AI-Powered Defect Segmentation in Industrial CT Data

Tim Schanz[1], Robin Tenscher-Philipp[2], Fabian Marschall[3], Martin Simon[4]

[1] Tim Schanz (M.Sc)
tim.schanz@h-ka.de
[2] Robin Tenscher-Phlipp (M.Sc)
robin.tenscher-philipp@h-ka.de
[3] Fabian Marschall (B.Eng)
fabian.marschall@h-ka.de
[4] Martin Simon (Prof. Dr.-Ing.)
martin.simon@h-ka.de

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

Abstract. Non-destructive quality testing using CT plays an important role in industrial quality assurance. However, manual analysis of the large voxel data sets is not efficient. AI-powered processes have already shown that successful segmentation of defects in industrial voxel data is possible. In this paper, we show an AI-solution to detect small defects in industrial CT data. Therefore, we propose two new network architectures, PU-Net and PCU-Net based on an U-Net architecture. For this purpose, we first conducted a parameter study to determine the parameters with the greatest impact on the segmentation performance and incorporated them into the new architecture. In addition, we improved a reference dataset by introducing a data augmentation and also improved the annotation of the real data in this dataset. The evaluation of the new architectures showed very good results.

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

1 Introduction

With ever increasing demand for resource-saving and thus more environmentally friendly production, the need for optimized processes and rising quality is growing. In modern production systems for highly stressed or safety-relevant components, the internal condition of the part is of high importance for quality assurance. The method of choice is Computed Tomography (CT), which is one of the most important non-destructive testing methods. The CT provides 3D image data of the scanned component, consisting of volumetric pixels (voxels). The gray values of the voxels correlate with the density of the part. Thus, defective areas can be determined based on the gray value. Until now, this process has required a trained worker to inspect the components individually. There are also algorithmic software solutions which lack in reliability because of the user influence on manually set thresholds. In this process, artifacts and pixel defects complicate the work. Particularly in inline applications, where the CT is integrated into a production process, the quality of the images suffers. As a result, algorithmic methods for detecting defects on the basis of gray value distributions are not an option.

With the help of artificial intelligence, it is possible to automize these tasks in order to save personnel in production. This is particularly important in view of the shortage of skilled workers, which will become even more acute in the coming years. The progress of image processing with neural networks in medical technology can be adapted to industrial applications. With the U-Net architecture [1] [2], a promising possibility for the segmentation of medical CT data was created, which is used in a modified way. This u-shaped architecture was the baseline for several new derivatives for image segmentation. For industrial CT data analysis researchers [3] [4] also used the classical U-Net architectures and applied changes to fulfill the task. In this paper, besides the architecture, the data set will be considered in detail. Here, the representation and distribution of the pores, the variance of the data as well as the generation of the synthetic data are important. Because training such a neural network requires a very large amount of data, which is difficult to obtain for various reasons. A major challenge is that real industrial CT data is not publicly available because companies don't distribute their CT data and even if they make them public the data must be annotated by an expert. In addition, the effort for annotation is enormous and requires trained personnel. It is very likely that some defects will be incorrectly annotated or even completely overlooked.

In a previous work, we have already published different approaches for the segmentation of CT data, which yielded promising results [5]. The goal of this work is to further improve the performance of the Network. In the following, these methods are presented.

2 CT data for training and evaluation

In this work we consider the detection of defects described as pores. These are gas inclusions that are perceptible as dark spots in the 3D volume data. Their gray value is therefore lower than the surrounding material. In this research pores can have a minimum extent of 2^3 voxel and up to of 10^3 voxel, depending the resolution of the CT scanner in microns per voxel. The very small size of the defects leads to a problem with class balance, because the number of voxels assigned to the background is very large compared to the voxels of the defect class. For example, in a training dataset of ~10.000 samples, a resolution of 64^3 per sample, an average of 35 pores per sample and an average size of 3^3 voxel per pore, the pores take an average of 0.36% of one whole volume. Thus, loss functions as well as metrics must be used to account for this problem. In addition to this, care can be taken when generating the synthetic data to oversample the errors in order to address this problem. For the reasons already mentioned, only a small amount of annotated real data is available for the training. To enable the training, a considerably larger number of synthetically generated data are added to the real data. The approach for the generation is algorithmic and is approximated as closely as possible to the real data. Where the main feature is the grey scale gradient from material to the inner of the pore as well as the shape. During the data development and the data generation process, it is necessary to have an eye on the network, the network architecture and the application, which the network must fulfill, to further improve the quality and the authentic representation of synthetic data.

To further increase the variation in the industrial CT data set used in [5] (hereafter referred to as reference data set reflCTDS) [5], Table 1), various augmentation methods were used. With data augmentation, the variation of the 3D data could be significantly increased by randomly applying rotation, flipping, cropping and elastic deformation on the complete dataset. The ratio between real and synthetic data stays the same but the overall amount of data is increased with higher variation to address the data issue. This will also variationally multiply our real data to a useful amount. With these methods we were able to create a dataset (ICTDS) (Table 1) with around 16000 samples where 156 available samples of real data are augmented to receive 329 samples, and 7300 synthetic samples are augmented to receive 15000 samples. The number of needed samples can be chosen, as well as split in training data, validation data and test data. Additionally, the annotation of the real data samples was improved.

Table 1 Overview of used datasets

Name	Res	No. of samples	No. of real samples included	No. of training samples	No. of evaluation samples	No. of test samples
reflCTDS	64^3	7405	156	6334	703	368
ICTDS	64^3	16190	318	12666	1408	2116

The following Figures show a comparison of real and synthetic samples of the two datasets. Figure 1 and Figure 2 are from the reflCTDS dataset. Figure 3 and Figure 4 belong to the augmented ICTDS dataset where the massive impact of elastic deformation can be seen. The deformation leads especially in the synthetic dataset to a much more realistic visualization of the pores and guarantee a higher variation in geometric shapes.

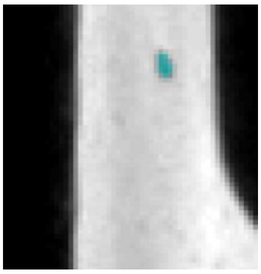


Figure 1: Real data symple reflCTDS

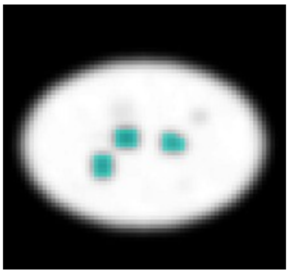


Figure 2: Syntehtic data sample reflCTDS

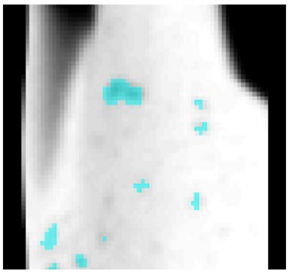


Figure 3: Real data reflCTDS with elastic deformation

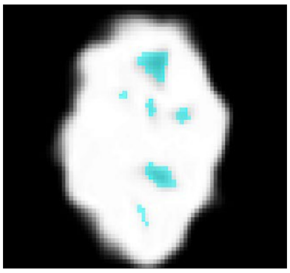


Figure 4: Synthetic data ICTDS with elastic deformation

3 Hardware and software setup

Training our models with different parameters and large 3D datasets requires specific hardware. In our case we are using a PC system with a Nvidia RTX Titan (24GB VRAM), 64 GB system memory and an Intel Core i9-10940X (14 cores). To train a model we used Tensorflow and needed to guarantee, that the model and data fits into the VRAM. To reserve the GPU memory for the model, we used the Tensorflow dataset API. This allows us to provide the path to the data and load it batchwise on the fly from HDD into VRAM when the previous batch is processed. For this purpose, an own separate nested load function has to be developed where the sample and groundtruh volume could be loaded together.

4 Neural network architecture

In our publication [5] different classical neural network architectures like U-Net [2] and V-Net [6] were compared with respect to object segmentation. Based on the earlier shown results, this project was started and developed on the basis of the best architecture U-Net-Gdata [5] (hereafter referred to as refU-Net). Thus, the models in this work are mainly based on the U-Net architectures. As described in [1] [2] [7], the encoder consists of several convolutional layers followed by down sampling, which can be performed in several different ways. This results in the input image being encoded into feature representations at several different levels. The deeper a model is, the more relevant features are extracted and the less significant ones are discarded. But if small details matter to finally distinguish between the classes this could lead to misclassification/-segmentation. The decoder semantically projects the low-resolution features learned by the encoder onto the higher-resolution pixel space to obtain dense pixel-wise segmentation. The decoder consists of up sampling and concatenation followed by regular convolution operations. The up sampling is also needed to obtain a segmentation with the resolution of the input image. The concatenation in the decoding path is important to marry classification with the localization obtained from the encoder at the corresponding level. The number of convolutions per depth level determine combined the filter kernel size the field of view for every voxel. Which means we are able to summarize larger features to a single value with more convolutions per layer and larger kernel sizes. To extend the results and progress shown, the classic U-Net architecture is used and significantly developed. The optimization is based purely on performance and detection probability. So, we investigate different parameters like encoding and decoding steps, number of convolutions per step, kernel size and filter amount, with and without batch normalization and dropout as well as activation functions [8] [9] and losses. The number of filters in the decoder is normally doubled every encoding step. A higher amount of feature maps could lead to a higher variation in separable features. The size of the filter kernels is adjusted depending on the spatial size of the features to learn. To train a model the model needs to be compiled where the optimizer algorithm, the loss function and evaluations metrics are set. These are some of the most important parameters, but the sheer number of variations of all these parameters is enormous.

Target ranges could be defined for most of the hyperparameters by performing some restrictive examinations. Thus, a theoretical parameter field can be spanned, in which optimal combinations of the parameters should be settled. With the help of the theoretical preliminary work, the enormous variety can be counteracted. The final solution is a combination of the theoretical parameter field and an hyperparameter optimization. The hyperparameter tuning is targeted against the evaluation metric which is the binary mean IoU. Additionally, to plain hyperparameter optimization we evaluated several hyperparameter to analyze which parameter has the highest influence in reaching a high evaluation score. For that we evaluated every single parameter. We also evaluated different loss functions based on a survey for image segmentation [10] [11]. The higher the score achieved with a specific hyperparameter variation, the higher the influence for our application and with our data. Also, the factor of score change depending on changing to another hyperparameter value within the evaluated set is taken into account. In addition, the correlation of the parameter change in dependence on the change of the detection probability must be correlated. In the following diagram (Figure 5), the most important parameters are shown. The farer outside the respective parameter is, the larger is the change in this parameter in terms of the achievable score.

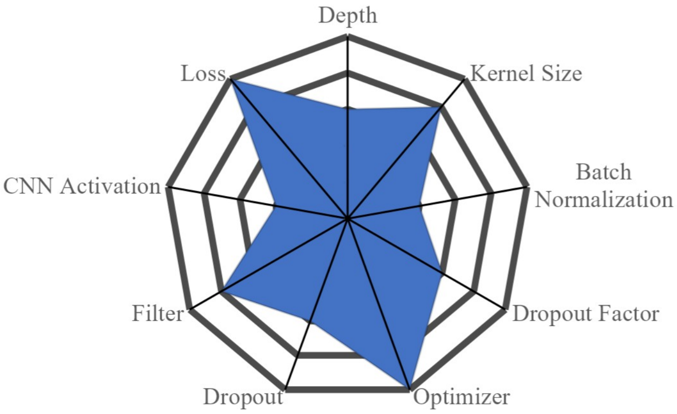


Figure 5: Impact of different hyperparameters on achievable IoU score

With this evaluation we were also able to determine specific values for every evaluated parameter which led us to the following values shown in Table 2.

Table 2: Determined hyperparameter set

Depth	**Filter base**	**Kernel size**	**Layer activation**	**Dropout**	**Dropout factor**	**BatchNorm**	**Optimizer**	**Loss function**
3	16	3 and 5	ReLU	False	0.1	True	Adam	Dice

In the following Table 3 we show a parameter comparison between the refU-Net and the two best models developed in this work. We moved the activation from the convolutional layer to a separate activation layer after normalization and created a parallel U-Net (PU-Net) model where we used the shown layer order per depth level in parallel with the kernel sizes 3 and 5 and concatenated the feature maps before down pooling. It has been shown [12] [13] [14] [15] that a parallelization of convolutions with different kernel sizes help to train features appearing in different sizes where as these mechanisms often used for natural language processing and training the feature maps for classification tasks. We made use of this technique to take into account that pores have volumetric sizes between 2^3 and 10^3 voxels. In the parallel paths we applied 2 convolutions to enlarge the receptive field of view for the resulting feature maps.

Table 3: Hyperparameters layer order of U-Net of previous work and new developed models

Model versions	**Down steps**	**Conv layers per depth lvl**	**Kernel size**	**Layer activation**	**Layer order per depth lvl**	**Loss function**	**Params**
refU-Net	3	2	3	ELU	Conv (incl. Act), Conv (incl. Act), MaxPool	BCE	1.4Mio
PU-Net	3	2	3+5	ReLU	Conv, BN, Act, Conv, BN, Act, Concat, MaxPool	Dice	6.3Mio
PCU-Net	3	2	3+5	ReLU	Conv, BN, Act, Concat, Conv, BN, Act, Concat, MaxPool	Dice	7.1Mio

The second model we developed, parallel cross U-Net (PCU-Net) (Figure 6), we concatenated the parallel generated feature maps after the first convolution for feature map mixture before leading into the second parallel convolutions. The parallelization is only used in the encoder path.

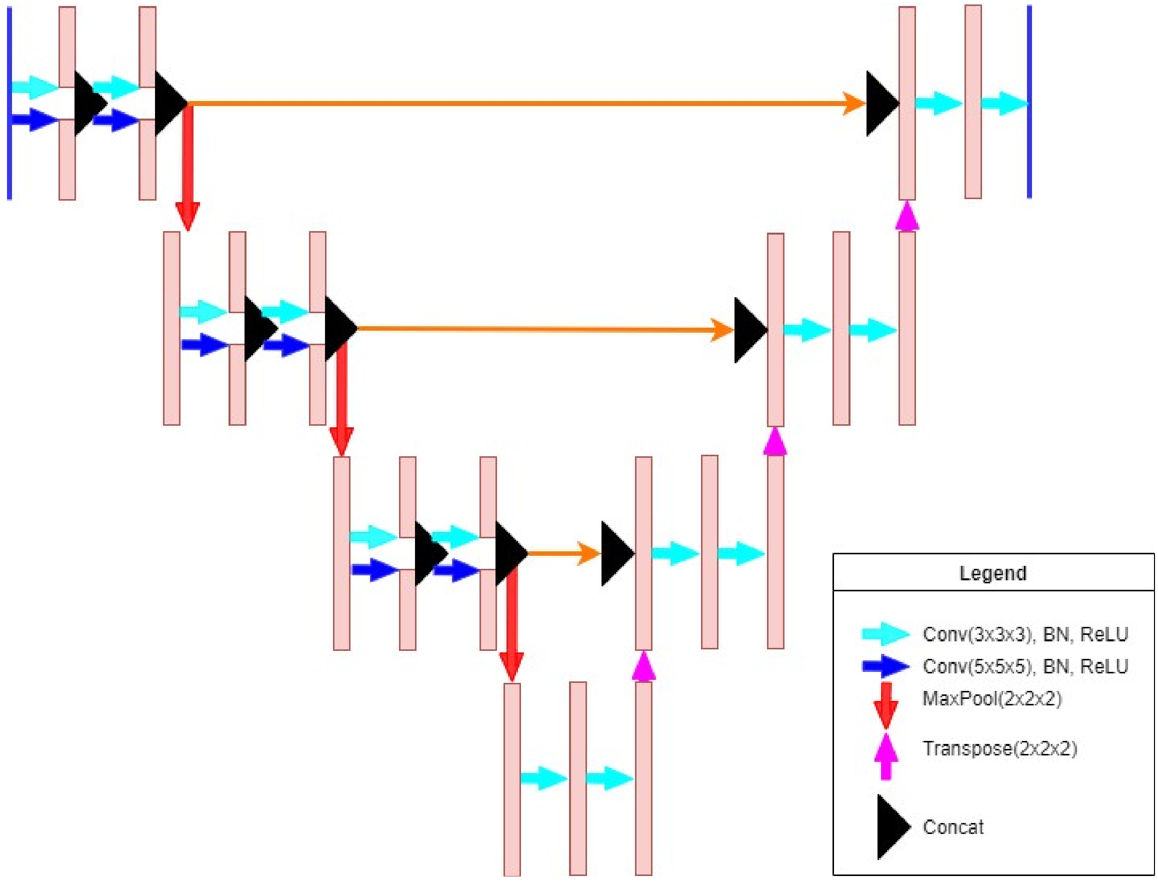


Figure 6: Schematic of our proposed PCU-Net

5 Training and evaluation of the neural networks

After defining the most impacting parameters, evaluating the best values for each parameter and building up the model architectures, the models were trained with different datasets. The datasets and their properties are described in chapter 2. During the training callbacks for early stopping, checkpointing, learning rate reduction and logging are used. At the beginning of the trainings, a learning rate of 0.001 was used.

Table 4 shows our results for the different models trained on different data sets (mixed and synthetic) in comparison to the -Net. First, the networks were trained with the reference dataset refICTDS to verify the changes and differences to the reference model. Both of our two new models PU-Net and PCU-Net achieved a higher BIoU score on the reference dataset. The PCU-Net achieved the highest value of 99.876% [BIoU]. In the confusion matrix (Figure 7) we can see that the false predicted background (419 voxel) and the false predicted pores (207) voxel is vanishig small compared to correct predicted background an pore voxels which shows an even more distictive good result. Consequently, an increase of 7.301% [BIoU] could be reached just by the new architectures and their parallelization of the convolutions. Thus, a first confirmation of the improvement of the new architecture has been achieved.

Table 4: Model evaluation on all types of samples of reference data set refICTDS. Tag in brackets describe which part of the dataset was used.

Model	**Training dataset**	**Evaluation dataset**	**BIoU [%]↑**
refU-Net	refICTDS (synth)	refICTDS (mixed)	92.575
PU-Net	refICTDS (mixed)	refICTDS (mixed)	99.838
PCU-Net	refICTDS (mixed)	refICTDS (mixed)	**99.876**

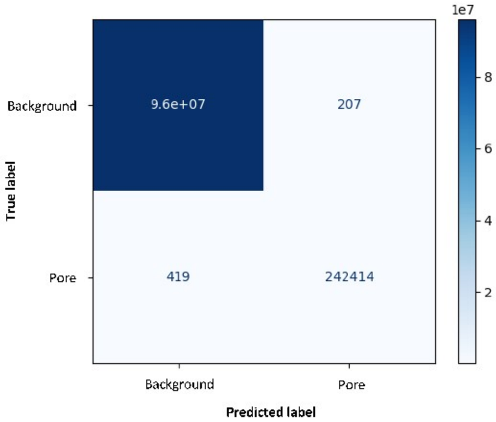


Figure 7: Confusion matrix of PCU-Net evaluation on reflCTDS (mixed)

In the second step we trained the models on our new improved dataset ICTDS. With our new paralell architectures we achieved reasonably higher IoU scores where as models trained on the new dataset have slightly lower scores. This behavior can be traced back to the more extensive and much more complex dataset (ICTDS) dataset. The results (Table 5) varify that even on the new dataset the architectures performe better that refU-Net with 93.017% for PU-Net and 93.567% for PCU-Net. In the confusion matrix (Figure 8) we can see again vanishing less false predictions for both classes.

Table 5: Model evaluation on all types of samples of dataset ICTDS. Tag in brackets describe which part of the dataset was used.

Model	**Training dataset**	**Evaluation dataset**	**BIoU [%]↑**
refU-Net	ICTDS (mixed)	ICTDS (mixed)	90.762
PU-Net	ICTDS (mixed)	ICTDS (mixed)	93.017
PCU-Net	ICTDS (mixed)	ICTDS (mixed)	**93.567**

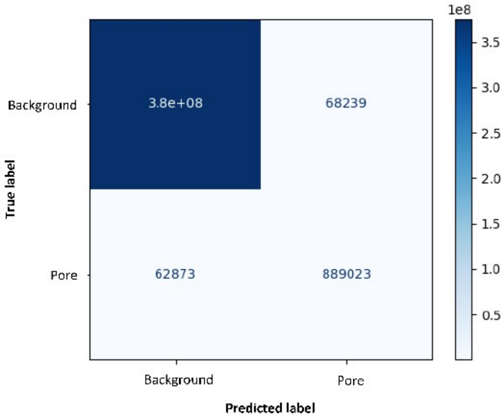


Figure 8: Confusion matrix of PCU-Net evaluation on ICTDS (mixed)

We also evaluated the models against reflCTDS dataset (Table 6) and against ICTDS dataset (Table 7) with only real samples . We can see that there is still a reality gap between synthetic and real data but our improved architectures, which can be deducted from Table 6 and data, which can be deducted from Table 7, achieved higher scores than we achieved in our prevous work [5].

Table 6: Model evaluation on real samples of dataset reflCTDS. Tag in brackets describe which part of the dataset was used.

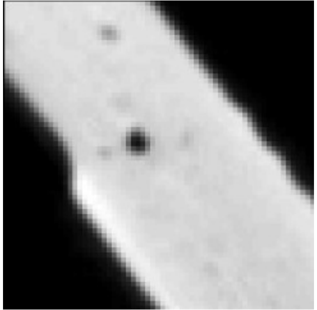
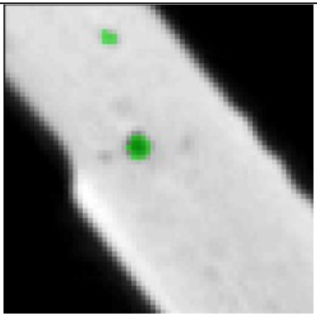
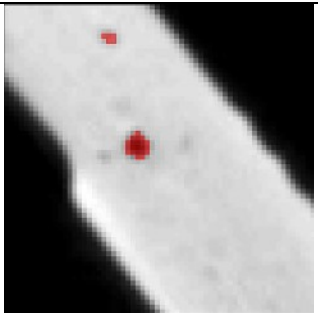
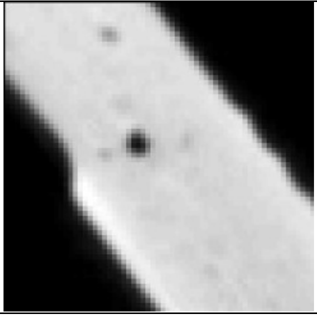
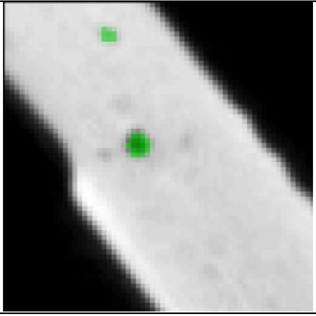
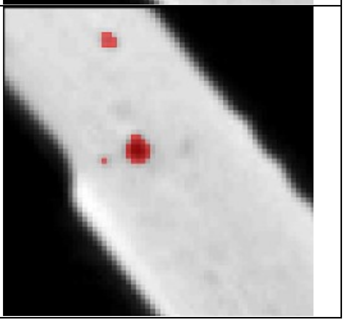
Model	**Training dataset**	**Evaluation dataset**	**BIoU [%]↑**
refU-Net	reflCTDS (mixed)	reflCTDS (real)	55.285
PU-Net	reflCTDS (mixed)	reflCTDS (real)	61.881
PCU-Net	reflCTDS (mixed)	reflCTDS (real)	**62.504**

Table 7: Model evaluation on real samples of dataset ICTDS. Tag in brackets describe which part of the dataset was used.

Model	**Training dataset**	**Evaluation dataset**	**BIoU [%]↑**
refU-Net	ICTDS (mixed)	ICTDS (real)	59.958
PU-Net	ICTDS (mixed)	ICTDS (real)	62.305
PCU-Net	ICTDS (mixed)	ICTDS (real)	**63.899**

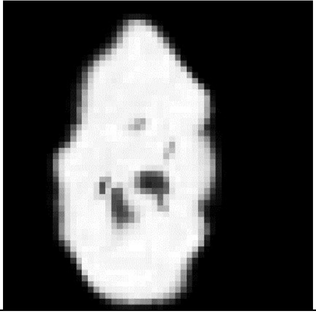
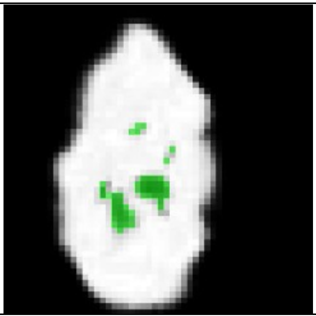
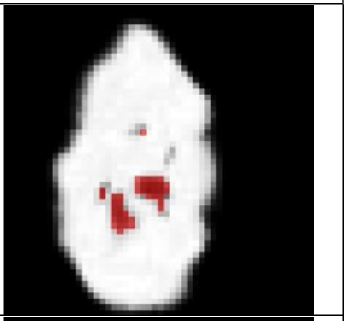
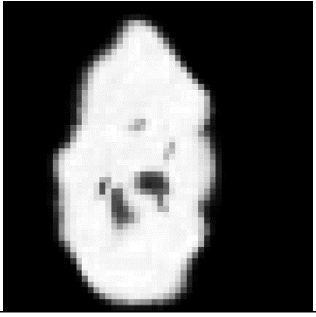
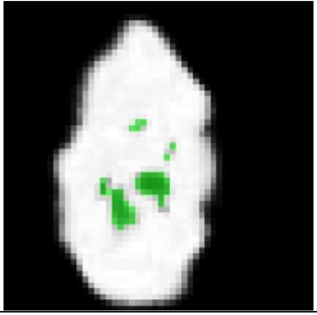
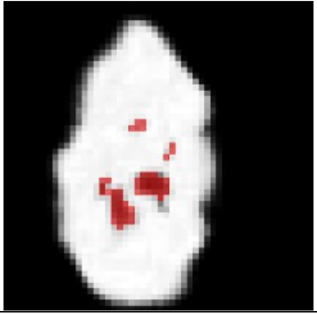
In order to make the pure numerical results more tangible, individual sections through samples are shown in the following overview (Table 8). This makes the results of refU-Net and PCU-Net visually comparable. The samples are taken from the evaluation set of ICTDS. In the comparison on the first sample which is real data we can see that our PCU-Net has a slightly more accurate segmentation prediction. As mentioned in the introduction, human accuracy in annotating data is not always consistent. This is because the decision whether a defect is present and which voxels belong to it are subjective. The PCU-Net model found an aditional pore right next to the large one which was not annotated but seems as valid segmentation. This is because the model learns from a large variety of data which is in case of real data an average of experts decission.

Table 8: Visual comparison of prediction results of refU-Net and PCU-Net taken from ICTDS evaluation dataset. Real data sample.

Model	**Sample**	**Groundtruth**	**Prediction**
refU-Net			
PCU-Net			

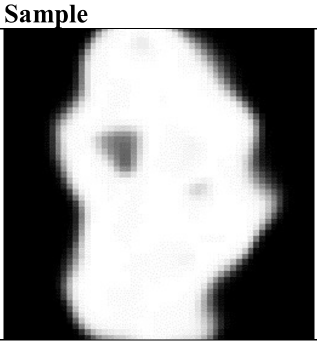
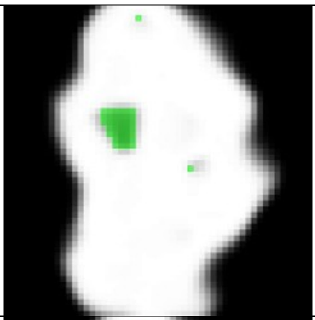
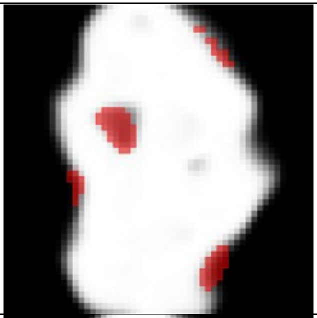
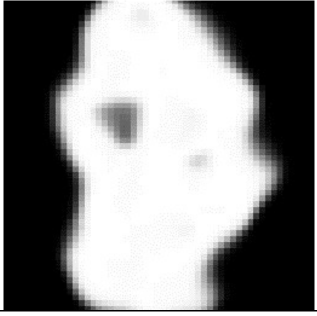
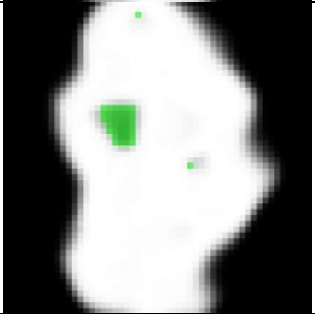
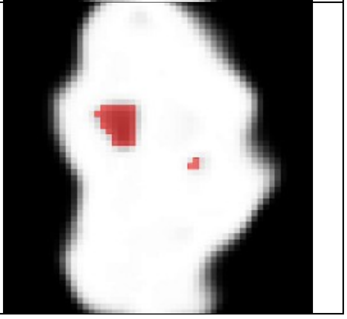
The second sample (Table 9) has highly deformed pores. What seems very unrealistic here is in reality often found in the overlaying of pores in pore nests or due to the deformation of pores. Additionally the visualisation of small structures could be influenced by noise in CT data, artifacts during reconstruction or cannot be resolved cleanly. Also here both models were able to segment most of the pores correctly but again slightly better results achieved by PCU-Net.

Table 9: Visual comparison of prediction results of refU-Net and PCU-Net taken from ICTDS evaluation dataset. Synthetic data sample.

Model	Sample	Groundtruth	Prediction
refU-Net			
PCU-Net			

In the last sample (Table 10) we can see that refU-Net has massive false predictions at the surface of the object which does not occur with our new model. Both models achieved reasonable results in the evaluation of over 90% but relying on just a metric could be dangerous because outliers in large evaluation sets vanisch in the metric. If the model refU-Net would be used in an application for example in a production processes it would be important to manualy inspect the quality. If such massive segmentation faults are automaticaly evaluated this could lead to waste of supposedly defective components in production. In some cases a basic algorithmic plausibility check could avoid wasting parts. Nevertheless, AI-based pre-processing of data can significantly reduce the effort required for manual quality assurance.

Table 10: Visual comparison of prediction results of refU-Net and PCU-Net taken from ICTDS evaluation dataset. Synthetic data sample with artifacts predicted from refU-Net.

Model	Sample	Groundtruth	Prediction
refU-Net			
PCU-Net			

6 Conclusion

In this work, we proposed two neural network models PU-Net and PCU-Net that are able to predict pore segmentation in industrial CT data. To determine the optimization possibilities, we evaluated the most important parameters and defined the most promising parameter values. These results allowed us to further develop a reference model to derive two new architectures. To verify our new models, we trained them on a reference dataset, which showed that the improvements were valid (BIoU of 99.876 %). In the next step, we created a new dataset ICTDS to further improve the prediction quality of PU-Net (BIoU of 93.017 % mixed data and BIoU of 62.305 % real data) and PCU-Net (BIoU of 93.567 % mixed data and BIoU of 63.899 % real data). Unfortunately, there is no publicly available industrial CT dataset with this type of defects to evaluate our results against other proposals, but with our stepwise evaluation and improvement strategy, we have shown that one way to successfully improve AI networks on specific data is to first improve the model by training with an existing dataset to verify the model improvements, and then use a new dataset. This brings us one step closer to AI-based defect segmentation of small defects in industrial CT data. With an achievable defect segmentation accuracy of ~63%, our model already provides good pre-segmentation for CT application engineers evaluating production data and saving time for quality assurance. The accuracy can be further enhanced with more realistic synthetic and additional real data. In technical tasks which should solved with AI unbalanced data is often the case. Using a methodology like proposed here could be beneficial on solving these problems. Furthermore, our method could be applied on similar industrial segmentation tasks or even for 2D analysis. Our method could be applied on similar industrial segmentation tasks or even for 2D analysis. In summary this paper shows that our AI solution achieved promising results solving automated non-destructive quality inspection needs.

7 References

[1] O. Ronneberger, P. Fischer und T. Brox, „U-Net: Convolutional Networks for Biomedical Image Segmentation," 18 May 2015.

[2] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox und O. Ronneberger, *3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation,* arXiv, 2016.

[3] P. Fuchs, T. Kröger und 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),* November 2019.

[4] P. Fuchs, T. Kröger, T. Dierig und C. S. Garbe, „Defect Detection in CT Scans of Cast Aluminum Parts: A Machine Vision Perspective," *9th Conference on Industrial Computed Tomography (iCT) 2019, 13-15 Feb, Padova, Italy (iCT 2019),* March 2019.

[5] T. Schanz, R. Tenscher-Philipp und M. Simon, „AI-Powered Analysis of Industrial CT Data," 2020.

[6] F. Milletari, N. Navab und S.-A. Ahmadi, „V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," 15 June 2016.

[7] N. Siddique, P. Sidike, C. Elkin und V. Devabhaktuni, „U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications," *IEEE Access,* Bd. PP, pp. 1-1, June 2021.

[8] D. P. Kingma und J. Ba, „Adam: A Method for Stochastic Optimization," 22 December 2014.

[9] C. Nwankpa, W. Ijomah, A. Gachagan und S. Marshall, „Activation Functions: Comparison of trends in Practice and Research for Deep Learning," 8 November 2018.

[10] S. Jadon, „A survey of loss functions for semantic segmentation," *2020 IEEE International Conference on Computational Intelligence in Bioinformatics and Computational Biology,* June 2020.

[11] S. S. M. Salehi, D. Erdogmus und A. Gholipour, „Tversky loss function for image segmentation using 3D fully convolutional deep networks," 18 June 2017.

[12] X. Chen, B. Xu und H. Lu, „Effects of Parallel Structure and Serial Structure on Convolutional Neural Networks," in *Journal of Physics Conference Series*, 2021.

[13] H. W. Fentaw und T.-H. Kim, „Design and Investigation of Capsule Networks for Sentence Classification," *Applied Sciences,* Bd. 9, p. 2200, May 2019.

[14] L. Jia, H. Zhai, X. Yuan, Y. Jiang und J. Ding, „A Parallel Convolution and Decision Fusion-Based Flower Classification Method," *Mathematics,* Bd. 10, p. 2767, August 2022.

[15] A. Krizhevsky, I. Sutskever und G. E. Hinton, „ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, 2012.

[16] S. R. Dubey, S. Chakraborty, S. K. Roy, S. Mukherjee, S. K. Singh und B. B. Chaudhuri, „diffGrad: An Optimization Method for Convolutional Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems, 2019,* 12 September 2019.