

Automatic Classification of XCT Images in Manufacturing

Bertram Sabrowsky-Hirsch, Roxana-Maria Holom, Christian Gusenbauer, Michael Reiter, Florian Reiterer, Ricardo Fernández Gutiérrez, Josef Scharinger

▶ **To cite this version:**

Bertram Sabrowsky-Hirsch, Roxana-Maria Holom, Christian Gusenbauer, Michael Reiter, Florian Reiterer, et al.. Automatic Classification of XCT Images in Manufacturing. 17th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI), Jun 2021, Hersonissos, Crete, Greece. pp.220-231, 10.1007/978-3-030-79150-6_18 . hal-03287706

HAL Id: hal-03287706
https://inria.hal.science/hal-03287706

Submitted on 15 Jul 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Automatic Classification of XCT Images in Manufacturing*

Bertram Sabrowsky-Hirsch[1][0000−0001−7081−5182], Roxana-Maria Holom[1][0000−0002−1321−0475], Christian Gusenbauer[2], Michael Reiter[2][0000−0003−0368−9758], Florian Reiterer[2][0000−0001−9672−4107], Ricardo Fernández Gutiérrez[2][0000−0002−6385−712X], and Josef Scharinger[3][0000−0001−6502−7501]

[1] RISC Software GmbH, Softwarepark 35, 4232 Hagenberg, Austria
office@risc-software.at
https://risc-software.at/
[2] Nemak Linz GmbH, Zeppelinstraße 24, 4030 Linz, Austria
https://www.nemak.com/
[3] Johannes Kepler University, Altenbergerstraße 69, 4040 Linz, Austria
https://www.jku.at/

Abstract. X-ray computed tomography (XCT) is an established non-destructive testing (NDT) method that, in combination with automatic evaluation routines, can be successfully used to establish a reliable 100% inline inspection system for defect detection of cast parts. While these systems are robust in automatically localizing suspected defects, human know-how in a secondary assessment and decision-making step remains indispensable to avoid an excess of rejected parts. Rather than changing the existing defect detection system and risking difficult to anticipate changes to a solid evaluation process, we propose the integration of human know-how in a subsequent support system through end-to-end learning. Using XCT data and the corresponding decisions performed by the XCT operator, we aim to support and possibly automate the secondary quality assessment process. In our paper we present a Convolutional Neural Network (CNN) architecture to predict both, the final decision of the XCT operator and a defect class indication, for cast parts rejected by the defect detection system based on XCT slice images. On a dataset of 19,459 defect records categorized in 7 classes, we achieved an accuracy of 92% for the decision and 93% for the defect class indication on the testing split. We further show that, by binding decisions to the reliability of the predicted defect class, our model has the potential to enhance also a production process with a near-faultless condition. Based on production-line data, we estimate that our model can reliably relabel 11% of defects reported during production and provide a defect class indication for another 57%.

Keywords: Machine Learning · Deep Learning · X-ray Computed Tomography · Quality Control · Casting · Manufacturing.

* Supported by EU H2020 and "Innovatives OÖ 2020" programs.

1 Introduction

Manufacturing, similar to other areas, is going through a transformation period. This is mainly because of the digitalization activities that manufacturing companies are experiencing. Therefore, terms like smart industry and zero defect manufacturing are very popular nowadays because of the cutting edge technologies involved, such as Internet of Things (IoT), big data, data analytics and machine learning (ML).

In the context of the European project called Big Data Value Spaces for COmpetitiveness of European COnnected Smart FacTories 4.0 (BOOST 4.0) [2] Nemak Linz GmbH and RISC Software GmbH, together with other partners, tried to achieve a better understanding of the data recorded along the production process and implement different data analytics applications to optimize the light metal casting processes. Nemak Linz GmbH focusses on the production of high-quality cylinder heads for engines by using the Nemak-owned Rotacast® [15] technology. One of the use cases identified in the project was empowering the quality assessment of cast parts through data analytics. For quality control, Nemak Linz GmbH provides extensive equipment for the assessment of the cast parts, including automatic X-ray computed tomography (XCT) inspection systems, as well as equipment for conventional X-ray imaging, leakage testing, etc.

In this paper, we are making use of ML to automatically reassess cast parts based on XCT images, which have been previously identified as faulty by an XCT inspection system. Hence, the expected impact is an optimization of the casting process at Nemak Linz GmbH and, consequentially, making more human resources available, since the XCT slice images are typically assessed by XCT and casting experts. However, this is not a straightforward solution. The XCT data is counter-balanced as some defects only occur at specific casting situations and at substantially different rates. In addition, XCT data is only collected for faulty parts, whereas the vast majority of defect-free parts is not recorded. Therefore, the imbalanced XCT data represents a challenge for the success of the ML algorithm [4]. In combination with strict requirements for quality assurance, it is important to identify conditions that allow for optimization without the possibility of disregarding actual defects.

The paper is structured as follows: section 2 summarizes the background information and the related work, section 3 details the challenges and enumerates the objectives, section 4 describes the core aspects of the proposed solution and section 5 concludes the paper by presenting the outcomes of the research and some observations related to future work.

2 Background

2.1 Quality Assessment using X-ray Computed Tomography (XCT)

Serial inline XCT inspection is used to optimise cast parts by automatically identifying defects that determine their quality. In the context of the use case presented in this paper, the ZEISS VoluMax 1500 G2 XCT system is used to

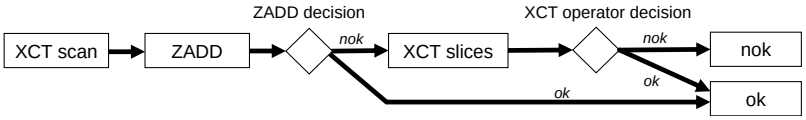


Fig. 1. The decision process used for XCT inspection outlined: Scanned production parts are analysed by the ZADD, which rates each part either as *ok* or *nok*. For every *nok* part, the system generates XCT slice images for the XCT operator, who will yield a final manual decision.

generate *ok* and *nok* (not OK) part decisions in combination with the ZEISS Automated Defect Detection (ZADD) software [17]. The ZADD software automatically detects, localizes and classifies manufacturing defects based on the original CAD data and on a reference model to detect deviations in the actual part to be analyzed. This reference model is the combined result of XCT data acquired from several exceptionally flawless parts [15]. For each inspected cast part, deviations are localized and based on their properties predefined defect types assigned. Finally, the system makes a binary decision - *ok* or *nok* - by comparing the determined properties with the specified limit values. In case of a *nok* decision, a manual decision is requested from the XCT operator, who is presented with the defect information including orthogonal slices of the XCT images at the defect locations. If the XCT operator declares the defect as tolerable, the part is relabelled as *ok* and is returned to the next manufacturing step. Fig. 1 summarizes the previously described XCT inspection and quality control steps. The goal of the work presented in this paper is to extend the current system with a machine learning algorithm to support and possibly automate the manual decision process using end-to-end learning.

2.2 Related Work

The scenario described in section 2.1 is an instance of Non-destructive Inspection (NDI), where XCT imaging has been established as a tool for the image-based defect detection of casting defects [19]. While existing solutions still often rely on traditional computer vision methods, deep learning and especially Convolutional Neural Networks (CNNs) have been used for defect detection throughout many industrial fields [3, 5, 10]. In combination with methods like transfer learning [6] and data augmentation [14] to overcome training data limitations, CNNs typically outperform traditional methods, with the benefit of end-to-end learning. However, most works in the context of defect detection rather revolve around the extension of CNNs to segmentation or detection architectures, such as U-net [1] and Mask R-CNN [6]. In our scenario, the detection of defects is already handled by the upstream ZADD software that we do not seek to replace or change, but rather implement a secondary assessment based on its output. For this reason traditional classification (in disregard of detection and/or segmentation) much closer reflects the reality of our use case, with few comparable works in the field of XCT defect detection. However, Masci et al. [12] showed that CNNs can be

effective in a similar setup, reporting a 7% error rate on their data. In another work by Park et al. [13], the authors show that even strong class imbalances can be overcome with the use of sampling strategies and data augmentation. One aspect that we did not find discussed in related literature, is the evaluation of a proposed model towards its practical use for industrial quality assurance, which usually mandates a near-faultless requirement. We account for this by identifying reliably predicted defect classes and by assessing the rate at which the model can automate decisions as well as provide indications without the risk of disregarding actual defects.

3 Motivation

The quality assessment process of cast parts involves many assessment steps. As an example, 100% inline XCT inspection relies on visual inspection and manual decisions and re-evaluation of *nok* parts detected by XCT and casting experts that may consume a lot of working hours. Therefore, the quality assessment process is expensive (time- and resource-intensive), i.e. can contribute very significantly to the overall costs of an cast part. Hence, there is a significant potential for cost savings in this area by automatizing those tasks at least partially.

3.1 Challenges

XCT inspection in smart manufacturing is dealing with the challenges imposed by the complexity of cast parts. Gradual changes to the cast parts induced by tolerable casting variations during the lifetime of a product can be handled by periodically examining and updating the reference model. Although ZADD is reliable and efficient in creating binary decisions, its scope of identifying different defect classes is limited as its decisions are rule-based and rely on configured examinations for defined regions of interest and local limits. Therefore, the XCT operator has to inspect all *nok* parts to review the ZADD decision and determine a specific defect class (e.g.: core fracture, core defect, metal shavings, etc.). Aiming at a fully automated quality assessment process, a strategy for extending ZADD with a ML-based classification is essential.

The success of ML is massively influenced by the data quantity and quality. Automatically predicting the *ok*/*nok* decision and defect classes assessment of the XCT experts is challenging. As in many other real industry cases, faulty cast parts are outweighed by samples of good performance. This effect is further amplified, as the data for our use case can inherently only be collected for reported defects, which creates another division (definite *nok* and relabelled *ok*) for an already under-represented class. Further discrimination of different defect classes, some of which have an especially rare rate of occurrence, creates a major hindrance for the acquisition of sufficient samples, which leads to greatly imbalanced data. In this paper, we approach the data imbalances with a combination of resampling and data augmentation methods. While these methods can be very

effective, they are only able to compensate for imbalances up to a certain degree. This is why we focus on classes reasonably well represented in the data and reserve the omitted classes for future extensions subsequent to further focussed data acquisition towards these under-represented classes. One other distinctive challenge is the strong bias towards avoiding False Positives (FP, i.e., parts incorrectly relabelled as *ok*) over False Negatives (FN, i.e., parts incorrectly confirmed as *nok*) as the former case undermines quality control, while the latter "merely" translates to added costs. We adapted our model architecture in order to reflect this bias in the models decisions.

3.2 Objectives

In order to achieve the goal of supporting and minimizing the manual decision process by end-to-end learning, the following objectives have been defined:

(i) Automatic classification of XCT slice images via CNNs.
(ii) Support the domain experts in identifying the quality of the classification for each defect image and understanding the decisions of the algorithm.
(iii) The machine learning algorithm should be optimized to minimize the FP.

4 Solution

4.1 Data

Our dataset consists of real XCT and reference data exported by the ZADD software in use at the Nemak production site. The data has been collected over a time of eight months and consists of a series of ZADD detection records for each production part. Each record is comprised of three slices that correspond to the principal axes and intersect at the center of the identified defect region. Three channels are provided for each slice: The channels *mean* and *std* correspond to the mean and standard deviation values of the reference part, and the channel *xct* to the real production part. All channels have been registered to CAD data by the ZADD software and thus extend over the same region. Fig. 2 visualizes the *xct* channels of the three slices of a detection record in 3D space and compares the three different channels of a single slice. In addition to the slices, each record is associated with two annotation labels:

defect: the defect class assigned by the XCT operator
decision: the final assessment of the XCT operator, either *ok* or *nok*

Fig. 3 outlines the number of detection records for each of the 13 defect classes and the respective share of parts finally labelled as *ok* and *nok*. There is clearly a strong imbalance in the distribution of defect classes corresponding partially to the frequency of occurrence of the respective type of defect in the production process. For the total of 20,378 records reported as *nok* 42% have been relabelled to *ok* by the XCT operator.

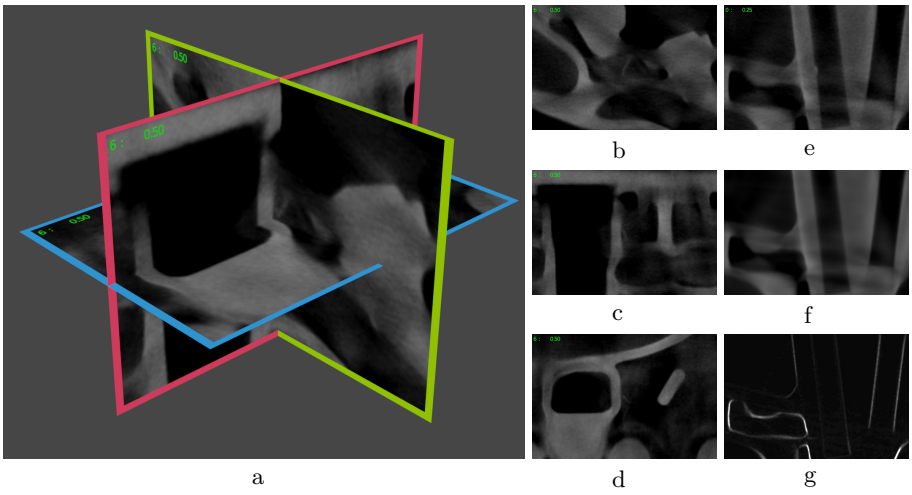


Fig. 2. Left: The channel *xct* for the slices of one detection record aligned in 3D space (a). Middle: for (a) the respective slices in 2D (b, c, d). Images of the channel *xct* do contain minimal annotations from the ZADD in form of text in the top left corner, which will be omitted through cropping during training and prediction. Right: the channels *xct* (e), *mean* (f) and *std* (g) of a single slice of another detection record. We applied strong contrast adjustments to the last channel *std* in order to make the contents visible and printable.

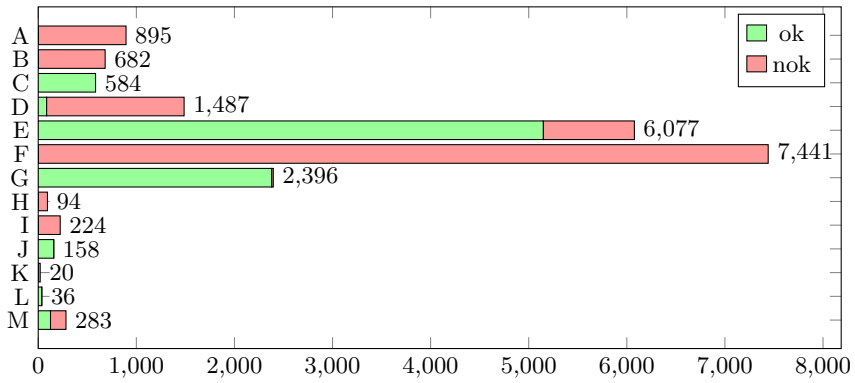


Fig. 3. Overview of the number of records for each of the 13 defect classes in our dataset, corresponding to the label *defect*. Each defect class is further split into records manually labelled as *ok* and *nok* by the XCT operator, corresponding to the label *decision*. Note, that we will later select a subset of 7 defect classes due to the strong imbalance in the dataset.

4.2 Model Architecture

Based on the dataset of annotated detection records, we train a CNN to automatically classify both annotation labels, *defect* and *decision*. By combining these

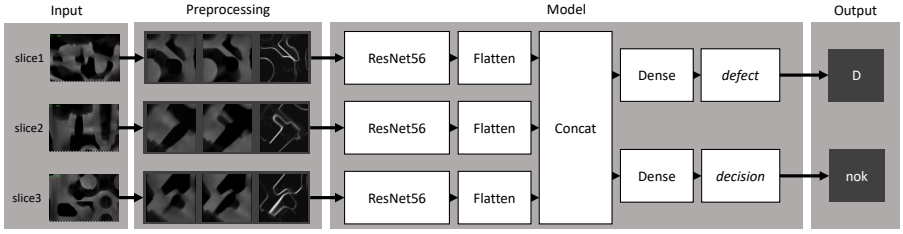


Fig. 4. The model architecture with exemplary input slice images and output values.

labels in one model, we hope to exploit label dependencies through multi-label learning as suggested by [2]. We empirically evaluated different configurations of the models *VGG*[18], *EfficientNet*[20] and *ResNet*[8], and found a combination of multiple ResNets to yield the best performance for our dataset. Fig. 4 illustrates the final model architecture. For a given defect, we process the corresponding three slices in separate input paths that are combined within the model to connect to the two output labels. Following the input path of one slice, first the related intensity images for its three channels *xct*, *mean* and *std* are processed separately using z-score normalization (based on the channel's deviation over the entire dataset). The preprocessed channels are then merged to a single vector image of size 1150x790(x3), which is rotated and flipped randomly for augmentation before extracting a central region of 448x448(x3). As the image is centered at the defect and spans a large extent, the extracted image still retains most of the relevant information. Finally, the resulting image is downscaled to an input size of 224x224(x3) using a local mean filter. This size is commonly used and standard for models pretrained on the well-known dataset *ImageNet* [11]. A standard ResNet56 is employed, excluding its top layer and instead connecting to a concatenation layer that aggregates all three input paths. Finally, each label is implemented through a dense layer using a softmax activation function that connects to the concatenation via an intermediate dense layer. An exemplary output of D/nok would indicate defect class D and a final decision: *nok*.

4.3 Training and Inference

We created stratified splits of the dataset for training, validation and testing in a 7:1:2 ratio. For stratification we used both labels, *defect* and *decision*. Groups of less than 500 records were discarded for our experiment to avoid excessive oversampling. This excluded the defect classes H-M as well as the combinations D/ok and G/nok, leaving a total of 19,459 records for 7 final defect classes (A-G). We used an Adam optimizer [6] with categorical cross-entropy loss to train the model in epochs of 2,000 batches of 10 records each. To counter the unbalanced nature of the dataset we used a combination of oversampling and augmentation during training. Samples were again grouped by both labels with minority groups being oversampled. Random rotations and flipping were used for augmentation as described in section 4.2.

Table 1. Evaluation for the *decision* label grouped by the 7 selected defect classes using the optimized threshold and class rules.

Actual-Defect	A	B	C	D	E	F	G	Total
Accuracy	100%	99%	99%	96%	79%	99%	95%	92%
FPs	0%	1.5%	0%	4.3%	2.9%	0.8%	0%	1.6%

Training converged within 100 epochs, an observation affirmed by the lack of further improvements of the validation score up to the termination of training at epoch 250. The accuracy scores of the model on the training, validation and testing splits were 97%, 93% and 94% respectively for the *decision* label and 98%, 92% and 93% respectively for the *defect* label. This indicates minor signs of overfitting to the training data, especially for the *defect* label and less so for the *decision* label, which could be mitigated by further augmentation methods or more preferably the extension of the dataset.

To bias the model towards the avoidance of FPs, we empirically evaluated the use of weighted outputs during training with no notable improvements. However, adjusting the probability threshold of the label *decision*, a technique described by Li et al. [9], yielded the desired effect. Based the model's softmax predictions on the training data, we optimized the threshold to balance a minimal ratio of FPs and high accuracy, changing the threshold from the default value of 0.5 to 0.79 (i.e., predictions with a softmax value below 0.79 for *ok* count as *nok*). We were able to lower the ratio of FPs for the *decision* label from 2.9% to 1.8% for a decreased accuracy of 92% on the testing split (or from 0.8% to 0.4% at 97% accuracy on the training split), which we deemed an acceptable compromise.

We achieved a final improvement with the use of class rules, which overrule the decision based on the predicted *decision* label, specifically for the defect classes without any *ok* representation in the dataset (e.g., A, B and F) to *nok*. This also reflects the production-line reality as these defects are without exception labelled *nok*. Introducing these class rules further decreased the rate of FPs to 1.6% (or 0.2% on the training split) for a insignificant 0.18% increase in FNs.

Table 1 outlines the final results on the testing split for each defect class. We evaluate the metrics Accuracy and FPs for the *decision* label using the optimized probability threshold and class rules over the testing split specified for each defect class. We further evaluated the confusion between defect classes as outlined in Table 2. The model struggles significantly with the class D, which can be difficult to distinguish even for experts as it is visually very similar to E and F. For class C on the other hand the model performs quite robust classifications. Similarly, the two related classes A and B, despite mutual confusion, can be predicted with high confidence.

4.4 Production-Line Evaluation

For the evaluation on actual production-line data we first continued to train the model for another 200 epochs on the detection records of all three splits to

Table 2. The confusion between defect classes for the testing split. The values are normalized by rows with the respective absolute number of Records for the actual defect class in the rightmost column.

		Predicted							
		A	B	C	D	E	F	G	Records
Actual	A	0.89	0.06	-	0.02	0.01	0.02	-	179
	B	0.05	0.90	-	0.01	0.04	-	-	136
	C	-	-	0.97	-	0.02	0.01	0.01	117
	D	-	-	-	0.77	0.11	0.09	0.03	280
	E	-	-	-	0.02	0.93	0.01	0.03	1216
	F	-	-	-	0.02	0.01	0.96	-	1488
	G	-	-	-	0.01	0.06	-	0.93	476

incorporate all available data into the model. The final accuracy values on the full dataset were 96% for the *decision* and 98% for the *defect* label. The evaluation was then conducted on an independent dataset of 1,299 detection records, of which 37% have been labelled *ok* by the XCT operator. The records have been collected over eight weeks from the production-line. The accuracy values for the production-line dataset were 92% for the *decision* and 87% for the *defect* label.

Table 3 illustrates evaluation metrics for the *part-ok* label over the production-line dataset for each *predicted* defect class. Grouping by the predicted instead of the true defect class allows us to assess the reliability of the model for each distinct defect class during real-time operation, when the true defect class is not known without further manual inspection. Other than standard metrics also used for the training evaluation, we evaluated the ratio of *Critical False Positives* (CFP). For the CFP metric, a domain expert reassessed all FPs manually, excluding those that are ambiguous, i.e., close to the decision limit. This makes the combination of FPs and CFPs the most important metrics to assess the potential use of the model in the production-line.

For the defect classes A, B, D and F the model does default to a *nok* decision, which can be explained by the omission of *ok* instances for these classes during training. Defects associated with these defect classes will have to be checked manually, with the benefit of a defect class indication by the model.

Class C only relates to *ok* instances, which is reliably predicted by the model. As the class does not show any FPs, defects can be automatically relabelled without further inspection. The case of Class D is curious as it does not yield any FPs in the production-line evaluation, a stark contrast to the 4.3% FPs in the previous evaluation. This can be explained by the FP cases being confused with the classes E-G and thus no longer contribute to the predicted class D.

Class E, which is arguably the most difficult class to correctly assess, shows a high 24.3% of FNs, but a relatively low 3.2% of FPs resp. 1.6% of CFPs. The class could become viable in the future with more training data to improve the model. Class G was problematic due to the high number of FPs (which exclusively relate to confusions with other defect classes) rendering the class unsuitable for automatic decision making.

Table 3. The production-line evaluation for the *decision* label grouped by the predicted defect classes.

Predicted-Defect	A	B	C	D	E	F	G	Total
Accuracy	98.6%	97.3%	100.0%	93.7%	72.5%	99.0%	85.6%	92.0%
FPs	**0%**	**0%**	**0%**	**0%**	3.2%	**0%**	14.4%	2.2%
CFPs	**0%**	**0%**	**0%**	**0%**	1.6%	**0%**	13.7%	1.8%
Overall Rate	6%	6%	11%	10%	17%	41%	9%	100%
Rate of TPs	0%	0%	**11%**	0%	11%	0%	9%	31%
Rate of TNs	**5%**	**5%**	0%	**8%**	3%	**39%**	0%	61%

To assess the potential for optimization in the production-line process, the table also outlines the *Overall Rate* for each defect class, i.e., the rate at which a specific defect occurs during production based on its proportion in the production-line dataset. *Rate of TPs* (i.e., *True Positives*) relates to the ratio of defects that the model correctly classified as *ok*. Although a total of 31% of defects were correctly relabelled, only class C can be deemed reliable, leaving effectively 11% of defects that can be automatically relabelled and thus require no further manual inspection. *Rate of TNs* (i.e., *True Negatives*) relates to the ratio of defects the model correctly confirmed as *nok*. In total 61% of the defects were correctly confirmed, counting the classes deemed reliable the model could give a defect indication to guide manual inspection for 57% of all defects.

5 Conclusion and Future Work

With regards to the objectives formulated in section 3.2, we have shown that it is possible to perform automatic classification of XCT slice images using our CNN model architecture. Using the XCT slice images provided by the defect detection system the model is able to predict a defect class with an accuracy of 93% for our dataset of seven selected defect classes. This assessment could traditionally only be determined through manual inspection at a considerable cost of time. Other than predicting the defect class, the model is also able to reproduce the expert decision and predict whether to keep or discard a production part with an accuracy of 92%. On actual data from the production-line, featuring a different distribution of defect classes, the model still yields an accuracy of 87% for the defect class. The model may not be reliable enough to fully replace manual inspections for all defects, it can however be used to decide for certain reliably predicted defect classes and to obtain defect class indications to guide the XCT operator. The model performs the expert decision reliably for 5 defect classes, which account for 74% of reported defects based on the production-line dataset. We further assessed the rate at which the model can be utilized given those five selected classes. We estimate that the model is able to correctly relabel 11% of reported defects to *ok* and thus significantly relieve the XCT operator. For another 57% of reported defects, the model confirms the *nok* decision and provide a defect class indication to guide the manual inspection.

The greatest challenge lay in the large imbalances between the defect classes in our training dataset, which is due to the rare occurrence of many of these types of defects during production. In combination with the strict quality requirements that do not tolerate any oversight of actual defects, we were still able to show that our model has the potential to optimize the inspection process. For near-term application, we suggest the use of the model as a support tool for the XCT operator to guide the manual decision and produce defect class indications. As more detection records are collected, especially for the under-represented defect classes, the model can be further trained to improve its reliability and extended to predict all of the original 13 defect classes. Fully automatic relabelling in lieu of manual inspections should be delayed to the point, when all defect classes have been integrated into the model. In order to reach a reasonable threshold for the integration of under-represented classes sooner, Generative Adversarial Networks (GANs)[7] might be an important concept to explore on our model. In the meantime, the addition of model-agnostic explanations [16] may further extend the supportive aspect of the model by providing context to the classification, i.e., outlining the supposed defect region in the XCT slice images.

Acknowledgements

This work was supported by the BOOST 4.0 project, which has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 780732 and by the strategic economic and research programme "Innovatives OÖ 2020" of the province of Upper Austria. RISC Software GmbH is a Member of UAR (Upper Austrian Research) Innovation Network.

References

1. Bellens, S., Vandewalle, P., Dewulf, W.: Deep learning based porosity segmentation in x-ray ct measurements of polymer additive manufacturing parts. Procedia CIRP **96**, 336–341 (2021). https://doi.org/10.1016/j.procir.2021.01.157
2. BOOST 4.0 Project Partners: Boost 4.0 - big data for factories, https://boost40.eu/
3. Du, W., Shen, H., Fu, J., Zhang, G., He, Q.: Approaches for improvement of the X-ray image defect detection of automobile casting aluminum parts based on deep learning. NDT & E International **107** (2019), https://doi.org/10.1016/j.ndteint.2019.102144
4. Fathy, Y., Jaber, M., Brintrup, A.: Learning with imbalanced data in smart manufacturing: A comparative analysis. IEEE Access **9**, 2734–2757 (2021). https://doi.org/10.1109/ACCESS.2020.3047838
5. Ferguson, M., Ak, R., Lee, Y.T., Law, K.H.: Automatic localization of casting defects with convolutional neural networks. In: 2017 IEEE International Conference on Big Data, Boston, MA, USA, December 11-14. pp. 1726–1735. https://doi.org/10.1109/BigData.2017.8258115
6. Ferguson, M.K., Ronay, A., Lee, Y.T.T., Law, K.H.: Detection and segmentation of manufacturing defects with convolutional neural networks and

transfer learning. Smart and sustainable manufacturing systems **2** (2018). https://doi.org/10.1520/ssms20180033
7. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. pp. 2672–2680. NIPS'14, MIT Press (2014)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
9. Li, X., Liu, Z., Luo, P., Loy, C.C., Tang, X.: Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 6459–6468 (2017). https://doi.org/10.1109/CVPR.2017.684
10. Lin, J., Yao, Y., Ma, L., Wang, Y.: Detection of a casting defect tracked by deep convolution neural network. The International Journal of Advanced Manufacturing Technology **97**, 573–581 (2018). https://doi.org/10.1007/s00170-018-1894-0
11. Marmanis, D., Datcu, M., Esch, T., Stilla, U.: Deep learning earth observation classification using imagenet pretrained networks. IEEE GEOSCIENCE AND REMOTE SENSING LETTERS **13**(1), 105 (2016)
12. Masci, J., Meier, U., Ciresan, D., Schmidhuber, J., Fricout, G.: Steel defect classification with max-pooling convolutional neural networks. In: The 2012 International Joint Conference on Neural Networks (IJCNN). pp. 1–6 (2012). https://doi.org/10.1109/IJCNN.2012.6252468
13. Park, J.K., An, W.H., Kang, D.J.: Convolutional neural network based surface inspection system for non-patterned welding defects. International Journal of Precision Engineering and Manufacturing **20**(3), 363–374 (2019)
14. Perez, L., Wang, J.: The effectiveness of data augmentation in image classification using deep learning. CoRR **abs/1712.04621** (2017), http://arxiv.org/abs/1712.04621
15. Reiter, M., Gusenbauer, C., Huemer, R., Kastner, J.: At-line x-ray computed tomography of serial parts optimized by numerical simulations. International Symposium on Digital Industrial Radiology and Computed Tomography - DIR 2019 (2019), https://www.ndt.net/article/dir2019/papers/Th.2.A.1.pdf
16. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why should I trust you?": Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 1135–1144. KDD '16, Association for Computing Machinery (2016). https://doi.org/10.1145/2939672.2939778
17. Schlotterbeck, M., Schulte, L., Alkhaldi, W., Krenkel, M., Toeppe, E., Tschechne, S., Wojek, C.: Automated defect detection for fast evaluation of real inline CT scans. Nondestructive Testing and Evaluation **35**(3), 266–275 (2020). https://doi.org/10.1080/10589759.2020.1785446
18. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv 1409.1556 (2015), http://arxiv.org/abs/1409.1556
19. Staude, A., Bartscher, M., Ehrig, K., Goebbels, J., Koch, M., Neuschaefer-Rube, U., Nötel, J.: Quantification of the capability of micro-CT to detect defects in castings using a new test piece and a voxel-based comparison method. NDT & E International **44**, 531–536 (2011). https://doi.org/10.1016/j.ndteint.2011.05.006
20. Tan, M., Le, Q.V.: EfficientNet: Rethinking model scaling for convolutional neural networks. In: Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 6105–6114 (2019)