRT, Fürth, Germany,
roland.gruber@iis.fraunhofer.de, tobias.schoen@iis.fraunhofer.de

Abstract

This work explores conditional Generative Adversarial Networks (cGANs) as a tool to create realistic appearing CT slice images with pre-defined imperfections for the verification of automated defect detection algorithms. We aim to create an effective and efficient technique to simulate defects in CT slice images by implementing a new method which can simulate realistic textures and shapes. The problem is stated as an image to image translation task, where a new image is generated based on a given semantic description of the desired realistic image. This semantic description is a material based segmentation of the image with additional circular segments indicating the rough size and position of the intended defect. Based upon related work on cGANs, a convolutional neural network architecture is introduced and applied to our task. This method was trained and evaluated using 2D slice images derived from industrial CT datasets of automotive pistons. Our method showed promising results, producing convincing images in under one minute of computation with image resolutions of 256 by 256 pixels.

Keywords: deep learning, verification defect detection image synthesis, cGAN, data augmentation

1 Introduction

Image processing operators in industrial X-ray Computed Tomography (CT) for defect detection must be verifiable to be consistent and reliable. A verification process necessitates a diverse dataset which depicts the features (e.g. defects, artifacts) of the future imaging objects as accurately as possible. This work focuses on cast metal (e.g. aluminum, steel) objects. Depending on the fabrication method and material used, a wide range of casting defects can occur. Aluminum casts often include unwanted pores and cavities which have to be detected using CT. Creating datasets for verification from real world objects is time consuming or sometimes even impossible, e.g. before the production line's launch where not enough specimens are available.

We demonstrate a method of generating new image samples for defect detection algorithms in CT using deep learning. Though we were unable to identify related work towards this topic, other possible solutions would entail e.g. the manual changing of pixel values to form defects or copying the sub-volume of one defect to a different volume in order to create defects there. Another approach would be to physically manipulate the object, for example by mechanically drilling into the desired positions to simulate defects. All of these solutions are time-intensive and the resulting defects often show characteristics not present in the actual data. This poses the risk that defect detection algorithms are optimized towards the wrong features.

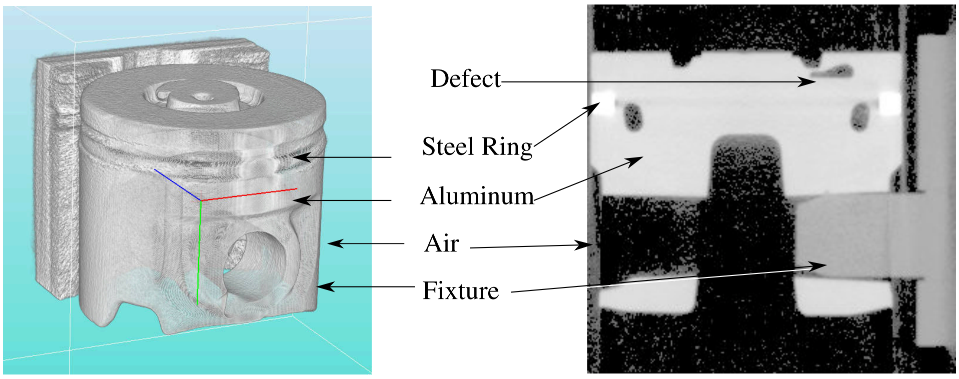
Our goal is to avoid these time intensive and non effective approaches by building a model of the mapping between a segmentation of the object with defect information and its actual CT slice image counterpart. This model can then be used to create CT slice images for segmentations of new specimen, while also providing the ability to specify defects in these new specimen.

In order to avoid the upfront parameterization of this problem, we propose a new approach to create such a model using deep learning – namely image to image translation – using cGANs based on the work of Goodfellow et al.[1], Chen and Koltun [2] and Isola et al. [3]. In recent research generative networks have shown impressive results for the synthesization of real world scenery e.g. faces, city scenes and many more. We extend those methods to create CT slice images with realistic appearing defects at specifiable positions based on a semantic description of an image from a combustion engine piston with an added specification of the desired defects. This way we can introduce new defects into CT images of non-defect parts just by in-painting a circular segment of the intended size and position into the semantic map. Additionally, new parts that are of similar properties to already known datasets (e.g. a new type of piston) can be simulated quickly based on the new objects semantic map which can e.g. be computed from its CAD design.

We will first describe the problem domain and imaging objects followed by our approach to this problem and lastly show the evaluation with test images.

2 Materials

We extracted our dataset from 1055 3D CT volumes with a resolution of [256, 256, 256] (see Figure 1) – with information on region of interest of the defects – from three different piston types. The volumes were segmented using stepwise Otsu threshold binarization [4] to segment into air, fixture (i.e. mounting) material, aluminum and steel. The intended size of the defect and its position are painted as a circular segment which masks the defect with a size of approximately 150% the defect size in order to mask the defect completely.



(a) 3D surface reconstruction

(b) 2D slice

Figure 1: Visualization of a volumetric piston image. The defect displayed in the slice (b) is not visible from the piston's surface (a)

The dataset contained only small amounts of defects since it was taken from a real world production environment. Consequently, in order to improve our dataset for the 2D use case, we selected slice images in three different directions in the volume and only chose slices that contain defects (see Figure 2) in order to balance the dataset. This procedure yielded ca. 2300 slice images which were divided into distinct training, validation and testing datasets.

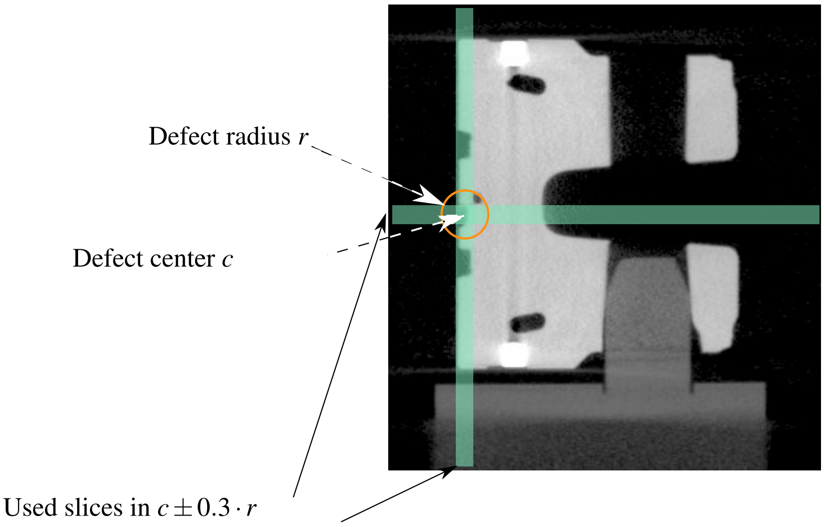
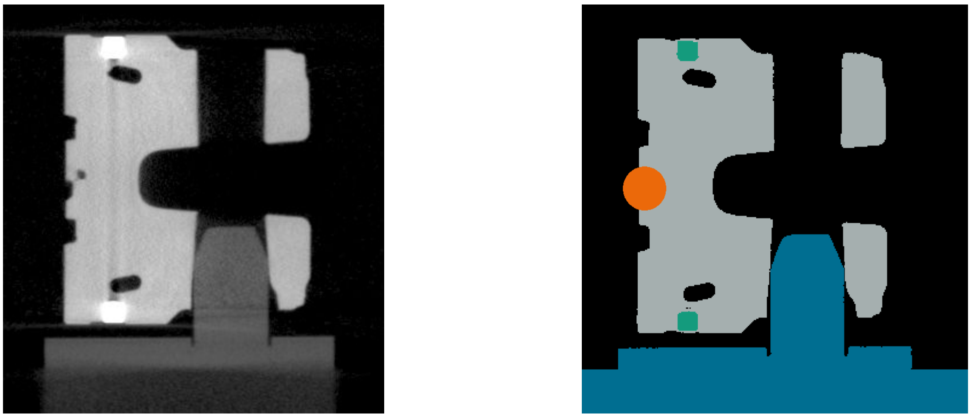


Figure 2: Visualization of the slices used (blue) in the different directions in the piston. Parameters r and c given in pixels.

3 Method

Our goal is to build a model which is able to learn the mapping of a semantic description from a CT slice image segmentation to a realistic looking CT image. Our method is based on deep learning using deep Convolutional Neural Networks (CNN).
The parameters of CNNs are optimized using gradient descent in order to minimize a pre-defined error metric. In our case the error metric has to be a measure on the realism of the generated images. The easiest possibility for this is the Mean Squared Error (MSE) of the predicted and the label image. In our as well as in related research (a. o. [2], [5]) it was observed that this metric cannot enforce high frequency features in the generator network's output. This would be needed for our use case e.g. for the graininess in the homogeneous areas of the images as visible in Figure 1. Parameterizing all features upon which the realism of an image depends upfront is a difficult and a very extensive task which can also be solved more efficiently using deep learning. Consequently, as mentioned before we formulate our task as an image to image translation problem which is typically solved using cGANs, which consist of two separate sub-networks [3]:

- Generator: creates images conditional to a given semantic input
- Discriminator: serves as loss function for the generator



(a) Label reconstruction

(b) Segmentation with masked defect (orange circle)

Figure 3: Original image slice with defect and according resulting segmentation as input for our model. The output is one-hot encoded, ergo shaped [256, 256, 5].

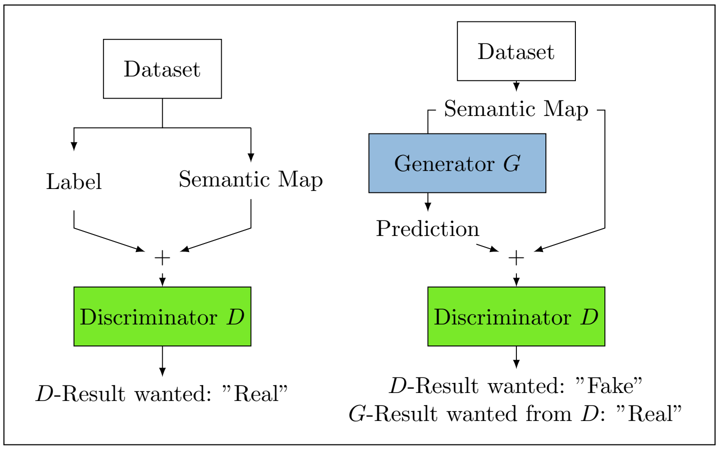


Figure 4: cGAN Model architecture as described in [3]. The model actually consists of two separate networks.

The discriminator receives either real samples or samples created by the generator and estimate the realness of the sample (see Figure 4). The estimation is then used to optimize the generator. This is commonly described as a two-player-game.

Adversarial training processes are difficult to optimize since the improvement rates of the discriminator and the generator have to match. Since both networks are structured differently this means the training process often becomes unstable, e.g. if the discriminator learns too quickly the generator's gradient will vanish, disabling the optimization step. For this reason, according to SRGAN, the training was stabilized by first pre-training the generator on MSE before employing the adversarial training process.

We used the generator architecture described in Figure 5, which we designed specifically for our dataset. It consists of several dilated convolutional kernels [6] that enable the generator to have a complete perception of the input image which we found is necessary in order to reproduce the characteristic appearance of noise of the label images. The generators input is the segmented piston slice image in a one-hot-transformed shape with the dimensions 256,256,5.

The discriminator used is an adapted version of the one described in SRGAN (Figure 6). Our addition to this architecture is that our discriminator uses the input segmentation map as additional input towards the computation of the realness of a predicted or real image. In our trials it was observed that otherwise the generator would often be driven to unlearn the production of defects.

4 Results

The model was trained on a machine with Intel Core i7-2600 CPU @ 3.40 GHz, RAM 32GB, NVIDIA GeForce GTX TITAN X 12 GB RAM, for 30 epochs ca 6h on MSE and for 70 epochs approx. 24h adversarially with a batch-size of 1. During the complete training process the Adam-optimizer was used [7] with the standard parameters as defined in Tensorflow [8] r1.11. The generation of images with the trained model is almost instantaneous. The following results were generated with our final model.

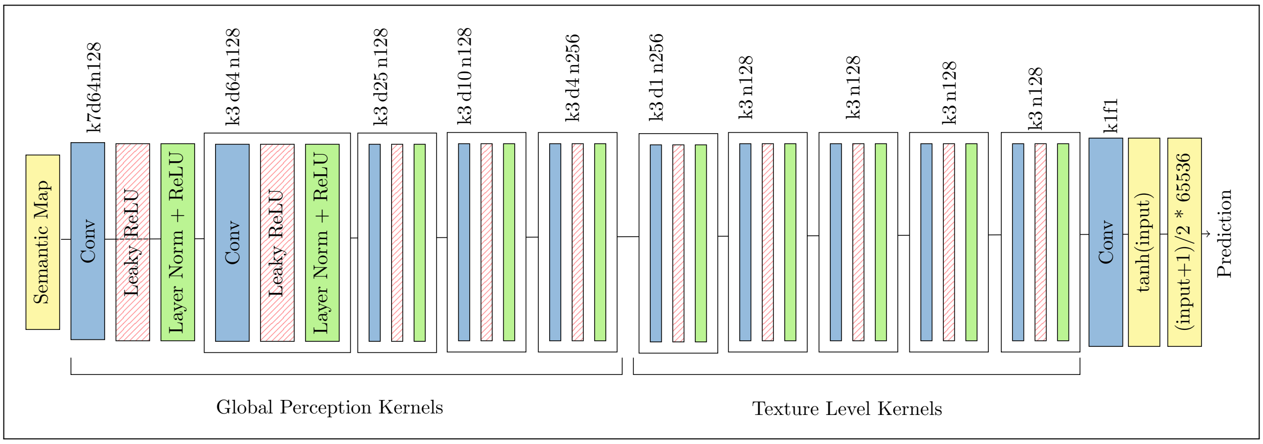


Figure 5: Our Generator Architecture using dilated convolutions

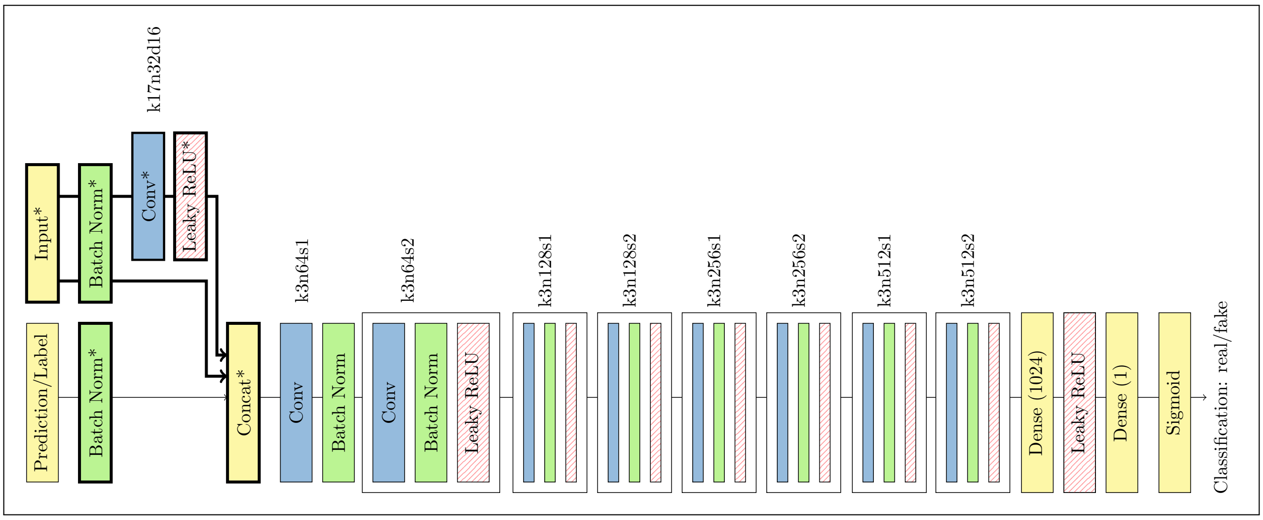


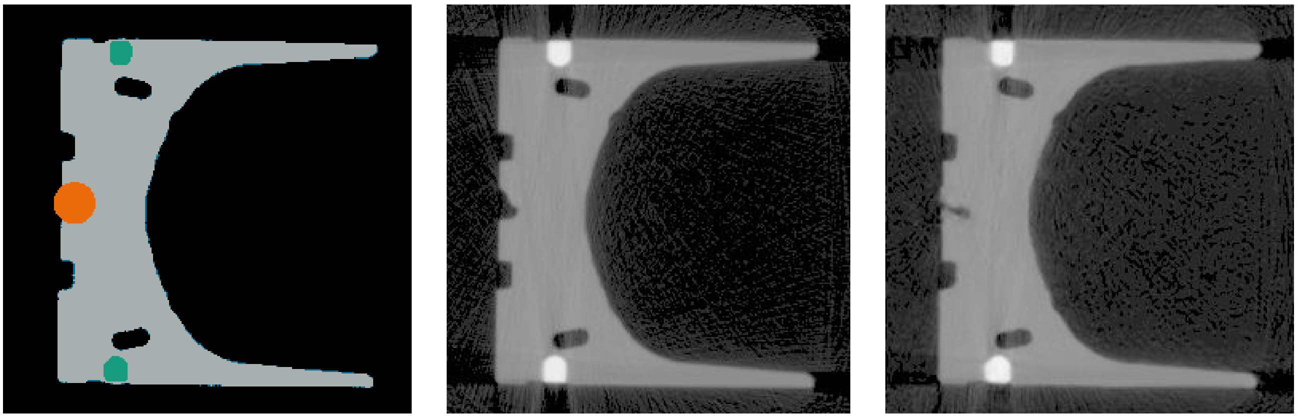
Figure 6: Used discriminator architecture adapted from SRGAN [5]. The boldly drawn elements (marked with *) are our modifications.

Figures 7 and 8 show one example from our testing dataset. Figures 9 shows images generated from hand-edited piston design segmentations, hence there is no given ground truth. Lastly, Figure 10 shows the networks output for a very freestyle type of image.

For standard images from the testing set the generation of the defects was done with great accuracy. In the hand-edited semantic maps the materials are interpreted correctly, though some of the defects do not correlate perfectly with the defect mask given. The textures of the images match the ones of real slices of our dataset very closely.

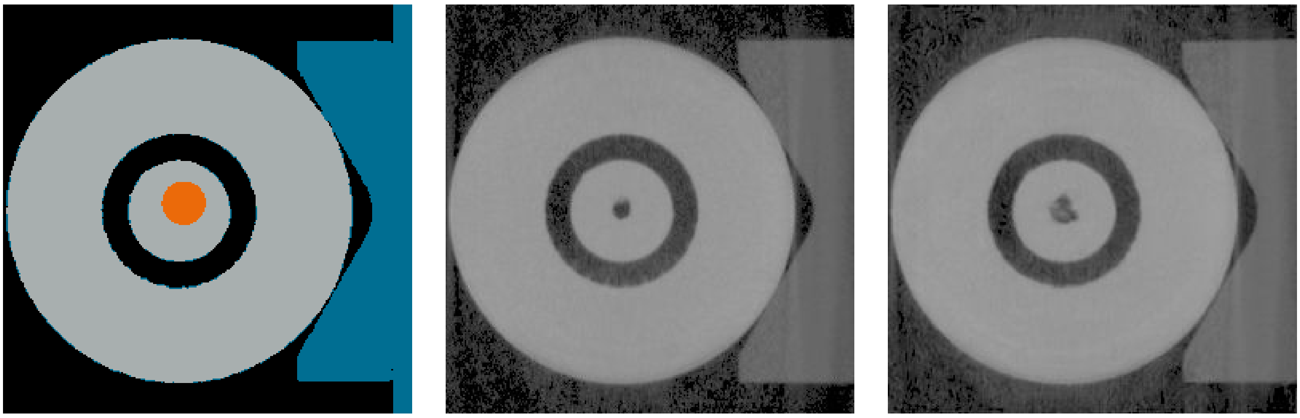
5 Discussion

An issue visible in some of our results is that the shape of the generated defects in some cases do not fill the given defect segment as intended (see Figure 9 with the right defect segment). These issues occur rarely and could be improved e.g. by further tweaking the loss functions used, by using a more diverse dataset, or by augmenting the training images.
Our method, similar to other deep learning approaches, depends on a well preprocessed and diverse dataset. Though we have demonstrated our approach with 2D slice images of automotive pistons, we haven't tested it with bigger volumes of other parts yet. Due to the curse of dimensionality, one has to provide far bigger GPU RAM for 3D images, more-so if the image resolution increases. Additionally, the generated images could not yet be tested with our given image processing operators since these operate in 3D. Nonetheless, training the network in 2D first was an important first step to perform an initial proof of concept on whether our goals were achievable. Furthermore, the operation of 2D is a type of prototyping which saves a lot of time in training.



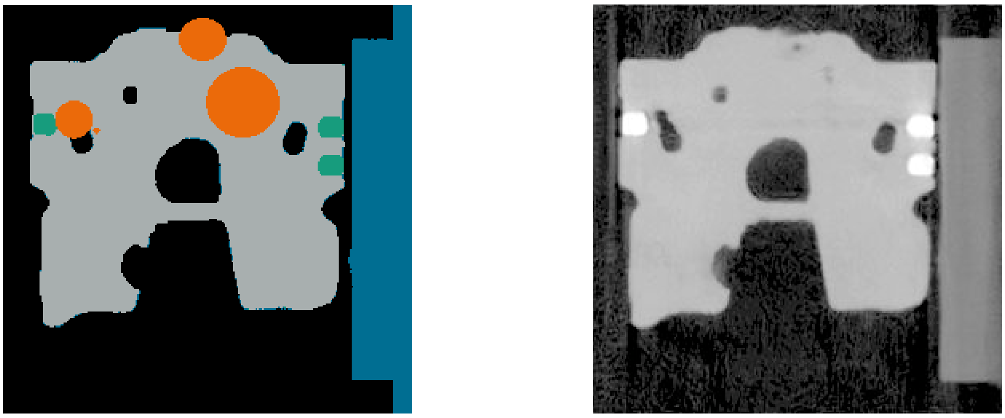
(a) Input (b) Label (c) Prediction

Figure 7: Comparison image on different axis from testing dataset



(a) Input (b) Label (c) Prediction

Figure 8: Comparison image on different axis from testing dataset



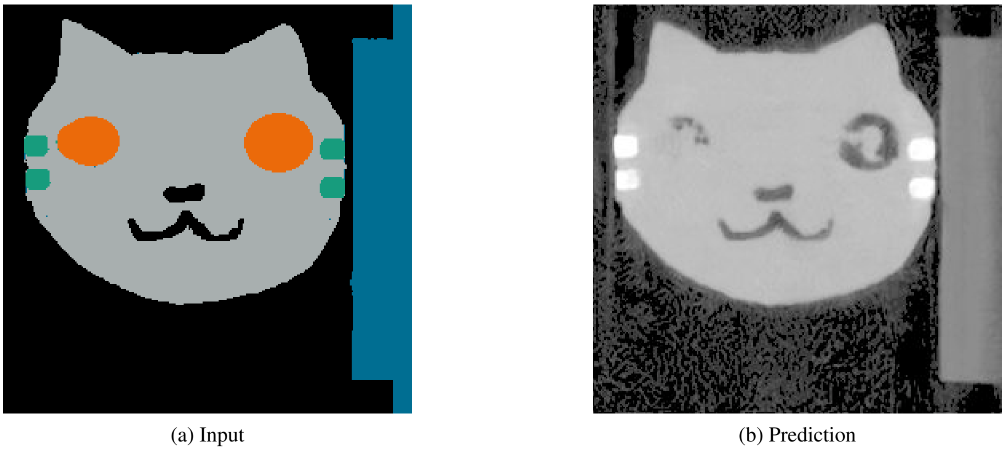
(a) Input (b) Prediction

Figure 9: Result with a hand-painted, more difficult piston.

6 Conclusion & Outlook

We have shown an efficient method to create new image samples for image processing operators using deep learning. The training of our network requires less time than traditional Monte Carlo simulation approaches (30h vs. several days). Generation of samples with a pre-trained model is possible in under 1 min.

We use deep learning to synthesize datasets with pre-definable defects, something not yet achieved by other pieces of related work. Our network is able to accurately learn the appearance and defect properties of our original dataset and can transfer this



(a) Input

(b) Prediction

Figure 10: Result with very different input image.

knowledge to create new samples with the same appearance. We were able to show the robustness of our approach by using it on input shapes very different from our original training dataset.

We expect that translating our approach to the 3D domain will be straightforward by exchanging the 2D-convolutions with 3D, though this could not yet be verified with our limited processing capability. Further work will be needed to accurately verify whether pattern recognition algorithms function as intended on the synthesized datasets.

Acknowledgements
The authors thank MAHLE GmbH and Fraunhofer EZRT for providing the objects and image samples used in this work.

References

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial NetworksarXiv:1406.2661.

[2] Q. Chen, V. Koltun, Photographic image synthesis with cascaded refinement networks, CoRR abs/1707.09405. arXiv:1707.09405.

[3] P. Isola, J. Zhu, T. Zhou, A. A. Efros, Image-to-image translation with conditional adversarial networks, CoRR abs/1611.07004.

[4] N. Otsu, A Threshold Selection Method from Gray-level Histograms, IEEE Transactions on Systems, Man and Cybernetics 9 (1) (1979) 62–66. doi:10.1109/TSMC.1979.4310076.

[5] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi, Photo-realistic single image super-resolution using a generative adversarial network, CoRR abs/1609.04802. arXiv:1609.04802.

[6] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, CoRR abs/1511.07122. arXiv:1511.07122.

[7] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, CoRR abs/1412.6980. arXiv:1412.6980.

[8] M. Abadi, A. Agarwal, P. B. et al., TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org (2015).
URL https://www.tensorflow.org/