Self-supervised Learning for Pore Detection in CT-Scans of Cast Aluminum Parts

Patrick FUCHS\textsuperscript{1,2}, Thorben KRÖGER\textsuperscript{1}, Christoph S. GARBE\textsuperscript{2}
\textsuperscript{1}Volume Graphics GmbH, Heidelberg, Germany
\textsuperscript{2}Interdisciplinary Center for Scientific Computing (IWR), Heidelberg University, Heidelberg, Germany
Contact e-mail: patrick.fuchs@iwr.uni-heidelberg.de

Abstract. Automatically detecting pores in computed tomography (CT) scans of cast aluminum parts is a difficult task. This is especially the case when the data quality suffers from image artifacts. These can arise due to constraints from the imaging process and the short scan times that are common in industry. In previous work we have shown that using the synthetic ground truth generated from a realistically simulated CT data set is sufficient to train modern machine learning algorithms to detect pores on real data sets—even when heavy image artifacts like noise, beam hardening, or ring artifacts are present.

Fully convolutional neural networks can yield promising results when moving from the simulated data set to real world examples. So, instead of first removing artifacts from the image and then applying defect detection algorithms, we propose to tackle the defect detection task directly without altering the original data.

In this paper we evaluate three models: (1) a fast slice-wise approach which does the prediction without the full 3D context, (2) an encoder-decoder-pair with an additional refinement step, which acquires context by reducing the spatial resolution, and (3) a plain convolutional neural network using dilated convolutions for context aggregation.

Further, we provide promising qualitative results on difficult real world cases, among others CT-scans from in-line scenarios and additively manufactured parts.

1. Introduction

The detection of pores, blowholes, cavities, and other types of defects in CT-scans of cast aluminum parts is known to be a challenging and time-consuming problem. Nevertheless, this has to be done extremely diligently to ensure the reliability of the inspected parts. The difficulties not only originate from having to cope with three-dimensional images, but especially arise from the limited image quality of real world CT-scans. The underlying physics of the scan process makes the CT-data very artifact-prone. Besides sampling artifacts from the digitization, notably noise, beam hardening, and ring artifacts affect CT-scans and impede an automated inspection. Recently, the first machine learning algorithms based on deep learning were reported to surpass human-level perception in common image classification\cite{1, 2, 3} and semantic segmentation tasks\cite{4, 5}. These modern machine-learning algorithms manage to detect and correctly classify their targets even in cluttered scenes. This accuracy is achieved by statistically evaluating huge data sets containing a broad variety of images with hand-labeled annotations. Due to the high accuracy and the invariance against disturbances like
image clutter, deep learning seem to be well suited to leverage the automated inspection of cast aluminum parts. We show that properly trained convolutional neural networks (CNNs) are able to sustain a stable detection rate when the scan-quality drops, i.e. the scan time is decreased. Having a scan of given quality, CNNs reach higher detection rates of small defects than simple filter-based methods. Fast, highly parallelized implementations utilizing modern general purpose graphic processing units (GPGPUs) allow for a high data throughput. Therefore, these methods hold the potential to bring production processes a step further towards an automated inspection of all parts.

However, as neural networks only learn a statistical representation of what they have seen during the training phase, we have to be careful when dealing with unseen data. In the end, this means if the target structures and target modalities are not covered by the training set, the deep learning algorithm cannot provide a reliable prediction. This is of special importance when the neural network not only localizes and labels the defects but also makes the decision if a defect has a critical impact on the structural integrity of a part. (The classification of defects is not part of this work.) Another downside of deep learning-based algorithms is the vast amount of labeled data which is necessary for proper training. Usually, this data is obtained by manually annotating thousands and thousands of samples. As this takes quite a while and can become very expensive depending on how qualified the labeling experts have to be, the trend goes towards training sets for which the ground truth can be derived automatically. These self-supervision approaches have been explored in several applications, e.g. by using simulated data [6, 7], or by exploiting other modalities like time-series [8]. Therefore, we propose a self-supervised method for pore detection in CT-scans of cast aluminum parts solely based on simulated data. We evaluate three different approaches to leverage the methods that have proven to be promising for two-dimensional image processing tasks to three-dimensional data. To validate the models, we evaluate their prediction performance on three hand-labeled CT-scans of cast aluminum parts. Further, we evaluate the models on hard cases like composed parts of materials with very heterogeneous absorption and additively manufactured parts with smaller defects.

2. Related Work – Image Processing In Medical Science And Industry

The crucial point for a well trained machine learning system is a sufficient amount of labeled data. For single radiographs the GDXray data set [9] contains radiographs and annotations for many different applications. The part relevant for pore detection comprises a few radiographs of rims, brakes, and welds with bounding boxes of defects as ground truth. So far, there is no publicly available data set of CT-scans with annotated pores, which especially inhibits training deep learning algorithms for this task. However, the precise detection of pores in three-dimensional CT-scans can be extremely useful, e.g. for further consideration in pursuing analyses like wall-thickness analyses and stress test simulations. A first learning-based approach operating on three-dimensionsal data is specialized in the detection of air and gas inclusions [10]. This method is based on a traditional machine learning workflow which involves candidate selection, the time-consuming extraction of hand-crafted features, and the subsequent classification in a machine-learning algorithm. To unleash the full potential of CT-scans, we need to process a huge amount of data in short time, though. That means the detection methods have to be sufficiently fast. Here, deep learning is more suited as it combines candidate selection, feature extraction, and classification in a single module, using highly parallelizable operations.

Another field of application of CT is medical imaging. Because of the limitation in power and exposure, these CT-scans only have the quality that is really necessary for a diagnosis.
Nevertheless, deep learning is successfully used to segment different parts of the human body like bones [11, 12], organs which consist of similar tissue [13, 14, 15, 16], or even individual tracts and neurons of the human brain [17, 18]. Moreover, CNNs are trained to detect defects in the human body, e.g. brain tumors [19] or lesions in vital organs like the liver [20]. In the industrial sector, first approaches use CNNs, for example, to segment individual short glass fibers in CT-scans of small samples of fiber composite materials [21]. Apart from this, generative deep learning methods are utilized to prepare CT-data for a subsequent inspection by a human expert. These methods are designed to remove common artifacts from the data, e.g. Compton scatter [22] or metal artifacts [23]. However, such pre-processing steps alter the original data and potentially induce unwanted side-effects like removing tiny structures along with the artifacts. Thus, we propose to head right for the actual task and learn to deal with image artifacts instead.

3. The Data Set – About Simulations And The Real World

When annotating casting defects in industrial CT-data we are confronted with several peculiarities complicating the task: (1) We need to handle three-dimensional data on a two-dimensional device, which makes it hard to grasp three-dimensional structures. Furthermore, in contrast to medical data, we do not have a specific model of the object under examination. Neither have the defects similar shapes like a liver always looks more or less the same, nor are they always in the same region. (2) CT yields artifact-prone data. In other domains, such as road scene segmentation, we only have to deal with slightly cluttered images. When searching for defects in CT-scans, however, we need to find tiny structures in images with high noise levels and other artifacts arising from the physics of the imaging process. (3) The structures we are looking for are very tiny. In common image processing data sets, small objects still are about 10 pixels large. Instances which are considered as too small belong to the background or an “undecided” class. However, we are exactly looking for anomalies which contain only very few image elements. (4) In different industrial branches different structures are considered as defective. We, however, want to find all defects, which are physically present in the part. Smaller defects are especially hard to spot and require scans of high image-quality.

Therefore, we use simulated data with precise per-voxel ground truth [24] for training and the small amount of hand-labeled real data we have for evaluation only. The training set consists of 675 simulated CT-scans of cast aluminum parts with defects, like pores or
shrinkage, of varying shape and size. For each of the 25 parts there are 27 CT-scans varying the artifact strength of noise, beam hardening, and ring artifacts. This spans an “artifact subspace” within the space of possible scan parameters, which has three gradations along each axis. Further, the data set comes with 54 additional CT-scans for validation. Figure 1a shows an example of the simulated data set with different artifacts and the according ground truth.

However, simulations are only an approximation of the real world and even though the data is considered to be realistic enough to serve as training data, we need to validate our models on real data to ensure that we do not overfit to the simulations. Therefore, we have three real CT-scans which were hand-labeled by eight experts. To ease the annotation process we make a high-quality scan of a defective cast aluminum part; then we scan the same part again under “normal” conditions to obtain a scan for evaluation holding the challenges for the trained defect detection models. Figure 1b shows an example of the real data and the corresponding hand-labeled ground truth.

4. Methods – A Comparison of Neural Network Architectures

The success of deep learning for classification and segmentation has its origin in the processing of two-dimensional images and time-series, i.e. videos. We, however, are dealing with three-dimensional volumetric data. When moving from two dimensions to three, there are several ways to do so: we can (1) process the volume slice by slice along a principal direction [18], (2) evaluate for each voxel the three slices which are aligned along the three axes [12], (3) process the volume slice-wise along each dimension and learn how to combine the results [17], or (4) make the smallest operation—the convolutions—three-dimensional [13]. In this section, we compare three model architectures: a slice-wise model following approach (3), a similar model but with three-dimensional convolutions following approach (4), and a flat model with a different kind of context aggregation, following approach (4), too.

As we have way less “defective” voxels than “non-defective” ones, we have to deal with a highly imbalanced training set. Therefore, we train all the models using dice loss [25] which does not reward correct “non-defective” predictions and thus puts the focus on the defects. Further, we distinguish between defects which penetrate the surface of the scanned part and defects which completely lie within the part. With this separation, we avoid false positive predictions on the surface of the part, for example in screw threads. This diminishes the need to segment the part from its background beforehand.

In the wild, different CT-scanners work with a huge variety of scan parameters which not only affect the amount of artifacts in the image, but global image properties like the contrast as well. To be able to deal with the different scan modalities, we heavily use data augmentation in addition to the artifact variation of the training set. We specifically focus on augmenting the brightness, contrast and noise of our data.

4.1 Slice-wise Computation

First, we examine a slightly modified U-Net architecture [18]. It consists of a down-sampling part culminating in the bottleneck layers and a symmetric up-sampling part producing an output of the same size as the input. Between these two parts, which form the eponymous U-shape, skip-connections are added to provide information from earlier layers for a detailed up-sampling. We use strided convolutions instead of maximum pooling layers to reduce the spatial resolution, which has a positive effect on the computational effort. During training, we randomly pick axis-aligned slices from the volumes in the training set. For prediction, we slide a window along each axis through the test volume and later combine the individual results using a simple maximum operation. Alternatively, it is possible to train a separate
model to combine the individual results, e.g. with the help of the original gray-values [17]. However, a separate classifier means more computational cost. Thus, we go for the maximum operation which already yields satisfying results in our two-class scenario. This method will further be referred to as “slice-wise detector”.

4.2 Three-dimensional Convolutions

The second architecture basically is a three-dimensional version of the U-Net, as it is used in [13, 15]. Additionally, we add a refinement step which is introduced in [26] to segment thin and semi-transparent structures in images. We utilize it to refine the boundaries of the detected defects. The complete architecture is introduced in [24]. During training, we randomly pick small blocks from the training volumes, as they do not fit on the GPU in one piece. For faster convergence it is helpful to pick only blocks containing defects or parts of the surface, because they contain the most information. Fully convolutional networks (FCNs) do not rely on a fixed size input. So, we can vary the block size depending on the capability of the hardware. We only need to take care that the blocks overlap during prediction by half of the receptive field of the model so that the model has the full information for the inner part of each block. This method will further be referred to as “deep defect detector”.

4.3 Dilated Convolutions

The two models above gather contextual information by reducing the spatial resolution of the data and subsequently upscale the results. These down-sampling steps might drop information necessary to detect tiny defects. A different way of context aggregation from dilated convolutions [27] which use dilated convolution kernels (see Figure 2a). A similar approach is used to detect small structures in remote sensing images [28]. The challenge when applying three-dimensional convolutions without any sub-sampling is the massive memory consumption. Therefore, we have to be content with less layers and less channels per layer. Thus, we only have nine convolutional layers. The full architecture is shown in Figure 2b. This method will further be referred to as “dilated defect detector”.

4.4 Evaluation

We evaluate the prediction performance of the models on the separate validation set and the three hand-labeled real CT-scans of [24]. For a holistic analysis, two measures are taken into account: On the one hand, we evaluate the quality of the segmentation mask using the Jaccard-index [29], or intersection over union (IoU) measure, (see Figure 3a) and, on the other hand, the probability of detection (POD) [30] to determine how many of the actual defects we find (see Figure 3b).

The differences in the precision of the segmentation—shown by the IoU—are not significant. However, in the probability of detection a gap between the models can be observed.
The “dilated defect detector” suffers from insufficient depth and channels and, therefore, still fails to detect small defects. For small defects, the “slice-wise detector” and the “deep defect detector” yield comparable results. For larger defects, the “deep defect detector” slightly benefits from the three-dimensional context, until the defects become big enough so that a combination of all three swipes of the “slice-wise detector” yield comparable results again.

5. Discussion – Advantages and Risks

Finally, we provide some promising results of the “deep defect detector” for three especially challenging scenarios to demonstrate the capability of deep learning methods as well as the risks involved. (1) We show that the results of our “deep defect detector” is less sensitive to changes in the image quality compared to traditional filter-based approaches; (2) we show that, having a scan of given image quality, we can find smaller defects compared to traditional filter-based approaches—without introducing disturbing false positive predictions; and (3) we show that a pre-trained model can be adapted easily to more difficult scenarios. The task of defect detection is not limited to aluminum. What happens if we need to detect defects in a less absorbing material of a multi-material part? To provide an example, we have a CT-scan of a part made of three materials and want to detect defects in the material that has a way smaller attenuation coefficient than the enclosing aluminum.

Trade image quality for speed. As mentioned before, CNNs are able to yield consistent results despite the presence of image clutter. Moreover, in [24] it is shown that the impact of noise on the results of the deep learning method is not significant compared to the other methods. So, it should be possible to save time by reducing the scan time and accepting the resulting increase in image noise. To validate this, we scan a cast aluminum part making 3300 projections with an exposure of 1 s each. Then, we reconstruct the volume several times using a decreasing amount of projections down to only 33 projections. This comes up to a scan time of half a minute compared to the original 55 minutes. The scanned cast aluminum part has a diameter of about 12 cm and a height of about 4 cm. The scan was done at 175 kV with 30 µA. Despite noise, we introduce artifacts due to under-sampling to the image, as we do not consider enough projection angles. The top row of Figure 4 shows a slice of the resulting data in varying quality.

Then, we use the filter-based method from [24] and the “deep defect detector” to find pores in the data. The results are shown in the second and third row of Figure 4, respectively. The filter results get brittle when the quality drops. Moreover, tiny false positive responses
emerge. In contrast, until about 110 projections, the results of the “deep defect detector” remain consistent without introducing further false positive responses. The detected defects are recognizable as individual instances even in the image of the worst quality. However, some of the smaller defects become missed out. After that, the neural network starts to hallucinate false positives in the streaking artifacts. The benefits become clearer when evaluating the IoU measure (Figure 5): The results of the filter-based method drop rapidly as the number of projection decreases, the results of the “deep defect detector” remain almost constant.

That hand labeling pores in CT-scans of cast aluminum parts is a hard task can be seen in Figure 6. In the upper row, in red, we see a structural loosening which is annotated as defect but was not recognized as such. However, in the bottom row a similar structural loosening—which this time was recognized as defect—was not annotated as defect (colored in blue indicating false positive predictions). These inconsistencies lessen the overall numbers of the evaluation. Thus, using hand-labeled data for training can be problematic.
Find tiny defects. In the additively manufactured parts of the aerospace industry much smaller pores have to be detected than in the cast aluminum parts of the automotive industry. So, we evaluate the “deep defect detector” on an additively manufactured part looking for tiny pores and again compare it to the traditional filter-based method. Figure 7a shows the quantitative results by means of the POD at a confidence level of 90% per defect. We see that the “deep defect detector” starts to yield reliable predictions (of more than 50%) for defects of an equivalent sphere diameter of 3 voxels, while the filter based method needs about 5 voxels. This difference can be explained by looking at the qualitative results in Figure 7b. In the top row we see that the filter-based method does not catch all the defects. If we configure it to be more sensitive more false positive responses emerge. The “deep defect detector”, in contrast, detects more of the small defects and separates them more precisely from the material—without introducing more false positive responses. The bottom row shows that the filter-based method introduces many tiny false positive responses which are especially irritating when looking for tiny pores. Another advantage of the “deep defect detector” is that we do not necessarily need to segment material and background to restrict the analysis area.

Fine-tune for different data. Lastly, we consider some data which widely differs from anything we have in the training set. To see what happens when we encounter CT-scans of parts made not solely of aluminum, we have a look at a CT-scan of a thermal sensor which is composed of three materials with very different absorption properties: A hollow aluminum cylinder filled with wax, some kind of plastics, and another aluminum cylinder. To draw conclusions about the reliability of the part we need to detect all pores and cracks in the wax part, which has the smallest attenuation coefficient. The leftmost image of Figure 8 shows a slice of the CT-scan and a detailed view with a few highlighted defects.

When training CNNs for different image classification tasks, especially the early layers of a CNN tend to learn similar convolution kernels. These resemble Gabor-like filters or look like color-blob detectors. Sharing these weights across different tasks not only reduces the convergence time but can increase the accuracy of the models [31]. Therefore, we use the pre-trained “deep defect detector” as basis for fine-tuning and adapt it to the new problem by re-training the refinement step of the model. With the simulation pipeline from [24] we create a single CT-scan similar to the thermal sensor with about 500 defects. Adapting the pipeline to the new scenario requires only little manual effort. As the pipeline provides us with precise labels for the defects, too, we can use it as training input for the fine-tuning process without hesitation. The results of the fine-tuned model are shown in the rightmost image of Figure 8 and they significantly improved to the original “deep defect detector” (middle image).
Figure 8: If we encounter data which widely differs from the training data, like the multi-material part on the left, the original “deep defect detector” fails to yield any reliable results (center image). However, when fine-tuning on a single simulation similar to the target, we can obtain better results again (right image). The orange outlines on the left indicate what we consider to be a defect.

As significant the results are, as dangerous the process of fine-tuning can be. We dramatically change the behavior of the model with the input of a single data set. We know that human annotators are not always consistent in their decision. Thus, we assume, if we fine-tune the model based on the decisions of a human inspector, e.g. in an in-line scenario, a bad day can have a tremendous impact on the prediction performance. It might even corrupt the whole system and all the effort would be in vain. We, however, use simulated data for training and fine-tuning. So, we can be sure to have precise and correct labels. Furthermore, as it is generated by a fully automated pipeline we only need to input a new model and the properties of the materials involved. This reduces the necessary manual effort to a minimum.

6. Conclusion

Deep learning methods are able to deal with noisy and artifact-polluted data and to detect smaller defects than filter-based approaches—if trained properly. For other tasks, fine-tuning might be necessary, which has to be done carefully. Already a small amount of data has a huge impact on the prediction performance. Of most importance for reliably trained machine learning algorithms is a precisely labeled ground truth. This might not be given if untrained human annotators produce new data in-line, but can be generated on the fly for simulated data with little effort. We demonstrate that deep learning methods are suitable for non-destructive testing by showing their capability to detect defects not only in CT-scans of cast aluminum parts but in those of additively manufactured parts, too. Further, we demonstrate that the good results are transferable to more complex scenarios, i.e. the testing of composed parts.

References
