

AI-Powered Analysis of Industrial CT Data

Tim Schanz¹, Robin Tenscher-Philipp², Martin Simon³

¹ Tim Schanz (M.Sc)

`tim.schanz@hs-karlsruhe.de`

² Robin Tenscher-Philipp (M.Sc)

`robin.tenscher-philipp@hs-karlsruhe.de`

³ Martin Simon (Prof. Dr.-Ing.)

`martin.simon@hs-karlsruhe.de`

Hochschule Karlsruhe - Technik und Wirtschaft
University of Applied Sciences
Fakultät für Maschinenbau und Mechatronik
Moltkestr. 30
76133 Karlsruhe

Abstract.

Extensive research and development has been conducted in the field of AI-powered analysis of medical CT data during the past years – with significant progress. Although voxel data from industrial parts differ from medical data in the contrast level and the resolution, the medical DL approach is promising. Therefore, the aim of this work is exploring and developing neural network models for detecting defects in industrial CT data. Network architectures, successfully applied to medical CT data, were investigated and derivatives were developed. Different neural network models were trained utilising a mixture of synthetic data and real data. The evaluation showed very good results for a modified U-Net neural network.

Keywords: AI; CT; INDUSTRIAL, CNN, DEEP LEARNING

1 Introduction

Quality assurance is one of the key issues for modern production technologies. Especially new production methods like additive manufacturing and composite materials require high resolution 3D quality assurance methods. Computed tomography (CT) is one of the most promising technologies to acquire material and geometry data non-destructively at the same time.

With CT it is possible to digitalize subjects in 3D, also allowing to visualize their inner structure. A 3D-CT scanner produces voxel data, comprising of volumetric pixels that correlate with material properties. The voxel value (grey value) is approximately proportional to the material density. Nowadays it is still common to analyse the data by manually inspecting the voxel data set, searching for and manually annotating defects. The drawback is that for high-resolution CT data, this process is very time consuming and the result is operator-dependent. Therefore, there is a high motivation to establish automatic defect detection methods.

There are established methods for automatic defect detection using algorithmic approaches. However, these methods show a low reliability in several practical applications. At this point

artificial neural networks come into play that have been already implemented successfully in medical applications [1]. The most common networks, developed for medical data segmentation, are by Ronneberger et al., the U-Net [2] and by Milletari et al., the V-Net [3] and their derivatives. These networks are widely used for segmentation tasks. Fuchs et al. describes three different ways of analysing industrial CT Data [4]. One of these contains a 3D-CNN. This CNN is based on the U-Net architecture and is shown in their previous paper [5]. The authors enhance and combine the U-Net and V-Net architecture to build a new network for examination of 3D volumes. In contrast, we investigate in our work how the networks introduced by Ronneberger et al. and Milletari et al. perform in industrial environments. Furthermore, we investigate if derivatives of these architectures are able to identify small features in industrial CT data.

2 Industrial vs. medical CT data

In industrial CT systems, not only in the hardware design but also in the resulting 3D imaging data differs from medical CT systems. Voxel data from industrial parts differ from medical data in the contrast level and the resolution. State-of-the-art industrial CT scanner produce one to two order of magnitude larger data sets compared to medical CT systems. The corresponding resolution is necessary to resolve small defects. Medical CT scanners are optimised for a low x-ray dose for the patient, the energy of x-ray photons are typically up to 150 keV, industrial scanner typically use energies up to 450 KeV. In combination with the difference of the scan “object”, the datasets differ significantly in size and image content.

To store volume data there are a lot of different file formats. Some of them are mainly used in medical applications like DICOM [6], NifTi¹ or RAW. In industrial applications VGL³, RAW and TIFF⁴ are commonly used. Also depending on the format, it is possible to store the data slice wise or as a complete volume stack.

3 CT data for training and evaluation

Industrial CT data, as mentioned in previous section, has some differences to medical CT data. One aspect is the size of the features to be detected or learned by the neural network. Our target is to find defects in industrial parts. As an example, we analyse pores in casting parts. These features may be very small, down to 1 to 7 voxels in each dimension. Compared to the size of the complete data volume (typically larger than 512 x 512 x 512 voxel), the feature size is very small. The density difference between material and pores may be as low as 2% of the maximum grey value. Thus, it is difficult to annotate the data even for human experts. The availability of real industrial data of good quality, annotated by experts, is very low. Most companies don't reveal their quality analysis data. Training a neural network with a small quantity of data is not possible. For medical applications, especially AI applications, there are several public datasets available. Yet these datasets are not always sufficient and researchers are creating synthetic medical data [7].

¹ Details available at: <https://nifti.nimh.nih.gov/> - 20/03/12

³ Details available at: <https://www.volumegraphics.com/> - 20/03/12

⁴ Details available at: <https://kb.iu.edu/d/afjn> - 20/03/12

Therefore, we decided to create synthetic industrial CT data. Another important reason for synthetic data is the quality of annotations done by human experts. The consistency of results is not given for different experts. Fuchs et al. have shown that training on synthetic data and predicting on real data lead to good results [4]. However, synthetic data may not reflect all properties of real data. Some of the properties are not obvious, which may lead to ignoring some varieties in the data. In order to achieve a high associability, we use a large numbers of synthetic data mixed with a small number of real data. To achieve this, we developed an algorithm which generates large amounts of data, containing a large variation of aspects, needed to generalize a neural network. The variation includes material density, pore density, pore size, pore amount, pore shape and size of the part.

There are some samples that could be learned easily, because the pores are clearly visible inside the material. However, some samples are more difficult to be learned, because the pores are nearly invisible. This allows us to generate data with a wide variety and hence the network can predict on different data. To train the neural networks, we can mix the real and synthetic data or use them separately. The real data was annotated manually by two operators.

To create a dataset of this volume we sliced it into 64x64x64 blocks. Only the blocks with a mean density greater than 50% of the grayscale range are used, to avoid too much empty volumes in the training data. Another advantage of synthetic data is the class balance. We have two classes, where 0 corresponds to material and surrounding air and 1 for the defects. Because of the size of the defects there is a high imbalance between the classes. By generating data with more features than in the real data, we could reduce the imbalance. Reducing the size of the volume to 64x64x64 also leads to better balance between the size of defects compared to full volume. In **Table 1** details of our dataset for training, evaluation and testing are shown. The synthetic data will not be recombined to a larger volume as they represent separate small components or full material units.

Table 1: Overview of used datasets.

Name	Description	Resolution	No. of samples	No. of training samples	No. of evaluation samples	No. of test samples
Gdata	synthetic	64x64x64	7249	6198	688	363
Rdata	real	64x64x64	156	135	15	6
Mdata	mixed	64x64x64	7405	6334	703	368

The following two slices of real data (**Figure 1**) and synthetic data (**Figure 2**) with annotated defects show the conformity between the data.

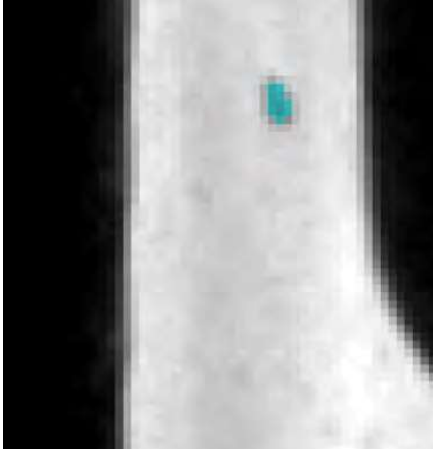


Figure 1: Sample slice of real data with size of 64x64x64 voxel.



Figure 2: Sample slice of synthetic data with size of 64x64x64 voxel.

4 Hardware and software setup

Deep learning (DL) consist of two phases: The training and its application. While DL models can be executed very fast, the training of the neural network can be very time-consuming, depending on several factors. One major factor is the hardware. The time consumed can be reduced by the factor of around ten when graphics cards (GPUs) are used. [8] To cache the training data, before it is given into the model, calculated on the GPU, a lot of random-access memory (RAM) is used [9] [10] [11]. Our system is built on a dual CPU hardware with 10 cores each running at 2.1 GHz and a Nvidia GPU Titan RTX⁵ with 24GB of VRAM and 64GB of regular RAM. All measurements in this work concerning training and execution time are related to this hardware setup.

The operating system is Ubuntu 18.4LTS. Anaconda is used for python package management and deployment. The DL-Framework is Tensorflow⁶ 2.1 and Keras as a submodule in Python⁷.

5 Neural network architecture

Based on the 3DU-Net [12] and 3DV-Net [3] architecture compared from Paichao et al. [13] we created modified versions which differ in number of layers and their hyperparameters. Due to the small size of our data, no patch division is necessary. Instead the training is performed on the full volumes. We actually do not use the Z-Net enhancement proposed in their paper. The input size, depending on our data, is defined to 64x64x64x1 with 1 dimension for channel. The incoming data will be normalized. As we have a binary segmentation task, our output activation is the sigmoid [14] function. Based on Paichao et al. [13] the convolutional layer of our 3DU-Nets have a kernel size of (3, 3, 3) and the 3DV-Nets have a kernel size of (5, 5, 5). As convolution activation function we are using ELU [14] [15] and he_normal [16] as kernel initialization [17]. The ADAM optimisation method [18] [19] is used with a starting learning

⁵ Product page: <https://www.nvidia.com/de-at/titan/titan-rtx/> - 20/03/12

⁶ Details available at: <https://www.tensorflow.org/> - 20/03/12

⁷ Details available at: <https://keras.io/> - 20/03/12

rate of 0.0001, a decay factor of 0.1 and the loss function is the binary cross-entropy [20]. **Figure 3** shows a sample 3DU-Net architecture where downwards max pooling and upwards transposed convolution are used. Compared to **Figure 4**, the 3DV-Net, where we have a fully convolutional neural network, the descend is done with a (2, 2, 2) convolution and a stride of 2 and ascent with transposed convolution. It also has a layer level addition of the input of this level added to the last convolution output of the same level, as marked by the blue arrows. To adapt the shapes of the tensors for adding them, the down-convolution and the last convolution of the same level, have to have the same number of kernel filters.

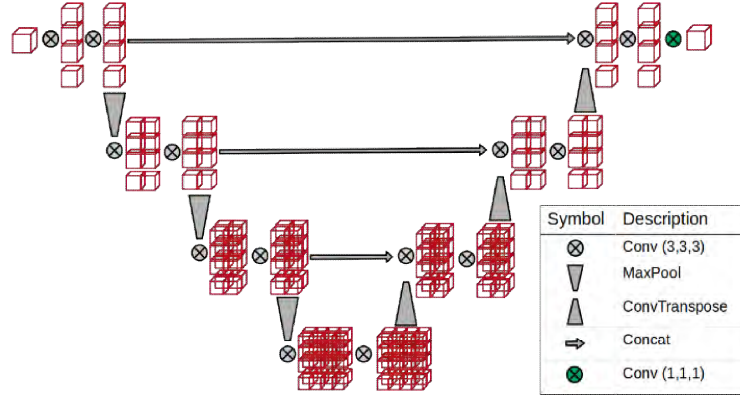


Figure 3: Sample U-Net architecture for building reference.

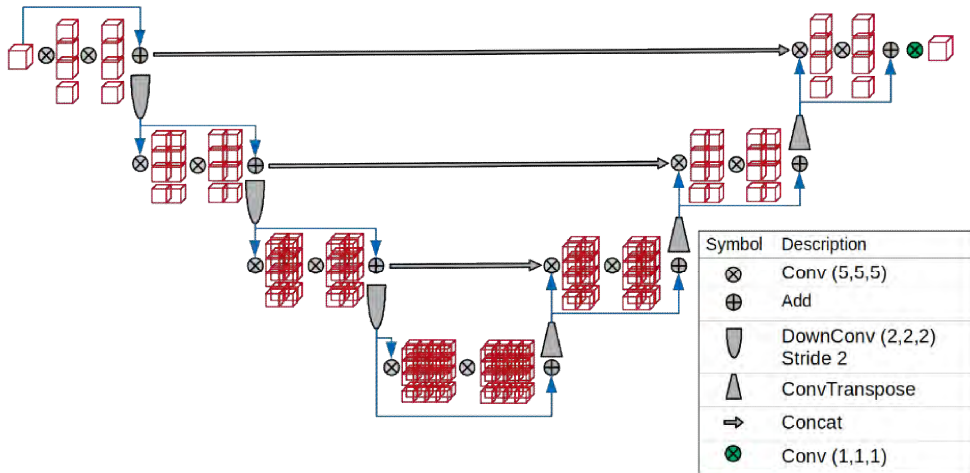


Figure 4: Sample V-Net architecture for building reference.

Our modified neural network differ in the levels of de-/ascending, the convolution filter kernel size and their hyperparameters, shown in **Table 2**. The convolutions on one level have the same number of filter kernel. After every down convolution the number of filters is multiplied by 2 and on the way up divided by 2.

Table 2: Tested models and their base specifications.

Architecture	Starting Kernel Number	Kernel Size	Down/Up-Levels	Hyperparameter
U-Net_small	16	(3, 3, 3)	3_BOTTOM_3	$1,40 \times 10^6$
U-Net_large	16	(3, 3, 3)	5_BOTTOM_5	$3,41 \times 10^7$
V-Net_small	16	(5, 5, 5)	3_BOTTOM_3	$1,06 \times 10^7$
V-Net_large	16	(5, 5, 5)	5_BOTTOM_5	$1,77 \times 10^8$

6 Training and evaluation of the neural networks

The conditions of a training and a careful parameters selection is important. In **Table 3** the training conditions fitted to our system and networks are shown. We are also taking into account that different network architectures and number of layers are better performing on different learning rates, batch size, etc.

Table 3: Training conditions.

Parameter	Description	Value
Batchsize	Number of sample per iteration	5
Epochs	Number of iterations of all samples	15-90
Learning rate	Factor for weight adjustment	0.001
Shuffle data	Shuffle data before loading in batches	True
Learning rate decay	Reduction of learning rate when reaching plateau	0.1

To evaluate our trained models, we are mainly focusing on the IoU metric, also called Jackard Index, which is the intersection over union. This metric is widely used for segmentation tasks and compares the intersection over union between the prediction and ground truth for each voxel. The value of IoU range between 0 and 1, whereas the loss values range between 0 and infinite. Therefore, the IoU is a much clearer indicator. An IoU close to 1 indicates a high intersection-precision between the prediction and the groundtruth. Our networks where trained between 30 and 90 epochs until no more improvement could be achieved. Both datasets consist of a similar number of samples, which means the epoch time is equivalent. One epoch took around 4 minutes.

Figure 5 shows the loss determined based on the evaluation data. As described in Fehler! Verweisquelle konnte nicht gefunden werden., all models are trained on and evaluated against the synthetic dataset Gdata and on the mixed dataset Mdata. In general, the loss achieved by all models is higher on Mdata because the real data is harder to learn. A direct comparison between the models is only possible between models with the same architecture. The IoU metric shown in **Figure 6**. Here the evaluation is sorted based on the IoU metric. If we compare the loss of UNET-Mdata with UNET-Gdata, which are nearly the same for Mdata, with their corresponding IoU (UNET-Mdata (~ 0.8) and UNET-Gdata (~ 0.93)), we can see that a lower loss does not necessarily lead to higher IoU score. If only the loss and IoU are considered, the UNets tend to

be better than the VNNets. As a conclusion, considering the IoU metric for model selection, the UNET-Gdata is the best performing model and VNET-Gdata the least performing.

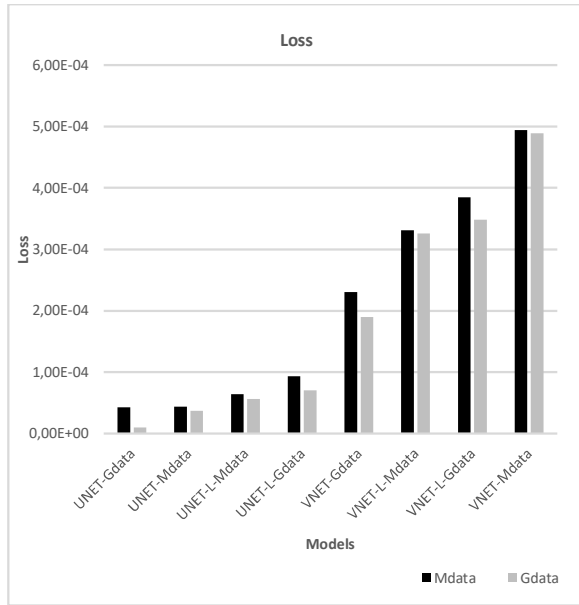


Figure 5: The evaluation loss determined based on the evaluation data sorted from lowest to highest.

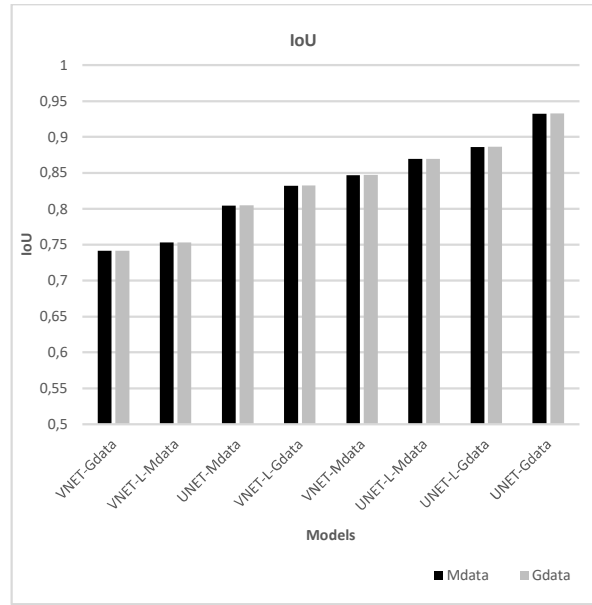
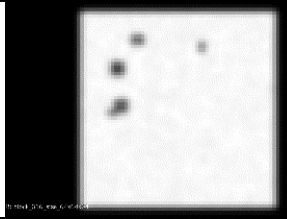
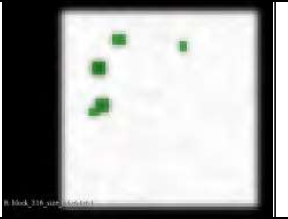
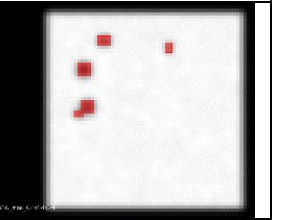
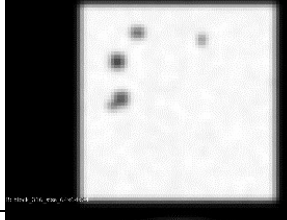
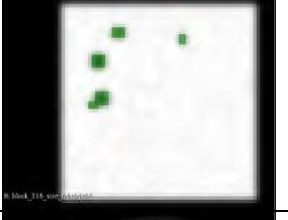
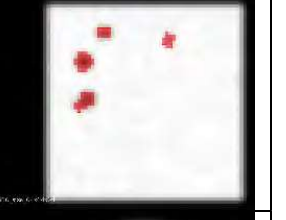
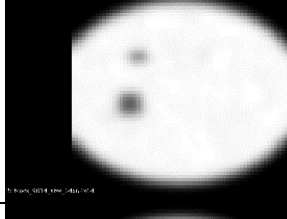
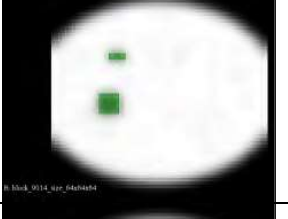
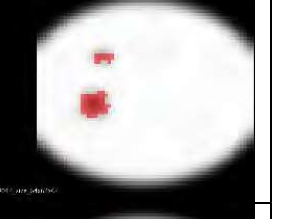
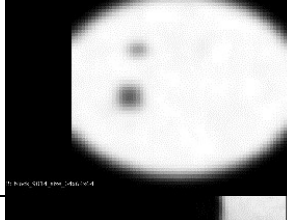
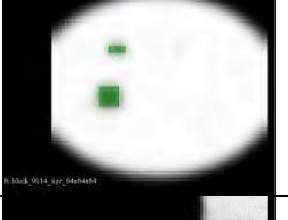
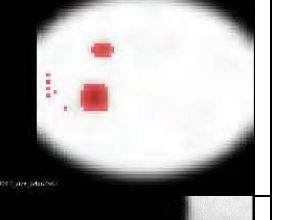

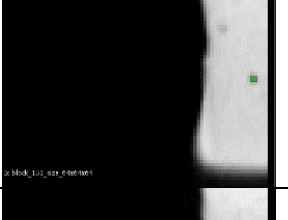
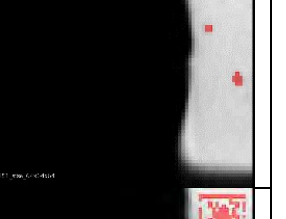





Figure 6: The evaluation IoU determined based on the evaluation data sorted from lowest to highest.

After comparing the automatic evaluation, we show prediction samples of different models on real and synthetic data (**Table 4**). Rows 1 and 2 show the comparison between UNET-Gdata and VNET-Gdata, predicting on a synthetic test sample. The result of UNET-Gdata exactly hits the groundtruth, whereas the VNET-Gdata prediction has a 100% overlap to the groundtruth but with surrounding false positive segmentations. In row 3 and 4 both models predict the groundtruth plus some false positive segmentations in the close neighbourhood. In row 5 and 6 the prediction results of the same two models on real data is shown, taking into account that both models are not trained on real data. UNET-Gdata delivers a good precision with some false positive segmentations in the groundtruth area and one additional segmented defect. This shows that the model was able to find a defect which was missed by the expert. VNET-Gdata shows a very high number of false positive segmentations.

Table 4: Overview of predictions by different models on synthetic and real test samples.

Model	Data type	Sample	Groundtruth	Prediction
UNET-Gdata	synthetic			
VNET-Gdata	synthetic			
UNET-Gdata	synthetic			
VNET-Gdata	synthetic			
UNET-Gdata	real			
VNET-Gdata	real			

7 Conclusion

In this paper, we have proposed a neural network to find defects in real and synthetic industrial CT volumes. We have shown that neural networks, developed for medical applications can be adapted to industrial applications. To achieve high accuracy, we used a large variety of features in our data. Based on the evaluation and manually reviewing random samples we have chosen the UNET architecture for further research. This model achieved great performance on our real and synthetic dataset. In summery this paper shows that the artificial intelligence and their neural networks will take an import enrichment in industrial issues.

References

- [1] P. F. Christ, F. Ettlinger, F. Grün, M. E. A. Elshaera, J. Lipkova, S. Schlecht, F. Ahmaddy, S. Tataavarty, M. Bickel, P. Bilic, M. Rempfler, F. Hofmann, M. D. Anastasi, S.-A. Ahmadi, G. Kaissis, J. Holch, W. Sommer, R. Braren, V. Heinemann and B. Menze, “Automatic Liver and Tumor Segmentation of CT and MRI Volumes using Cascaded Fully Convolutional Neural Networks,” 20 2 2017.
- [2] O. Ronneberger, P. Fischer and T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, 2015.
- [3] F. Milletari, N. Navab and S.-A. Ahmadi, V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation, 2016.
- [4] P. Fuchs, T. Kröger and C. S. Garbe, “Self-supervised Learning for Pore Detection in CT-Scans of Cast Aluminum Parts,” International Symposium on Digital Industrial Radiology and Computed Tomography, 2 – 4 July 2019 in Fürth, Germany (DIR 2019), 11 2019.
- [5] P. Fuchs, T. Kröger, T. Dierig and C. S. Garbe, “Generating Meaningful Synthetic Ground Truth for Pore Detection in Cast Aluminum Parts,” 9th Conference on Industrial Computed Tomography (iCT) 2019, 13-15 Feb, Padova, Italy (iCT 2019), 3 2019.
- [6] NEMA PS3 / ISO 12052, Digital Imaging and Communications in Medicine (DICOM) Standard, National Electrical Manufacturers Association, Rosslyn, VA, USA.
- [7] D. Jin, Z. Xu, Y. Tang, A. P. Harrison and D. J. Mollura, “CT-Realistic Lung Nodule Simulation from 3D Conditional Generative Adversarial Networks for Robust Lung Segmentation,” 11 6 2018.
- [8] Y. LeCun, 1.1 deep learning hardware: Past, Present, and Future. In 2019 IEEE International Solid-State Circuits Conference-(ISSCC), IEEE, 2019.
- [9] N. Jangamreddy, “A Survey on Specialised Hardware for Machine Learning,” 2019.
- [10] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen and J. S. Rellermeier, “A Survey on Distributed Machine Learning,” 20 12 2019.
- [11] V. Sze, Y.-H. Chen, J. Emer, A. Suleiman and Z. Zhang, “Hardware for machine learning: Challenges and opportunities,” in 2017 IEEE Custom Integrated Circuits Conference (CICC), 2017.
- [12] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox and O. Ronneberger, 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation, 2016.
- [13] P. Li, X.-Y. Zhou, Z.-Y. Wang and G.-Z. Yang, Z-Net: an Anisotropic 3D DCNN for Medical CT Volume Segmentation, 2019.

- [14] C. Nwankpa, W. Ijomah, A. Gachagan and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," 8 11 2018.
- [15] D.-A. Clevert, T. Unterthiner and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," 23 11 2015.
- [16] K. He, X. Zhang, S. Ren and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," 6 2 2015.
- [17] A. Daniely, R. Frostig and Y. Singer, "Toward Deeper Understanding of Neural Networks: The Power of Initialization and a Dual View on Expressivity," 18 2 2016.
- [18] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 22 12 2014.
- [19] S. R. Dubey, S. Chakraborty, S. K. Roy, S. Mukherjee, S. K. Singh and B. B. Chaudhuri, "diffGrad: An Optimization Method for Convolutional Neural Networks," IEEE Transactions on Neural Networks and Learning Systems, 2019, 12 9 2019.
- [20] S. S. M. Salehi, D. Erdogmus and A. Gholipour, "Tversky loss function for image segmentation using 3D fully convolutional deep networks," 18 6 2017.