Information Extraction from Industrial CT Scans Using 3D Deep Learning

Patrick FUCHS [1], Sven GONDROM-LINKE [1]
[1] Volume Graphics GmbH, Heidelberg, Germany

Contact e-mail: patrick.fuchs@volumegraphics.com

Abstract. Data is at the heart of the fourth industrial revolution and with the spread of automated non-destructive evaluation we have an excellent driver of data generation at hand. However, the acquired information needs to be parsed to make it machine-readable. Especially, imaging data like 3D CT scans offer plenty of useful information. Yet, the impeding influence of image artifacts complicates the interpretation of this data.

Modern 3D deep learning with its processing speed and accuracy is a promising tool to efficiently automate the information distillation from imaging data. It allows us to solve high-level classification tasks, e.g. OK/NOK-decisions, as well as low-level semantic segmentation tasks which build the foundation for the extraction of more detailed information.

Unfortunately, as helpful machine learning is, as many risks it poses. We present the implementation of a machine-learning-based in-line inspection system for light metal cast parts in terms of a detailed case study which explains the typical machine learning project life cycle, unveils the potential pitfalls on the way to a solution, and explores the vast number of possibilities:

First, we examine the creation of a proper data set pointing out the importance of a consistent labeling. Here, we go in more detail about how the digital twin of the imaging system provides a shortcut to an accurately labeled training set via simulations. Then, we discuss the need for a proper validation set for the project to be successful and to build the necessary trust in machine learning systems. Finally, we share our considerations of model deployment and how to monitor the inspection system dealing with concept drift.

1. Introduction

In the course of the fourth industrial revolution, non-destructive evaluation (NDE) methods using imaging techniques pervade the shop floor to establish a comprehensive at-line and in-line inspection. Being primarily known from the product development stage and from quality assurance laboratories doing sample checks, new challenges arise due to the short cycle-times in production. The images are encumbered with much more artifacts. Nonetheless, these methods offer the advantage of allowing both dimensional measuring and structural integrity checks. To ensure a fully automated inspection and thus make the leap to NDE 4.0, the relevant information contained in the images needs to be extracted first—at best without any human intervention. Here, great hope lies in the methods of modern machine learning (ML), primarily due to the unprecedented successes of the promising deep learning (DL) approaches which emerged during the last decade. With the possibility of dealing with highly

artifact-affected data [1], DL helps with the challenging data we encounter in at-line and in-line scenarios. To name just a few examples, in the automotive sector we face the inspection of large light metal cast parts like cylinder housings with a cycle time of five minutes and less or the inspection of an entire production of batteries with a cycle time of only a few seconds per part. In the first case strong artifacts from scattering and beam hardening arise in the data, complicating the proper determination of the material boundaries and the reliable detection of porosity. In the latter case strong artifacts from image noise reduce the contrast resolution and, thus, impede the consistent segmentation of the anode overhang and with that the computation of its length and bending.

With properly trained deep neural networks, however, we are able to handle these challenges and can extract the necessary information from the imaging data. Unfortunately, getting a ML system for NDE in place is not an easy task: Especially, the scarce availability of good and consistently labeled training sets poses a huge challenge. Almost all the overwhelming successes presented in the media are achieved by big tech companies, often referred to as "AI first" companies like Google, Meta, or Amazon, which have access to a vast amount of data and compute power (see Figure 1). Compared to that the extent of the data we have available to train a model, for example, to segment porosity in cast light metal parts or the anode overhang in batteries, is rather limited. Despite being able to rent large amounts of compute power in the cloud, it is often considered being too expensive. Furthermore, transferring the data to that cloud servers during an in-line inspection might take too long. However, most of the actual use cases in NDE 4.0 fall into that category, which is often referred to as ML "at reasonable scale" [2] (see Figure 1).

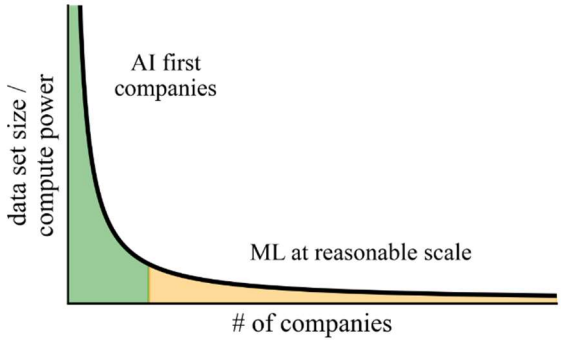


Figure 1. Only a handful companies have the data, the people, and the computing power to develop novel ML applications setting new trends. The vast majority needs help to apply ML at reasonable scale *[2]*.

The second issue we have to address when establishing ML for NDE tasks is the lack of explainability of how a ML system arrives at a decision. In contrast to traditional image processing approaches ML systems do not offer many possibilities for configuration (if they do at all). While the parameters of image processing methods, for example, can be modified on the fly providing the inspector with a feeling of control, DL models cannot be changed that easily. Once trained, they can only be modified by changing the training data and re-training or fine-tuning them. To increase the trust in ML systems they need to be properly evaluated before putting them into production. That means we need to define an appropriate validation set and suitable evaluation metrics which match the targeted task. Of course, there are efforts to provide more explainability to ML systems and DL models in particular [3]. However, these explanations add an additional layer of complexity [4, 5, 6].

From the reduced configurability another issue arises: If there are changes in the inspection system or the test specifications, the ML system needs to be adapted, too. This so-called concept drift [7] can heavily affect the accuracy and predictiveness of the overall inspection and should be understood before implementing a ML system for an inspection task. With every change to the system the ML system should at least be re-evaluated—which requires a new validation set acquired with the new setup. This also applies when individual

components of the systems are supposedly improved, for example, switching the detector of the CT system to a more sensitive one with increased spatial resolution.

In this paper we focus on these three aspects of a ML project which we find to have the most significant impact on its success:

- We explore the effect of inconsistent and noisy labels on the outcome, advocating a data-centric ML development in the field of NDE, mainly driven by the scarcity of available data.
- We argue to only learn what cannot be measured in favor of explainability of the overall method and increasing the trust in the DL model by using proper evaluation.
- We raise the awareness that the data used during development of the ML model needs to be from the same distribution as the actual production data and we warn of concept drift.

We first address these issues from a theoretical point of view in Sections 2 and 3, and later, in Section 4, we show their actual effect in a case study of detecting porosity in CT scans of a light metal cast cylinder housing.

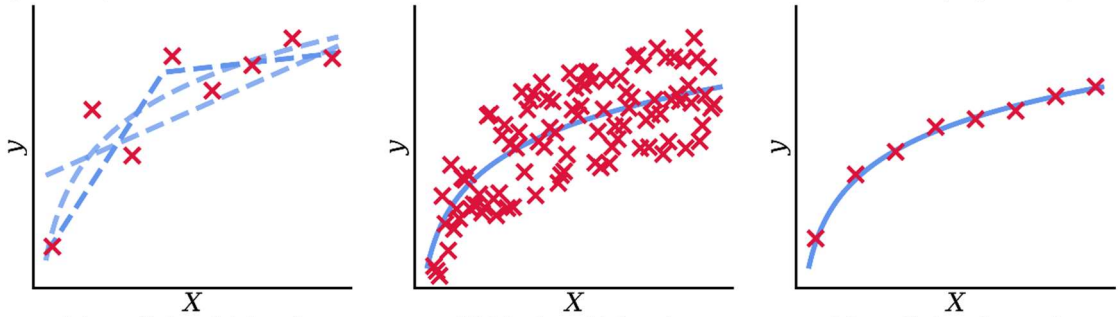
2. Data-centric ML Development in NDE

Assuming an in-line system that produces a CT scan of two gigabytes every five minutes, which to our knowledge is a rather conservative estimate, the amount of data would total to more than half a terabyte a day. So, it is hard to imagine that there is not enough training data. However, the situation is more complex: To be usable for training, the data needs to be labeled appropriately. In our experience, the data is usually processed by skilled domain experts who visually inspect the data by quickly scrolling through the CT scan. That often leaves us with only an OK/NOK decision per CT scan—without an indication of what rendered the part NOK—or we only get a sparsely labeled data set which might contain more critical flaws than the ones indicated [8]. In addition, the produced data set usually is highly imbalanced as—luckily—most of the parts are OK. In order to train a DL model that reproduces these decisions we would need to process an entire CT scan at once and we would need tens of thousands of examples for solving the task, which would require a tremendous hardware effort that currently is not feasible at all.

Another important issue contributing to the scarcity of data is the different handling of data ownership compared to the B2C-applications we know from our daily lives. There, millions of users willingly donate fresh labeled data by uploading images to web-platforms and assigning tags to them. In the terms and conditions which the users have to accept, the providers of those platforms grant themselves the rights to use these images for the purpose of developing new ML systems. As an effect, this data can be used to solve ever more data-hungry applications. In the B2B-world the data is at much higher stake. Here, the data is protected by non-disclosure agreements such that the data of one customer must not be used for another customer. Despite there being good reasons like (i) the CT scans containing sensitive information about intellectual property or (ii) the fact that the experts labelling the data are expensive and the customers are not willing to make this information advantage available to others, it severely complicates the creation of large, labeled data sets from which all can benefit. So, we do not develop one ML system for millions of users but hundreds (or thousands) of ML systems for hundreds of customers.

However, for most of the NDE tasks we need to solve, it is not absolutely necessary to resort to large data sets—on one condition: The labels need to be consistent! Sounding rather obvious, in practice it turns out to be more difficult. The labels not only differ between

different domain experts but also between different sessions of the same expert (note that one goal of establishing automated systems is to avoid such variation) [1, 9, 10]. This leads to noise and inconsistencies in the training data and dampens the prediction results. Despite consistency being such an important factor, even in well-established benchmark data sets we encounter errors in the labeling. For example, the famous ImageNet data set, which is widely used to pre-train 2D computer vision models, contains about 6 % incorrect labels [11]. As it is a really large data set, this error quote remained unrecognized for quite a long time. The sheer number of training samples mitigated the effect of the erroneous labels. Figure 2 demonstrates the severe effect of noise in the training data on the outcome: If the data is noisy and we only have a small number of sample points, it is hard to estimate the true distribution (Figure 2a). To safely determine the true distribution, we either need a lot more sample points (Figure 2b) or we need to reduce the amount of noise in the labels we have (Figure 2c) [12].



(a) small data, high noise (b) big data, high noise (c) small data, low noise

Figure 2. The effect of noise on the training process: (a) Scarce data with noisy labels makes it very hard to find the true distribution. (b) Having more training data at hand makes it easier to detect the true distribution under the presence of noise. (c) A consistent labelling allows for good results even on scarce data sets. *[12]*

Following the data-centric AI movement, initiated by Andrew Ng [12, 13], we propose to put more emphasis on a good training set to solve the challenges of establishing ML in NDE. In his recent talks, Andrew Ng argues that (i) state-of-the-art DL models are good enough to solve a wide range of tasks we encounter in B2B scenarios and (ii) that it is more beneficial for the success of a ML project to use such a standard DL model and to focus more on the training data, "moving from big data to good data" [12].

A ML system always consists of code and data. In the last decade, a lot of effort has been put in the code part. Researchers were coming up with ever more powerful DL models and we eventually arrived in a state which provides a good "default model" that allows to achieve high precision results for almost all computer vision tasks. For classification tasks there is the ResNet architecture [14] (and for hardware limited use cases the MobileNet architecture [15]); for localization tasks we have the RCNN and YOLO architectures [16, 17] which add bounding box proposals upon the classification; and for segmentation tasks we have different flavors of the UNet architecture [18] with up-sampling layers and skip-connections which form its distinctive U-shape. In the B2B-usecase the goal is not to beat a high score by another 0.1 % of accuracy but to build a precise, fast, and robust ML system for a specific use case. So, we are better off by properly training a proved "default model" instead of designing a complex and hardly maintainable ivory tower.

For a proper training process, we need a proper training set that fulfills the following criteria: (i) It covers the entire problem domain. DL models are rather bad at extrapolating data [19] so it is important that the training set contains examples of every flaw the model should detect.[1] (ii) It is labeled consistently. As discussed above, noisy labels have a significant impact on the training results. (iii) It is representative of the production data. The training data needs to come from the same data distribution, i.e. data source, as the production

[1] A different approach would be anomaly detection models that can spot deviations from the standard but do not categorize them [20].

data [21]. Particularly, this means that it is necessary to fix the scan parameters of the CT system before starting to train the DL model.

A good and increasingly common approach to arrive at consistent labels is the use of synthetic training data [1, 22, 23, 24]. When using synthetic data, however, it is important to keep the gap to the real-world data as small as possible [25], i.e. the synthetic data needs to be realistic enough so that the model does not degenerate. Then, it is also possible to train with synthetic data only. For imaging methods, like CT, this requires a good digital twin not only of the part under examination but also of the imaging system. Figure 3 compares the precision of the predictions of differently trained porosity segmentation model in terms of their intersection over union (IoU) [26]: (i) The model trained with inconsistently labeled, real CT scans (red curve) performs worst; (ii) The model trained with poorly created synthetic data, yet, with precise labels yields more confident—but not necessarily better—results (blue curve); (iii) The model trained with an elaborated, simulated training set using the digital twin of a real CT system significantly outperforms the other models (green curve). Note that there is probably still some noise in the labels of the evaluation data that dampens the results of the models.

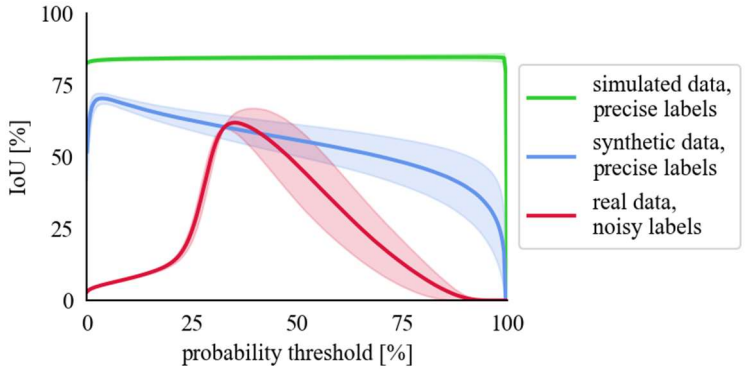


Figure 3. With the help of a realistically simulated training set (green curve) we are able to overcome the issues we have with the noisy real data (red curve). However, synthetic data has to resemble all aspects of reality well enough, otherwise we see no benefit (blue curve) compared to training on real data with inconsistent labels.

We will explore the impact of noise in the training data, in the example of porosity detection in cast aluminum parts, in Section 5.1. There we will also discuss the impact on the probability of detection (POD).

These significant fluctuations in the quality of the segmentation results demonstrate that a proper validation of the trained DL model is absolutely necessary before deploying it to production. A proper validation set, i.e. a labeled data set that is not used during training, follows the same rules we described above for the training data. Note that the point is not to have a benchmark to beat, but merely to see if the model does what it is supposed to. Besides a well-defined validation set, a feedback loop between the data scientists developing the model and the domain experts at customer's site during the training process, showing progress on a regular basis and allowing justified interventions, is key for building trust in the ML system.

3. Concept Drift – About Harmful Adjustments

Unfortunately, the pitfalls do not end with the deployment of the ML system. Even after running successfully for years, the prediction results can start to degenerate. In the field of machine learning this effect is summarized by the term concept drift. In general, concept drift describes a change in the relation between the input data and the target output that the model has learned [7]. On the one hand, these changes happen gradually or incrementally staying unrecognized for plenty of time. For example, the data can become noisier over time due to

the attrition of the imaging system (compare Figure 4a). On the other hand, they happen suddenly or abruptly and lead to an instant change in the prediction accuracy. For example, when exchanging individual components of the imaging system (compare Figure 4b) or due to the emergence of new types of flaws during production.

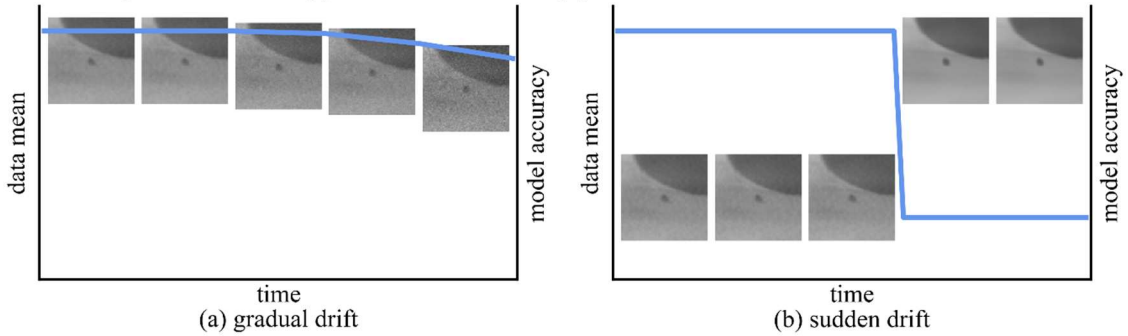


Figure 4. With the imaging system being operated 24/7 it can come to signs of wear which gradually worsen the data quality. While the accuracy of the DL model might remain at a high level for a while, the gradual concept drift already begins (blue curve) (a). On the other hand, exchanging a component of the imaging system to a better one might lead to a significant change in data quality which will have a direct impact on the accuracy, we have to deal with a sudden concept drift (b) (compare *[7]*).

ML systems are a static snapshot of the world[2] and even though we are very careful when selecting the training data, making sure that the training set resembles the real-world data distribution, we only capture a single point in time. The world, however, is constantly affected by changes which we do not cover with the training set and so the model accuracy will decrease over time.

Before dealing with concept drift, we need to detect it. A straightforward monitoring approach is scanning the same sample part from time to time and comparing prediction results either to known parameters or to earlier results. This requires the proactive involvement of an operator. Another monitoring approach is using ensembles [32], i.e. multiple, differently trained models. If their consensus starts to decrease, we have an indication for concept drift. This can be automated but is computationally more expensive.

Using synthetic training data, we can improve the robustness of the DL model against certain types of concept drift from the beginning. For example, it is possible to artificially worsen the data quality of the training data which allows the DL model to cope with signs of wear of the CT system, or we can vary the positioning of the gates of a cast light metal part to be less sensitive to changes in the shape of a part. This allows us to strengthen the DL model against most of the gradually occurring effects, but we cannot arm it against all eventualities. When making changes to the overall system, like changing a component of the CT system or changing the specifications, it is necessary to re-evaluate the DL model and probably it is also necessary to re-train it.

However, it is not so easy to simply re-train and re-deploy a DL model. Besides having to collect a new training set, we encounter another difference between the B2C- and the B2B-world here: While for user applications in the B2C-world the DL model is usually hosted by the service provider, in the B2B-world the DL models typically run on-premises —probably on an isolated system. The hosted DL models can be exchanged at any time and most of the users will not even notice. Updating an isolated system is associated with increased effort. So, unnecessarily induced drift should be avoided, which means all system relevant parameters should be fixed before starting the training of a ML system.

We will explore the impact of concept drift to the model in Section 5.2, by simulating the exchange of a component of the CT system towards better contrast resolution and see how it affects the precision.

[2] If we do not explicitly model some active learning system with a human in the loop.

4. Pore Detection in Light Metal Casting – A Case Study

To demonstrate the effect of the above-mentioned challenges, we show a small case study in which we analyze the cylinder housing of an engine (see Figure 5). The object has a size of about 240 x 140 x 460 mm and is made of an aluminum alloy with an admixture of copper which impedes the penetrability by X-rays. We scan the object using a tube with 170 kV, 8 mA, and a filter of 3 mm of tin. The flat field detector has a pixel size of 0.4 x 0.4 mm and we set up the geometry to get a magnification factor of 1.36. To work in a controlled environment with an accurate ground truth at hand, we run our experiments on simulated data which, as shown in Section 2 and described in [1], sufficiently reflect reality.

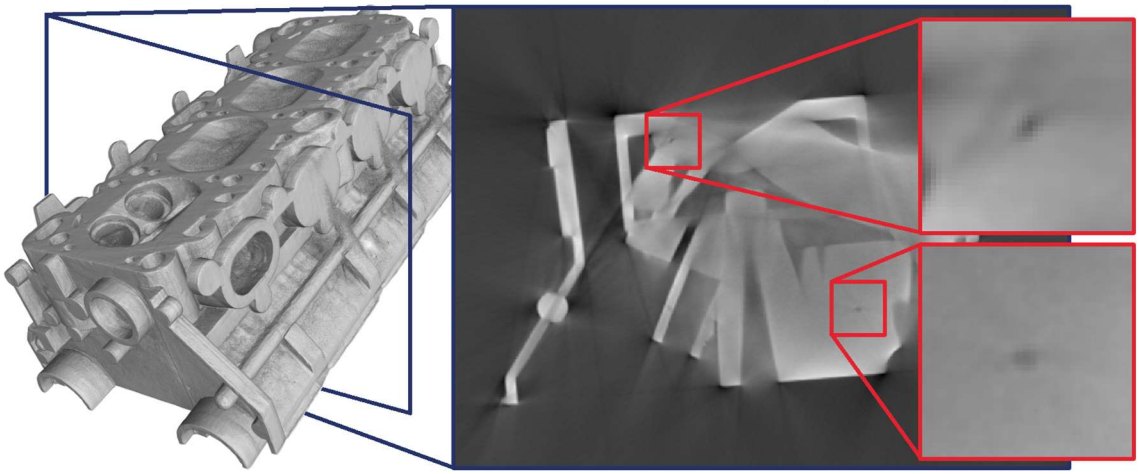


Figure 5. This CT scan of a cylinder housing was done in only few minutes, accordingly high is the artifact level in the image data. Nonetheless, we like to reliably detect even small defects of about 1 mm in diameter (which corresponds to about three voxels).

4.1 The Effect of Noise in the Labels

The first experiment deals with the effect of noisy labels on the prediction performance. Therefore, we prepare (i) a small training set of three CT scans with noisy labels which are either too small or too big at random, as we have seen in the expert labels, but contain all pores, (ii) a big data set of 16 CT scans with noisy labels, and (iii) a small training set of three CT scans with accurate labels. Then, we train the defect detection model of [1] on each of the data sets until convergence. We achieve an IoU of 65.3 %, 78.1 %, and 86.7 % for the models (i), (ii), and (iii), respectively. This confirms the assumptions postulated in Section 2. The differences originate from the fuzziness of the predictions reported by the models trained on the noisy data (compare Figure 6).

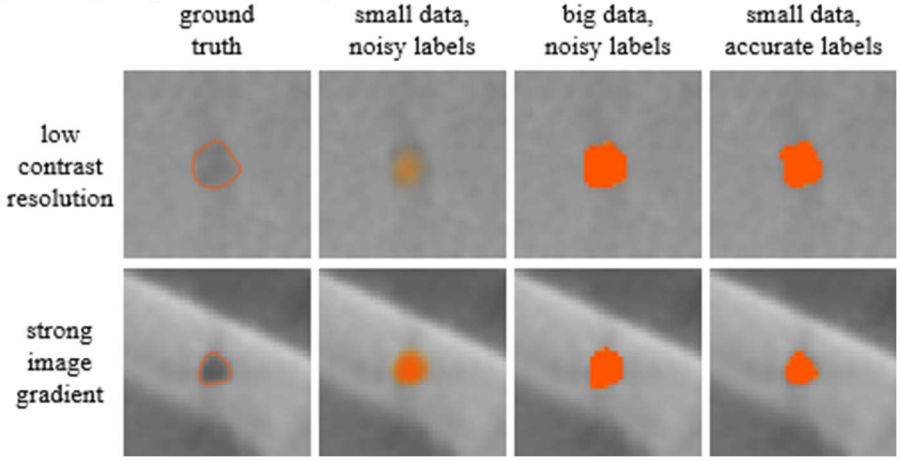


Figure 6. The results of the differently trained models on two types of defects (with their ground truth in the first column): one with a low contrast resolution (top row) and one taken from a region with a high image

gradient (bottom row). We see that the model which was trained on the small data set with noisy labels has problems in finding the correct boundary and yields fuzzier results (second column). The model which was trained on a bigger data set with noisy labels yields crisp edges but tends to overestimate the size of the defects (third column). Finally, the model which was trained on the small data set but with accurate labels yields the best results, precisely segmenting al the defects (last column).

In addition, this has a significant effect on the POD: As shown in Figure 7, particularly the detection rate for small pores significantly decreases when training with noisy labels. The detection probability of model (i) decreases for larger defects due to the fuzzy boundaries. The fact that model (ii) saturates at 100 % earlier than model (iii) is owed to the overestimation of the defect size and, therefore, has a negative influence on the false positive rate. Considering that we are using the "a vs. â" method to compute the POD and the high IoU value, we come to the conclusion that model (iii) trained on a small data set with accurate labels performs best.

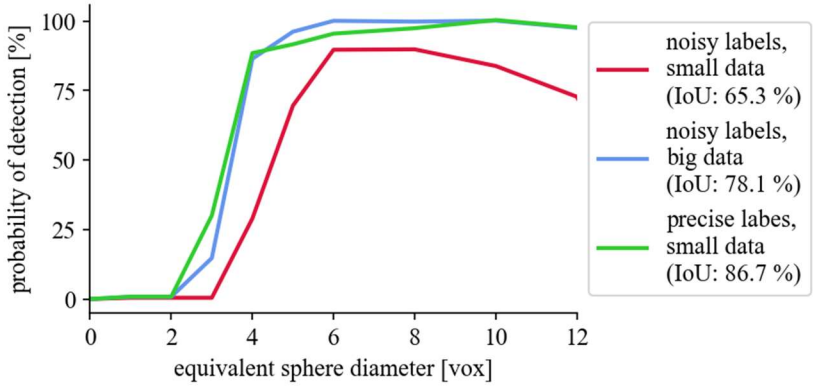


Figure 7. We compute the POD for the three models using the "a vs. â" method. We see that the model which was trained on a small data set with noisy labels struggles to find small defects at all and has problems in precisely segmenting larger defects to their full extent (red curve). This effect is mitigated by adding more data (blue curve). But this can also be mitigated by using accurate labels (green curve).

While in this experiment—which was carried out in a controlled environment—we needed more than five times more training data to mitigate the effect of noisy labels, we experience that when working with real data even more training data is necessary, because the labels are generally noisier and the real CT scans contain a little more variation in data quality.

4.2 The Effect of Improvements to the System

In the second experiment we address sudden concept drift. Imagine we detected a decrease in the amount of light in the CT system and, therefore, exchanged the source. As the market evolved in the meantime, we switch to a new type of X-ray source. With this new source we can go up to 450 kV and with the new amount of light arriving at the detector we can refine the binning of the detector. The new image quality looks very pleasing to the human eye; most of the defects are clearly visible with crisp edges. However, our DL model cannot deal with the new data: The IoU on the validation set decreases to 74.0 % and we also recognize a decrease in the POD (see Figure 8). The reason is simple: The model has not seen a comparable data quality during the training process.

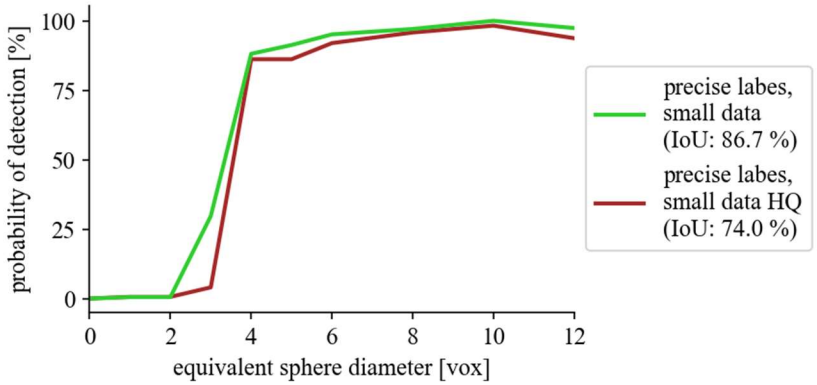


Figure 8. For the model that was trained on the small data set with accurate labels we examine the POD under the presence of sudden domain drift. We see that even though we switched to a perceived higher image quality the prediction performance of the ML model decreases. This is since the model has not seen comparable data during training.

This experiment also emphasizes the necessity to sample the training data from the actual production data. When setting up an inspection system it is important to first fix the parameters of the CT system before starting to train a DL model. However, the development process allows to iterate back and forth between tuning the CT system and tuning the ML system, so that we can optimize the scan-time and the prediction quality appropriately.

5. Conclusion

We see a huge difference between the world of B2C-ML, which we know from our daily lives and its broad presence in the media, and the world of B2B-ML, which we encounter in NDE. To successfully establish ML systems in NDE we advocate a data-centric workflow to avoid common pitfalls that could lead to the failure of a project and to only learn what cannot be measured.

We showed (i) how the data-centric view helps dealing with data scarcity and how inconsistent and noisy labels lead to worse prediction results; (ii) that, in any case, ML needs to be used sensible, learning only what cannot be measured, and validated properly; and (iii) it is advisable to monitor ML systems in production and to not expose the ML system to intentional concept drift. This particularly comprises the need of the ML system to be trained with the same data that it will encounter in the later production scenario.

In the end, machine learning is all about the data and if the data does not fit the use case, machine learning will not be able to solve anything. The use of digital twins and synthetic data can help to guide the way towards better ML systems.

Acknowledgements

This work did not receive any external fundings. Nonetheless, we would like to shut out to our colleagues at Volume Graphics GmbH for their fruitful discussions. Kudos also goes to the AI Working Group of the German Society of Non-destructive Testing (DGZfP), who is always looking for active members.

References

[1] P. Fuchs, T. Kröger and C. S. Garbe, "Defect detection in CT scans of cast aluminum parts: A machine vision perspective," *Neurocomputing,* vol. 453, pp. 85-96, 2021.

[2] J. Tagliabue, "You Do Not Need a Bigger Boat: Recommendations at Reasonable Scale in a (Mostly) Serverless and Open Stack," *ACM Conference on Recommender Systems,* pp. 598-600, 3 2021.

[3] A. Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," *IEEE Access,* vol. 6, pp. 52138-52160, 9 2018.

[4] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," *International Conference on Machine Learning,* vol. 48, pp. 1050-1059, 2016.

[5] G. Alain and Y. Bengio, "Understanding intermediate layers using linear classifier probes," *arXiv,* 10 2016.

[6] H. Zhang, I. Goodfellow, D. Metaxas and A. Odena, "Self-Attention Generative Adversarial Networks," *arXiv,* 5 2018.

[7] I. Žliobaitė, "Learning under Concept Drift: an Overview," *arXiv,* 10 2010.

[8] S. Jain, A. Smit, A. Y. Ng and P. Rajpurkar, "Effect of Radiology Report Labeler Quality on Deep Learning Models for Chest X-Ray Interpretation," *Neural Information Processing Systems Workshop,* 2021.

[9] D. Kahneman, O. Sibony and C. R. Sunstein, Noise: A Flaw in Human Judgment, Hachette Book Group, 2021.

[10] M. Motamedi, N. Sakharnykh and T. Kaldewey, "A Data-Centric Approach for Training Deep Neural Networks with Less Data," *Neural Information Processing Systems Workshop,* 2021.

[11] C. G. Northcutt, A. Athalye and J. Mueller, "Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks," *Conference on Neural Information Processing Systems,* 3 2021.

[12] A. Ng, "MLOps: From Model-centric to Data-centric AI," 2021. [Online]. Available: https://youtu.be/06-AZXmwHjo.

[13] E. Strickland, "Andrew Ng: Unbiggen AI," 2022. [Online]. Available: https://spectrum.ieee.org/andrew-ng-data-centric-ai.

[14] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *IEEE Conference on Computer Vision and Pattern Recognition,* pp. 770-778, 12 2016.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *IEEE Conference on Computer Vision and Pattern Recognition,* 2018.

[16] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," *IEEE International Conference on Computer Vision,* 2017.

[17] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition,* pp. 779-788, 6 2016.

[18] O. Ronneberger, P. Fischer and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Lecture Notes in Bioinformatics,* vol. 9351, pp. 234-241, 2015.

[19] U. Hasson, S. A. Nastase and A. Goldstein, "Direct Fit to Nature: An Evolutionary Perspective on Biological and Artificial Neural Networks," *Neuron,* vol. 105, no. 3, pp. 416-434, 2 2020.

[20] M. Salehi, H. Mirzaei, D. Hendrycks, Y. Li, M. H. Rohban and M. Sabokrou, "A Unified Survey on Anomaly, Novelty, Open-Set, and Out-of-Distribution Detection: Solutions and Future Challenges," *arXiv,* 10 2021.

[21] A. Ng, Machine Learning Yearning, Self-published online resource, 2017.

[22] A. White, "By 2024, 60% of the data used for the development of AI and analytics projects will be synthetically generated," 2021. [Online]. Available: https://blogs.gartner.com/andrew_white/2021/07/24/by-2024-60-of-the-data-used-for-the-development-of-ai-and-analytics-projects-will-be-synthetically-generated/.

[23] S. Castellanos, "Fake It to Make It: Companies Beef Up AI Models With Synthetic Data," 2021. [Online]. Available: https://www.wsj.com/articles/fake-it-to-make-it-companies-beef-up-ai-models-with-synthetic-data-11627032601.

[24] T. Pollok, L. Junglas, B. Ruf and A. Schumann, "UnrealGT: Using Unreal Engine to Generate Ground Truth Datasets," *International Symposium on Visual Computing,* pp. 670-682, 2019.

[25] T. Jaunet, G. Bono, R. Vuillemot and C. Wolf, "SIM2REALVIZ: Visualizing the Sim2Real Gap in Robot Ego-Pose Estimation," *Conference on Neural Information Processing Systems,* 9 2021.

[26] V. K. Rentala, D. Kanzler and P. Fuchs, "POD Evaluation: The Key Performance Indicator for NDE 4.0," *Journal of Nondestructive Evaluation,* vol. 41, no. 1, 3 2022.

[27] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, B. Kim and G. Brain, "Sanity Checks for Saliency Maps," *Conference on Neural Information Processing Systems,* 2018.

[28] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl and B. Yu, "Definitions, methods, and applications in interpretable machine learning," *Proceedings of the National Academy of Sciences,* vol. 116, no. 44, pp. 22071-22080, 1 2019.

[29] C. Rudin, "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead," *Nature Machine Intelligence,* vol. 1, pp. 206-2015, 5 2019.

[30] C. Molnar, G. König, J. Herbinger, T. Freiesleben, S. Dandl, C. A. Scholbeck, G. Casalicchio, M. Grosse-Wentrup and B. Bischl, "General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models," *International Conference on Machine Learning,* 7 2020.

[31] C. Molnar, G. Casalicchio and B. Bischl, "Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges," *Communications in Computer and Information Science,* 10 2020.

[32] J. Linmans, J. van der Laak and G. Litjens, "Efficient Out-of-Distribution Detection in Digital Pathology Using Multi-Head Convolutional Neural Networks," *Proceedings of Machine Learning Research,* pp. 1 - 15, 2020.