Discover Data

Research

Challenges and approaches when realizing online surface inspection systems with deep learning algorithms

Henrike Stephani[1] · **Thomas Weibel**[1] · **Ronald Rösch**[1] · **Ali Moghiseh**[1]

Received: 21 December 2022 / Accepted: 22 March 2023
Published online: 30 March 2023

Abstract
Using deep learning in complex online surface inspection systems is challenging due to different framework conditions. First, time restrictions in production are usually fixed in terms of clock rate and response time. Furthermore, these methods need a lot of data, while typically the data situation is thin in the beginning as well as continuously unbalanced: defects occur rarely and thereby providing few example data for learning, while the desired detection rate is 100%. Another important issue is that although defect catalogues exist, they often change, especially when automatic inspection is applied for the first time. This is due to imaging systems usually being able to detect more defects than visual-manual inspection, therefore production, management, and quality assurance usually reiterate their prior defect catalogues. However, data driven methods depend heavily on consistent annotation. Therefore, respective parties must be made aware of this issue on the one hand, on the other hand, annotation and reannotation must be easy and useable by non-experts. Related is the issue of parametrization and traceability. Both are not inherent to neural networks but must be provided to some level to help building trust in machine learning methods. In this paper, we present a quality inspection system that uses deep neural networks for defect detection under real production conditions in wood manufacturing. We will address how we systematically deal with the above issues both in terms of process and algorithm.

1 Introduction

In many industrial production sites, manual inspection is still the method of choice when it comes to surface control. Among the reasons for this are the flexibility of manual laborers and their ability to make decisions under unexpected circumstances.

However, there are several reasons why computer vision (CV) based inspection systems are constantly on the rise [1]. The main reason is that computers mostly outperform humans in repetitive calculations, in speed as well as consistency. Furthermore, long running systems are more cost efficient than manual inspection [2].

Up until today, CV based inline inspection algorithms are often done with so-called "classical" CV algorithms. That means experts build algorithm chains. These chains consist of filtering algorithms, statistical methods, adaptive thresholding, modelled feature extraction and usually some classification step in the end [3, 4]. This approach has various advantages as can be seen on the left-hand side of Table 1.

The disadvantages however (right hand side of Table 1) are increasing. The most important one is that these algorithms do not perform well in certain cases. Additionally, algorithms for data driven methods, have been constantly improved in the last 15 years. Therefore, these "classical" algorithms have been replaced more and more by data driven

✉ Henrike Stephani, henrike.stephani@itwm.fraunhofer.de; Thomas Weibel, thomas.weibel@itwm.fraunhofer.de; Ronald Rösch, ronald.roesch@itwm.fraunhofer.de; Ali Moghiseh, ali.moghiseh@itwm.fraunhofer.de | [1]Fraunhofer ITWM, Kaiserslautern, Germany.

Table 1 Classical image processing

Advantages	Disadvantages
With a few example images, first approaches can be developed Lean solutions are possible. Simple (sub-) tasks can be solved with very efficient algorithms Single steps of the solution are understandable, explainable, and parametrizable. New variants can be included with reparametrization	High number of parameters to be manually set Design and maintenance of these algorithm chains is an expert task Image quality should be high, i.e., defects should have high contrast compared to the rest of image Performance is sometimes poor and limited Problems such as inspection of natural materials almost impossible

deep learning approaches [3]. Several problems, such as the detection of cars in images or the identification of faces, are exclusively solved with these methods and often even outperform human recognition [5, 6]. On the left-hand side of Table 2 several advantages of data driven methods are shown. Various deep learning based defect detection systems have been developed and have shown good results in several application areas. However, the applicability of these methods highly depends on the precise definition of the application domain (e.g. steel [7, 8]) and the defect types and example datasets (e.g., cracks [9–11]). Although there is a vast amount of literature on surface detection with CNNs, there are only few industrial inline inspection solutions presented. Mostly because translating the industrial expert knowledge into a necessary database presents several difficulties [12], as displayed on the right hand side of Table 2. Because of these reasons, industrial deep learning solutions for defect detection are still scarce and only applied in areas with very well-defined defect types such as cracks. However, in many application domains defect inspection is done fully manual where human interpretation—often including knowledge of the production process—is key for defect assessment. Applying state of the art CNNs to the final dataset is not the key problem. The key problem is the collection of said dataset in accordance with the specific requirements of industrial production on the one hand and with knowledge about the workings of CNNs on the other hand.

In this paper, we will present a process on how to implement a data driven solution for image -based surface inspection under industrial conditions. This process will especially focus on the issue of data collection and annotation (second and third disadvantage in Table 2). We will discuss challenges and present an application example.

In the following, we will first describe the pipeline of classical image processing and deep learning for developing an inline surface inspection system, and then propose a hybrid solution for such inspection systems. In the next section, we explain an application example for a real-world problem and show the main parts and steps of our application. In the discussion section, we will reveal the challenges of such systems and in particular discuss the data cleaning process. In the last section, we summarize our results and further work.

2 Methods

2.1 Classical image processing for inline surface inspection

A classical image processing pipeline usually consist of filtering the image, adaptive thresholding to identify candidates for defects and finally classification of these defects [13].

A typical process of developing image processing solutions for inline surface inspection involves expert knowledge from domain experts but only at specific points. In Fig. 1 the process steps displayed in blue represent their input. Steps

Table 2 Data driven methods

Advantages	Disadvantages
Generic and depending mostly on good data Openly available architectures and datasets that can be adapted to the respective application domain Powerful on poor quality images Outperform classical image processing/machine learning methods if enough data is available	The data is very unbalanced. Defects occur rarely (in comparison to non-defective parts) and vary in importance Data collection must be done in accordance with the production schedule—as not every variety is produced at the same time. Production is usually not changed for the purpose of data collection Data annotation is necessary and needs precise and consistent definition of defect classes

in yellow represent CV expert work. Initially, defect catalogues are shared, and production/manual inspection expert interviews are conducted. However, after this, even with only very limited image material, a first algorithm chain can be developed. Image processing experts combine parametrizable algorithms. A first iteration is then run on the system, at this point again, production experts check the outcome of the automatic analysis. Typically, too many defects are found on the one hand (false positives), and some might also be missed (false negatives). This leads to a reparameterization of the developed algorithms. If the system is well designed, unknown variants of the product/new kinds of defects can be included into the system by adapting only a few parameters.

However, challenges of this expert modelled image processing are that the advantage of being able to parametrize each step also is a disadvantage: parametrization has to be done manually and usually with image processing expert knowledge.

2.2 Deep learning

We will focus on deep learning in this work as this is the dominant form of data driven approaches. Within deep learning approaches convolutional neural networks (CNN) are the most relevant architectures. Images are not a random accumulation of pixel values, but rather contain spatial information. Features should therefore represent this spatial connection. CNNs are designed to encode this information on several resolution layers. They consist first of a image encoding through so-called convolutional layers—in which image information is encoded—and max pooling layers—where most relevant features are selected. And then, depending on the desired output, for example of a classification step done by a so-called fully connected layer, where the network decides how to sort the images into classes [14], or of a decoder step, where not only classification but also segmentation can be performed [15].

In both cases thousands to millions of weights must be learned. Not only efficient algorithms and computational power are needed for this purpose, but also a lot of data. Data acquisition, however, is only a small part of the task. For the networks to learn their weights properly they must know "how" to learn them. In case of classification on an image level they need to be annotated. In case of detection, annotation is not performed on an image level, but the various classes contained in an image must also be localized.

2.3 Deep learning for inline surface inspection

The most important task for data driven methods is to build a representative and large enough database. This, however, changes the procedure of developing surface inspection systems. In a classical system the work of designing the surface inspection system is mostly done by image processing experts who "translate" the defect catalogue and the domain expert input into algorithms—see the yellow part of the pipeline in Fig. 1. This modelling can be avoided in deep learning solutions, as it is done by the data itself. Data must therefore be reliable and well categorized. Therefore, domain experts must be involved to a higher degree in building the database as illustrated by the green parts in Fig. 2. This can be very time-consuming and difficult. Another issue is that in the traditional approach interpretability and parametrization are inherent to the solution. In contrast to that, in data driven methods most of this is "hidden" in the annotation. This annotation must be consistent over the whole dataset. Future changes of requirements and variants that might influence current annotation must therefore be anticipated.

2.4 Hybrid setup for inline surface inspection

The most problematic part of deep learning systems is the collection and annotation of data. Therefore, instead of exclusively developing a data-driven system, we propose to use a hybrid solution as displayed in Fig. 3. A candidate detector is developed by means of classical image processing. A focus should be in detecting rather too many than too little and

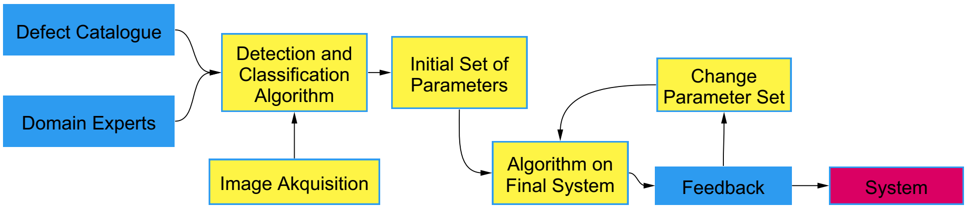

Defect Catalogue
Domain Experts
Detection and Classification Algorithm
Image Akquisition
Initial Set of Parameters
Change Parameter Set
Algorithm on Final System
Feedback
System

Fig. 1 Typical development steps of a defect detection algorithm with classical image processing. Domain experts: blue, image processing experts: yellow

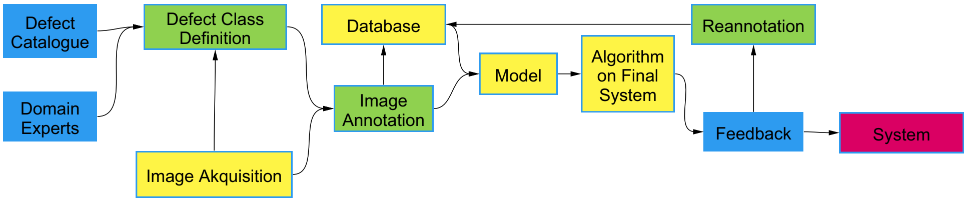


Fig. 2 Process of building a defect detection based on deep learning algorithms. Domain experts: blue, image processing experts: yellow, both: green

on building a first solution. The candidate annotation is then usually much easier and faster than the original image annotation. The discussion of how to annotate defects is similar in the full deep learning solution. However, first defect candidates can be used as a basis for discussion. This is significant mainly for two reasons: First, to get an impression of the expression of defects in the images—as they are usually different from the human-visual impression. Secondly the number of detected defects can be estimated—often automatic inspection systems detect more defects than is desirable for production management. The database that is build can either be used for classification or can be supplemented with "background" images for additional localization as described above. Even if some "raw image annotation" has to be done too (marked here as "subsampled image annotation"), the bulk of the necessary data is done by candidate annotation.

3 Application example

3.1 Inspection system environment

We present an inline-surface inspection system for shelve panels. The system will replace a visual manual inspection. There are several fixed parameters such as the production pace (about 50 m/min) and response time (depending on length of panel (7–12 s). Requirements to the system are that it meet the production times and be as good as the manual inspection (or better). The reasons for exchanging the current inspection are the higher reliability and consistency of an automatic system.

3.2 Image acquisition

The product varieties consist of unicoloured panels as well as wood pattern. The wood pattern is not just applied by colouring the panels but also by imprinting wooden structures. The panels have different sizes. Ranging from about 50 cm up to 2 m in length and 20 cm to 50 cm width.

The defect catalogue consists of typical defects such as scratches and cavities and several specific defects such as discolouring, excess of glue or lines. The higher adaptability as well as the irregular pattern of the surfaces are the reason this system was developed with a deep learning algorithm. Independently of the algorithmic approach, the first task is to define an imaging setup that ensures that defects are covered. We decided to us three different kinds of images. A brightfield setup for scratches and cavities, a darkfield setup for even surface defects and a full reflection setup that is specifically integrated for defects in the imprinted wooden structure. The setup can be seen in Fig. 4.

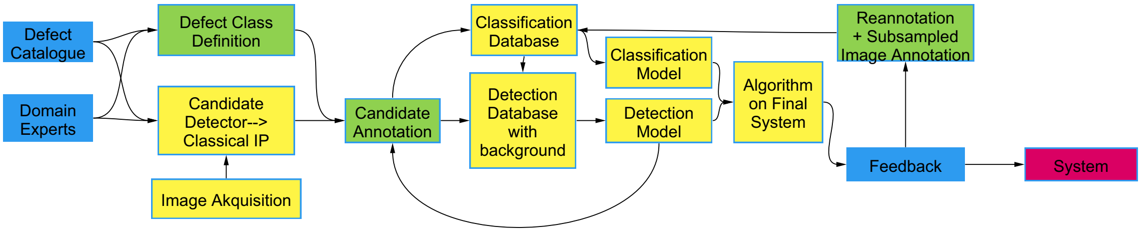


Fig. 3 Hybrid processing for inline surface inspection. Domain experts: blue, image processing experts: yellow, both: green

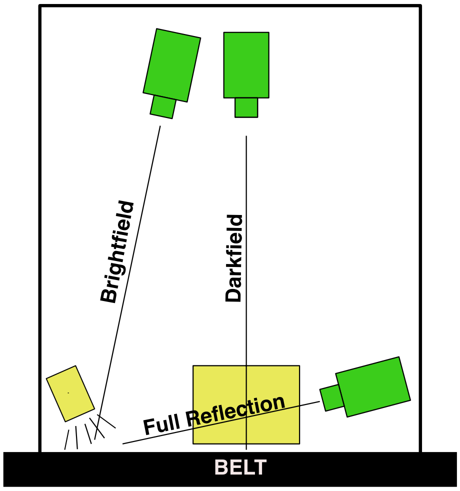


Fig. 4 Imaging Setup: light sources are in yellow (left one spotlight, right one incident light), cameras in green

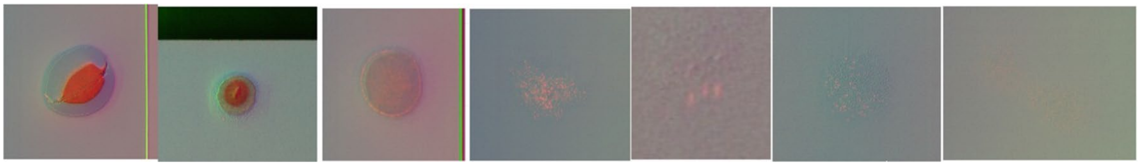


Fig. 5 Customer defined defect type "sinkhole" with different intensities, strong (left) to weak (right)

3.3 Defect definition

Defect definition must be done in discussion between production/domain experts and CV experts. The former usually have a focus on the genesis of the defect as well as the possible consequences (e.g. fixable vs. unfixable). Image processing experts must consider how to define classes that comply with the mathematical logic of the deep learning algorithm. In Fig. 5 one can see an example of the defect class "sinkhole". The expression of these defects is very different: on the left-hand side strong sinkholes can be seen that are characterized by round, clear structures, while weak sinkholes appear through a cluster of dots. Sinkholes are one example of defects that have a varying appearance and to ensure a detection by a data-driven method the full variety must be covered with multiple examples.

Therefore, in this system, we decided to build a classifier that is based on the image expression of defects as much as possible while still fulfilling the customers' requirements. We have four main categories that are of most interest to the customer. In Fig. 6 a variety of sample pictures for these four categories are shown. We include another category, that is called "negative" which is applied to false positives of the detection dataset.

3.4 Candidate detection

Tasks that can be solved by classical image processing in a stable manner, should be solved thus, because it reduces the amount of data needed. In many production cases the segmentation of object and belt is such a pre-processing step. This step is also executed for the three images from this system. After this, we have three exact segmentations of the same object. One could treat them as separate grey-scale images, process them separately and join the results in the end. Instead of this, we decided to register them onto each other, to directly deal with one multidimensional image instead of three one-dimensional ones. Furthermore, as we have exactly three channels, it is very useful to treat them as RGB images. Libraries for image processing and image display are generally available for RGB images and can directly be used on this "mixed" image. Additionally, pretraining data bases are more often available for RGB than for general multidimensional images. The CNN might therefore be easier to apply.

After foreground detection and registration, the algorithm pipeline consists of typical steps edge preserving filtering, adaptive thresholding, and mathematical morphology, all done on this RGB image.

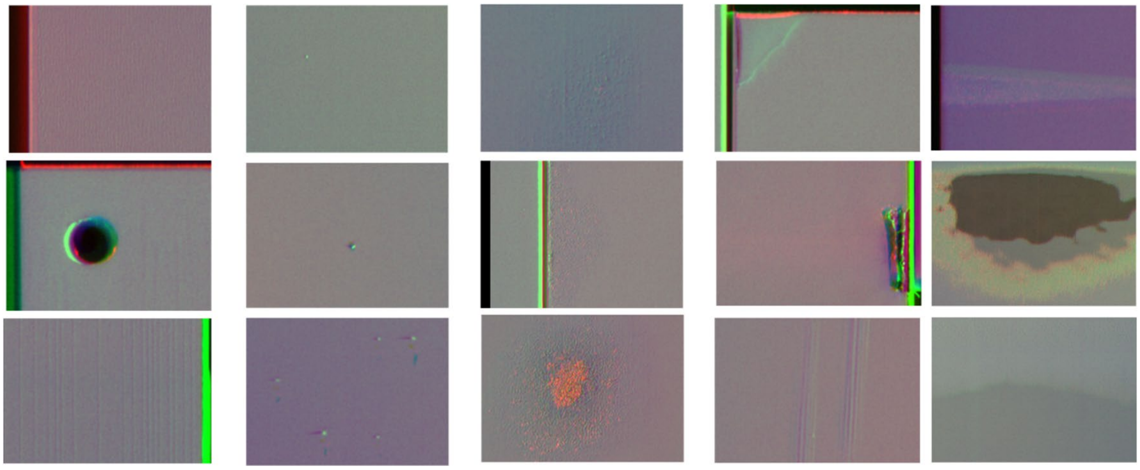
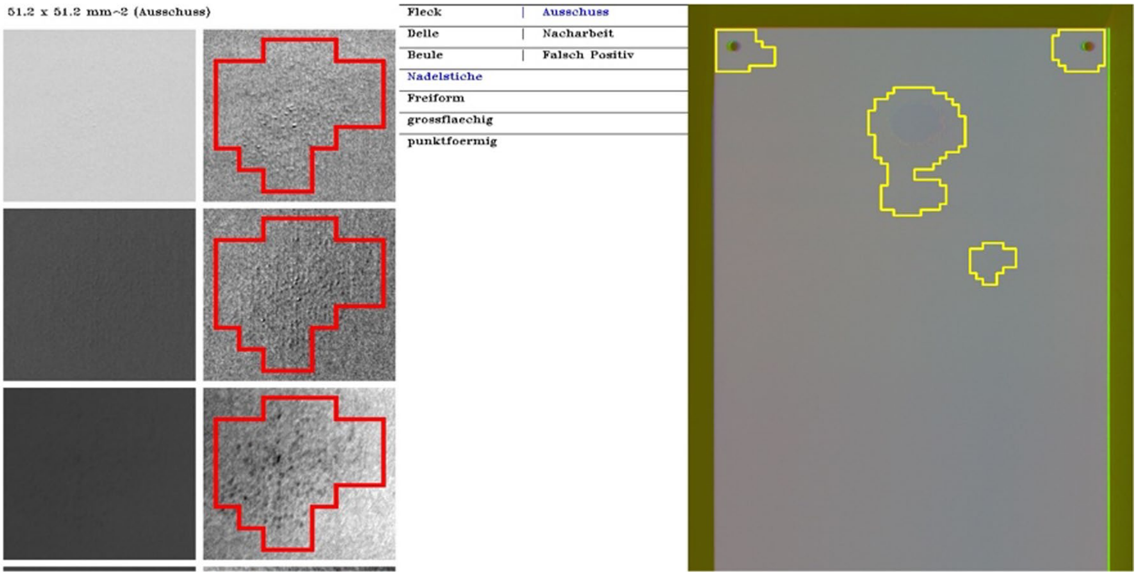


Fig. 6 In the leftmost column, the category "Rillispur, i.e., line defects is displayed. Next to it "Staub", i.e., dust, next to that "Nest", i.e. dotted local defects. They are displayed in three different intensities (top to bottom). The rightmost two columns represent the category "positive"; those are reserved for very clear or very global defects that occur less often

3.5 Candidate annotation

A panel of 2 m consists of about 9000 pixels in length, defects down to 10 pixels need to be detected (without our RGB image registration on three images per panel) this is tedious at best, not doable at worst. We built a custom annotation tool as displayed in Fig. 7. A visualization of the detailed defect itself (three different images, original and normalized for clearer visibility) is displayed alongside—as classification decision depends on size and position on the panel—the RGB image of the whole panel. The annotation expert should have the definition of the defect classes at hand. Annotation itself can then be done without further image processing/AI expertise with one or two clicks per defect. The annotation tool was programmed in python with standard open source libraries for simple GUI implementation such as HighGUI from OpenCV.



51.2 x 51.2 mm~2 (Ausschuss)

Fig. 7 Custom annotation tool. Left: three different images of candidate (brightfield, dark field, full reflection) original and normalized. Right side: position of candidate in panel, middle: annotation choices

Table 3 Precision, recall and support for CV-expert defined categories

Category	Precision	Recall	Support
Nest	0.970	0.780	1072
Rillispur	0.972	0.946	2013
Staub	0.812	0.623	334
Negative	0.830	0.967	2095
Positive	0.728	0.725	630
Accuracy	0.890	0.884	6144

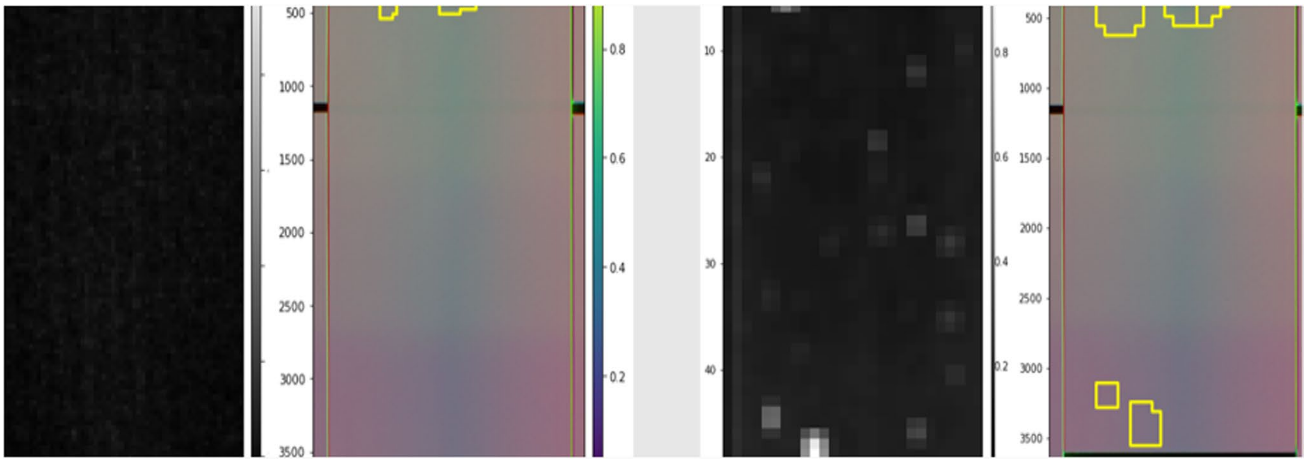


Fig. 8 Defect detection with classical image processing (left) and sliding window NN on the right

3.6 Deep learning

Our goal is to build a database that enables the use of different deep learning techniques. The model used for classification is the VGG16 [16] architecture, pretrained on the ImageNet [5] database. In Table 3 the recall and precision of the CV-expert defined annotation is shown. While the overall accuracy of the domain expert-defined classes was 58%, the overall accuracy of the CV-expert defined classes is above 88%. Careful labelling with understanding of the underlying mathematical algorithms can therefore make a big difference. Additionally, some confusion are in between the classes of "Staub" and "negative" which do not count as defects. They can therefore be neglected. It must be said, that the database for CV-expert labelling was slightly bigger and not as biased. This might also have influenced the results.

For detection we use the same model with an additional background class and combine all the labelled foreground classes. On the right-hand side of Fig. 8 one can see the resulting response of the classifier of this foreground class after applying a sliding window with overlap to detect areas of interest.

Processing time depends on the length of the panel and the number of defect candidate. However, in industrial inline inspection systems, the average processing time is of limited importance. The main goal is to always comply with the fixed production rate (50 m/min) and response time (between 5 and 8 s, depending on the length of the panel). Within this time all system coordination as well as defect detection take place. A local server is designed to comply with these requirements (no cloud computing possible).

4 Discussion

The used methods enable a fast annotation of the database. However, in contrast to classical approaches it remains difficult to change annotations iteratively. Therefore, the definition of defect classes is one of the most crucial points in designing such a system. In our approach, it became clear that the previously defined defect classes are not necessarily suitable for a databased defect detection algorithm. Especially the dataset bias and the need for sufficient representation of all classes leads to poor results in the beginning. In addition to well defined defect classes, we therefore advise a two-step approach, where a first coarse classification is performed. It is also our experience that the importance of defects changes over time, with changing production facilities and changing supplier/customers of the production. This is also a reason why a graded defect annotation is sensible. This way reannotation can also be minimized by only targeting subgroups of the whole dataset, if necessary.

The collection of additional data without any candidate detection first, also has to be performed. In our case this was done by monitoring a graphical display of the system by image processing experts. In the future we will additionally implement an automatic feedback system that enables production to do feedback sessions and thereby improve the system.

Processing time, as stated in results, satisfies the customers' response time and production rate demands. The big improvement to the prior approach (visual-manual control) is that control is now fully automatized, consistent, and transparent. Furthermore, no more manual labour is need for quality inspection.

5 Conclusion

We have shown that using deep learning for inline defect detection systems is possible but at the same time has various challenges in comparison to a traditional inspection automatization approach. Production experts and image processing experts must communicate constantly to ensure that the defect definition is correct. We have proposed a process on how such database generation can be done for industrial inline surface inspection. The main advantage of having such a database is that constant advances in deep learning can easily applied on this database. While classical solutions are adaptable only be experts and with a lot of effort, the effort of interchanging a CNN architecture for a given database is minor. Even tasks that might not be solvable for example due to time restrictions might be solvable in the future by a new architecture/hardware setup based on the existing dataset. Careful dataset generation can therefore be seen as a solid investment into the future of quality inspection as well as a means to realize a current solution.

Our experience with industrial production (especially small and medium sized companies (SME)) is, that they have very traditional inspection systems or mostly no automatic inspection at all. This is reflected by the lack of literature on Deep Learning application in industrial inline surface inspection systems.

Acknowledgements Not applicable.

Author contributions HS and AM prepared the main manuscript text. TW implemented the software solution and provided the results from Table 3. RR coordinated the application example. All authors developed the proposed hybrid process solution together. All authors reviewed the manuscript. All authors have read and approved the final manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability Data subject to third party restrictions. The IP of the data lies with the industrial customer, it cannot be shared. Not applicable due to confidentiality with industrial customer.

Declarations

Ethics approval and consent to participate Not applicable.

Consent for publication Not applicable.

Competing interests No, I declare that the authors have no competing interests as defined by Springer, or other interests that might be perceived to influence the results and/or discussion reported in this paper.

References

1. Shirmohammadi S, Ferrero A. Camera as the instrument: the rising trend of vision based measurement. IEEE Instrum Meas Mag. 2014;17:41–7. https://doi.org/10.1109/mim.2014.6825388.
2. Rauhut M, Stephani H (2017) Konzeption und Aufbau eines Online-Oberflächeninspektionssystems. In: Sackewitz M (ed) Handbuch zur industriellen Bildverarbeitung: Qualitätssicherung in der Praxis, 3., vollständig überarbeitet und aktualisierte Auflage, vol 17. Fraunhofer Verlag, Stuttgart
3. O'Mahony N, Campbell S, Carvalho A et al. (2020) Deep Learning vs. Traditional Computer Vision. In: Springer, Cham, pp. 128–144
4. Sun X, Gu J, Tang S, et al. Research progress of visual inspection technology of steel products—a review. Appl Sci. 2018;8:2195. https://doi.org/10.3390/app8112195.
5. Russakovsky O, Deng J, Su H, et al. ImageNet large scale visual recognition challenge. Int J Comput Vis. 2015;115:211–52. https://doi.org/10.1007/s11263-015-0816-y.
6. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Commun ACM. 2017;60:84–90. https://doi.org/10.1145/3065386.
7. Wang Y, Liu M, Zheng P, et al. A smart surface inspection system using faster R-CNN in cloud-edge computing environment. Adv Eng Inform. 2020;43:101037. https://doi.org/10.1016/j.aei.2020.101037.
8. Neogi N, Mohanta DK, Dutta PK. Review of vision-based steel surface inspection systems. J Image Video Proc. 2014;2014:1–19. https://doi.org/10.1186/1687-5281-2014-50.
9. Kim B, Yuvaraj N, Sri Preethaa KR, et al. Surface crack detection using deep learning with shallow CNN architecture for enhanced computation. Neural Comput & Applic. 2021;33:9289–305. https://doi.org/10.1007/s00521-021-05690-8.
10. Chen F-C, Jahanshahi MR. NB-CNN: deep learning-based crack detection using convolutional neural network and Naïve Bayes data fusion. IEEE Trans Ind Electron. 2018;65:4392–400. https://doi.org/10.1109/tie.2017.2764844.
11. Müller O (2018) Application of deep learning for crack segmentation on concrete surface. In: Forum Bildverarbeitung 2018. KIT Scientific Publishing
12. Bhatt PM, Malhan RK, Rajendran P, et al. Image-based surface defect detection using deep learning: a review. J Comput Inf Sci Eng. 2021. https://doi.org/10.1115/1.4049535.
13. Gonzalez RC, Woods RE (2018) Digital image processing, Fourth edition, Global edition. New York: Pearson.
14. Wang T, Chen Y, Qiao M, et al. A fast and robust convolutional neural network-based defect detection model in product quality control. Int J Adv Manuf Technol. 2018;94:3465–71. https://doi.org/10.1007/s00170-017-0882-0.
15. Jing J, Wang Z, Rätsch M, et al. Mobile-Unet: an efficient convolutional neural network for fabric defect detection. Text Res J. 2020;92:30–42. https://doi.org/10.1177/0040517520928604.
16. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition