



UNIVERSIDAD NACIONAL DE COLOMBIA

Automated defect detection approach for production processes of patterned steel plates using computer vision and deep learning

Claudia Kicker

Universidad Nacional de Colombia
Faculty of Engineering, Department of Mechanical Engineering and Mechatronics
Bogotá, Colombia
2022

Automated defect detection approach for production processes of patterned steel plates using computer vision and deep learning

Claudia Kicker

Graduation work presented as partial requirement to obtain the degree:
Master of Mechanical Engineering

Director:
Ph.D. Flavio Augusto Prieto Ortiz

Area of Research:
Automation, Control and Mechatronics

Universidad Nacional de Colombia
Faculty of Engineering, Department of Mechanical Engineering and Mechatronics
Bogotá, Colombia
2022

Dedication

A special thanks to my family for their unconditional support on this long journey.

Acknowledgements

I would like to express my gratitude to:

The IMS Messsysteme GmbH for the great opportunity of realizing this work in the industrial context.

My director Flavio Prieto for his guidance as well as his professional support and precious technical advices.

Profesor Ricardo Ramirez for his considerable help in the admission process, which paved the way for my professional career at the National University of Colombia.

The University of Duisburg-Essen, especially the Chair of Intelligent Systems, who brought me in contact with the research topic and supported me scientifically throughout large parts at the beginning of the project.

My family for their unconditional support in presence and at great distance, especially my mother Lilly and my father Klaus who encouraged me to pursue my goals whenever I was doubting.

My colombian family Kevin, Lorena and Aleida for having been an indispensable part of my life in Colombia and for having become a second family to me.

Resumen

Método para la detección automatizada de defectos en la producción de láminas de acero alfajor mediante visión artificial y aprendizaje profundo

La detección de anomalías es de gran importancia en la producción de placas de acero para garantizar que los productos no tengan defectos. En los últimos años han surgido varios métodos de aprendizaje profundo para la detección de defectos en superficies de acero limitándose principalmente a superficies de acero planas. Además, la detección de anomalías basada en el aprendizaje profundo sigue siendo una tarea difícil si no se dispone de suficientes muestras de entrenamiento, lo que suele ocurrir en escenarios del mundo real. En cuanto a las placas de acero texturizadas, como las láminas alfajor, la disponibilidad de muestras anómalas es baja, ya que las producciones están optimizadas para minimizar la aparición de defectos. Por lo tanto, el objetivo principal de este trabajo es la determinación de un método adecuado basado en el aprendizaje profundo, para la detección de anomalías superficiales en placas de acero texturizadas. Se entrenaron varios modelos, los que se compararon en términos de capacidad de segmentación y precisión de clasificación. Por un lado, se adaptó una red neuronal convolucional pre-entrenada en defectos artificiales a imágenes procedentes de una línea de producción diferente, de la que solo se disponía de datos libres de anomalías para su entrenamiento. Por otro lado, se entrenó un autocodificador de forma semi-supervisada para reconstruir imágenes libres de anomalías, con el fin de identificar las regiones defectuosas midiendo el error de reconstrucción. Además, se realiza un análisis del espectro de frecuencias para las imágenes de placas de acero texturizadas bajo la aplicación de la transformada discreta de Fourier. Se descubrió que un autocodificador de reconstrucción entrenado con una función de pérdida que mide la similitud estructural, proporciona las localizaciones más precisas de las anomalías superficiales.

Palabras clave: Aprendizaje profundo, detección de anomalías, autoencoder, CNN, similitud estructural.

Abstract

Automated defect detection approach for production processes of patterned steel plates using computer vision and deep learning

Anomaly detection is of great importance in the production of steel plates, in order to guarantee that the products are defect-free. Various deep-learning approaches for defect-detection in steel surfaces have emerged in the recent years, however, they are mainly limited to plain steel surfaces. Furthermore, deep-learning-based anomaly detection is still a challenging task if not enough training samples are available, which is often the case in real world scenarios. As for patterned steel plates, the availability anomalous samples is low, as productions are optimized to minimize the occurrence of defects. Hence, the main purpose of this work is the determination of a suitable deep learning-based method for the detection of surface anomalies in patterned steel plates. Several methods were trained and compared in terms of segmentation ability and classification accuracy. On the one hand, a convolutional neural network pretrained on artificial defects was adapted to images from a different production line, of which only anomaly-free data was available for training. On the other hand, an autoencoder was trained in a semi-supervised fashion to reconstruct anomaly-free images, in order to identify defective regions by measuring the reconstruction error. Moreover, an analysis of the frequency spectrum for images of patterned steel plates under the application of discrete fourier transform is provided. It was found out that a reconstructing autoencoder trained with a structural similarity loss provided the most accurate localizations of surface anomalies.

Keywords: Deep learning, anomaly detection, autoencoders, CNN, structural similarity

Contents

Acknowledgements	vii
Abstract	ix
List of Symbols and Abbreviations	xv
1 Introduction	1
1.1 Motivation	1
1.2 Problem formulation	2
1.3 Aims	3
2 Background	5
2.1 Introduction to the production of patterned steel	5
2.2 Basic Concepts	6
2.2.1 Computer vision	6
2.2.2 Machine Learning and Deep Learning	7
2.2.3 Convolutional Neural networks	7
2.2.4 Autoencoders	7
2.2.5 Anomaly Detection and Surface Defect Detection	8
2.2.6 Anomaly Detection Techniques	8
2.3 Literature review	9
2.3.1 Supervised approaches	9
2.3.2 Unsupervised approaches	10
2.3.3 Semi-supervised approaches	11
2.4 Previous work	12
2.4.1 Data preparation	12
2.4.2 Architecture	13
2.4.3 Evaluation of previous work	16
3 Methodology	19
3.1 Available Data	19
3.2 Methods	21
3.2.1 Transfer learning on U-Net	21
3.2.2 L2 and SSIM Autoencoder	21

3.2.3	Excursus: Analysis of Frequency Spectrum	24
3.3	Metrics	26
3.3.1	True Positive Rate (TPR)	27
3.3.2	True Negative Rate (TNR)	27
3.3.3	False Positive Rate (FPR)	27
3.3.4	Accuracy (ACC)	28
3.3.5	AUROC	28
3.3.6	Intersection over Union (IoU)	28
3.4	Implementation Details	28
4	Results and Discussion	29
4.1	Transfer Learning on U-Net	30
4.2	L2 and SSIM autoencoder	30
4.2.1	Variation of threshold for the SSIM autoencoder	35
4.2.2	Variation of the structuring element during post-processing for the SSIM autoencoder	37
4.3	Autoencoder with Fourier Transform	39
5	Conclusions and Future Research	40
5.1	Conclusions	40
5.2	Future Research	41
	Bibliography	43

List of Figures

2-1	Example of two different designs of patterned steel plates, design T and design R according to DIN EN 10363	5
2-2	Schematic representation of the scanning process of patterned steel plates by parallelly installed line scan cameras	6
2-3	Defect labelling	13
2-4	(a) Defect inserted into a patterned steel image (b) Ground truth	13
2-5	CompactCNN network architecture	14
2-6	Combined network consisting of U-Net and the CompactCNN classification stage	15
2-7	Examples of network outputs for CompactCNN and U-Net+CompactCNN. From left to right: Input image, ground truth, CompactCNN prediction, U-Net + CompactCNN prediction	17
3-1	Example images for each dataset from left to right: (a) Teardrop1, (b) Teardrop2, (c) Teardrop3, (d) Teardrop4, (e) Diamond. The top row shows a defect-free, the two bottom rows a defective sample and a close-up of the defective region	20
3-2	Network structure of the L2 and SSIM Autoencoder	22
3-3	Magnitude spectrum in frequency domain for all datasets. Top row: image in spatial domain, Bottom row: frequency spectrum	25
3-4	Structure of Fourier-based Autoencoder	26
4-1	Predictions on dataset 1 of U-Net after transfer learning and the L2 and SSIM Autoencoder	29
4-2	Segmentation examples of the L2 and SSIM autoencoder	32
4-3	ROC-Curves for SSIM and L2 autoencoder	34
4-4	Different segmentation results based on different percentiles for threshold selection	36
4-5	Different segmentation results based on different sizes of the structuring element for post-processing	38
4-6	Results of reconstructing the magnitude spectrum of a patterned steel plate image with an autoencoder	39

List of Tables

2-1	Segmentation results: AUC Score for different datasets and image sizes . . .	17
2-2	Classification accuracies for different datasets and image sizes	17
4-1	Classification results for the SSIM and L2 autoencoder	31
4-2	Segmentation results for the SSIM and L2 autoencoder	31
4-3	Classification results with different percentiles for threshold estimation . . .	35
4-4	Segmentation results: Intersection over Union (IoU) based on different percentiles for threshold estimation	36
4-5	Classification results with respect to varying the radius of the circular structuring element during post-processing	37
4-6	Segmentation results: Intersection over Union (IoU) with respect to varying the radius of the circular structuring element during post-processing	38

List of Symbols and Abbreviations

Mathematical Symbols

Symbol	Description
α	Weighting coefficient for luminance of SSIM
b	Image sample
B	Total number of samples in image batch
β	Weighting coefficient for contrast of SSIM
γ	Weighting coefficient for structure of SSIM
i	Pixel position in vertical direction
j	Pixel position in horizontal direction
K	Patch size
μ	Mean value
M	Number of vertical pixels
N	Number of horizontal pixels
σ	Variance
σ_{pq}	Covariance between p and q
x	Ground truth image
\hat{x}	Predicted image
y	True label
\hat{y}	Predicted label
z	Latent vector size of the autoencoder

Abbreviations

Abbreviation	Description
ACC	Accuracy
AUC/AUROC	Area Under The Curve
CNN	Convolutional Neural Network
FN	False Negatives
FNR	False Negative Rate
FP	False Positives
FPR	False Positive Rate
GAN	Generative Adversarial Network
IoU	Intersection Over Union
LBP	Local Binary Pattern
ReLU	Rectified Linear Unit
ROC	Receiver Operating Characteristics
SSIM	Structural Similarity
SVM	Support Vector Machine
TN	True Negatives
TNR	True Negative Rate
TP	True Positives
TPR	True Positive Rate

1 Introduction

1.1. Motivation

Quality control plays an important role in industrial settings. It aims to verify that the produced goods are defect-free, in order to avoid losses for the company, dissatisfaction of the clients and prevent safety issues.

One typical method of quality control is surface inspection, which is used to find superficial irregularities, such as cracks, scratches or dents [36]. Traditionally, such quality related inspection tasks are often carried out by human workers who are trained on detecting complex surface anomalies [46]. But investigations have shown that automated solutions also have great potential [13] [57]. The benefits of automated solutions are numerous: On the one hand, surface inspection tasks are often monotonous and tiring. Hence, human inspection is susceptible to errors, as defects can easily be overlooked due to fatigue or distraction. In fact, observations have shown that humans present a failure rate of 20 % to 30 % in complex visual inspection tasks [42]. Automated systems are not influenced by these and other individual factors, such as the inspector's experience, visual acuity or personal condition. On the other hand, automated systems have a 24/7 availability. Once implemented, they can run for many years and save running costs.

Hence, it is not surprising that a lot of research has been carried out in the area of automated defect detection. Many approaches focus on the analysis of 2D images of produced goods [23] [28] [47] [54]. Therefore, the products are scanned with camera systems and afterwards processed automatically by means of computer vision, a sub-discipline of machine learning, that deals with the computational processing and analysis of digital images in order to understand their content.

Traditional computer-vision methods, including structural, statistical, filter-based or model-based methods [53], focus on the manual extraction of significant features. The selection and application of these techniques require a lot of domain knowledge and a profound understanding of the process.

More recently, deep learning architectures, which are based on neural networks, have become a very popular tool for visual detection problems and an alternative to manual

feature extraction techniques, as these architectures are able to learn relevant features autonomously. Their growing popularity was enabled by the increasing computational capacities of the recent years. Especially Convolutional Neural networks (CNNs) are widely used in image processing tasks. Several investigations have shown that neural network approaches can achieve promising results in image processing tasks or even outperform traditional methods [49] [52].

One industrial sector, in which computer vision-based defect detection solutions are applied, is the steel industry, for example for the inspection of steel strips and steel plates. The German company IMS Messsysteme GmbH develops and manufactures camera systems which monitor such steel production processes. Their systems are already capable of detecting defects in plain steel plates, but are not yet developed to analyse structured surfaces, such as patterned steel plates. Also in the scientific literature, the focus is mainly drawn on plain steel surfaces, for which there can be found numerous publications (e.g. [13] [18] [20] [26] [43] [27] [57]), whereas there is still a lack of information about automated defect detection solutions for patterned steel.

This is taken as motivation to investigate the applicability of computer-vision techniques to images in patterned steel plates, in order to detect anomalies in their surfaces. Hence, the goal of this thesis is the development of a computer-vision based solution, which can detect such anomalous regions in patterned steel plates. Following the current state of the art, the focus is hereby set on deep-learning architectures.

The success of such deep learning methods is highly dependent on the availability of suitable data. In anomaly detection tasks, a large image data set which contains both defect-free and anomalous examples is the optimal case. However, in industrial defect detection problems, the availability of such data is often limited, as the companies try to minimize the appearance of defects. Hence, anomalous data is often scarce. For the realization of this work, an image data set of patterned steel plates was collected, which faces the same problem, containing almost no defective image samples. In order to overcome this problem, the main focus of this work is the implementation of a semi-supervised deep-learning method, which can be trained without anomalous data.

1.2. Problem formulation

Currently, automated defect detection in patterned steel plates, such as diamond plates, is still a challenging task. This is due to their complex surface structure, for which existing algorithms for plain steel defect detection based on gray level analysis methods and thresholding can not be directly applied. Inhomogeneous lighting and dirt particles on the camera lens complicate automatic detection even more. But recent investigations have

shown that deep-learning algorithms obtain promising results in the area of texture analysis [36] and anomaly detection [13] [18]. Inspired by this, a previous project closely related to this work was realized by the author as master's thesis at the University of Duisburg-Essen to develop a supervised deep-learning approach for defect detection in diamond plates.

However, the approach presents one great difficulty: the developed method is a supervised deep-learning method, which depends heavily on the availability of both defective and defect-free image samples. This is a problem, as defective data is difficult to obtain. The reasons are, on the one hand, that defects occur at very rare intervals and, on the other hand, that the data captured by the camera systems is not necessarily saved for later use, as this would require a lot of computational storage capacity. Consequently, the probability that recording is activated precisely when a defect occurs, is so small, that the data sets lack of sufficient defective images. In the previous work a workaround was found by creating defective data synthetically. However, this solution is not ideal, as the creation of suitable defective images is a laborious and time-consuming process that may require manual adjustment to make the defects look realistic. It also complicates the adaptability to other production lines, for which usually also no defective data is available and a synthetical creation of defect data for each unknown production line is little practical. Furthermore, it is difficult to verify how good the artificial defect samples represent real defects and hence, if they are an adequate instrument to train a system that is to be applied in a real world scenario.

Consequently, the focus of this work is drawn on finding a deep-learning algorithm for patterned steel plates with a good adaptability to new production lines, which does not depend on the creation of synthetical defect data for each production line. In this case, semi-supervised deep learning methods, which do only require normal but no anomalous data for training, are a suitable instrument. Inspired by publications on other anomaly detection problems [5] [4] [25] several deep learning methods were elaborated and the performance on anomaly detection in patterned steel plates evaluated with the aim to integrate the best solution into existing vision-based defect inspection systems of the IMS Messsysteme GmbH.

1.3. Aims

This work aims to contribute a comparison of deep-learning based computer vision solutions, which can automate the defect detection in patterned steel plates. Hence, this work provides a promising approach for the optimization of quality control in steel production. Several architectures were investigated and evaluated. In the industrial context, this is interesting for the companies that desire to improve their production

processes. From the investigational point of view, the aim of this work is to present advances in semi-supervised deep learning, exploring suitable methods on a patterned steel product, which is so far not widely investigated. To reach this goal, this work presents the following contributions:

General objective: Development of deep-learning based solution for the defect detection in patterned steel plates, which is easily adaptable to unknown data of different production lines and only depends on anomaly-free data.

Specific objectives:

- Selection, implementation and evaluation of different deep-learning approaches:
 - Transfer learning with only anomaly-free data of a pretrained supervised architecture,
 - Investigation on semi-supervised deep-learning architectures, which can be trained on exclusively defect-free data from scratch,
- Analysis and optimization of the investigated approach by varying the deep learning parameters,
- Systematic evaluation of the different approaches.

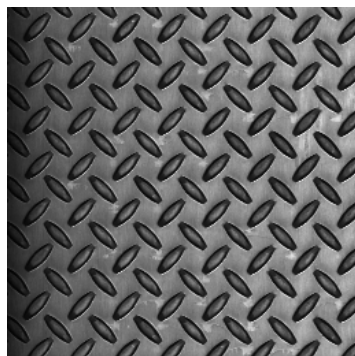
2 Background

In this chapter, some relevant background information related to defect detection in patterned steel plates is provided. A short introduction to the production process of patterned steel is given. Furthermore, a literature review provides an overview over significant work that has already been published in the field of surface defect detection and the concepts of supervised, semi-supervised and unsupervised learning are introduced. Finally, a general summary of preceding work related to defect detection in patterned steel plates carried out by the author of this work, as master's thesis for the University of Duisburg-Essen, is presented at the end of this chapter.

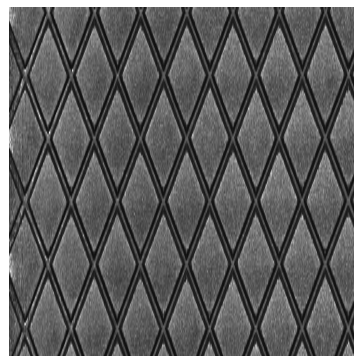
2.1. Introduction to the production of patterned steel

Patterned steel plates, also known as diamond plates, checker plates or floor plates, are a steel product with diamond or teardrop shaped elevations. This texture gives the plates their anti-slip properties, for which they are often used as anti-slip floors.

According to the German norm DIN EN 10363 [11] “Continuously hot-rolled patterned steel strip and plate/sheet cut from wide strip - Tolerances on dimensions and shape”, there exist three designs of steel diamond plates, with either a diamond/rhombus pattern (type R) or a teardrop-shaped pattern (type T and A, of which one has teardrops with sharp and the other with flattened corners). An example for both a teardrop patterned and a diamond patterned plate is given in Figure 2-1.



(a) Teardrop pattern



(b) Diamond pattern

Figure 2-1: Example of two different designs of patterned steel plates, design T and design R according to DIN EN 10363

The steel plates are usually produced in a hot rolling process, during which steel is pressed into the desired shape in by a rolling mill at high temperatures. In such processes, steel plates with lengths of several hundred meters or even various kilometers length can be produced continuously. The systems of the IMS Messsysteme GmbH are designed to be installed as part of the production line, in order to scan the plates directly after rolling. The plates are passed through the camera system and can be scanned from both the top and the bottom. Several line scan cameras are installed in a row in order to capture the whole width of the steel plates. Figure 2-2 shows a simplified schematic representation of the scanning process with several cameras scanning the top side of a patterned plate.

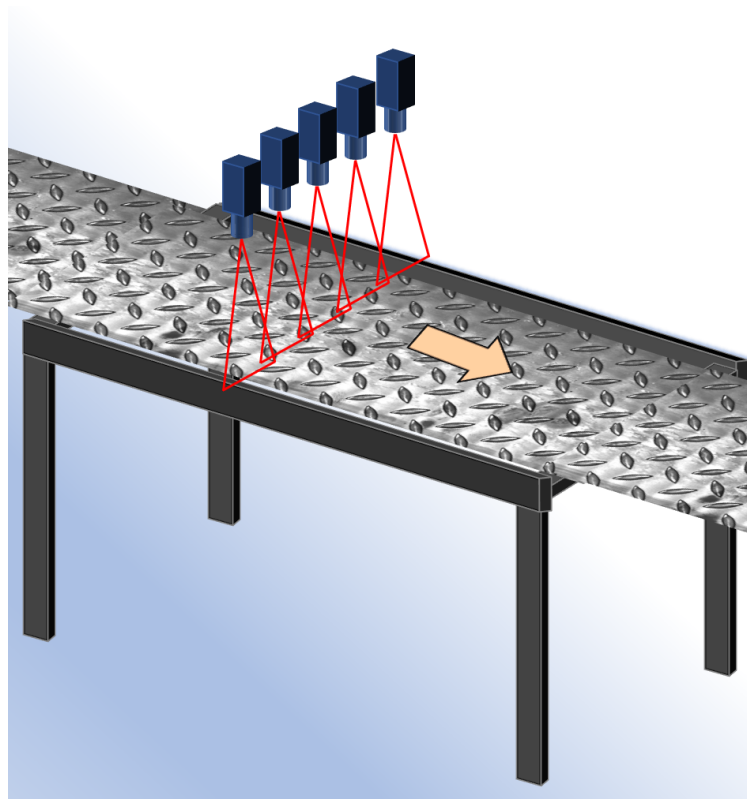


Figure 2-2: Schematic representation of the scanning process of patterned steel plates by parallelly installed line scan cameras

2.2. Basic Concepts

2.2.1. Computer vision

Computer vision is a scientific field that deals with the extraction of meaningful characteristics from images or videos [10]. Traditional computer vision methods focus on the manual feature extraction from the images. The extracted features, also called

hand-crafted features, can then be combined with classifiers like SVM [45], decision trees or random forest classifiers [7]. Many modern approaches apply artificial neural networks, which are able to learn relevant features and classification tasks automatically.

2.2.2. Machine Learning and Deep Learning

Machine Learning can be described as an application of artificial intelligence that can “automatically learn and improve from experience without being explicitly programmed to do so” [29]. Within this field, deep learning is a sub-discipline of machine learning, which makes use of deep neural networks in order to solve machine learning tasks. Neural networks which are considered as “deep”, consist of multiple trainable layers. This enables the network to learn highly complex characteristics and find complex patterns and structures in the exploited data [14]. Some typical applications of deep learning are speech recognition, autonomous driving, fraud detection [8] or computer vision including defect detection [5].

2.2.3. Convolutional Neural networks

Within computer vision tasks, Convolutional Neural Networks (CNNs) are a widely used network type, being named after the convolutional layers they are composed of. These layers consist of a trainable kernel, which is stridden over the layer’s input, e.g. a 2D or 3D image, producing an output called feature maps [35]. Their kernels allow the network to extract meaningful features while maintaining the number of trainable parameters relatively low in comparison to other layer types such as fully connected layers. A CNN can be used for different image processing tasks, for example image classification or image segmentation.

Classification networks: Classification networks are used to classify the images into different categories or classes by assigning a label or classification score to each input image.

Segmentation Networks: Segmentation networks are used to localize objects in an image. In contrast to classification networks, which output a single class label for the whole image, the segmentation network returns a mask that labels every pixel of an image and assigns it to a certain class [35].

2.2.4. Autoencoders

Autoencoders are a special type of neural network that consist of an encoding and a decoding part. The encoder is used to create a compressed representation of the input data, usually for dimensionality reduction [1]. The decoder tries to reconstruct the original

input from the reduced encoded representation. Both the encoder and the decoder can consist of multiple trainable layers. If the network layers are convolutional layers, the network is called Convolutional Autoencoder (CAE) [56]. They are commonly used in image segmentation or image reconstruction.

2.2.5. Anomaly Detection and Surface Defect Detection

Anomaly detection describes the process of identifying untypical data points whose characteristics differ from the normal data [58]. Surface defect detection, on the other hand, describes the identification of anomalies in the surface structure of materials or objects. Many surface defect detection methods focus on computer vision. While some approaches are based on neural networks, others use more traditional techniques, where significant features are extracted manually instead of being self-learned by the system [53]. Surface defect detection can be considered a type of anomaly detection, when classifying into defective and defect-free. Other approaches aim to distinguish between different types of defects [13] [18].

2.2.6. Anomaly Detection Techniques

Depending on the available data, anomaly detection techniques can be roughly classified into three groups, namely supervised, semi-supervised and unsupervised methods [8]. In anomaly detection, a data sample can be either normal or anomalous.

Supervised: Supervised algorithms for anomaly detection require labelled training data of both the normal as well as the anomalous class. This means, that for each training data sample the desired result is known and the network is trained on predicting a result that is as close as possible to the desired result.

Unsupervised: Unsupervised learning techniques do not require any labelled data. This has the benefit that the laborious labelling process can be avoided. The system is trained on data samples of which it is not known if they are normal or anomalous, however, making the assumption that normal samples are much more frequent than anomalous samples [8].

Semi-supervised: Semi-supervised methods depend only partially on labelled data. In the context of anomaly detection, it is often assumed that there is only labelled normal data, but no labelled anomalous data [8]. Such as unsupervised learning, these techniques are useful if anomalous data is hard to obtain, which is often the case in industrial use-cases.

2.3. Literature review

Automated defect detection in industrial settings has been widely investigated [53] and applied to many use cases, such as surface inspection of LED chips [24], wood [38], textiles [17] [33], or metals [54]. Within the latter, Neogi et al. [32] provide an overview over several defect detection and classification methods for the steel industry.

2.3.1. Supervised approaches

A great variety of different approaches has been proposed for supervised anomaly detection in steels and other materials. While Paulraj et. al [34] provided a technique for crack identification in steel plates based on vibration analysis, many other methods focus on computer-vision [13] [18]. An Inception-V4 [9] based CNN architecture for the classification of different defect types in hot rolled steel plates and steel strips was presented by He et al. [13], achieving classification results of around 95%. Zhou et al. [57] proposed a very simple network combining convolutional and pooling layers to classify self-collected images of seven defect types in hot-rolled steel sheets, achieving classification error rates of just 0,63%. Another approach, based on Local Binary Patterns combined with nearest neighbor and SVM classifiers was developed by Song et al. [43]. They introduced the NEU dataset, which contains images of six defect types from hot rolled steel plates.

The same dataset was used by He et al. [18], who extracted features of different scales from various layers of a pretrained CNN architecture and combined the features in a neural network they called multifeature network. In contrast to the previous mentioned approaches, their method is not only used for the classification of images, but also to segment the defect areas within the images. Another deep learning method for both segmentation and classification that was tested on the NEU-dataset, was based on a CNN architecture named DeCaf [38] and achieved classification accuracies of 99.27%. Huang et al. [20] used a dataset named SD-saliency-900, which contains 900 images of three defect types in hot rolled steel strips, to test a CNN with depth-wise convolutions for segmentation based on U-Net, an architecture that was introduced in a biomedical context by Ronneberger et al. [39] for the segmentation of neural cell structures. This architecture has become widely adopted for image segmentation problems, especially in the medical sector [16] [44]. The U-Net with depthwise convolutions achieved good segmentation results of steel defects in comparison with other variations of U-Net.

However, the aforementioned methods were tested on images of plain surfaces. Attempts for anomaly detection on structured surfaces applying CNNs were proposed by Weimer et al. [52], Huang et al. [19] or Racki et al. [36], who tested their models on the DAGM dataset, a synthetically created image collection which contains 10 types of different

textures, each providing normal and anomalous images. Racki et al [36] proposed a two-stage segmentation and classification network, based on a light-weight architecture that requires low computational costs. Their architecture was refined by Tabernik et al [46] and Bozic et al. [6] and tested on a dataset of electrical commutators as well as on another dataset of plain steel surface images named Severstal steel dataset.

Other investigations on textured structures were published for the textile sector. Li et al. [23] combine several small CNN architectures to a wide but compact network, in order to detect typical fabric defects. With another fabric dataset, Zhang et al. [55] tested several network architectures based on YOLO [37], a well known network for image segmentation. A method for defect classification which requires only a small amount of samples was proposed by Wei et al. [51]. However, although the amount of defective samples can be small for some approaches, all of the methods require defect samples for training. In order to overcome this problem, unsupervised or semi-supervised methods can be more suitable.

2.3.2. Unsupervised approaches

Unsupervised approaches, which do not require labelling of images, are often based on autoencoders. In a setting where defective samples are rare or not available, the autoencoder learns a feature representation that can reconstruct the pattern of defect-free samples and fails to reconstruct defective anomalies [25]. An architecture for patterned fabric anomaly detection was proposed by Mei et al. [28], who trained an architecture of stacked autoencoders on different levels of a gaussian pyramid.

In another approach an autoencoder was trained with an optimized loss function, adding some regularizers to it, in order to minimize the distance and restrict the spread range of defect-free samples [47]. It was trained on the MVTecAD dataset, an image collection of 15 sub-datasets of different patterns and objects, such as carpet, grid, bottleneck or screws [3] [2] collected by Bergmann et al. The dataset was tailored to unsupervised anomaly detection. Each sub-dataset consists of solely defect-free samples for training and both defect-free and defective samples and their corresponding ground truths for testing. The creators of this dataset themselves proposed two unsupervised deep learning approaches, which they tested on the dataset. On the one hand, they proposed to train a convolutional autoencoder with a structural similarity loss function, which encourages the network to minimize the differences in mean value, variance and covariance between the input image and its reconstruction [5]. They could show that this method led to improved results in comparison with the widely used L2-loss, which minimizes the squared difference between each pixel of the input image and its reconstruction. On the other hand, they proposed an approach where several student networks learn to predict the output of a pretrained teacher network. It is based on the idea that the student networks, which were only trained on regressing anomaly-free samples, fail

to reconstruct the teachers output for anomalous samples. The authors tested their methods against various unsupervised deep learning approaches [3]. One of these approaches was designed to extract features of good data with a CNN and cluster them in a dictionary, finding anomalies by checking how much a test sample deviates from the learned features in the dictionary [30]. Other approaches proposed by Schlegl et al. [41] [40] were based on General Adversarial Networks (GAN) and originally introduced in the context of medical image processing. GANs consist of a generator and a discriminator, which are trained in an adversarial way, such that the generator learns to create real looking images, while the discriminator is trained to distinguish between real images and the generator’s fake images. In comparison to these network types, both the student-teacher network [4], as well as the SSIM autoencoder [5] performed well in their respective comparative analyses.

2.3.3. Semi-supervised approaches

The MVTec dataset was further adapted to several semi-supervised learning approaches [25] [31] [48], which were also based on autoencoders. Liu et al. [25] designed an encoder-decoder-encoder structure, where the reconstructed image of the decoder is encoded again into a latent vector. A residual of the two latent vectors of both the input image and the reconstructed image was taken as a measure for the presence of anomalies. Wang et al. [48] proposed an autoencoder, where the ideal dimension of the latent space is determined by a probability model. Both Liu et al. [25] and Wang et al. [48] trained their models on solely defect-free data. Napoletano et al. [31] extended the autoencoder structure by interposing a normality pass filter between the encoder and the decoder in order to filter out anomalous features. They detected anomalies by comparing the output of an autoencoder with and an autoencoder without a normality pass filter. The autoencoder was trained primarily on a large independent image database and then adapted on only defect-free samples.

Other approaches apply another concept of semi-supervised learning. Instead of relating it to the presence of only the anomaly-free images, it is related to the kind of training procedure, which is partially executed in an unsupervised manner, and partially in a supervised manner. One example is the classification approach by He et al [12], who trained an autoencoder in an unsupervised manner on defect images in steel strips and steel plates, in order to extract significant features and then use the feature vector of the encoder as a discriminator which can classify different defect types, which was further trained in a supervised manner.

An example from the textile industry was provided by Li et al. [22]. Two stacked autoencoders were trained, one of them on positive and negative samples and the other on only negative samples, using the residual of their outputs for defect localization. The

autoencoders were firstly pretrained in an unsupervised manner before being stacked and fine-tuned on a labelled dataset. Although being designed for use cases with limited defective samples, the availability for at least some defective samples is crucial for this approach.

The above mentioned approaches provide a wide overview over the current state of the art with respect to anomaly detection in both plain as well as several patterned and structured surfaces. The introduction of the MVTecAD dataset as well as the corresponding investigations by Bergmann et al. [5] [3] provide a good baseline for anomaly detection approaches which can even be trained without anomalous images. However, the approaches do not cover investigations on patterned steel plates. For this reason, this work focuses precisely on patterned steel anomaly detection. Inspired by the investigation of Bergmann et al. [3] semi-supervised autoencoders are investigated as optional method. They are compared to the adaption of a supervisedly trained architecture which was developed in some pre-work by the author. It differs from the focus of this work by having been trained on artificially created defect data and is further introduced in the following section.

2.4. Previous work

Prior to this work, a supervised deep learning method for the detection of defects in patterned steel plates has been elaborated. Inspired by the architecture of Racki et al. [36] a CNN-based network structure was implemented that produces two outputs: a segmentation map, which provides a pixel-wise location of defects and a classification score, which labels an analysed image as either defect-free or defective. As only defect-free image material was available, a synthetical data set of defective images was generated by merging error-free images of patterned steel plates with images of defective regions from plain steel plates. A special attention was drawn to periodic defects, which are errors that repeat themselves at a certain interval, for example because of a defect in the cylindrical steel roller. They are of high interest, as these types of errors can damage the whole production batch. The developed network structure showed very good results in terms of defect localization and classification, with a classification accuracy from over 99 % in detecting defects and over 95 % in detecting if their occurrence is periodic. The part of that approach which covers the CNN-based anomaly detection is presented in more detail in the following sections.

2.4.1. Data preparation

For the realization of the work, two image datasets of patterned steel from industrial production lines were available, both of them depicting the teardrop shaped pattern (see section 2.1). The data, which was provided in the form of .avi video files, was cut into image patches of size 512x512 pixels and 256x256 pixels respectively. As the provided data

did not contain defective image samples, defects were inserted synthetically into some of the image patches. For this task, a dataset of defects in plain steel plates was used. The defect regions in the plain steel images were labelled, cut out and inserted into a random position in the patterned steel image. The steps are shown in figures **2-3** and **2-4**. Vertical and horizontal flipping and 180° rotating was applied as data augmentation, such that a total of 1756 defect images could be created. Together with the resulting defect patch (Figure **2-4a**), a ground truth image (Figure **2-4b**) showing the location of a defect was created.

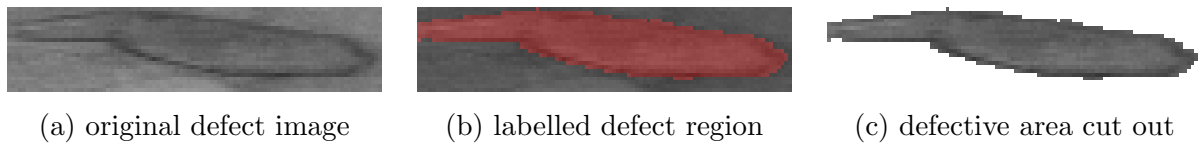


Figure 2-3: Defect labelling

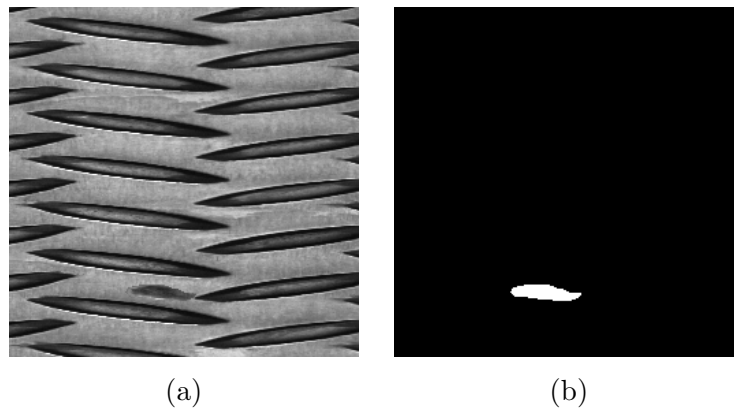


Figure 2-4: (a) Defect inserted into a patterned steel image (b) Ground truth

2.4.2. Architecture

The baseline for the deep learning approach was the two-stage CompactCNN architecture introduced by Racki et al. [36]. The first stage was designed to return a segmentation map which shows the location of a defect in the input image. If no defect is present in the image, the network returns a segmentation map in which all values are close to zero. The segmentation stage consists of nine convolutional layers clustered into three blocks and one additional convolutional layer, which returns the segmentation map. Within the blocks, every convolutional layer is followed by batch normalization and a rectified linear unit (ReLU) activation function. The final layer is followed by a hyperbolic tangent (tanh) activation function. In each block, the number of feature maps is doubled. The height and width of the features is reduced by half in the first and second block by applying a stride of

2 in the block's first convolutional layers. Hence, the total height and width of the output segmentation map is reduced to a quarter of the input image size.

The second stage was designed to return a classification score, which classifies an image as defect-free or defective. Both the outputs from the penultimate and the ultimate layer of the segmentation stage are used as input for this stage. The outputs from the penultimate layer are passed through another convolutional layer which returns 32 feature maps. Global max-pooling and global average pooling is performed on these feature maps and on the output from the last segmentation layer. The results are concatenated and passed through a fully connected layer to return one single classification score. The score denotes the probability that an image is defective, being 0 the lowest and 1 the highest probability. The whole architecture of the CompactCNN is shown in Figure 2-5.

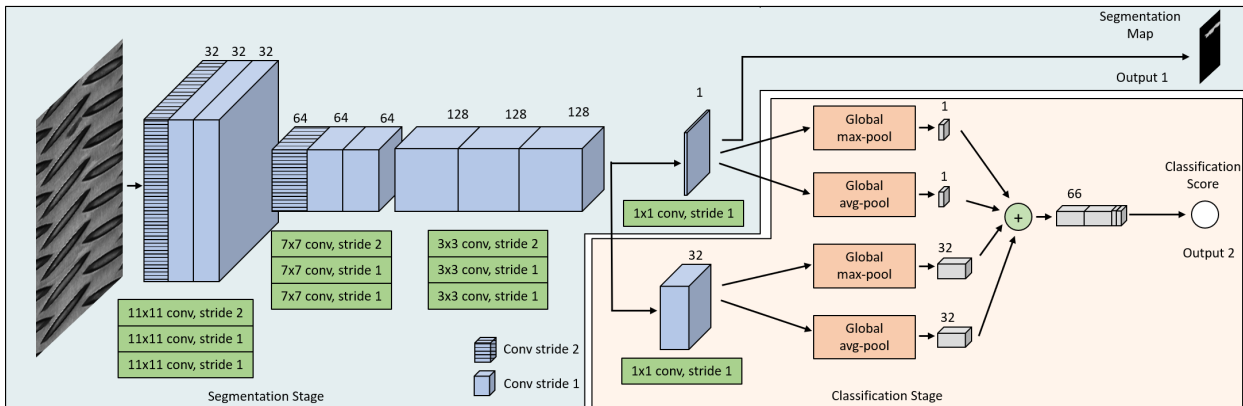


Figure 2-5: CompactCNN network architecture

The original CompactCNN architecture was compared to a modified CompactCNN architecture, in which the segmentation stage was replaced by a CNN called U-Net [39]. This network was originally developed for biomedical image processing, in order to segment neural cell structures. It consists of a contracting path and an expanding path. The contracting path is made of five convolutional blocks with two convolutional layers each. The number of features is doubled in each block, while the height and width is halved by a max-pooling operation with a stride of 2 after each block except for the last one. The four blocks of the expanding path are the counterpart of the contracting path, in which the number of features is halved in each block while their width and height is doubled by a 2x2 upsampling operation. Furthermore, there are skip connections between the blocks of the contracting path and the corresponding blocks of the expanding path. This means that the blocks in the expanding path receive two inputs: the output from the previous block as well as the output from their counterpart in the contracting path. Both inputs are concatenated before being passed into the first convolutional layer of the block. Just as in the

CompactCNN, the blocks are followed by one last convolutional layer that returns one single feature map, which is the desired segmentation map. Each convolutional layer is followed by batch norm and ReLU activation, except for the last layer that uses tanh as activation, just as in the original CompactCNN. A combination of the CNN with U-Net as segmentation network is depicted in Figure 2-6.

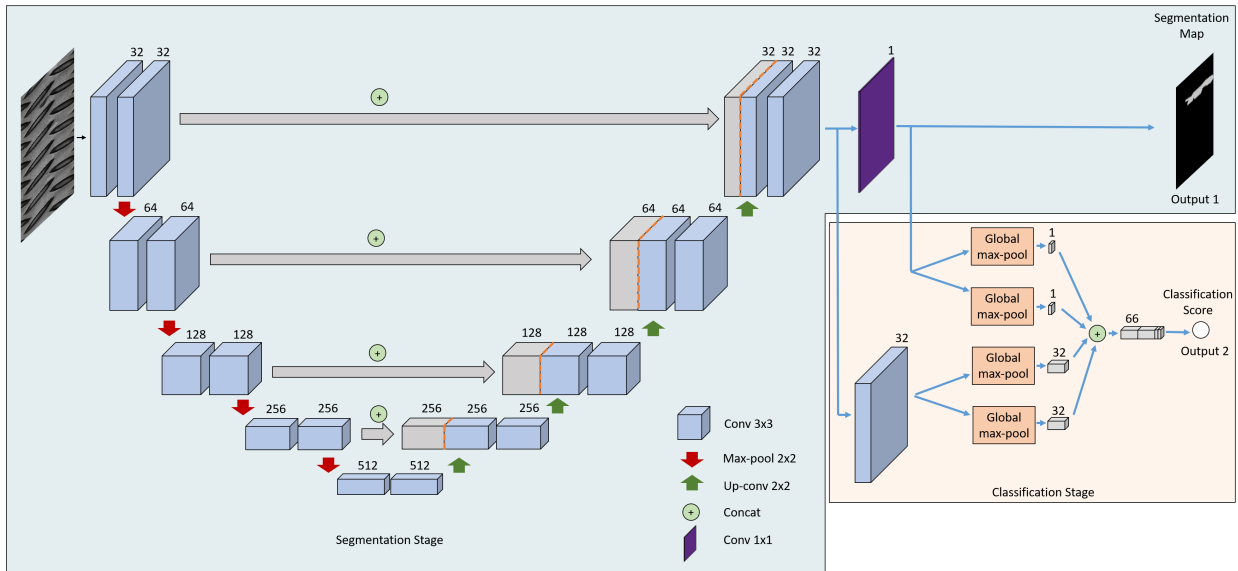


Figure 2-6: Combined network consisting of U-Net and the CompactCNN classification stage

For both architectures, the two stages were trained separately. For training of both stages the Adadelta optimizer with default settings was used. Firstly, the segmentation stage was trained, in order provide meaningful segmentation maps that can be used for classification training. Especially for the segmentation stage, several hyperparameters were varied, such as:

- the number of training epochs,
- the loss function,
- the patch size of the input images,
- the number of training samples.

The segmentation stage worked best when trained to minimize a mean squared error (MSE) loss function according to Equation 2-1. The losses are computed per image batch, where B denotes the number of images in the batch. M and N denote the the number of pixels in vertical and horizontal direction, x_b the label value from the ground truth for sample b at

pixel position i, j and \hat{x}_b the predicted pixel value. Accordingly, the segmentation stage loss function calculates a pixel-wise loss. The ground truth labels are in the value set $\{0,1\}$.

$$\mathcal{L}_s = \frac{1}{BMN} \sum_{b=1}^B \sum_{i=1}^M \sum_{j=1}^N \left\| x_b^{(i,j)} - \hat{x}_b^{(i,j)} \right\|^2 \quad (2-1)$$

The classification stage was trained for 10 epochs with a binary cross-entropy (BCE) loss:

$$\mathcal{L}_C = \frac{1}{B} \sum_{b=1}^B [y_b \log(\hat{y}_b) + (1 - y_b) \log(1 - \hat{y}_b)] \quad (2-2)$$

Again, B denotes the number of samples, y_b the ground truth label of sample b , which is either 0 or 1, and \hat{y}_b the predicted label.

The segmentation stage was either trained on only defective samples or on both defect-free and defective samples with a ratio of 1:1 or 1:3 (defective : defect-free), where the number of defective samples was fixed to 1756. When segmentation training was performed on only defective samples, the classification training was performed on a 1:1 ratio, in order provide some defect-free samples for teaching the network to distinguish between the two types of images. In the other cases the ratios for segmentation and classification were the same.

2.4.3. Evaluation of previous work

In general, the U-Net architecture performed better on image segmentation than the CompactCNN segmentation stage, offering a more precise pixel based localization of the defective area. This is not surprising, as the U-Net is the deeper architecture with more trainable parameters.

It was further found out that a 25 epoch training was sufficient, as a longer training on 100 epochs did not lead to a significant improvement of segmentation results. Furthermore, training tended to work best when the segmentation stage was trained on a mix of defect-free and defective images.

The best results for each dataset and each image size are summarized in tables **2-1** and **2-2** when trained on the CompactCNN architecture and the combined U-Net+CompactCNN architecture. Table **2-1** shows the AUC-score as a measure for the segmentation performance of the network. The AUC-score gives information on how well the defective and the defect-free pixels in the images can be separated. Its score ranges from 0 to 1, being 1 the optimal score.

Table 2-1: Segmentation results: AUC Score for different datasets and image sizes

Network	Dataset 1		Dataset 2	
	256x256	512x512	256x256	512x512
CompactCNN	0.9942	0.9889	0.9912	0.9958
U-Net + CompactCNN	0.9986	0.9967	0.9993	0.9976

Table 2-2 shows the classification accuracy, which denotes the ratio of correctly classified images among all tested images. A further description of both the AUC and the accuracy metric is given in chapter 3.3. In all the cases the U-Net based architecture outperformed the CompactCNN with AUC-scores and classification accuracies of over 99,5.

Table 2-2: Classification accuracies for different datasets and image sizes

Network	Dataset 1		Dataset 2	
	256x256	512x512	256x256	512x512
CompactCNN	0.9942	0.9889	0.9912	0.9958
U-Net + CompactCNN	0.9986	0.9967	0.9993	0.9976

A visualization of the predictions from both the CompactCNN and the U-Net+CompactCNN is given in Figure 2-7.

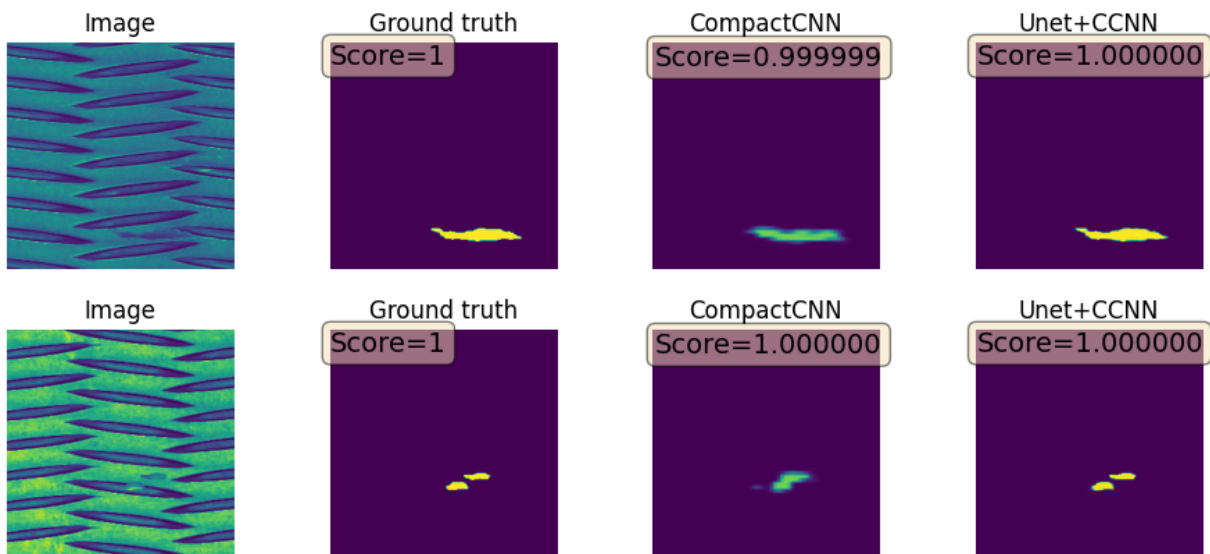


Figure 2-7: Examples of network outputs for CompactCNN and U-Net+CompactCNN. From left to right: Input image, ground truth, CompactCNN prediction, U-Net + CompactCNN prediction

Although the images are originally grayscale images, a colormap ranging from dark purple to yellow was chosen for a clearer indication of the different shades in the predictions. The images show, how the U-Net segmentation maps provide a clearer localization of the defective areas, whereas the prediction of the CompactCNN is rather weak. The score on the top of the images indicates the probability that the analyzed image is defective, being 1 the maximum probability.

3 Methodology

This chapter provides information on the methodology applied to detect anomalies in patterned steel plates under the assumption that no defect data is available from the production lines under investigation. This is the main difference to the previously introduced work. The chapter is structured as follows: Firstly, a description of the available image data is given. Afterwards, the investigated methods are described in detail. Finally, the metrics used for evaluation of the methods are explained and the implementation details are provided.

3.1. Available Data

Grayscale 2D images from five different production batches were collected to conduct the experiments. Four of these represent a teardrop shaped pattern according to Figure 2-1 and the last one a diamond shaped pattern. The data was cropped into patches of size 256 x 256 pixels. A representative image collection of 2000 defect-free images per dataset was chosen as training set. The test datasets contain 200 images each, of which one half is composed of anomaly-free images and the other half of anomalous images. Whenever real anomalies could be found in the image material, they were included into the test set. For the datasets with no or not sufficient real anomalous data, the anomalous test data was complemented with artificial defect data created with the same method introduced in chapter 2.4.1. Figure 3-1 shows one anomaly-free (green margin) and one anomalous image as well as a close-up of the defect region (red margin) from each dataset.

Datasets

- 1) **Teardrop 1:** The first teardrop test set contains 50 real and 50 artificial anomalous images. The real anomalous data mainly depicts several defective regions that occurred periodically as well as some other irregularities found in the image material.
- 2) **Teardrop 2:** The image material for the second dataset does not contain any real anomalies, such that defects were added synthetically.

3) Teardrop 3: Noteworthy is the inhomogeneous background texture of this dataset, which appears in many of the anomaly-free images. This makes the distinguishability between non-defective and defective textures more difficult. As for the previous mentioned dataset, all anomalous images were created artificially.

4) Teardrop 4: This dataset also exhibits a rather inhomogeneous background structure. Large parts of the available image material were excluded due to excessive amount of impurities in front of the camera lenses. Some smaller impurities such as watermarks were included into the test set (Figure 3-1d). They are not a production defect, but still they should be detected as anomalies by automated deep-learning solutions.

5) Diamond: This dataset differs from the others in the type of pattern. The defects were inserted artificially.

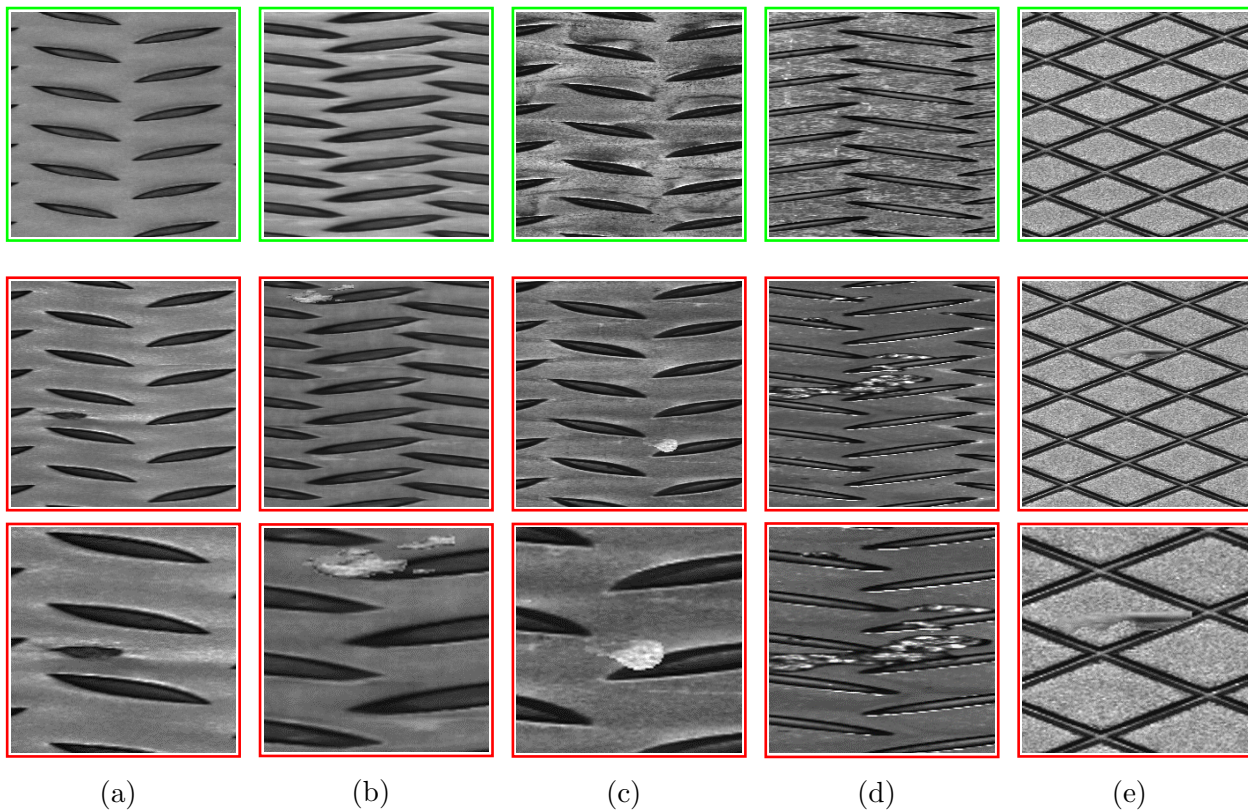


Figure 3-1: Example images for each dataset from left to right: (a) Teardrop1, (b) Teardrop2, (c) Teardrop3, (d) Teardrop4, (e) Diamond. The top row shows a defect-free, the two bottom rows a defective sample and a close-up of the defective region

3.2. Methods

3.2.1. Transfer learning on U-Net

As the previously introduced U-Net architecture provided very accurate segmentation results when it was trained in a supervised manner, it was tested to do a transfer learning on this architecture with the aim to adapt the network to the images of a different production line, of which only good data is available. This method is not semi-supervised as it assumes that of at least one production line there are defective images available to serve as a pretraining dataset. However, once this condition is fulfilled, no defect data of other production lines shall be needed for the adaptation. In our case, the teardrop 2 dataset was used as pretraining dataset. It was modified in a way that artificial defects were inserted into half of the training images applying the same method as introduced in Chapter 2.4.1. That way, 1000 defective and 1000 defect-free samples were obtained for training. It was taken care that none of the artificial defects for training appeared in the test datasets. The network architecture was pretrained on that dataset, in order to learn to extract meaningful characteristics which define the defective areas. Pretraining was done in the same way as mentioned in chapter 2.4.2 for 25 epochs with an MSE loss function and Adadelta optimizer. After pretraining, the parameters learned by the network were kept and the data was mixed with exclusively defect-free data of another production line. In these experiments, the teardrop 1 dataset served as the anomaly-free transfer learning dataset, to which the network should adapt. Training was carried out for another 5 epochs. The performance of the adapted network architecture was measured on the test images of the teardrop 1 dataset.

3.2.2. L2 and SSIM Autoencoder

As autoencoders are a common tool for semi- and unsupervised learning and have shown superior results in anomaly detection than other approaches, such as GANs [3], an autoencoder architecture was tested on the patterned steel dataset. The architecture and training procedure were adapted from the publications of Bergmann et al. [5] [3]. 10.000 randomly cropped image patches of size 128x128 were created from the training dataset with images of size 256x256. These patches were used to train the autoencoder, whose architecture is shown in Figure 3-2.

The encoding part consists of nine convolutional layers of which some have a stride of 2, in order to reduce the output feature size. The stride defines the distances at which the layer's kernel is applied to the input image, meaning that for a stride of 2 the kernel is applied every second pixel. The last layer of the encoding part is a latent vector of variable size z . In our experiments the latent vector's size is set to $z = 100$ by default. The decoding path is the inverse version of the encoding path using deconvolutions in order to upscale the features. Every convolutional layer is followed by a LeakyReLU activation function with slope 0.2,

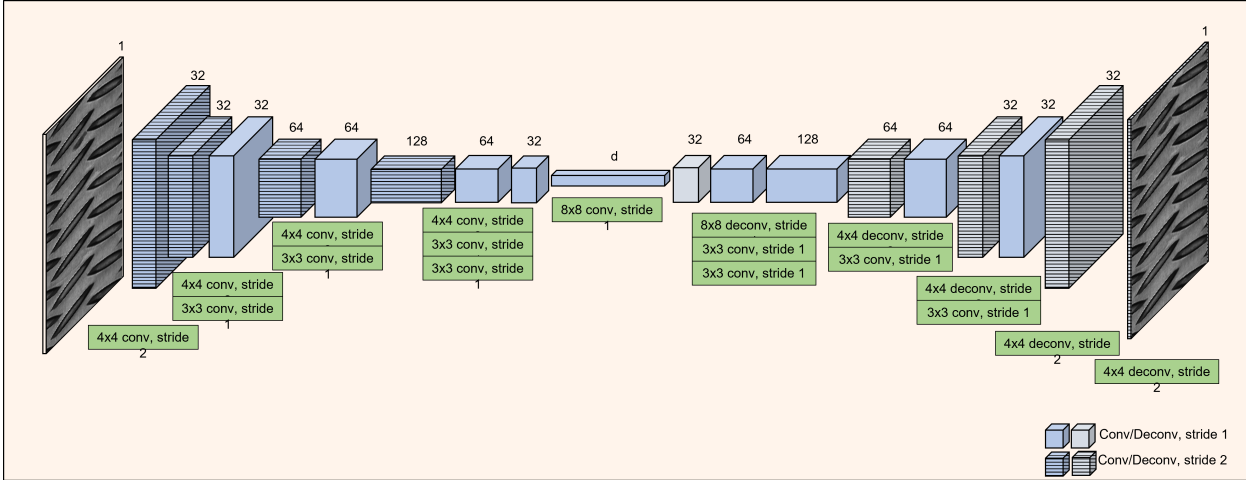


Figure 3-2: Network structure of the L2 and SSIM Autoencoder

except for the last layers of both the encoder and the decoder, which use a linear activation function and a sigmoid activation function respectively.

Training

An Adam optimizer with an initial learning rate of $2 \cdot 10^{-4}$ and a weight decay of 10^{-5} was used for training, which was carried out for 200 epochs. From the 2000 images contained in the training dataset, 80% were used for training and the other 20% for validation.

The architecture was trained with two different loss functions: On the one hand with a squared L2 loss, on the other hand with a structural similarity loss. The **L2 loss** computes the squared difference between input image and the reconstruction image at each pixel [5]. The resulting difference image is called residual map. To apply it as a loss function, which requires a single scalar value, the results over all the pixels were averaged. This is also known as mean squared error, as introduced in Equation 2-1. In the following sections, the non-squared version of the difference image is referred to as L2 residual map.

The **structural similarity** [50] of two image patches p and q is a measure of luminance $l(p, q)$, contrast $c(p, q)$ and structure $s(p, q)$, where the luminance is a function of the mean intensities μ , the contrast of the variance σ and the structure of the covariance σ_{pq} of p and q :

$$l(p, q) = \frac{2\mu_p\mu_q + c_1}{\mu_p^2 + \mu_q^2 + c_1} \quad (3-1)$$

$$c(p, q) = \frac{2\sigma_p\sigma_q + c_2}{\sigma_p^2 + \sigma_q^2 + c_2} \quad (3-2)$$

$$s(p, q) = \frac{\sigma_{pq} + c_2}{2\sigma_p^2\sigma_q^2 + c_2} \quad (3-3)$$

$$SSIM(p, q) = l(p, q)^\alpha c(p, q)^\beta s(p, q)^\gamma \quad (3-4)$$

α , β and γ define the weighting of each component. If they are set equal to 1, the components can be substituted and the structural similarity can be expressed as:

$$SSIM(p, q) = \frac{(2\mu_p\mu_q + c_1)(\sigma_{pq} + c_2)}{(\mu_p^2 + \mu_q^2 + c_1)(2\sigma_p\sigma_q + c_2)} \quad (3-5)$$

The constants c_1 and c_3 are set to $c_1 = 0,01$ and $c_2 = 0,03$ by default. p denotes a window of size $K \times K$ in the input image, while q denotes the corresponding window of the same size in the reconstruction image. By default, K is set to 11 [5]. The windows are slid over the images, i.e. displaced pixel per pixel, and the SSIM score is computed for every possible position. This creates a residual map with the same height and width as the input and reconstruction image. The values of the SSIM index range between -1 and 1, where -1 indicates a low similarity and 1 a high similarity. In order to obtain a function that can be minimized, $1 - SSIM$ is computed. When applied as a loss function, the mean is calculated over all the values in the residual map.

Thresholding

The residual maps were also used as evaluation criterion, in order to decide if an area is anomalous or not. A location is considered to be anomalous, if the residual exceeds a certain threshold. The threshold is to be selected, such that it accurately separates the values of the anomalous areas from the values of the anomaly-free areas, though defining a suitable threshold based on only anomaly-free data is challenging. Under the assumption that an anomaly-free region should produce a smaller residual than an anomalous region, one could attempt to choose biggest residual value among the good data as threshold, such that 100 % of the anomaly-free pixels fall below it. However, this selection is very sensitive to outliers and noise. Hence, it is more robust to choose a threshold, which allows a certain percentage of outliers among the good data. By default, we chose a threshold which located 98 % of the anomaly-free data below the threshold and allowed an outlier rate of 2 %. In statistical terms, this threshold can be described as the 98th percentile of the data. The threshold was selected as the 98th percentile of the anomaly-free validation data. Other percentile selections were also investigated within the experiment.

Testing

During testing, a prediction on the images from the test data was executed, in order to obtain their reconstruction. As the test images have a size of 265x256, but the network takes an input of size 128x128, the input images were predicted section-wise. It would be possible to apply a stride of 128, i.e. to execute one prediction every 128 pixels, such that one section begins precisely where the previous one ends. However, according to Bergmann et al. [5], it results more beneficial to choose overlapping sections, as the reconstructions produced by network can vary slightly with their spatial location. Therefore, a reconstruction was predicted every 32 pixels and the overlapping reconstructions were averaged. Afterwards, the L2 and SSIM residual maps were calculated based on the averaged reconstructions and the thresholds determined. For the autoencoder trained on the L2 loss, the L2 residual map was thresholded during evaluation. For the SSIM autoencoder both the L2 and the SSIM residuals were computed.

Post-Processing

After thresholding, the segmented areas were post-processed, in order to eliminate single pixels or small areas, whose values surpass the thresholds, although they are just caused by outliers or noise. Their elimination was carried out with an opening operation. Opening is a morphological operation which removes small areas from an image - in this case from the thresholded residual map - while preserving larger objects. Openings require a structural element, e.g. a small rectangle or a disk, which can be slid over the residual map. Any element so small that it can be completely covered by the structuring element, is removed from the image. Following the proposal of Bergmann et al. [3] a circular structuring element (disk) was used. For the experiments in this work, a default disk radius of 4 pixels was applied.

3.2.3. Excursus: Analysis of Frequency Spectrum

The analysis of the frequency spectrum is a common tool in image processing, which can be useful for a variety of applications such as filtering or image reconstruction. Images can be transformed into the frequency domain by applying Fourier transform, a technique which decomposes an image into its sine and cosine components. For 2D images, a 2-dimensional discrete Fourier transform is applied. It represents the image by a fixed number of frequencies, which corresponds to the number of pixels. The mathematical expression for the 2D discrete Fourier transform [15] for an image of size $M \times N$ is given as:

$$F(u, v) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i, j) e^{-i2\pi \frac{ui}{M} + \frac{vj}{N}} \quad (3-6)$$

$F(u, v)$ denotes the representation of the image in the frequency domain, which is a

function of the horizontal and vertical frequencies u and v . $f(i, j)$ describes the image in spatial domain as a function of the pixel positions i and j . M is the number of rows and N the number of columns. The resulting function is conformed of complex numbers, which consist of a real part R and an imaginary part I . The imaginary unit is denoted with i . In order to visualize the magnitudes of the frequencies u and v , the amplitudes of the complex numbers can be calculated. Representing a complex number by their amplitude and angle is an alternative to representing it by its real and imaginary component. The amplitudes $|F(u, v)|$ and angles $\Phi(u, v)$ can be calculated as shown in equations 3-7 and 3-8. The logarithmic term is introduced to the amplitude spectrum in order to better capture and visualize the extremely big value range of the amplitudes.

$$|F(u, v)| = \log(\sqrt{R^2(u, v) + I^2(u, v)}) \quad (3-7)$$

$$\Phi(u, v) = \tan^{-1} \frac{I(u, v)}{R(u, v)} \quad (3-8)$$

For highly repetitive structures, such as the patterned steel plates, the images are usually characterised by a very specific set of frequencies, which can be identified in the frequency domain. Figure 3-3 visualizes the magnitude spectrum for an example image from each of the five datasets.

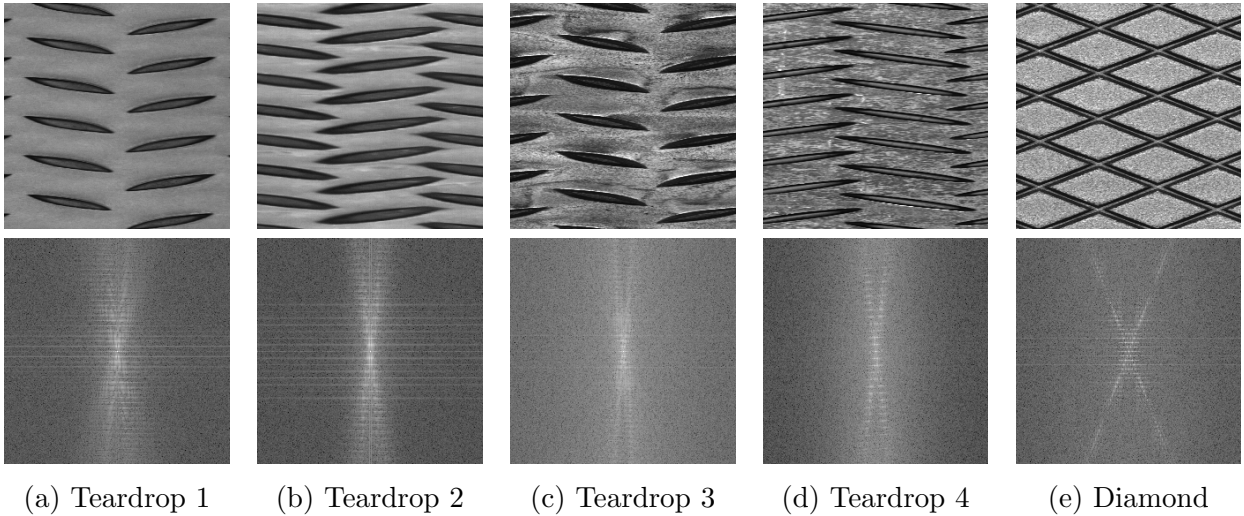


Figure 3-3: Magnitude spectrum in frequency domain for all datasets. Top row: image in spatial domain, Bottom row: frequency spectrum

The frequency images show how the frequency spectra vary with the data sets. The dark areas in the images correspond to frequencies with low amplitudes and the light areas to high amplitudes. The teardrop-shaped images possess a narrow frequency spectrum in

horizontal direction and a spreading spectrum in vertical direction that opens up like a cone beginning at the center point. For each dataset the appearance varies slightly. The diamond shaped pattern on the other hand stretches along both axes with the shape of an X.

In order to make use of this characteristic information, it was investigated if a deep learning architecture could be applied to reconstruct an image in the frequency domain instead of reconstructing it in the spatial domain. This idea was inspired by Lappas et al. [21], who introduced Fourier transform to autoencoders for anomaly detection. Their approach was based on encoding the spatial image itself in combination with the real and imaginary parts of the Fourier transform with separate encoders. In contrast, the approach in this work was based on encoding solely the logarithmic magnitude spectrum, in order to train the autoencoder on identifying and reconstructing only the most dominant frequencies of the corresponding pattern. To achieve this, the autoencoder introduced in the previous section 3.2.2 was adapted and trained on the the frequency images instead of images in the spatial domain. Training was done with the same hyperparameters as for the images in spatial domain. The schematic of this method is visualized in Figure 3-4.

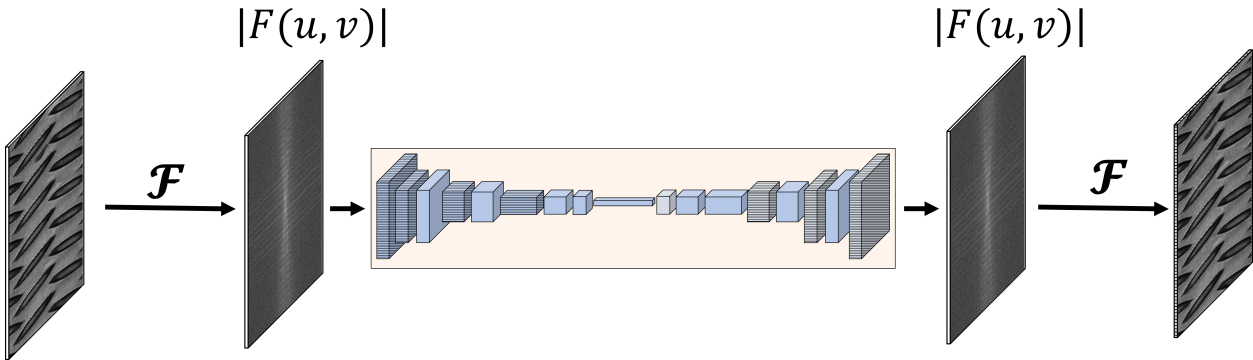


Figure 3-4: Structure of Fourier-based Autoencoder

The information of the angle Φ of the frequency spectrum was preserved as necessary information to transform the reconstructed amplitudes back to the spatial domain. This was done by applying inverse discrete Fourier transform [15]:

$$f(i, j) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} F(u, v) e^{i2\pi \frac{ui}{M} + \frac{vj}{N}} \quad (3-9)$$

3.3. Metrics

Metrics are measures that are used to evaluate the performance of a machine learning model on a given task. This section provides an overview over the performance metrics

used in this work. The outputs of the model can be either positive or negative and are denoted as follows:

True positives (TP): defective instances which are classified correctly (true label *positive*, prediction *positive*).

False positives (FP): defect-free instances that are erroneously predicted as defective (true label *negative*, prediction *positive*).

True negatives (TN): defect-free instances which are classified correctly (true label *negative*, prediction *negative*).

False negatives (FN): defective instances that are misclassified as defect-free (true label *positive*, prediction *negative*).

The definition of what is an instance depends on the task. For classification, one image is considered as one instance, which receives either the label *negative* if there is no anomaly in the image or *positive* if an anomaly is present. For segmentation tasks, each pixel in the image receives an individual label. The definitions given above are the basis for the following evaluation metrics.

3.3.1. True Positive Rate (TPR)

The true positive rate, also known as *sensitivity* or *recall* is the ratio describing how many of all existing positive instances were accurately detected, i.e. predicted as positive. It is defined as:

$$TPR = \frac{TP}{TP + FN} \quad (3-10)$$

3.3.2. True Negative Rate (TNR)

Similar to the TPR, the TNR or *specificity* describes the ratio of all correctly identified true negatives:

$$TNR = \frac{TN}{TN + FP} \quad (3-11)$$

3.3.3. False Positive Rate (FPR)

This measure is the counterpart of the TNR, measuring the ratio of all incorrectly classified negatives, i.e. the negatives which are falsely classified as positives:

$$FPR = \frac{FP}{TN + FP} = 1 - TNR \quad (3-12)$$

3.3.4. Accuracy (ACC)

The accuracy is a measure to evaluate the overall classification performance, hence it measures the ratio of all correctly classified instances, which includes both TNs and TPs, and is defined as:

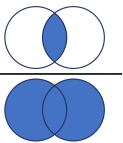
$$ACC = \frac{TN + TP}{TN + FP + TP + FN} \quad (3-13)$$

3.3.5. AUROC

The area under the receiver operating characteristic curve (AUC-ROC or AUROC) is a measure of separability for binary classification problems. The ROC-curve is a graphical chart that plots the TPR against the FPR for various thresholds. The AUC-score measures the area under the ROC-curve and operates in the range of $[0,1)$. A score close to one is desirable, as it indicates that the two classes can be separated very well.

3.3.6. Intersection over Union (IoU)

The intersection over union is a measure that can be used to describe how well the predicted anomaly region and the true anomaly region overlap. Therefore, the overlap of the area is divided by the union of the area.

$$IoU = \frac{A \cap B}{A \cup B} = \frac{\text{Diagram 1}}{\text{Diagram 2}}$$


For binary classification, the IoU, or Jaccard score can be expressed as:

$$IoU = \frac{TP}{FP + TP + FN} \quad (3-14)$$

3.4. Implementation Details

All experiments were executed on a PC with an Intel(R) Xeon(R) CPU E5-2670, 32GB RAM and an NVIDIA GeForce GTX 1080 Graphics Card. The SSIM and L2 Autoencoder were implemented in Tensorflow 2.2 with Python 3.8. The U-Net+CompactCNN architecture was adapted from a previous implementation which was realized with Pytorch 1.9.

4 Results and Discussion

In the following section, the results on the investigated methods are presented and discussed. A comparison of the different methods is given below. Afterwards, the methods are discussed in more detail.

The autoencoder trained on the SSIM loss provided the by far best segmentation and classification results. Therefore, the most extensive studies were executed on this architecture. The L2 autoencoder and the transfer learning on U-Net offer much room for improvement, as they only provided a very rough location of the defects and tended to miss some defects in numerous cases. Figure 4-1 provides an overview over the segmentations of all three methods. The manually annotated ground truths are indicated with green borders and the segmentation results of the methods are marked red within the image. The examples are taken from dataset 1, which contains the most real anomalies and was tested on all three methods. The first example in Figure 4-1 shows a production defect in the material, while the other is no production defect, but an anomaly in the image caused by impurities, which should also be marked, as it is a deviation from the normal pattern.

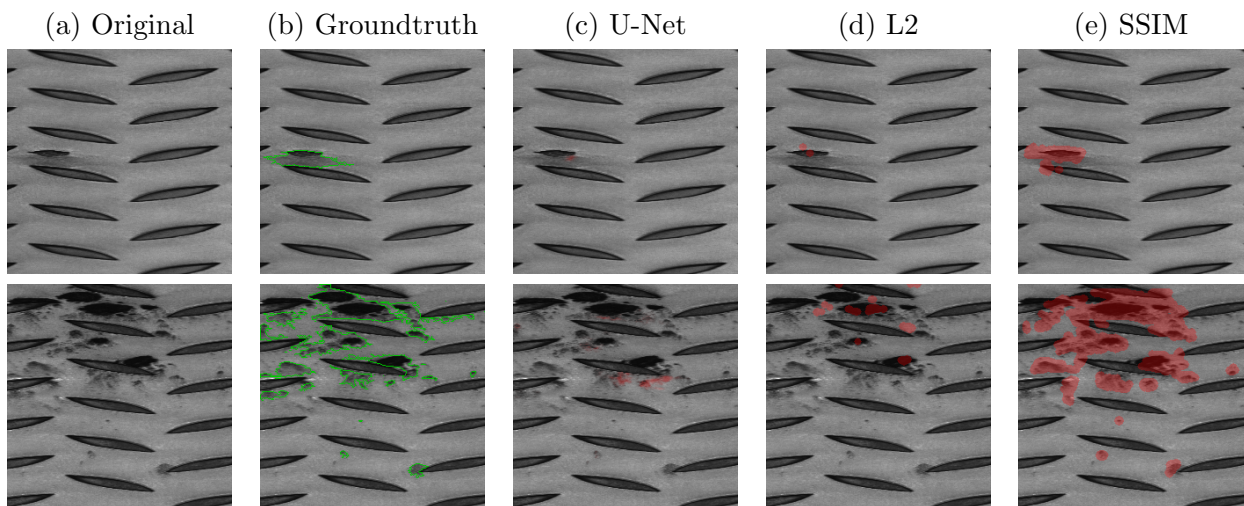


Figure 4-1: Predictions on dataset 1 of U-Net after transfer learning and the L2 and SSIM Autoencoder

4.1. Transfer Learning on U-Net

The transfer learning approach was based on pretraining the network in a supervised fashion on one dataset with artificial defects and then adapt it to another dataset by mixing solely anomaly-free images of this second dataset among the training data. The main idea, to enable the network to learn meaningful defect features in the pretraining step and finally adapt to the slightly altered base texture of another dataset in the second training step, did not provide very accurate results. Even after training only for a small number of 5 epochs during the transfer learning, the network had learned to classify most of the images from the “new” dataset as anomaly-free, even if they had anomalies in it. This is not surprising, as the network only got to see anomaly-free images of that dataset during training. Nonetheless, it is an indicator that the anomaly characteristics learned during pretraining were not transferred very well on images with a slightly different texture.

4.2. L2 and SSIM autoencoder

The L2 and SSIM autoencoder segmentation and classification ability varied with the dataset. In general, the autoencoder tended to segment and classify better when trained on an SSIM loss. These results are quantified in tables **4-2** and **4-1**.

The accuracy scores for each dataset show that the total ratio of correctly classified images is higher for the SSIM autoencoder. The accuracy, as well as the true positive rate and true negative rate, vary greatly depending on the threshold selection. As mentioned in Section 3.2.2, the threshold was selected as the 98-th percentile of a subset of the anomaly-free training data, which means that 98% of the values in the residual maps of the good data had to be below the selected threshold. For the SSIM autoencoder, this threshold selection provided a rather good trade-off for the true positive rate and true negative rate, which means that both anomalous and anomaly-free samples were classified correctly by a similar proportion. As the optimal threshold can vary for each dataset, a more detailed threshold selection is presented in Section 4.2.1 for the SSIM autoencoder.

For the L2 autoencoder, the threshold selection based on the 98-th percentile did not lead to satisfying results, as most of the defects were overlooked. A lower threshold selection could augment the number of correctly detected anomalous areas, but would also lead to a much higher rate of misclassified normal samples (false positives), such that no threshold with a satisfying number of total correct classifications could be obtained. Furthermore, for two of the datasets (Teardrop 3 and Diamond) the L2 autoencoder failed to converge during training. Hence, there is no evaluation listed in tables **4-1** and **4-2** for these datasets, as no reconstruction image could be produced by the L2 autoencoder in those cases. The convergence failure may be due to the very noisy background texture, which is characteristic for both the Teardrop 3 and Diamond dataset and makes it more difficult for

the encoder to learn a latent space representation from which a reconstruction can be created accurately. We conclude that per-pixel based losses like the L2 loss are not a suitable loss function for datasets with a very noisy background texture like the Teardrop 3 and Diamond dataset.

Table 4-1: Classification results for the SSIM and L2 autoencoder

Method	Metric	Teardrop 1	Teardrop 2	Teardrop 3	Teardrop 4	Diamond
SSIM-AE	TNR	0.99	0.88	0.41	0.64	0.93
	TPR	0.92	0.79	0.61	0.72	0.57
	ACC	0.955	0.835	0.51	0.68	0.75
L2-AE	TNR	1.0	1.0	-	1.00	-
	TPR	0.26	0.11	-	0.06	-
	ACC	0.63	0.55	-	0.53	-

Table 4-2: Segmentation results for the SSIM and L2 autoencoder

Method	Metric	Teardrop 1	Teardrop 2	Teardrop 3	Teardrop 4	Diamond
SSIM-AE	IOU	0.202	0.267	0.066	0.121	0.148
	AUC	0.922	0.965	0.740	0.870	0.829
L2-AE	IOU	0.031	0.023	-	0.001	-
	AUC	0.916	0.965	-	0.909	-

For all the other datasets, a reconstruction image could be produced by the both the SSIM and L2 autoencoder. The reconstruction images of both autoencoders can be seen in Figure 4-2, in which one example is given for each dataset. On the top row, the original images are shown for comparison. The graphic also displays the corresponding residual maps which visualize the differences between the original images and the reconstructions. The L2 residual is basically the per-pixel distance, which can be calculated as the absolute values of subtracting the original image from its reconstruction of the L2 autoencoder. For the datasets Teardrop 3 and Diamond, where the L2 autoencoder did not achieve to build a reconstruction image, the displayed L2 residual map was calculated as the difference between the input image and the reconstruction from the SSIM autoencoder. In the L2 residual map, the white pixels denote great differences between original and reconstruction.

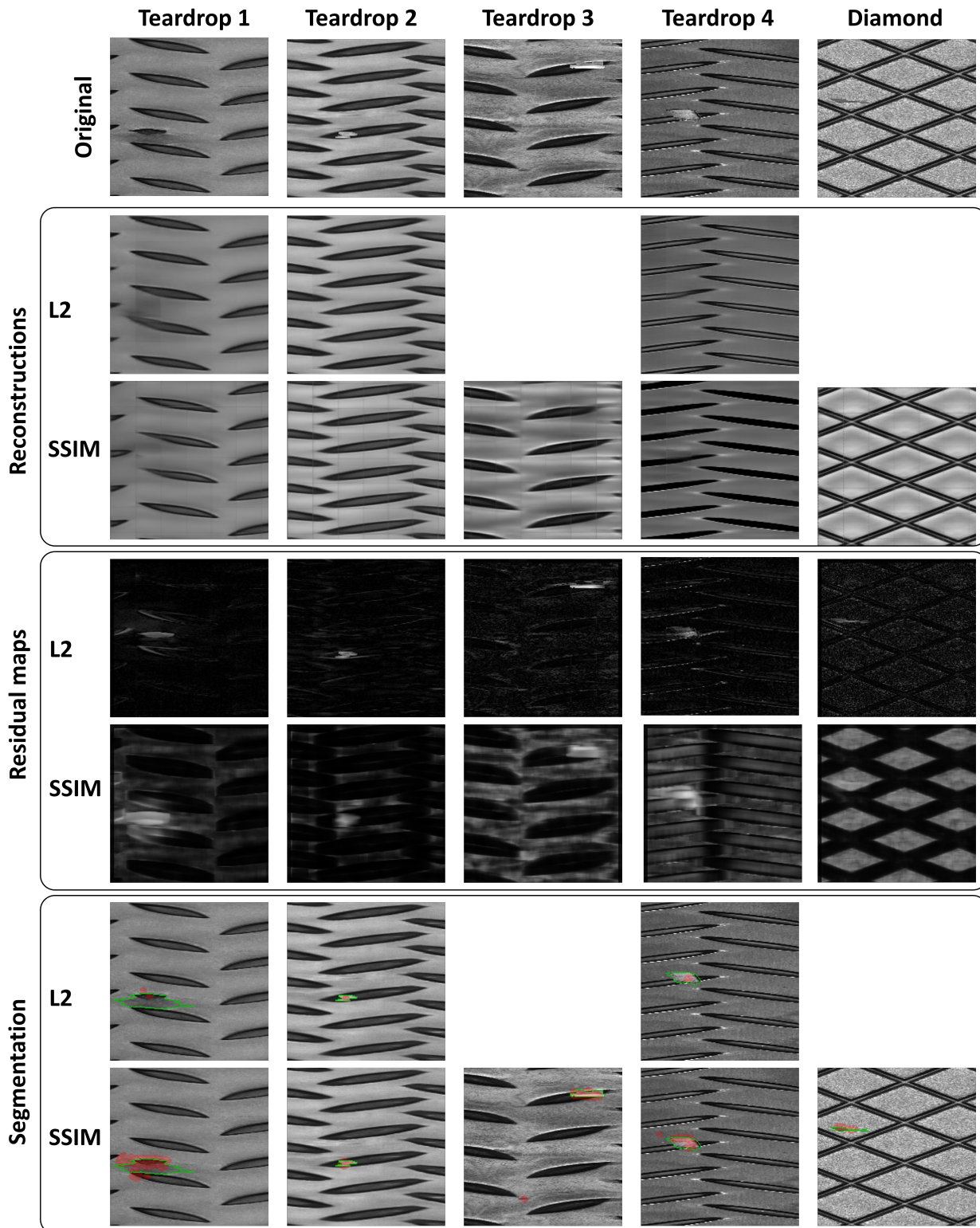


Figure 4-2: Segmentation examples of the L2 and SSIM autoencoder

Differently from the L2 residual map, the SSIM residual map provides a measure for the structural similarity at each point according to Equation 3-5. At each pixel position, the similarity score between an 11x11 pixels sized area around that position in the original image and the reconstruction is computed. White areas in the SSIM residual map indicate a low similarity between original and reconstruction.

The segmentation images provided in the two bottom rows of Figure 4-2 visualize the final segmentation result after post-processing. The L2 segmentation result was based on thresholding the L2 residual map from the L2 autoencoder. For the SSIM autoencoder, both the L2 residual map and the SSIM residual map were calculated and an anomalous region was detected if either of the residuals surpassed their corresponding threshold. Subsequently, the results for each dataset are discussed in more detail.

Teardrop 1: This dataset has the most homogeneous texture of all the datasets and, hence, showed the best classification results. While 99% of the anomaly-free images were classified correctly (TNR), also 92% of the anomalous samples were recognized (TPR) correctly by the SSIM autoencoder. This gives an overall accuracy of 95.5%. The example for this dataset in Figure 4-2 illustrates that the reconstruction of the SSIM autoencoder achieves to eliminate the defective area more accurately than the reconstruction of the L2 autoencoder. The figure points out the difference between the resulting L2 and SSIM residual maps. While the L2 residual map emphasizes very slim defect regions, which are prone to be filtered out by the post-processing algorithm, the SSIM residual map highlights a broader area around the defect. The results can be observed in the segmentation images. In the L2 segmentation, most areas have been eliminated during post-processing, such that only a very rough localization is provided, whereas the SSIM autoencoder highlights a broader region as anomalous. From the SSIM segmentation image it can be seen, why the IoU score is only 0.202 despite the high classification accuracy. Although the defective region is recognized roughly, the ground truth and the prediction do not overlap perfectly. This is difficult to achieve, as on the one hand, an accurate ground truth definition itself is a challenging task, as the transitions between anomalous and non-anomalous regions are smooth. This makes it difficult to define optimal contours, which can be precisely predicted with one single threshold. On the other hand, the post-processing for outlier elimination smoothens the contours of the predicted defect areas whereas the ground truth has sharp contours. However, a low IoU is not a serious issue as long as the defect gets generally detected.

Teardrop 2: For this dataset, both autoencoders provide an accurate reconstruction. The AUC-score, which is 96.5 for both the L2 and the SSIM autoencoder reflects the similar good performance. In comparison with the Teardrop 1 dataset, the texture has slightly more variations which results in less accurate classification results.

Teardrop 3: For this dataset, the background texture contains a lot of noise and especially the anomaly-free sample shown in Figure 3-1 in the previous chapter shows how even among the good data, the pattern is interspersed with black irregular lines, such that more good samples are erroneously classified as anomalous. This is reflected by the low true negative rate of 41%. In general, it is more difficult to select a suitable threshold which distinguishes between the anomalous and anomaly-free areas. This is reflected in the lower accuracy as well as the significantly lower AUC-value.

Teardrop 4: The fourth dataset also tends to exhibit an inhomogeneous background texture, for which the overall classification accuracy of just 68% was reached with the SSIM autoencoder. This indicates that a the SSIM autoencoder works well on the homogeneous structures but leaves room for improvement for production lines, whose image texture is not that even. Accordingly, the AUC-score and IoU-score indicate lower segmentation performance than for the evenly structured datasets 1 and 2. A comparison of the ROC-curves can be found in Figure 4-3.

Diamond: The diamond dataset is the only one with a different pattern. Noticeable is again the noisy background which gets reconstructed in a smoothed form by the autoencoder. As the L2 autoencoder failed to converge, the L2 residual map and the SSIM residual map both compute the differences between the SSIM reconstruction and the original image. The residual maps in Figure 4-2 show that it can be valuable to take the L2 residual map into account as well.

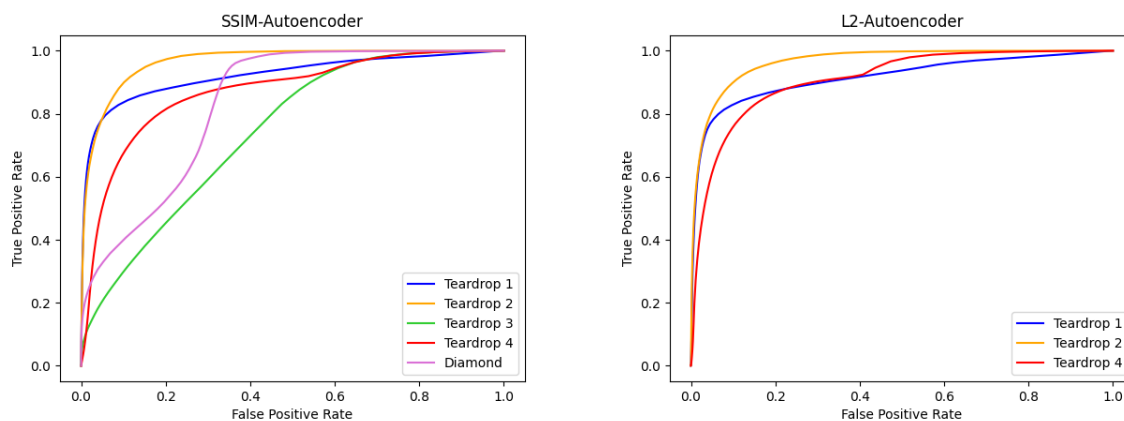


Figure 4-3: ROC-Curves for SSIM and L2 autoencoder

4.2.1. Variation of threshold for the SSIM autoencoder

The choice of a suitable threshold is a key element for an accurate anomaly detection as it greatly influences the results. If it is chosen too low, areas are classified as defective, which are actually defect-free. On the contrary, if it is chosen too high, real anomalous regions might be missed. If no threshold can be found which separates the two classes accurately at all points, a threshold that provides a good trade-off between the number of falsely classified anomaly-free samples and anomalous samples has to be found. This section provides an overview over the influence of different threshold selections on the classification and segmentation results tested on the SSIM autoencoder. For all datasets, the classification results are summarized in Table 4-3 and the segmentation results in Table 4-4. As introduced in Chapter 3.2.2, the threshold was calculated solely on the residual maps of anomaly-free data, such that a certain percentage q of the pixel values lies below this threshold, called the q -th percentile. The percentage q was varied between 95 % and 99 %, where a higher percentage is associated with a higher threshold.

The results listed in Table 4-3 show for all the datasets how the true negative rate drops with a rising threshold, while, in contrast, the true positive rate increases. The highest overall classification accuracies were achieved selecting the 97 % or 98 % percentile as threshold.

Table 4-3: Classification results with different percentiles for threshold estimation

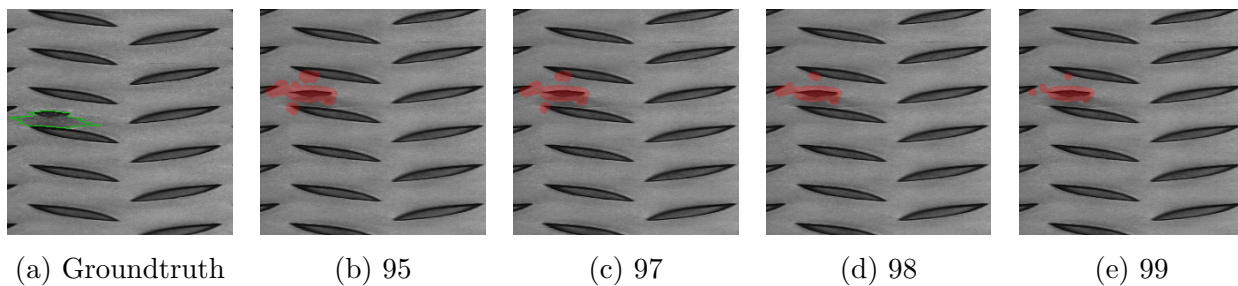
Percentile	Metric	Datasets				
		Teardrop 1	Teardrop 2	Teardrop 3	Teardrop 4	Diamond
0.95	TNR	0.94	0.54	0.08	0.46	0.44
	TPR	0.97	0.94	0.92	0.83	0.88
	ACC	0.955	0.74	0.50	0.645	0.66
0.97	TNR	0.97	0.71	0.24	0.55	0.79
	TPR	0.96	0.90	0.75	0.79	0.73
	ACC	0.965	0.805	0.495	0.67	0.76
0.98	TNR	0.99	0.88	0.41	0.64	0.93
	TPR	0.92	0.79	0.61	0.72	0.57
	ACC	0.955	0.835	0.51	0.68	0.75
0.99	TNR	1.00	0.95	0.80	0.77	1.00
	TPR	0.80	0.56	0.30	0.53	0.36
	ACC	0.90	0.755	0.505	0.65	0.68

Table 4-4: Segmentation results: Intersection over Union (IoU) based on different percentiles for threshold estimation

Percentile	Datasets				
	Teardrop 1	Teardrop 2	Teardrop 3	Teardrop 4	Diamond
0.95	0.186	0.253	0.043	0.217	0.161
0.97	0.196	0.288	0.058	0.168	0.171
0.98	0.202	0.267	0.066	0.121	0.148
0.99	0.211	0.164	0.044	0.054	0.098

For the IoU score, which quantifies the precision of the segmented areas, the tendency is a bit less clear. While the IoU for dataset 1 improves with higher thresholds, other datasets show an improved IoU on lower threshold selection. Just for the Teardrop 3 dataset with the very inhomogeneous texture, the network failed to locate the defects precisely, which is visible by the extremely low IoU as well as the classification accuracy just around 50 percent for all thresholds. It is also visible that the threshold with the best classification accuracy did not necessarily match the threshold for the precisest defect segmentation. In general, it can be concluded that the threshold selection is a flexible procedure, which has to be adjusted for each dataset individually.

Figure 4-4 provides an examples of the Teardrop 1 dataset whose IoU improves with a rising threshold. The figure shows how excessive areas around the defective area are eliminated with increasing thresholds, such that the predicted defect area fits better into the annotated ground truth area.

**Figure 4-4:** Different segmentation results based on different percentiles for threshold selection

4.2.2. Variation of the structuring element during post-processing for the SSIM autoencoder

Similar to the threshold selection, the post-processing technique also greatly influences the segmentation and classification. As the the post-processing was executed with an opening technique with a circular structuring element (see Section 3.2.2), the size of this element was varied and the changes observed. Its radius was varied between 1 and 5. The results for the usage of no structuring element are also listed in tables 4-5 and 4-6.

Table 4-5: Classification results with respect to varying the radius of the circular structuring element during post-processing

Datasets	Metric	Size of structuring Element					
		none	1	2	3	4	5
Teardrop 1	TNR	0.00	0.49	0.82	0.97	0.99	0.99
	TPR	1.00	1.00	0.99	0.96	0.92	0.79
	ACC	0.50	0.745	0.905	0.965	0.955	0.89
Teardrop 2	TNR	0.00	0.16	0.45	0.71	0.88	0.97
	TPR	1.00	1.00	0.99	0.94	0.79	0.59
	ACC	0.50	0.58	0.72	0.825	0.835	0.78
Teardrop 3	TNR	0.00	0.00	0.07	0.22	0.41	0.71
	TPR	1.00	1.00	0.98	0.84	0.61	0.35
	ACC	0.50	0.50	0.525	0.53	0.51	0.53
Teardrop 4	TNR	0.00	0.03	0.40	0.54	0.64	0.74
	TPR	1.00	0.99	0.91	0.81	0.72	0.55
	ACC	0.50	0.51	0.65	0.675	0.68	0.645
Diamond	TNR	0.00	0.06	0.21	0.60	0.93	0.98
	TPR	1.00	1.00	0.97	0.84	0.57	0.33
	ACC	0.50	0.53	0.59	0.72	0.75	0.68

Table 4-6: Segmentation results: Intersection over Union (IoU) with respect to varying the radius of the circular structuring element during post-processing

Datasets	Metric	Size of structuring Element					
		0	1	2	3	4	5
Teardrop 1		0.128	0.174	0.190	0.197	0.202	0.201
Teardrop 2		0.157	0.245	0.268	0.276	0.267	0.229
Teardrop 3	IoU	0.043	0.064	0.067	0.070	0.066	0.056
Teardrop 4		0.155	0.180	0.170	0.149	0.121	0.095
Diamond		0.071	0.157	0.169	0.173	0.148	0.101

The best classification results were achieved using structuring elements with a radius of 3 or 4 pixels, which is a bit higher than the circular element with radius 2 that was used in the publication of Bergmann et al. [3]. The elements of this size provided also the most accurate segmentation areas with the highest IoU scores. However, as most of the IoU scores range around 0.2, there is still a lot of room for improvement, in order to match the predicted defect areas more accurately with the ground truths. Figure 4-5 visualizes the changes of the prediction in relation to the size of the structuring element of the opening procedure. The ground truth is shown in green (Figure 4-5a). Figure 4-5b shows the segmentation result, if no post-processing is performed. Additionally to the defect areas, several other pixels are highlighted red, especially at the outer borders of the teardrops. Applying a radius of 2 already removes these small areas. Applying an opening with a disk radius of 4 even leads to the total elimination of one of the defect areas on the bottom right side, which was still detected with smaller disk sizes. This image was taken from the Teardrop 1 dataset.

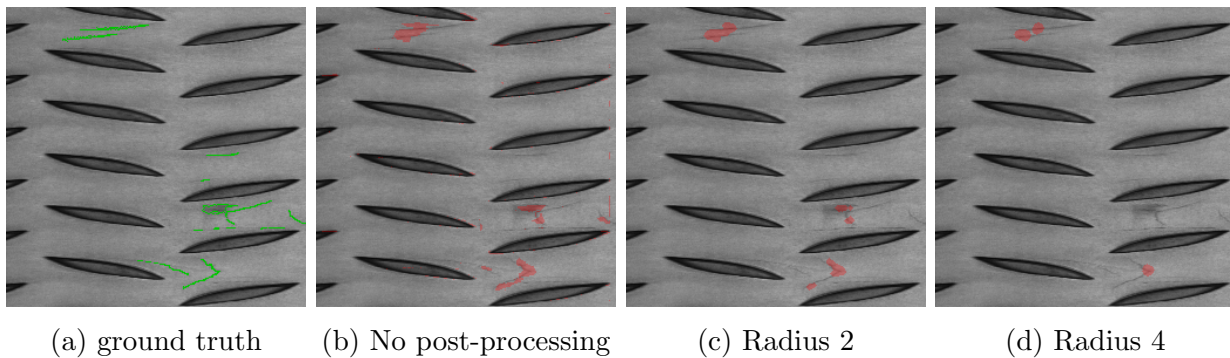


Figure 4-5: Different segmentation results based on different sizes of the structuring element for post-processing

5 Conclusions and Future Research

5.1. Conclusions

In this work, a deep learning solution for the detection of anomalies in patterned steel surfaces under the assumption that only anomaly-free data is available has been presented. For this, several deep learning approaches have been implemented and compared, in order to check their suitability for the given problem.

Transfer learning of an architecture, which had been pretrained on artificial defects in a supervised manner, has been performed on images of another production line and compared to semi-supervised approaches, which were trained on only anomaly-free data from scratch.

The transfer learning was done under the assumption that both non-defective and defective samples were available for at least one production line and the samples were mixed with solely non-defective samples from another production line. This idea did not prove to be ideal, as the deep learning model tended to overlook defects in the test data.

More accurate results could be achieved with an autoencoder trained in a semi-supervised manner with a structural similarity loss function. The autoencoder was trained to build a defect-free reconstruction of the input images. It was shown that applying structural similarity as loss function and as method to build the residual between an image and its reconstruction leads to better segmentation and classification results than an autoencoder trained on a pixel-based loss function such as L2.

For datasets with a very homogeneous structure and few noise, up to 95.5% of the images could be classified correctly. The precision of the segmented defect area leaves room for improvement, as it matches the annotated ground truths only roughly.

Furthermore, the images were investigated in the frequency domain applying Fourier transform. An autoencoder was trained to reconstruct the magnitude spectrum of anomaly-free data. As for the SSIM and L2 autoencoder trained on spatial domain images, the autoencoder should return an anomaly-free reconstruction of any anomalous input image. However, the reconstruction was too detailed, such that the defective areas were not

eliminated in the reconstruction. Further investigations on the setting are recommended in order to improve the results.

In general, the presented results on patterned steel plates live up to other state-of-the-art investigations in semi- and unsupervised anomaly detection, which focus on training on only anomaly-free data [3]. Comparing the methods to supervised deep learning approaches, they have the advantage that no tedious collection or artificial creation of defect data is necessary. However, at the current point, supervised deep learning methods provide more accurate segmentation and classification results with regard to automated defect detection in patterned steel.

5.2. Future Research

Among the investigated methods, the SSIM based autoencoder provided the best results. On datasets with very inhomogeneous background textures, however, the applied method tended to misclassify the anomaly-free images, as the non-defective inhomogeneities were marked as possible defects. This problem could be minimized by the development of an additional classifier, which receives all marked anomalous areas as possible defect candidates and classifies them into uncritical anomalies and critical production defects. In this manner, false positive areas could be eliminated. Furthermore, uncritical anomalies, such as dirt on the camera or watermarks could be identified and distinguished from production defects.

Another option for optimizing the segmentation results could be the improvement of the post-processing. The currently applied opening technique with a circular structuring element can filter out very small areas and single pixels, which are erroneously classified as defective, but the resulting predicted areas deviate from the ground truths. An improved morphological post-processing could increase the precision of the segmented defect areas, in order to match the ground truths more perfectly.

Furthermore, the Fourier based deep learning can be further investigated. The presented investigations show that the autoencoders are able to reconstruct the most dominant frequencies from the encoder's compressed representation, but the reconstruction is still too detailed. It could be tested if changes in the architecture or the latent space dimension can encourage the autoencoder to predict an anomaly-free reconstruction.

Finally, it is planned to transfer the existing deep learning model developed in Python into a C# based environment, in order to integrate it into the defect detection systems installed in the production lines. Once this is done, it is recommended to execute experiments on the inference time on these systems, in order to evaluate the model's suitability for real-time applications.

Bibliography

- [1] ATIENZA, Rowel: *Advanced Deep Learning with Keras: Apply deep learning techniques, autoencoders, GANs, variational autoencoders, deep reinforcement learning, policy gradients, and more.* 1. Birmingham : Packt Publishing Limited, 2018. – ISBN 9781788624534
- [2] BERGMANN, Paul ; BATZNER, Kilian ; FAUSER, Michael ; SATTLEGGGER, David ; STEGER, Carsten: The MVTEC Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In: *International Journal of Computer Vision* 129 (2021), No. 4, p. 1038–1059. – ISSN 0920–5691
- [3] BERGMANN, Paul ; FAUSER, Michael ; SATTLEGGGER, David ; STEGER, Carsten: MVTEC AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2019. – ISBN 978–1–7281–3293–8, p. 9584–9592
- [4] BERGMANN, Paul ; FAUSER, Michael ; SATTLEGGGER, David ; STEGER, Carsten: Uninformed Students: Student-Teacher Anomaly Detection With Discriminative Latent Embeddings. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2020. – ISBN 978–1–7281–7168–5, p. 4182–4191
- [5] BERGMANN, Paul ; LÖWE, Sindy ; FAUSER, Michael ; SATTLEGGGER, David ; STEGER, Carsten: Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders. (2019), p. 372–380
- [6] BOŽIČ, Jakob ; TABERNIK, Domen ; SKOČAJ, Danijel: End-to-end training of a two-stage neural network for defect detection. In: *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, p. 5619–5626
- [7] BREIMAN, Leo: Random Forests. In: *Machine Learning* 45 (2001), No. 1, p. 5–32. – ISSN 08856125
- [8] CHANDOLA, Varun ; BANERJEE, Arindam ; KUMAR, Vipin: Anomaly detection. In: *ACM Computing Surveys* 41 (2009), No. 3, p. 1–58. – ISSN 0360–0300
- [9] CHRISTIAN SZEGEDY, SERGEY IOFFE, VINCENT VANHOUCHE, ALEXANDER A. ALEMI: Inception-v4, Inception-ResNet and the Impact of Residual Connections on

- Learning. In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)* (2017), p. 4279–4284
- [10] DAWSON-HOWE, Kenneth: *A practical introduction to computer vision with OpenCV*. Online edition. Chichester, England : Wiley, 2014. – ISBN 9781118848784
- [11] DEUTSCHES INSTITUT FÜR NORMUNG E.V. *DIN EN 10363:2016 D: Continuously hot-rolled patterned steel strip and plate/sheet cut from wide strip - Tolerances on dimensions and shape*. October 2016
- [12] DI, He ; KE, Xu ; PENG, Zhou ; DONGDONG, Zhou: Surface defect classification of steels with a new semi-supervised learning method. In: *Optics and Lasers in Engineering* 117 (2019), p. 40–48. – ISSN 01438166
- [13] DI HE ; XU, Ke ; WANG, Dadong: Design of multi-scale receptive field convolutional neural network for surface inspection of hot rolled steels. In: *Image and Vision Computing* 89 (2019), p. 12–20. – ISSN 02628856
- [14] GLASSNER, Andrew: *Deep Learning: A Visual Approach*. No Starch Press, 2021. – ISBN 9781718500730
- [15] GONZALEZ, Rafael C. ; WOODS, Richard E.: *Digital image processing*. Fourth, global edition. New York, New York : Pearson Education, 2018. – ISBN 9781292223070
- [16] HAN, Lin ; CHEN, Yuanhao ; LI, Jiaming ; ZHONG, Bowei ; LEI, Yuzhu ; SUN, Minghui: Liver segmentation with 2.5D perpendicular UNets. In: *Computers & Electrical Engineering* 91 (2021), p. 107118. – ISSN 00457906
- [17] HANBAY, Kazım ; TALU, Muhammed F. ; ÖZGÜVEN, Ömer F.: Fabric defect detection systems and methods—A systematic literature review. In: *Optik* 127 (2016), No. 24, p. 11960–11973. – ISSN 00304026
- [18] HE, Yu ; SONG, Kechen ; MENG, Qinggang ; YAN, Yunhui: An End-to-End Steel Surface Defect Detection Approach via Fusing Multiple Hierarchical Features. In: *IEEE Transactions on Instrumentation and Measurement* 69 (2020), No. 4, p. 1493–1504. – ISSN 0018–9456
- [19] HUANG, Yibin ; QIU, Congying ; WANG, Xiaonan ; WANG, Shijun ; YUAN, Kui: A Compact Convolutional Neural Network for Surface Defect Inspection. In: *Sensors (Basel, Switzerland)* 20 (2020), No. 7
- [20] HUANG, Zheng ; WU, Jiajun ; XIE, Feng: Automatic surface defect segmentation for hot-rolled steel strip using depth-wise separable U-shape network. In: *Materials Letters* 301 (2021), p. 130271. – ISSN 0167577X

-
- [21] LAPPAS, Demetris ; ARGYRIOU, Vasileios ; MAKRIS, Dimitrios: Fourier Transformation Autoencoders for Anomaly Detection. In: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021. – ISBN 978-1-7281-7605-5, p. 1475–1479
- [22] LI, Yundong ; ZHAO, Weigang ; PAN, Jiahao: Deformable Patterned Fabric Defect Detection With Fisher Criterion-Based Deep Learning. In: *IEEE Transactions on Automation Science and Engineering* 14 (2017), No. 2, p. 1256–1264. – ISSN 1545-5955
- [23] LI, Yuyuan ; ZHANG, Dong ; LEE, Dah-Jye: Automatic fabric defect detection with a wide-and-compact network. In: *Neurocomputing* 329 (2019), p. 329–338. – ISSN 09252312
- [24] LIN, Hui ; LI, Bin ; WANG, Xinggang ; SHU, Yufeng ; NIU, Shuanglong: Automated defect inspection of LED chip using deep convolutional neural network. In: *Journal of Intelligent Manufacturing* 30 (2019), No. 6, p. 2525–2534. – ISSN 0956-5515
- [25] LIU, Jie ; SONG, Kechen ; FENG, Mingzheng ; YAN, Yunhui ; TU, Zhibiao ; ZHU, Liu: Semi-supervised anomaly detection with dual prototypes autoencoder for industrial surface inspection. In: *Optics and Lasers in Engineering* 136 (2021), p. 106324. – ISSN 01438166
- [26] LIU, Yang ; XU, Ke ; XU, Jinwu: Periodic Surface Defect Detection in Steel Plates Based on Deep Learning. In: *Applied Sciences* 9 (2019), No. 15, p. 3127
- [27] LUO, Qiwu ; LIU, Kexin ; SU, Jiaojiao ; YANG, Chunhua ; GUI, Weihua ; LIU, Li ; SILVEN, Olli: Waterdrop Removal From Hot-Rolled Steel Strip Surfaces Based on Progressive Recurrent Generative Adversarial Networks. In: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), p. 1–11. – ISSN 0018-9456
- [28] MEI, Shuang ; WANG, Yudan ; WEN, Guojun: Automatic Fabric Defect Detection with a Multi-Scale Convolutional Denoising Autoencoder Network Model. In: *Sensors (Basel, Switzerland)* 18 (2018), No. 4
- [29] MUELLER, John P. ; MASSARON, Luca: *Deep learning for dummies*. Hoboken, NJ : John Wiley & Sons Inc, 2019 (For dummies). – ISBN 9781119543039
- [30] NAPOLETANO, Paolo ; PICCOLI, Flavio ; SCETTINI, Raimondo: Anomaly Detection in Nanofibrous Materials by CNN-Based Self-Similarity. In: *Sensors (Basel, Switzerland)* 18 (2018), No. 1
- [31] NAPOLETANO, Paolo ; PICCOLI, Flavio ; SCETTINI, Raimondo: Semi-supervised anomaly detection for visual quality inspection. In: *Expert Systems with Applications* 183 (2021), p. 115275. – ISSN 09574174

- [32] NEOGI, Nirbhar ; MOHANTA, Dusmanta K. ; DUTTA, Pranab K.: Review of vision-based steel surface inspection systems. In: *EURASIP Journal on Image and Video Processing* 2014 (2014), No. 1
- [33] NGAN, Henry Y. ; PANG, Grantham K. ; YUNG, Nelson H.: Automated fabric defect detection—A review. In: *Image and Vision Computing* 29 (2011), No. 7, p. 442–458. – ISSN 02628856
- [34] PAULRAJ, M. P. ; SHUKRY, A. M. M. ; YAACOB, S. ; ADOM, A. H. ; KRISHNAN, R. P.: Structural steel plate damage detection using DFT spectral energy and artificial neural network. In: *2010 6th International Colloquium on Signal Processing & its Applications*, IEEE, 2010. – ISBN 978–1–4244–7121–8, p. 1–6
- [35] PLANCHE, Benjamin: *Hands-On Computer Vision with TensorFlow 2: Leverage deep learning to create powerful image processing apps with TensorFlow 2.0 and Keras*. 1. Birmingham : Packt Publishing Limited, 2019. – ISBN 9781788839266
- [36] RACKI, Domen ; TOMAZEVIC, Dejan ; SKOCAJ, Danijel: A Compact Convolutional Neural Network for Textured Surface Anomaly Detection. In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 32018. – ISBN 978–1–5386–4886–5, p. 1331–1339
- [37] REDMON, Joseph ; DIVVALA, Santosh ; GIRSHICK, Ross ; FARHADI, Ali: You Only Look Once: Unified, Real-Time Object Detection. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2016
- [38] REN, Ruoxu ; HUNG, Terence ; TAN, Kay C.: A Generic Deep-Learning-Based Approach for Automated Surface Inspection. In: *IEEE transactions on cybernetics* 48 (2018), No. 3, p. 929–940
- [39] RONNEBERGER, Olaf ; FISCHER, Philipp ; BROX, Thomas: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: NAVAB, Nassir (Ed.) ; HORNEGGER, Joachim (Ed.) ; WELLS, William M. (Ed.) ; FRANGI, Alejandro F. (Ed.): *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Cham : Springer International Publishing, 2015. – ISBN 978–3–319–24574–4, p. 234–241
- [40] SCHLEGL, Thomas ; SEEBÖCK, Philipp ; WALDSTEIN, Sebastian M. ; LANGS, Georg ; SCHMIDT-ERFURTH, Ursula: f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. In: *Medical image analysis* 54 (2019), p. 30–44
- [41] SCHLEGL, Thomas ; SEEBÖCK, Philipp ; WALDSTEIN, Sebastian M. ; SCHMIDT-ERFURTH, Ursula ; LANGS, Georg: Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery, Springer, Cham, 2017, p. 146–157

- [42] SEE, Judi E. ; DRURY, Colin G. ; SPEED, Ann ; WILLIAMS, Allison ; KHALANDI, Negar: The Role of Visual Inspection in the 21 st Century. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 61 (2017), No. 1, p. 262–266. – ISSN 2169–5067
- [43] SONG, Kechen ; YAN, Yunhui: A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. In: *Applied Surface Science* 285 (2013), p. 858–864. – ISSN 01694332
- [44] SOULAMI, Khaoula B. ; KAABOUCHE, Naima ; SAIDI, Mohamed N. ; TAMTAOUI, Ahmed: Breast cancer: One-stage automated detection, segmentation, and classification of digital mammograms using UNet model based-semantic segmentation. In: *Biomedical Signal Processing and Control* 66 (2021), p. 102481. – ISSN 17468094
- [45] STEINWART, Ingo ; CHRISTMANN, Andreas: *Support vector machines*. New York, NY : Springer, 2008 (Information science and statistics). – ISBN 978–0–387–77241–7
- [46] TABERNIK, Domen ; ŠELA, Samo ; SKVARČ, Jure ; SKOČAJ, Danijel: Segmentation-based deep-learning approach for surface-defect detection. In: *Journal of Intelligent Manufacturing* 31 (2020), No. 3, p. 759–776. – ISSN 0956–5515
- [47] TSAI, Du-Ming ; JEN, Po-Hao: Autoencoder-based anomaly detection for surface defect inspection. In: *Advanced Engineering Informatics* 48 (2021), p. 101272. – ISSN 14740346
- [48] WANG, Lu ; ZHANG, Dongkai ; GUO, Jiahao ; HAN, Yuexing: Image Anomaly Detection Using Normal Data Only by Latent Space Resampling. In: *Applied Sciences* 10 (2020), No. 23, p. 8660
- [49] WANG, Tian ; CHEN, Yang ; QIAO, Meina ; SNOUSSI, Hichem: A fast and robust convolutional neural network-based defect detection model in product quality control. In: *The International Journal of Advanced Manufacturing Technology* 94 (2018), No. 9-12, p. 3465–3471. – ISSN 0268–3768
- [50] WANG, Z. ; SIMONCELLI, E. P. ; BOVIK, A. C.: Multiscale structural similarity for image quality assessment. In: *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, IEEE, 2003. – ISBN 0–7803–8104–1, p. 1398–1402
- [51] WEI, Bing ; HAO, Kuangrong ; TANG, Xue-song ; DING, Yongsheng: A new method using the convolutional neural network with compressive sensing for fabric defect classification based on small sample sizes. In: *Textile Research Journal* 89 (2019), No. 17, p. 3539–3555. – ISSN 0040–5175
- [52] WEIMER, Daniel ; SCHOLZ-REITER, Bernd ; SHPITALNI, Moshe: Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. In: *CIRP Annals* 65 (2016), No. 1, p. 417–420. – ISSN 00078506

-
- [53] XIE, Xianghua: A Review of Recent Advances in Surface Defect Detection using Texture analysis Techniques. In: *ELCVIA Electronic Letters on Computer Vision and Image Analysis* 7 (2008), No. 3, p. 1
- [54] ZHANG, Defu ; SONG, Kechen ; XU, Jing ; HE, Yu ; YAN, Yunhui: Unified detection method of aluminium profile surface defects: Common and rare defect categories. In: *Optics and Lasers in Engineering* 126 (2020), p. 105936. – ISSN 01438166
- [55] ZHANG, Hong-wei ; ZHANG, Ling-jie ; LI, Peng-fei ; GU: Yarn-dyed Fabric Defect Detection with YOLOV2 Based on Deep Convolution Neural Networks. In: *2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, IEEE, 2018
- [56] ZHANG, Yifei: *A better autoencoder for image: Convolutional autoencoder*, Australian National University, Dissertation
- [57] ZHOU, Shiyang ; CHEN, Youping ; ZHANG, Dailin ; XIE, Jingming ; ZHOU, Yunfei: Classification of surface defects on steel sheet using convolutional neural networks. In: *Materiali in tehnologije* 51 (2017), No. 1, p. 123–131. – ISSN 15802949
- [58] ZIMEK, Arthur ; SCHUBERT, Erich: Outlier Detection. In: LIU, Ling (Ed.) ; ÖZSU, M. T. (Ed.): *Encyclopedia of Database Systems*. New York, NY : Springer New York, 2017. – ISBN 978–1–4899–7993–3, p. 1–5