

# Automated Defect Detection of Screws in the Manufacturing Industry Using Convolutional Neural Networks

Johannes Breitenbach  
University of Bayreuth  
[johannes.breitenbach@uni-bayreuth.de](mailto:johannes.breitenbach@uni-bayreuth.de)

Isabelle Eckert  
Aalen University  
[isabelle.eckert@studmail.htw-aalen.de](mailto:isabelle.eckert@studmail.htw-aalen.de)

Vanessa Mahal  
Aalen University  
[vanessa.mahal@studmail.htw-aalen.de](mailto:vanessa.mahal@studmail.htw-aalen.de)

Hermann Baumgartl  
Aalen University  
[hbaumgartl@ieee.org](mailto:hbaumgartl@ieee.org)

Ricardo Buettner  
University of Bayreuth  
Fraunhofer FIT  
[ricardo.buettner@uni-bayreuth.de](mailto:ricardo.buettner@uni-bayreuth.de)

## Abstract

*Defect detection in industrial production processes is an important and necessary part of quality control. Many defects can occur during the manufacturing process, causing high manufacturing costs. Thus the inspection of screws, which represent an indispensable element of many mechanical components, is a critical process. To reduce manufacturing costs and increase efficiency, a reliable method for inspection is Deep Learning. It can help simplify the process of quality control and increase the velocity and volume of detected defects in screws. This approach uses a CNN model to classify non-defective and defective screws with different types of defects. Instead of manual quality control methods, which can be easily biased, our CNN approach is accurate, cost-efficient, and fast, with an accuracy of over 97 percent. With this approach corresponding to industrial production processes, different defects in screws and non-defective screws can be classified from images according to a real-world industrial inspection scenario.*

## 1. Introduction

In the industrial production processes of the fastener industry, several defects can occur and cause high manufacturing costs as well as a high amount of rejects [1, 2]. The quality inspection of screws is a critical process due to the complex structure and variety of defects [2–4]. Screws are indispensable elements of many mechanical components, and it is crucial that those elements do not negatively affect the product. Therefore, it is necessary to inspect all manufactured screws [5].

Screw parts like the threads are one of the most important components of machinery. Threaded elements amount to 15% of the mechanical parts in equipment and machines. In industrial production, the usage of screw threads is very high. Real-time and high-

precision measurement, as well as the improvement of accessories in this field, are important [6].

One of the most popular quality management systems globally is ISO 9001, which aims to improve processes, access to foreign markets, and increase competitiveness. Another goal is an optimum quality level which equals zero defects. Many manufacturers follow the zero-defect strategy for their products. Hence, the total cost of quality can be decreased with quality improvement processes [7–10].

Thus, it is necessary to perform quality control to detect defect screws in order to reduce manufacturing costs and to improve the quality and yield [1, 2]. A high level of quality in products is necessary for manufacturers to be competitive globally and efficient [11]. The detection of defects by manual work is costly, time-consuming as well as low efficiency. It can also lead to a high error rate, and not all produced screws can be inspected [1, 2, 5, 12, 13]. Especially for a large number of screws that are produced in a short time, a high-speed inspection is necessary [1, 5]. For an automated classification and quality inspection of the produced goods, an Automated Visual Inspection (AVI) system is often used to replace the manual quality inspection [14, 15]. The system recognizes the products lying on the conveyor belt and uses an algorithm for object detection and recognition [15, 16].

Automatic defect recognition is a reliable method for quality control in the production processes [2]. Thus, visual analysis for surface defect detection is a standard method [17]. This important part of quality control is becoming more interesting for industrial manufacturing processes [12]. Sensors and the Internet of Things cause an increase in data recorded in the manufacturing process. Machine learning approaches have proven to be an effective tool for the evaluation of different sensor data [18, 19]. Especially Deep Learning shows superior performance for the analysis of image data [20]. As industrial automation advances, Deep Learning can help simplify the process of quality control. Based on

images, defects in the manufactured objects can be detected, and defect products can be sorted out [2, 21]. Also, it contributes to more reliable and efficient defect detection in terms of the zero-defect strategy [7–10, 13].

Therefore, good results were already achieved with Deep Learning approaches such as the work by Song et al. [1], where a Convolutional Neural Network (CNN) approach was successfully used to detect micro defects of metal screw surfaces with images collected from an industrial camera. However, no realistic production scenario was considered. Wu et al. [17] concentrated on surface defect detection and created an adaptable CNN-based model. Nevertheless, the location of different defects was not considered like they can occur in the quality inspection [22].

There are already CNN-based approaches for the defect detection of small parts for manufacturing, which have considered more realistic images. The work by Yang et al. [23] considered the real-world manufacturing constraints and parameters for four different defect classes. However, separate models were used for the different defects, which is not suitable for real production conditions in quality control.

The various studies show that a CNN-based approach is an auspicious method for defect detection of screws in quality control. CNNs are able to identify defects based on automatically learned deep features and do not have to be designed by human engineers [24].

Therefore, this paper investigates if a CNN approach can classify different defects in screws and defect-free screw images with realistic angles of lying screws and five defect types according to a real-world industrial inspection scenario. To achieve this, the Deep Learning method CNN with a VGG16 architecture is applied to test and train the screw data set [22].

As several defects can occur in the manufacturing process, it is necessary to focus on defects that can happen in real-world manufacturing scenarios. The threads as an important part should therefore be considered as well as other common defects like scratches and dents on the screw head, top or neck [6, 22]. In order to be able to detect realistic defects reliably, we chose a real-world scenario approach.

Other approaches detect differences in the size of screws [25] or classify different screw types [26], as well as identify several similar items like screws, washers, and nuts [2]. Some focus on the detection of defects in different materials such as texture, metallic gasket, and screws but consider only one defect type [17] or detect defects of screw surfaces using only top view images [1]. We focus on recognizing five screw defects with side-view images like in a realistic production scenario. In this case, the screw front, head, neck, thread side, and thread top are considered, and defects like scratches and dents on the surface as well as

distorted and missing object parts can be recognized. The most important results of this work are:

1. We develop an automated classification method based on a CNN model with VGG16 architecture for a reliable classification of defective and non-defective screws.
2. Our accurate and robust CNN-based approach can automatically detect faulty screws as part of camera-based in-line inspection during production using only a single image of the object. This minimizes the negative impact of quality inspection on production logistics performance, being of high practical relevance.
3. With a high level of accuracy (97.92%) based on real-world images, our approach outperforms previous studies.

This paper is organized as follows: First, we provide an overview of the research background and the related work. Subsequently, the methodology is described with the used method as well as the used data set and the preprocessing. Then we present the results of our implemented method. Finally, we discuss our results, present the limitations, and propose ideas for future work.

## 2. Research Background

### 2.1 Quality Inspection of Screws

During the production process of material components, cracks and scratches can occur. To prevent defective products from getting into the market a quality control after production is necessary. One of the difficulties of these inspections is the parts' complex structure, resulting in various workpiece defects [2, 3].

Many manufacturers use manual inspections to detect defects in the quality control process which is time-consuming and affected by the energy level and the worker's experience [1, 27]. Quality control in the fastener industry is often carried out by measuring with calipers or optical amplifiers [1]. Some screw manufacturers use human resources to do manual inspections [28]. Trained workers are required during this process, which is costly in terms of time and labor [1, 29]. Also, the energy levels of employees are narrowed, which can cause reduced efficiency on long working hours [1]. Often, not every single product is inspected, and samples are used for quality control, resulting in defective parts leaving the manufacturing process [11]. Mistakes of manual methods then lead to the loss of economic efficiency [30].

AVI is a quality control technique that utilizes cameras connected to computer systems. The AVI system enhances the quality of the produced goods.

Therefore, the inspection checks whether a product differs from specific production rules and specifications. The system captures images of the parts while the conveyor belt is moving. The images are filtered, objects are recognized, and features are extracted. Then the detected non-conforming parts are separated [14–16]. For this method, CNNs are promising because they are able to perform feature extraction and defect recognition on a single network. Hence, the preprocessing of images for certain applications is not necessary. Also, the transferability of the results is facilitated [31].

## 2.2 Quality Control via Machine Learning

A more effective method is automated defect detection in production processes. For detecting surface defects, diverse image processing techniques have been developed [1, 12, 32]. An overview of the relevant related work is shown in Table 1.

Often, manufacturers work with inspection systems like high-speed inspection machines using a conveyor and vision cameras to detect surface defects [33]. Challenges of visual defect detection with images are the greatly heterogeneous appearances of the object surfaces and the defective areas, as well as different rotations and angles of the parts in the inspection process [16, 34]. All parameters can influence the visual defect detection results [16, 35], whereas a robust solution is CNN [35–37].

Song et al. [1] developed a deep CNN-based method to detect micro defects of metal screw surfaces while first locating the screw surface. The data set contained 3,000 samples of defect-free screws, stripped screws, surface-damaged and surface-dirty screws. The model achieved an accuracy of 98.40% and showed the superiority of the DCNN-based approach compared to traditional template matching methods and the LeNet-5-based method. The limitation of their work is that only images of mounted screws with a top view were considered. So, this approach is only suitable for special applications like ready-made products with mounted screws.

One study used a detection method based on CNN with automatic image feature extraction to detect surface defects in industry production. The study of Wu et al. [17] was able to achieve 90.00% accuracy for screw images with a CNN approach, completed with a voting mechanism for final classification and location. CNN was applied to detect defects in texture images, special structural images such as metallic gasket and screw images. The result was an adaptable model to different data sets. The main limitation of this work is that only single defects were considered, which is not the case in a real-work quality control scenario.

Also, an accurate real-time defect detection system for tiny parts was developed for manufacturing using an end-to-end CNN algorithm. The data set includes four different defect classes of 0.8 cm darning needles and one class of defect-free needles. They use side-view images of the darning needles on a conveyor belt taken from an industrial camera. The different accuracies refer to the different defect types: 98.00%, 99.00%, 97.80%, and 79.40%, so each defect corresponds to a separate model. In contrast to this study, our approach aims to detect all screw defects [23]. The separate models limit the suitability of this approach for a realistic quality inspection process.

Bergmann et al. [22] proposed a method for an automated and real-time image classification for different defect types on screws. A variety of defects such as scratches, dents, structural defects like distorted objects, and the absence of parts were considered. Several approaches were tested to improve the results. AnoGan, L2 Autoencoder (AE), SSIM AE, CNN feature dictionary, and variation model were considered. The best result was achieved with the L2 AE with a ratio of 98.00% for correctly classified defect-free screws and 39.00% for defect screws. This corresponds to an accuracy of 83.37%. For accurate quality control in the industrial field, the accuracy is a very important indicator and could be considered as a limitation for this model.

Another anomaly detection approach on the MVTec Anomaly Detection data set of Bergmann et al. [22] uses a technique to visually explain variational AEs via gradient-based attention. By using attention maps, they can localize anomalies in images. They reported the area under the receiver operating characteristic curve (ROC-AUC) with 0.97 for the screw data set [38]. Furthermore, a third approach on the same data set was made by Bergmann et al. [39]. They present a student-teacher framework for unsupervised anomaly detection as well as pixel-precise anomaly segmentation. They report the normalized area under the PRO-curve of 0.928 for the screw anomaly detection. The best result achieved by those anomaly detection approaches is a ROC AUC of 0.97, which is shown in Table 1.

**Table 1. Approaches for automated defect detection of screws.**

Year	Reference	Method	Performance
2018	Song et al. [1]	DCNN	98.40%
2017	Wu et al. [17]	CNN	90.00%
2019	Yang et al. [23]	CNN	79.40% -99.00%
2019	Bergmann et al. [22]	L2 AE	83.37%
2020	Liu et al. [38]	Variational AE	AUC-ROC: 0.97

In conclusion, Song et al. [1] applied a deep CNN-based method to detect micro defects of metal screw surfaces from a top view of mounted screws. An adaptable model was created by Wu et al. [17] by using a CNN approach to detect defects in texture images, special structural images such as metallic gasket and screw images. Only one defect type was taken into account for this approach. Defect darning needles can be detected using several single models with the approach by Yang et al. [23], which is not practical in an industrial process.

### 3. Methodology

For the Deep Learning approach, we carried out detailed literature research [40, 41] and considered the specific machine learning guidelines [24, 42] to be able to evaluate the results thoroughly.

#### 3.1 Model Architecture

Our network architecture is based on VGG16 CNN network architecture, which was presented by Simonyan and Zissermann [36]. We adjusted the VGG16 base model to achieve the best result. We chose the VGG16 architecture because of its ability to solve complicated object detection problems, and because of its representation depth, it is suitable for the accuracy of classification. As a result of the depth and number of fully connected nodes, the architecture requires much memory, but this does not pose a problem as powerful hardware is available for industrial inspection and manufacturing [43]. The architecture consists of five blocks of convolutional layers. They all have a kernel size of  $3 \times 3$ . After every block, a max pooling 2D is applied to sample down the input. By using pooling layers, the resolution of the feature maps is decreased, allowing a spatial invariance to be generated [44]. We used two dense layers before the classifier, one with a filter size of 128 and the other one with 32. In addition, two dropout layers were applied, both with a rate of 0.5, to prevent the model from overfitting [45]. For the layers, the activation function rectified linear unit (ReLU) was used. With layer weights, regularizers penalties are imposed on the layer parameters. Using the kernel regularizer L2, penalties are applied to the layer's kernel. The regularizer was set to the default value of  $l2=0.01$  [32]. For the output layer, which is a dense layer, the sigmoid activation function was applied, which is commonly used for binary classification as it returns a value in the range of 0 to 1. The filter size of the output layer is 1.

For the training, the layers of our network were frozen so that the weights do not differ. The Adam optimizer [46] is applied because the optimizer has the

benefits of two other optimizers, which are the RMSprop [45] and Adagrad [46]. Our architecture has a total of 14,784,513 parameters. After the training, the model can be optimized with Fine Tuning methods. For our model, we unfroze the last ten layers to make small adjustments to improve the performance. Furthermore, the Adam optimizer was set to a learning rate of  $5 \times 10^{-5}$ . The training and Fine Tuning were done with 200 epochs and a batch size of 32. Subsequently to the Fine Tuning, the model can be evaluated by calculating several performance indicators such as balanced accuracy, kappa, and precision of the model and providing a confusion matrix.

#### 3.2 Convolutional Neural Network and Transfer Learning

A widely used Deep Learning method are CNNs. In the past, CNNs already achieved outstanding results in the detection, segmentation, and recognition of objects and areas in images [49, 50]. This approach is also cost-efficient, robust, and fast [2]. CNNs are used to convert data that is in the form of multiple arrays, such as a color picture consisting of 2D arrays containing pixel intensities in the three-color channels.

A CNN is organized into several stages. These stages consist of different convolutional and pooling layers. The units of convolutional layers are organized into feature maps. Each unit is connected to local patches in the feature map of the previous layer by a filter bank. The result of this locally weighted sum is then passed through a non-linearity, e.g., a ReLU [24].

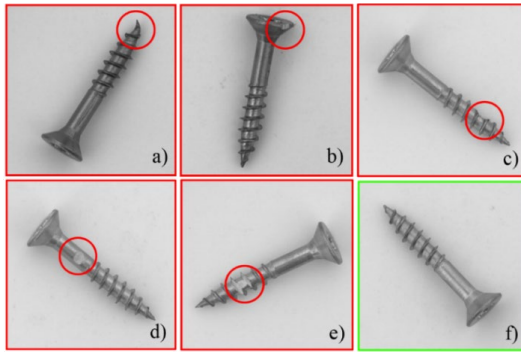
CNNs need a large number of images to be able to classify reliably. This has the effect that training an architecture requires a lot of processing capacity and time. Sometimes it is not possible to collect a large quantity of training data. With the help of high-performance learners, it is possible to train a CNN successfully even with only a small amount of data. This technique is called Transfer Learning [51]. Transfer Learning has already been effectively applied for image classification [52, 53].

The training of a CNN consists of two steps. First, the relevant structures and shapes are learned, from which more abstract objects can be derived and recognized. In the second step, a fully trained network is established based on the features already learned. Here Transfer Learning is being applied. During this process, Fine Tuning is utilized. Here, a transfer of the trained layers only occurs on the output layer. The output layer is adapted by adjusting the number of object classes of the new model to be detected and newly trained [54, 55]. Successful implementation of Fine Tuning has already been taken place in several applications [56, 57].

For the CNN architecture, we used the network architecture VGG16. This network was introduced by Simonyan and Zissermann [36]. The architecture consists of a total of 16 weighted layers and is used for large image recognition. The size of the convolutional filters is only  $3 \times 3$ , and the depth of the network can be increased. All convolutional layers have the same filter size. The five max pooling layers are in a kernel size of  $2 \times 2$ . The activation of all hidden layers takes place with ReLU. In total, the network consists of 13 convolutional layers and two fully connected layers, each of which has a total of 4,096 units. The final layer is activated with softmax and with 1,000 nodes. The default input size for the VGG16 architecture is  $224 \times 224 \times 3$  [58, 59].

### 3.3 Dataset and Preprocessing

The data set used for our training and evaluation of our Deep Learning approach is the MVTec Anomaly Detection data set by Bergmann et al. [22]. The data set was collected in 2019 and consists of 15 categories, including images of several objects and textures. The data set contains a total of 3,354 images of several objects and textures. Out of the 15 categories, we consider the screw images to detect defective and non-defective parts. The screw images can be categorized into two classes. The first class consists of correct screws, whereas the second class includes screws that have defects on the surface of the objects, like scratches and dents.



**Figure 1. Defective screws with defects of the front a), head b), thread side c), neck d), and thread top e), and a non-defective screw f).**

The defects were created manually with the aim to produce realistic anomalies similar to real-world industrial inspection scenarios. The images were acquired with a high-resolution industrial RGB sensor with  $2,048 \times 2,048$  pixels. The screws have different damage types, which can be seen in Figure 1. The damage types on the screws include: a) the screw front, b) the screw head, c) the screw thread side, d) the screw

neck, e) the screw thread top. Image f) shows an example of a defect-free screw.

The data set for screws consists of 119 images of defective and 361 images of non-defective screws. For preprocessing, a balanced data set of 119 images for each class has been selected. Therefore, all images of the defect class were used, and 119 images out of the non-defective screw class were chosen randomly. Afterward, a split into a train, test, and validation set was carried out. The data set was split into a ratio of 60:20:20. Thus, the number of training images is  $n=142$ , the number of validation images is  $n=48$ , and the number of test images is  $n=48$ . The model was trained with the training and validation images. For the calculation of the performance indicators, only the unseen test images were used.

The RGB-colored images were resized to  $224 \times 224$  pixels and normalized into the range of  $[-1,1]$ . Resize and rescale were applied to all images of the balanced data set. Furthermore, Data Augmentation was used to prevent overfitting and to improve the performance of the classification. Therefore, only the training images were rotated [37].

### 4. Results

For the training of the CNN, Keras 2.3.1 was used with Tensorflow 2.0 backend. For training of the model, we used the publicly available Google Collaboration service. It uses an Nvidia Tesla K80 12 GB GPU. To be able to detect the defect screw, the model was trained with 200 epochs and a batch size of 32. For the Fine Tuning, the learning rate was set to  $5 \times 10^{-5}$ . To evaluate the model, a confusion matrix has been calculated as well as several performance indicators like the balanced accuracy, kappa, precision, recall and the area under the receiver operating characteristic curve (AUC-ROC).

Thus, it was possible to achieve a balanced accuracy of 97.92%. The model predicted all 24 defective screws right. Unfortunately, one screw was wrong classified as a defective screw, and 23 were correctly predicted as defect-free screws. Table 2. shows the summarized classification results in a confusion matrix.

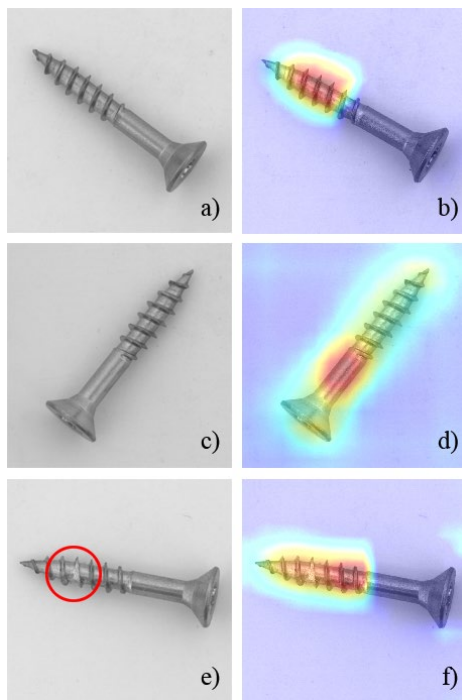
**Table 2. Confusion matrix.**

		Reference	
		<i>Defective</i>	<i>Non-Defective</i>
Prediction	<i>Defective</i>	50.00% (24)	2.08% (1)
	<i>Non-Defective</i>	0.00% (0)	47.92% (23)

In Table 3., the performance indicators are presented. Cohen's kappa has a value of 93.96%, and the model reached a true positive rate of 100.00%. The positive predictive value is 96.00% and the AUC-ROC equals 0.990. These results prove a good performance of the model.

**Table 3. Performance indicators.**

Performance Indicator	Value
Accuracy	97.92%
True positive rate	100.00%
Positive predictive value	96.00%
Prevalence	50.00%
Balanced Accuracy	97.92%
Kappa	93.96%
AUC-ROC	0.990



**Figure 2. Examples of a false classified non-defective screw (image a) and heatmap b)), correctly classified non-defective screw (image c) and heatmap d)), and correctly classified defective screw (image e) and heatmap f)).**

To understand why the model falsely predicted one defect-free screw as defective, we used the Grad-CAM heatmap algorithm [60] to visualize the important areas. In order to create the heatmaps, the images for the Grad-CAM are up- and downscaled and therefore not pixel accurate [61].

The one falsely predicted screw image a) in Figure 2. is defect-free but was predicted as defective. The heatmap b) in Figure 2. shows that the model mainly focused on the thread of the screw, which is colored in red. The rest of the image, which is colored in blue, is not predictive for the model. Image c) shows a correctly classified non-defective screw. Heatmap d) shows that the whole screw is recognized, but the focus is mainly on the neck of the screw. A correctly predicted defective screw is shown in image e) in Figure 2. The heatmap f) indicates the focus on the thread of the screw, where the defect is located. Thus, the area of the defect is clearly shown.

## 5. Discussion

As shown in Table 3., our CNN model performs efficiently and shows excellent classification outcomes. The performance is also shown in the confusion matrix in Table 2., supporting the good results. Further, the confusion matrix displays that all types of defective screws were correctly classified, while only one image out of 24 defect-free images is falsely classified.

This might be because some of the screw defects are just slight defects. Furthermore, the lighting can make it appear as if the surface has been scratched [62, 63]. Therefore, the model has probably classified the screw as faulty. The evaluation of the heatmaps shows that the network focuses especially on the parts where the defects are located. This includes the screw neck and thread, as shown in Figure 2.

The related work shows that a CNN approach is highly promising for the defect detection of screws and tiny metal parts. Also, an anomaly detection with an L2 AE approach was used to detect such defects in screws [22]. The current studies achieve an accuracy from the range of 79.40% to 99.00% for defect detection. However, those approaches rely on additional imaging techniques or simply focus on the detection of only single defect types [19]. If different defects are considered, several models were needed to detect those with a high level of accuracy [23].

Therefore, with an accuracy of 97.92%, our model, which can detect different defect types, also shows clear advantages compared to models with better performance results like the work by Yang et al. [23]. Even though they considered realistic quality inspection scenarios in their experiment, the four defects could only be detected with separate models. This means that only one defect type could be predicted with a 99.00% accuracy, which might not be practicable in real-world manufacturing.

Furthermore, with an accuracy of 98.40%, high-performance results were also achieved by Song et al. [1]. However, they only considered very specific



perspectives. The top view perspective of mounted screws used in this work is not suitable for the realistic screw production process.

We compared the prediction time of our model on CPU and GPU with different batch sizes to evaluate the suitability of our approach for an application during production. Table 4. shows good inference times for CPU as well as for GPU. With an increase of the batch size, the time the computer needs for each step is doubled. This applies to the performance speed of CPU and GPU. A fast performance speed enables the possibility to be used in real-time applications.

Smaller and lighter models are very useful for portable solutions but less suitable for very complex models [64]. The larger model VGG16 is able to solve very complex object recognition problems. In industrial inspection and manufacturing, high-performance hardware is available, making our approach highly practicable [36, 43, 65].

**Table 4. Comparison of inference time on CPU and GPU.**

Batch Size	CPU	GPU
1	320 ms	24 ms
2	614 ms	44 ms
4	1166 ms	75 ms
8	2104 ms	92 ms
16	3225 ms	167 ms

**Table 5. Performance indicators of other transfer-learning architectures.**

Architecture	Accuracy
Xception [42]	83.30%
VGG19 [36]	71.67%
InceptionResNetV2 [67]	61.65%
ResNet50 [49]	55.00%
EfficientNetB3 [68]	53.36%
EfficientNetB4 [68]	52.81%

Our model outperformed several other architectures, which were tested and evaluated for this approach. The pre-trained VGG16 network performed overall with the best results and an accuracy of 97.92%. Only one screw image was falsely predicted as defective. Table 5. shows that the Xception architecture performed with an accuracy of 83.30% and achieved the best results of the compared networks. The light Xception architecture is very easy to be defined and modified. Also, architectures such as VGG16 have those properties [36, 42, 66].

Other networks like VGG19 and InceptionResNetV2 performed with an accuracy below

80% and therefore were not considered any further [36, 67]. The ResNet50 architecture did not perform well, and so this approach was not considered in detail [49]. The EfficientNetB3 and EfficientNetB4 are mobile-sized models and can be scaled up, but in our use-case only achieved an accuracy of under 60% [68].

## 6. Conclusion

We built a highly effective defect detection model that achieves an excellent defect detection performance based on a CNN approach. Further, the performance of our model based on a pre-trained VGG16 architecture outperforms the previous studies and shows that a CNN network can be used to detect and identify several defects during quality inspection for screws. The small and simple model concerning computational costs accomplishes a fast training and outcome [36].

It consists of a pre-trained VGG16 architecture used for Transfer Learning. Furthermore, Data Augmentation was performed on the training data. This work is one of the first approaches with images in a real-world industrial inspection scenario to detect the defect screw class with five realistic defect types and the defect-free screw class with an accuracy of 97.92%. The model is accurate, cost-efficient, and fast compared to manual quality control methods. For training and testing, side-view images were used, which were taken with a random rotation of the screws and realistic damages [22].

Our fast and reliable model is able to detect five realistic defect types like scratches and dents on the surface as well as distorted and missing object parts located at the screw front, head, neck, thread side, and thread top as it occurs in a real-world industrial inspection scenario [22]. Therefore, our model is suitable for real-time in-situ defect detection [14, 15].

In addition, the heatmaps of our network show the high reliability in defect detection. The important areas and different damages on the screw surface are highlighted in the Grad-CAM heatmaps, as shown in Figure 2.

The internal validity of our model is high due to the train-test-validation-split, where testing is only carried out on unseen test images [69–71].

All defect parts were predicted correctly. Thus, our model contributes to zero-defect manufacturing. It is not only about the improvement of quality but also the learning process of quality inspection by identifying and analyzing any outliers in the process. Any potential risk for the quality has to be considered and not just an improvement of the yield [9, 10, 72].

The CNN model for the defect detection of screws has a practical relevance because of the realistic approach according to the industrial production

processes. The quality control can perform much faster and efficient, thus, to reduce manufacturing costs and improve the quality and yield [1, 2]. Our model is suitable for the large number of images and the high-speed inspection in the industrial environment.

## 6.1 Limitations

The main limitation of this work is based on the limited number of images that were provided by the MVTec Anomaly Detection dataset [22]. Only two classes could be predicted due to the small data set. This might not represent the whole complexity of the quality control process in the industry with large manufacturing data.

Changing lighting conditions, which were not considered, might be another limitation, as well as the manually generated defects of the data set [22, 62, 63]. The manual defects might not represent all real-world defects, like very fine scratches or dents caused by the machines. A problem could be that several defects can occur on one screw instead of only one defect per screw and image. Also, only one perspective of the screws was considered, which might be a limitation as the screws are moving in manufacturing processes. The occurrence of dust and/or dirt on the surface was not examined, which could occur in a manufacturing environment and lead to wrong predictions [2, 22]. This problem might limit the model and should be considered and evaluated.

A limitation is that k-fold cross-validation is not performed on the data set yet. This computationally intensive method could improve the robustness of our model and accuracy to a more accurate result [64–66].

More data could further improve the accuracy and enhance the prediction performance. Another aspect that can be viewed critically is the absence of external validation. The model needs to be tested in a real manufacturing environment.

## 6.2 Future Work

In the future, a more extensive dataset can be used to further evaluate the reliability of the proposed approach. It would also enable a more specific prediction for the different defect types. In the current classes of defective screws, five different defects are presented. A larger data set would make it possible to achieve a differentiated recognition of the defects. Thus, a more precise quality evaluation is achievable. Conclusions can be drawn, for example, to indicate errors on machines if one type of error occurs very frequently, in regards to the zero-defect manufacturing [7–10, 72]. Furthermore, the number of individual faults provides information about the production process and quality. An unsatisfactory result can lead to the

consequence that the production process is optimized to avoid the respective defect.

Another approach for the future could be the prediction of defective screws with more than one defect. It is possible that screws get more than one defect during the production process. In order to achieve accurate quality control, all defective parts of a screw should be recognized.

Another direction in which could be worked on in the future is to make the model available for mobile devices. Due to the fast performance speed of our architecture for CPU and GPU, it could be possible to run it on a mobile application. This would enable the usage of the model in further areas of quality control. Thus, it could be applied in every step of the production and delivery of screws. For instance, to check the screws randomly at the incoming goods inspection [25, 64].

To further improve the performance accuracy of the model, stacking could be useful. By combining two or more CNN network architectures, the prediction could be increased to create a more reliable and robust model for screw defect detection [73]. A further approach is to apply stronger Data Augmentation. Besides the already applied rotation, functions like brightness, blur or zoom can be useful to make the model more robust.

An additional line of research to be followed in the future is to analyze how human-computer interaction during workforce collaboration and automated quality control in production impacts user workload and user-oriented concepts [74]. We are also planning to carry out experimental work to evaluate the acceptance of technology and trust in our embedded module and confirm whether the automated method enhances coordination more efficiently [75].

## 7. References

- [1] L. Song, X. Li, Y. Yang, X. Zhu, Q. Guo, and H. Yang, "Detection of Micro-Defects on Metal Screw Surfaces Based on Deep Convolutional Neural Networks," *Sens.*, vol. 18, no. 11, pp. 3709–3723, 2018.
- [2] Q. Dong, A. Wu, N. Dong, W. Feng, and S. Wu, "A Convolution Neural Network for Parts Recognition Using Data Augmentation," in *WCICA '18 Proc.*, 2018, pp. 773–777.
- [3] M. Han, Q. Wu, and X. Zeng, "Single-Scale Workpiece Defect Detection Based on Deep Learning," 12th International Congress on Image and Signal Processing, in *CISP-BMEI '19 Proc.*, 2019, pp. 1–5.
- [4] C.-F. Lin, S.-F. Lin, C.-H. Hwang, H.-K. Tu, C.-Y. Chen, and C.-J. Weng, "Real-Time Image-Based Defect Inspection System of Internal Thread for Nut," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 8, pp. 2830–2848, 2019.
- [5] H. Ukida, "Visual Defect Inspection of Rotating Screw Heads," in *SICE '07 Proc.*, 2007, pp. 1478–1483.



- [6] E. S. Gademawla, "Computer vision algorithms for measurement and inspection of external screw threads," *Meas.*, vol. 100, pp. 36–49, 2017.
- [7] J. Priede, "Implementation of Quality Management System ISO 9001 in the World and its Strategic Necessity," *Procedia Soc. Behav. Sci.*, vol. 58, pp. 1466–1475, 2012.
- [8] S. Choi, O. Cho, and E. Kim, "Design of Automated Screw Blade Welding Defect Detection System for Image Processing-based Zero-defect Mass-production," in *CSCI '15 Proc.*, 2015, pp. 845–846.
- [9] F. Psarommatis, G. May, P.-A. Dreyfus, and D. Kiritsis, "Zero defect manufacturing: state-of-the-art review, shortcomings and future directions in research," *Int. J. Prod.*, vol. 58, no. 1, pp. 1–17, 2020.
- [10] P. Martinez, M. Al-Hussein, and R. Ahmad, "Intelligent vision-based online inspection system of screw-fastening operations in light-gauge steel frame manufacturing," *Int. J. Adv. Manuf. Technol.*, vol. 109, no. 3-4, pp. 645–657, 2020.
- [11] M. Rezaei-Malek, M. Mohammadi, J.-Y. Dantan, A. Siadat, and R. Tavakkoli-Moghaddam, "A review on optimisation of part quality inspection planning in a multi-stage manufacturing system," *Int. J. Prod. Res.*, vol. 57, no. 15-16, pp. 4880–4897, 2019.
- [12] T. Wang, Y. Chen, M. Qiao, and H. Snoussi, "A fast and robust convolutional neural network-based defect detection model in product quality control," *Int. J. Adv. Manuf. Technol.*, vol. 94, no. 9-12, pp. 3465–3471, 2018.
- [13] J. Chen, Z. Liu, H. Wang, A. Nunez, and Z. Han, "Automatic Defect Detection of Fasteners on the Catenary Support Device Using Deep Convolutional Neural Network," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 2, pp. 257–269, 2018.
- [14] A. S. Prabuwo et al., "Automated Visual Inspection for Bottle Caps Using Fuzzy Logic," *TEM J.*, vol. 8, no. 1, pp. 107–112, 2019.
- [15] S. H. Haider, A. S. Prabuwo, and N. H. S. A. Siti, "Metal Parts Visual Inspection Based on Production Rules," *Appl. Mech. Mater.*, vol. 110-116, pp. 4091–4095, 2011.
- [16] S. Arsovska, A. D. Cheoka, and C. Y. Yatb, "Automated Visual Inspection Model for Screw Detection on the Moving Objects in Industrial Quality Control," in *ISM '19 Proc.*, 2019, pp. 1–21.
- [17] X. Wu, K. Cao, and X. Gu, "A Surface Defect Detection Based on Convolutional Neural Network," in *ICVS '17 Proc.*, 2017, pp. 185–194.
- [18] M. Gaertner, D. Sauter, H. Baumgartl, T. Rieg, and R. Buettner, "Multi-Class Emotion Recognition within the Valence-Arousal-Dominance Space Using EEG," in *AMCIS '21 Proc.*, 2021, pp. 1–10.
- [19] J. Gross, J. Breitenbach, H. Baumgartl, and R. Buettner, "High-Performance Detection of Corneal Ulceration Using Image Classification with Convolutional Neural Networks," in *HICSS-54 Proc.*, 2021, pp. 1–10.
- [20] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *J. Manuf. Syst.*, vol. 48, pp. 144–156, 2018.
- [21] B. Staar, M. Lütjen, and M. Freitag, "Anomaly detection with convolutional neural networks for industrial surface inspection," *Procedia CIRP*, vol. 79, pp. 484–489, 2019.
- [22] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD -- A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection," in *IEEE/CVF CVPR Proc.*, 2019, pp. 9584–9592.
- [23] J. Yang, S. Li, Z. Wang, and G. Yang, "Real-Time Tiny Part Defect Detection System in Manufacturing Using Deep Learning," *IEEE Access*, vol. 7, pp. 89278–89291, 2019.
- [24] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [25] J. Lehr, M. Schlüter, and J. Krüger, "Classification of Similar Objects of Different Sizes Using a Reference Object by Means of Convolutional Neural Networks," in *ETFA '19 Proc.*, 2019, pp. 1519–1522.
- [26] E. Yildiz and F. Worgotter, "DCNN-Based Screw Detection for Automated Disassembly Processes," in *SITIS '19 Proc.*, 2019, pp. 187–192.
- [27] F. Zhou, G. Liu, F. Xu, and H. Deng, "A Generic Automated Surface Defect Detection Based on a Bilinear Model," *Appl. Sci.*, vol. 9, no. 15, pp. 1–17, 2019, Art. no. 3159.
- [28] B. Zhang, P. Jaiswal, R. Rai, P. Guerrier, and G. Baggs, "Convolutional neural network-based inspection of metal additive manufacturing parts," *Rapid Prototyp. J.*, vol. 25, no. 3, pp. 530–540, 2019.
- [29] K. A. Gross, J. Lungevics, J. Zavickis, and L. Pluduma, "A comparison of quality control methods for scratch detection on polished metal surfaces," *Meas.*, vol. 117, pp. 397–402, 2018.
- [30] K. D. Joshi, V. Chauhan, and B. Surgenor, "A flexible machine vision system for small part inspection based on a hybrid SVM/ANN approach," *J. Intell. Manuf.*, vol. 31, no. 1, pp. 103–125, 2020.
- [31] T. Konrad, L. Lohmann, and D. Abel, "Surface Defect Detection for Automated Inspection Systems using Convolutional Neural Networks," in *MED '19 Proc.*, 2019, pp. 75–80.
- [32] F. Tao, Y. Zuo, L. D. Xu, and L. Zhang, "IoT-Based Intelligent Perception and Access of Manufacturing Resource Toward Cloud Manufacturing," *IEEE Trans. Ind. Inform.*, vol. 10, no. 2, pp. 1547–1557, 2014.
- [33] K. Iyshwerya, B. Janani, S. Krithika, and T. Manikandan, "Defect Detection Algorithm for High Speed Inspection in Machine Vision," in *IEEE ICSSS '13 Proc.*, 2013, pp. 103–107.
- [34] W. Zhai, J. Zhu, Y. Cao, and Z. Wang, "A Generative Adversarial Network Based Framework for Unsupervised Visual Surface Inspection," in *IEEE ICASSP '18 Proc.*, 2018, pp. 1283–1287.
- [35] G. Hu, P. Ji, J. Zhu, B. Wei, Z. Yan, and L. He, "High Performance Visual Inspection Service Architecture - Squeezing the Most Out of Commodity Servers," in *IEEE ICWS '18 Proc.*, 2018, pp. 375–378.
- [36] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *ICLR '15 Proc.*, 2015.
- [37] F. Chollet, "Keras Documentation," <https://github.com/fchollet/keras>, 2015.
- [38] W. Liu et al., "Towards Visually Explaining Variational Autoencoders," in *IEEE/CVF CVPR '20 Proc.*, 2020, pp. 8642–8651.
- [39] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Uninformed Students: Student-Teacher Anomaly Detection with Discriminative Latent Embeddings," in *IEEE/CVF CVPR '20 Proc.*, 2020, pp. 4183–4192.
- [40] J. vom Brocke, A. Simons, B. Niehaves, K. Riemer, and A. Cleven, "Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process," in *17th European Conference on Information Systems Proc.*, 2009, pp. 2206–2217.

- [41] J. vom Brocke, A. Simons, K. Riemer, B. Niehaves, R. Plattfaut, and A. Cleven, "Standing on the Shoulders of Giants: Challenges and Recommendations of Literature Search in Information Systems Research," *Commun. Assoc. Inf. Syst.*, vol. 37, pp. 205–224, 2015.
- [42] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *IEEE CVPR '17 Proc.*, 2017, pp. 1800–1807.
- [43] D. E. Baz, "IoT and the Need for High Performance Computing," in *IIKI '14 Proc.*, 2014, pp. 1–6.
- [44] D. Scherer, A. Müller, and S. Behnke, "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition," in *ICANN '10 Proc.*, 2010, pp. 92–101.
- [45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [46] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *ICLR '15 Proc.*, 2015, pp. 1–15.
- [47] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," in *ICLR '18 Proc.*, 2018, pp. 1–23.
- [48] J. C. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, 2011.
- [49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE CVPR '16 Proc.*, 2016, pp. 770–778.
- [50] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Adv. Neural Inf. Process. Syst.*, vol. 60, no. 6, pp. 84–90, 2017.
- [51] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A Survey of Transfer Learning," *J. Big Data*, vol. 3, no. 1, pp. 1–40, 2016.
- [52] X. He, Y. Chen, and P. Ghamisi, "Heterogeneous Transfer Learning for Hyperspectral Image Classification Based on Convolutional Neural Network," *IEEE Trans. Geosci. Remote. Sens.*, vol. 58, no. 5, pp. 3246–3263, 2020.
- [53] L. Wen, D. Lixin, X. Dong, and I. W. Tsang, "Learning with Augmented Features for Supervised and Semi-Supervised Heterogeneous Domain Adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1134–1148, 2014.
- [54] C. Iorga and V.-E. Neagoe, "A Deep CNN Approach with Transfer Learning for Image Recognition," in *ECAI '19 Proc.*, 2019, pp. 1–6.
- [55] G. Wang et al., "Interactive Medical Image Segmentation Using Deep Learning with Image-Specific Fine Tuning," *IEEE Trans. Med. Imaging*, vol. 37, no. 7, pp. 1562–1573, 2018.
- [56] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "From Generic to Specific Deep Representations for Visual Recognition," in *IEEE CVPRW '15 Proc.*, 2015, pp. 36–45.
- [57] O. A. B. Penatti, K. Nogueira, and J. A. dos Santos, "Do Deep Features Generalize from Everyday Objects to Remote Sensing and Aerial Scenes Domains?" in *IEEE CVPRW '15 Proc.*, 2015, pp. 44–51.
- [58] H. Qassim, A. Verma, and D. Feinzimer, "Compressed Residual-VGG16 CNN Model for Big Data Places Image Recognition," in *IEEE CCWC '18 Proc.*, 2018, pp. 169–175.
- [59] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep Fisher Networks for Large-Scale Image Classification," *Adv. Neural Inf. Process. Syst.*, vol. 26, pp. 163–171, 2013.
- [60] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," in *IEEE ICCV '17 Proc.*, 2017, pp. 618–626.
- [61] P. Morbidelli, D. Carrera, B. Rossi, P. Fragneto, and G. Boracchi, "Augmented Grad-CAM: Heat-Maps Super Resolution Through Augmentation," in *IEEE ICASSP '20 Proc.*, 2020, pp. 4067–4071.
- [62] A. Zhou, H. Zheng, M. Li, and W. Shao, "Defect Inspection Algorithm of Metal Surface Based on Machine Vision," in *ICMTMA '20 Proc.*, 2020, pp. 45–49.
- [63] Z. Li, J. Zhang, T. Zhuang, and Q. Wang, "Metal surface defect detection based on MATLAB," in *IEEE IAEAC '18 Proc.*, 2018, pp. 2365–2371.
- [64] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *IEEE/CVF CVPR '18 Proc.*, 2018, pp. 4510–4520.
- [65] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning Transferable Architectures for Scalable Image Recognition," in *IEEE/CVF CVPR '18 Proc.*, 2018, pp. 8697–8710.
- [66] A. Canziani, E. Culurciello, and A. Paszke, "Evaluation of Neural Network Architectures for Embedded Systems," in *IEEE ISCAS '17 Proc.*, 2017, pp. 1–4.
- [67] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," in *AAAI '17 Proc.*, 2017, pp. 4278–4284.
- [68] M. Tan and Q. Le V, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *ICML '19 Proc.*, 2019, pp. 6105–6114.
- [69] S. Yadav and S. Shukla, "Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification," in *IEEE IACC '16 Proc.*, 2016, pp. 78–83.
- [70] Y. Bengio and Y. Grandvalet, "No Unbiased Estimator of the Variance of K-Fold Cross-Validation," *J. Mach. Learn. Res.*, vol. 5, pp. 1089–1105, 2004.
- [71] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Stat. Surv.*, vol. 4, pp. 40–79, 2010.
- [72] U. Gabler, I. Osterreicher, P. Bosk, and C. Nowak, "Zero defect manufacturing as a challenge for advanced failure analysis," in *IEEE/SEMI ASMC '07 Proc.*, 2007, pp. 341–344.
- [73] O. Korzh, G. Cook, T. Andersen, and E. Serra, "Stacking Approach for CNN Transfer Learning Ensemble for Remote Sensing Imagery," in *IntelliSys '17 Proc.*, 2017, pp. 599–608.
- [74] R. Buettner, "Cognitive Workload of Humans Using Artificial Intelligence Systems: Towards Objective Measurement Applying Eye-Tracking Technology," in *KI '13 Proc.*, vol. 8077 of LNCS, 2013, pp. 37–48.
- [75] R. Buettner, S. Sauer, C. Maier, and A. Eckhardt, "Real-time Prediction of User Performance based on Pupillary Assessment via Eye Tracking," *AIS Trans. Hum.-Comput. Interact.*, vol. 10, no. 1, pp. 26–56, 2018.