2017 16th IEEE International Conference on Machine Learning and Applications

Supervised machine learning based surface inspection by synthetizing artificial defects

M. Haselmann, D.P. Gruber
Optical and Haptical Material Characteristics
Polymer Competence Center Leoben GmbH
Leoben, Austria
matthias.haselmann@pccl.at; dieter.gruber@pccl.at

Abstract— **The preparation of labeled training data for supervised machine learning methods involves a lot of effort. Regarding surface inspection tasks, this endeavor is often not economically reasonable. In this paper, an artificial defect synthetization algorithm based on a multistep stochastic process is proposed. It adds defects to fault-free surface images, which can be used for supervised machine learning. By this means a deep convolutional neural network has been trained, achieving a detection rate of 94% of occurring real defects on the presented test surface.**

Keywords-inspection; surface images; image synthetization; artificial defects; supervised machine learning; convolutional neural nets; deep learning

I. INTRODUCTION

A. Motivation and Related Work

Whether a product surface is flawless or not, has a large impact on its perceived quality and therefore crucially contributes to a buying decision. For consumer products like cars, cell phones, etc., especially for the respective premium variants, a flawless surface is an absolute must and necessitates an inspection of all visible surfaces of all fabricated units. However, such an endeavor is a monotonous and exhaustive task for human raters. It is not possible to keep the concentration over a prolonged time period, which frequently leads to varying and inconsistent quality assessments. Consequently, automated quality inspection systems are of great interest from an industrial point of view.

While for plain and homogenous surfaces an inspection algorithm can be comparably easy implemented on a hard-coded rule base [1], there is far more effort needed for patterned, structured and/or complexly shaped surfaces. The distinction of patterns and structures including allowed part-to-part variations from visually perceived imperfections is a major challenge. Therefore, over the past few years more frequently machine learning techniques got the means of choice for complex inspection tasks [2,3,4].

One major drawback of machine learning based approaches, is their demand for a sufficiently large number of training data to achieve a good generalization on the problem [5,6]. For the supervised case there is also a manual labeling of occurring defects in the training data needed, which in addition should also appear in a sufficiently large number in the training data. For surface inspection tasks where the occurrence of some rare defects is often arbitrarily and not reproducible, these requirements are difficult to fulfill at a reasonable expense.

B. Approach

In this paper, an artificial defect synthetization algorithm is proposed with the aim to enable supervised machine learning for surface inspection tasks where no labeled defects can be provided for the model training. The presented approach is based on a multistep stochastic process that generates defect textures that are added to fault-free surface image patches. Therefore, in the manual selection process of training images only surface images that are mainly fault-free have to be included. Consequently, with the exception of a few outliers, all extracted image patches can be considered and labeled as fault-free. Image patches, labeled as defective, are then only provided by carefully manipulating their fault-free pendants by means of the proposed algorithm. For this reason, a manual labeling is not necessary.

In order to demonstrate the approach on a real world inspection task the proposed synthetization algorithm is applied on images of automotive interior parts described in section II. Image patches with synthetic defects together with non-manipulated fault-free image patches are fed to a deep convolutional neural network (DCNN) that is evaluated on real defects, eventually. The obtained results suggest a good generalization of the DCNN model between fault-free and defective looking image patches. The presented approach is demonstrated on one example surface, though the principle is supposed to be also applicable on other surface types with similar occurring defect morphologies.

II. TEST PARTS AND SURFACE DEFECTS

The example objects are complexly shaped decorative plastic parts that are manufactured with the so-called foil-insert-molding (FIM) process [7]. This is a special type of polymer injection-molding process, where a preformed decorated foil is inserted into the cavity of the injection molding machine. The polymer mass is injected afterwards. The surface design is apart from the surface geometry

0-7695-6321-X/17/31.00 ©2017 IEEE
DOI 10.1109/ICMLA.2017.0-130

exclusively determined by the imprinting process of the foil that takes place beforehand the actual injection molding process. It is therefore easily interchangeable without expensive mechanical modifications on the fabrication tools. Furthermore, FIM parts show a higher surface robustness and scratch-resistance than usual injection-molding products. Overall, the FIM process allows the fabrication of high-quality plastic parts that are frequently used for premium products like high-priced cars.

One significant characteristic of FIM products are slightly varying foil distortions due to the thermoforming process in which the geometry of the inlay foil is preformed with regard to the final surface geometry of the plastic parts. In case of an imprinted pattern this leads to slight random pattern distortions and shifts especially for complexly shaped surface geometries. Furthermore, when the foil consists of multiple imprinted layers, even the visual appearance of pattern primitives can vary from part to part. In fact, all manufactured parts are unique in their surface texture details (compare Fig. 1). For humans those part-to-part variations do not stand out at all and can only be noticed by comparing two different parts. However, from the perspective of computer-aided inspection those characteristics are challenging, since there is no constant reference part (golden sample) available for comparison with the parts to be inspected. An inspection algorithm has to reliably distinguish between allowed part-to-part variations and visually perceived imperfections [8].

The focus of this paper is mainly on tiny surface imperfections that are quite difficult to spot for a human. They usually extend over a few pixels only in a surface picture and stand out weakly in terms of contrast (compare Fig. 2). Depending on the defect type and the used measurement geometry they can appear bright or dark contrasted. The demonstrated example surface is photographed in a bright-field setup. The majority of the light originating of the illumination unit is reflected by the part's surface directly to the camera direction. Therefore, the majority of occurring surface disturbances appear as dark contrasted structures.

Since the presented test parts are complexly shaped they were photographed in four different positions to cover a large area of the surface. The training data contains pictures of 31 different parts where occasionally some positions were excluded that showed obvious defects. The region of interest of each of the four positions was determined manually by drawing four masks. Image patches of a size of 51x51 pixels were extracted that way that no corner was visible in the extracted patches. That way roughly 21 million different image patches could be extracted. It must be noted that this set of image patches is highly correlated, because they can differ from each other only by shifting the extraction window by one pixel. Hence, this patch set roughly corresponds to 8000 uncorrelated image patches.

The validation data comprises 9 parts, again with the exclusion of images showing image regions with defects. In comparison to that, the test data consists of 19 parts. In this data set occurring defects where manually labeled for the evaluation of the trained DCNN on real defects.

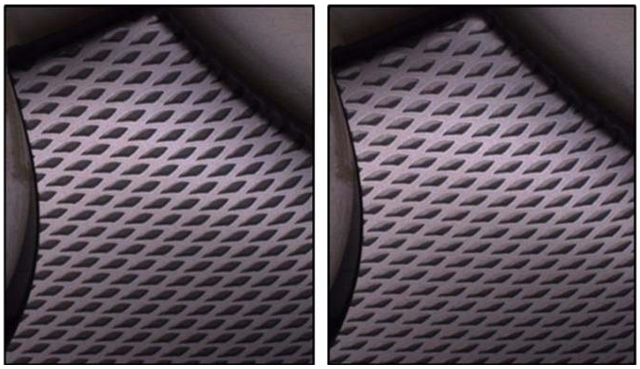

Figure 1. Surface pictures of two different Foil-Insert-Molding (FIM) parts that show the same region. Note the significant appearance variations of the surface pattern.

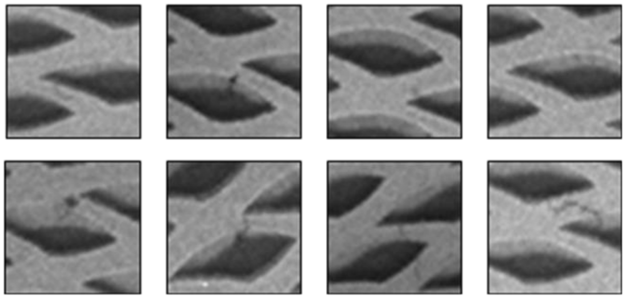

Figure 2. Image patches of size 51x51 pixels that show real defects. Since these defects are very small and often weakly contrasted, they are difficult to find for a human on a large surface region.

III. Synthetizing Artificial Defects on Surface Images

The generation of artificial image data as an approach for enabling machine-learning was already used in cases where the acquisition and labeling of a large set of real images comes with a great expense and/or is not possible for various reasons. One example is the automated recognition of number plates [9], where artificially generated images of number plates including random affine transformations are inserted in an arbitrarily selected image scene. The result is an image generator that can provide any count of unique images, which is helpful for training a complex machine learning model like a deep convolutional neural network (DCNN).

Regarding the synthetization of surface defects corresponding to a few pixels our approach can be roughly divided into four steps. The first step concerns the generation of a morphological defect skeleton by means of a stochastic process that can be considered as a random walk with direction momentum. In the second step, a texture of the skeleton is randomly generated and widened afterwards. In the third step, the synthetized image patch of the defect texture is used to modify a randomly picked fault-free image patch that shows the type of surface to be inspected. In the final step, the defect visibility in the modified image is analyzed.

A. Skeleton generation

The used skeleton generation algorithm resembles a random walker with momentum in a 2-dim space [10]. The random walker starts from a random point $(x^0, y^0) \in R^2$ with $a < x_0 < d_x - a - 1$, $a < y_0 < d_y - a - 1$ with a step size s and a total number of steps l, where dx and d_y are the size of the image patches and a is the minimum distance between the starting point and the image edges. The coordinates $(x^{(i)}, y^{(i)})$ of the i^{th} point resulting from the subsequent stochastic process are obtained by

$$\theta^{(i)} = \theta^{(i-1)} + X_\theta{}^{(i)} \Delta\theta^{(i)} \quad (1)$$

$$\varphi^{(i)} = \varphi^{(i-1)} + X_\varphi{}^{(i)} \Delta\varphi^{(i)} + \theta^{(i)} \quad (2)$$

$$x^{(i)} = x^{(i-1)} + s \cos \varphi^{(i)}, \quad y^{(i)} = y^{(i-1)} + s \sin \varphi^{(i)} \quad (3)$$

where $\varphi^{(i)}$ is the polar angle determining the movement direction of the random walker at the i^th^ step. The random variable $X_\varphi{}^{(i)}$ that follows a Bernoulli distribution with p_φ determines whether $\varphi^{(i-1)}$ is changed by $\Delta\varphi^{(i)}$. In addition a polar angle bias θ is introduced that causes a persistent change of the polar angle for several steps. θ itself is changed by $\Delta\theta$ with the probability $X_\theta{}^{(i)}$ which in turn follows a Bernoulli distribution with p_θ. $\Delta\varphi^{(i)}$ and $\Delta\theta^{(i)}$ are uniformly distributed random variables in the intervals [-ϑ,ϑ] and [-Θ,Θ], respectively. In addition, the parameters s, p_φ, p_θ, ϑ, Θ and l are also uniformly distributed random variables, which however are only measured once at the start of every stochastic defect generation process. The corresponding intervals are hyper parameters and have to be chosen manually.

The probability p_θ has a large effect on how frequently bulky defects or defects with a large curvature are generated. In the presented experiments its range is chosen low in comparison to p_φ so that also thin, scratch-like defects are generated. In general all hyper parameters should be chosen with the goal to generate the greatest possible diversity of artificially synthetized defects.

For every carried out stochastic skeleton generation process a binary skeleton image S is obtained by rounding all l points $(x^{(i)}, y^{(i)}) \in R^2$ to the nearest integer values $(m^{(i)}, n^{(i)}) \in Z^2$. This then determines the indices of the non-zero pixels. Indices that links to non-existent image regions are discarded.

B. Texture generation

In this step a texture image of the defect is generated on basis of the binary skeleton image. For all non-zero pixels $S_{m^{(i)},n^{(i)}}$ in the skeleton image the corresponding non-zero pixels in the preliminary texture image are obtained by:

$$T_{m^{(i)},n^{(i)}} = \tau_\mu + \tau^{(i)} \quad (4)$$

with τ_μ indicating the mean contrast of the defect texture and $\tau^{(i)}$ indicating a uniformly distributed random variable within the interval [-τ_ν,τ_ν]. The parameter -τ_μ and τ_ν are also uniformly distributed random variables that are measured once for every defect synthetization process. The preliminary texture image defined in (4) is finally blurred to obtain a smoother more natural looking defect.

C. Modification of a fault-free image patch

In the last synthetization step a fault-free surface image patch is manipulated by adding the previously generated defect texture. In this step a real surface images is required. Since our focus was so far only on the surface image introduced in 1.C it must be suspected that other surface types may require a modified version of the two last synthetization steps.

The fault-free image patch is manipulated by subtracting the texture image. In order to generate synthetized defects that appear as individual structures and do not strictly follow the gray levels of the regular surface image structure a gray level cutoff for the defective image region was used that results from the grey levels in the corresponding fault-free image region. In addition, a further random variable maintain the regular image noise in case that a large manipulated image region runs into the gray level cutoff.

The described modification step only leads to dark contrasted defect structures, which however in our case study represents the majority of real defect occurrences. In order to generate bright contrasted defect structures which may be necessary for other illumination setups or other surface types, the fault-free image patch is just inverted in advance of this synthetization step and inverted back afterwards. Of course, dark and bright contrasted defects can be mixed by introducing a binary random variable that decides between the two variants.

D. Analysis of the defect visibility

This analysis of the defect visibility is actually not part of the synthetization procedure. Nevertheless, it is necessary to discard synthetized defect images, where the added defect is so weak that it is simply not visible. If those images are kept, they negatively affect the training process of a machine learning algorithm leading to a significantly reduced classification accuracy.

Since for every synthetized defect image a fault-free clone exists, the analysis of the defect visibility was done by subtracting both images. The sum of the squared residual image provides an acceptable estimator for the defect visibility under consideration of the defect size. Although, this approach helps to discard the majority of badly synthetized defect images, there occur some exceptions. In example in our case study, when a very thin defect is exactly added to the edge of a diamond shaped surface element, it only get thicker by one or two pixels which does not change the character of the element. For a human such an image still appears fault-free, although the residual image shows a sufficiently large signal for the synthetized defect image being accepted. Although, this problem could be attenuated by slightly blurring the fault-free image before the image

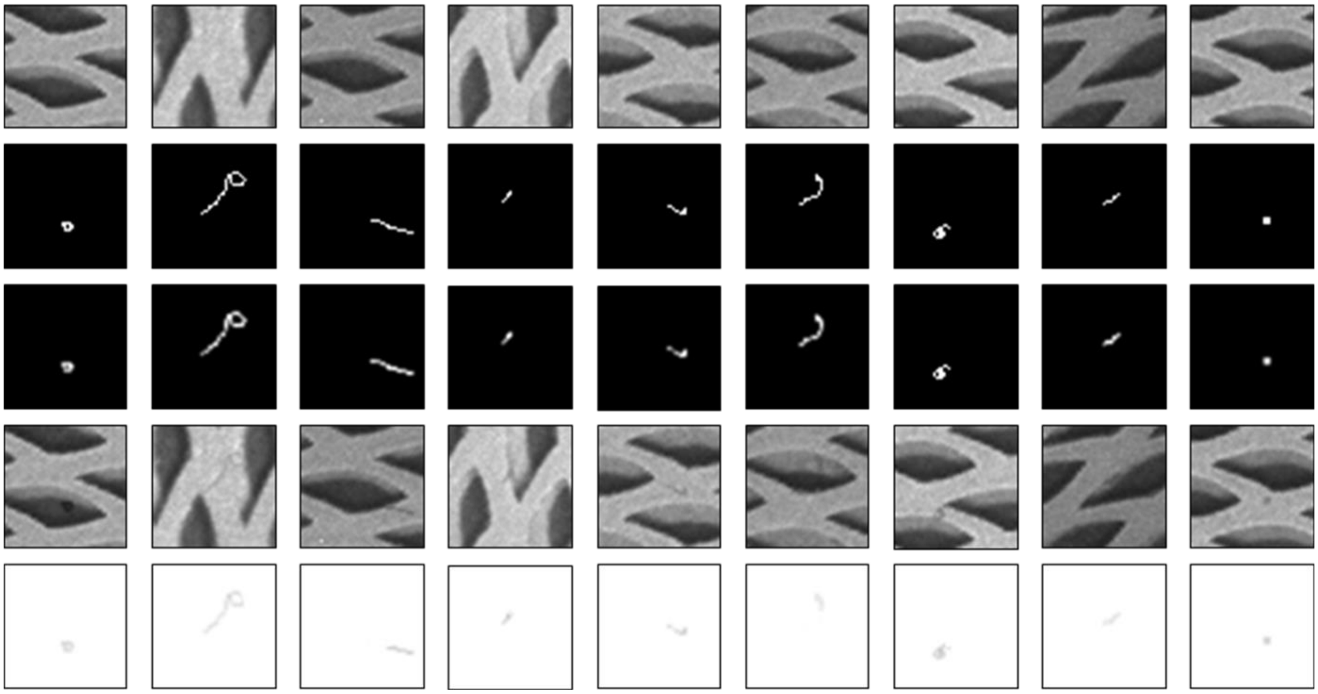

Figure 3. Stepwise synthetization of artificial defects on fault-free image patches that show the surface to be inspected. The first image row show the fault-free image patch to be manipulated. The second row shows the randomly generated defect skeleton. The third row shows the generated defect texture. The fourth row shows the artificial defect added to the fault-free source image. The 5th row shows the residual images used for analyzing the defect visibility.

subtraction, it could not be completely ensured that badly synthetized defect images are discarded with no exception.

Vice versa, there exists a reverse case where image patches – falsely labeled as fault-free – are fed to a machine learning algorithm. This happens particularly when some of the tiny defects are overlooked in the manual selection of the surface images.

IV. Machine Learning Model

A. Architecture

As machine learning methodology a deep convolutional neural networks (DCNN) was chosen. The corresponding network architecture was designed with consideration of the case study. Since the surface images are processed as gray-scale images due to defect synthetization algorithm, the input to the first convolutional layer is an l x l x 1 image, where l is the height and the width of the image. For the presented experiments the input image size was fixed to $l = 51$. Correspondingly, the surface images presented in 1.C were split into image patches of the same size.

The applied DCNN consists of seven layers which are (64C2)-(128C3-MP2)-(128C2)-(128C3-MP2)-(128C2)-(128C3-MP2)-(2N). This corresponds to six convolutional layers with max-pooling on every second layer and one fully-connected output layer with two units. Both, additional convolutional layers and one additional fully connected hidden layer did not show an increased performance of the DCNN for the test parts.

As activation functions Rectified Linear Units (ReLUs) were used over tanh-units except for the output layer, since they are generally known to show a shorter training time due to their more non-saturating behavior [11]. As output layer two softmax units were used – one each for the prediction of fault-free and defective image labels. This configuration is approximately analogous to an output layer with a single sigmoid unit, which would predict a binary variable linked to the binary classification task. However, the output layer with softmax units is easily extendable by additional units, in case of the requirement of a distinction between several defect classes.

In order to accelerate the training of the DCNN batch-normalization was used for all convolutional layers in advance to the activation functions. Batch-normalization is known to reduce the internal covariate shift. It generally facilitates the training process by allowing higher learning rates, making it less prone to bad initializations as well as acting as a regulizer [12].

B. Training

The DCNN model was learned from scratch using the Adam optimizer [13] with the hyperparameters $\alpha = 0.0005$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ $\varepsilon = 10^{-8}$ and a batch size of 128. The weights were initialized in each layer from a zero-mean Gaussian distribution with a standard deviation of 1. The biases were all initialized with the constant 0. An initialization of the biases in the same way as the weights showed a considerably worse training behavior. In order to enhance the model generalization a comprising data augmentation was implemented in the preprocessing step after the optional synthetization of defect images. It included random rotations, random shears, random zooms and random flips of the image patch. By using larger raw patches as input to the preprocessing pipeline no image extrapolation at the corners was necessary in the end. Such image artefacts, generated by extrapolation techniques would have a serious impact on the model's performance, since they often show similar characteristics as defects.

The network was trained for roughly 650k iterations which corresponds to a total number of 83.2 million image patches (compare Fig. 4). In view of a total number of 21 million raw image patches this corresponds to roughly 4 cycles. The training was performed in roughly 25 hours on a GTX1080. As deep learning framework TensorFlow [14] was used.

V. Experimental Results

A. *Classification result for synthetically generated defects*

The deep convolutional neural network described in section III achieved a maximum accuracy of 99.7% on the validation data set, which includes unseen surface images but only labeled defects that originate from the same defect synthetization algorithm used in the training stage. Fig. 4 indicates that the network is only slightly overfitting at the very end of the training process.

From this results it can be concluded that the model generalizes well on varying surface images. It is at least capable of a reliable distinction between unseen variations of surface patterns and synthetically generated defects. The question whether the model is also capable of detecting non-synthetized defects has to be answered on the test set.

B. *Result on the test set with real defects*

In comparison to the validation set the test set includes real defects that were labeled manually. Since, there is only a total of 35 labeled defects in the test set, the model performance was separately evaluated for defective and fault-free image patches. Fig. 5. illustrates the trained DCNN applied to the test set and also provide an idea of how large the image patches are in relation to one of the inspected surface regions.

In order to perform the inference on the test set in a practical manner, a sliding window with a jump distance of 20 pixel was used. Occurring defects were considered as detected (true positive defect detection results) when at least one overlapping window responds accordingly. If no overlapping window shows the proper response it is counted as a "not detected defect", otherwise as a "detected defect". On the other hand, if a fault-free image patch is predicted as defective it is counted as an "erroneously detected defect", otherwise as "correctly classified as fault-free".

The proportion of image patches "correctly classified as fault-free" is at 99.8% on the test set which is comparable to the maximum accuracy of the network on the validation set as expected. Regarding the 35 labeled defects in the test set, 33 could successfully detected by the DCNN, which corresponds to an accuracy of roughly 94%. Examples for "detected defects" and "not detected defects" and "erroneously detected defects" are shown in Fig. 6-8.

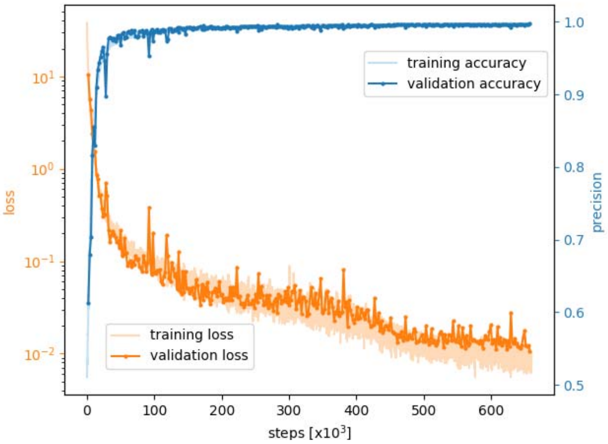

Figure 4. Training of the DCNN on surface images with artificially synthetized defects.

C. *Discussion and future directions*

The detection rate of real defects implies that the variety of synthetized defects of the proposed algorithm in section III leads to a solid generalization of the trained 7-layer DCNN between flawless and faulty looking images of the presented surface. However, there are two defects that were not properly detected by the network (see Fig. 8). On the other hand, there are defects properly detected that deviate significantly in its appearance from the majority of synthetized ones (compare the two image patch in the lower row of Fig. 6).

The results demonstrate the suitability of supervised machine learning methods for surface inspection tasks with the focus on tiny and weakly contrasted surface defects task despite the absence of real defects. Consequently, the total amount of surface images to be taken is much lower, since real defects have to be provided only for model testing. Especially for rare and randomly occurring defects the effort of the production of a large number of test parts, the time-consuming examination of the surface images and the manual drawing of corresponding ground truths for each image would be too large in an industrial environment.

While the initial results are very encouraging, the topic deserves further investigation, especially for other surface types or surface regions that show edges or other topographical structures. The inspection of defects that cover a larger surface region can be probably solved by a multiscale approach. Furthermore, a segmentation of detected defects would be useful for an automated inspection system, since it would allow a subsequent analysis of the defect morphology and even the visual perceptibility [15]. Former could be done with an accordingly modified DCNN and the same presented defect synthetization algorithm, since a detailed ground truth is automatically provided as a by-product.

VI. Conclusion

In this paper, an artificial defect synthetization algorithm is proposed to enable supervised machine learning for surface inspection tasks where no or not enough pictures

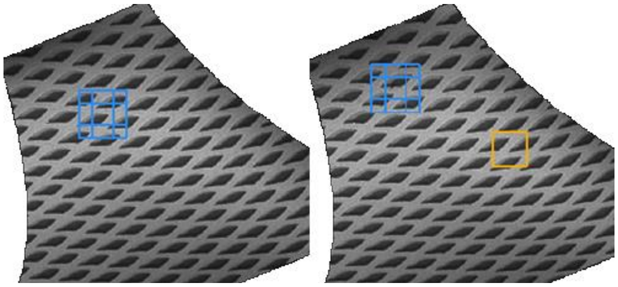

Figure 5. One out of four inspected regions of two different parts of the test set. The blue rectangular indicates a successfully detected defect. The yellow one indicates an erroneously defect detection event.

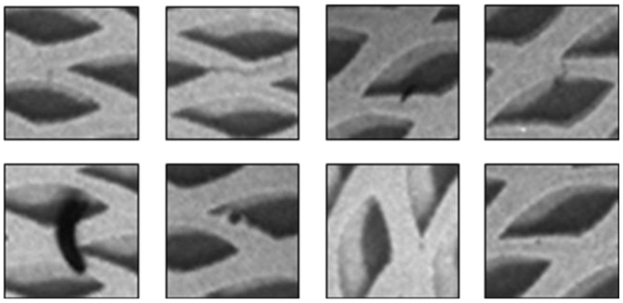

Figure 6. Some of the real defects in the test set that were successfully detected by the DCNN.

with labeled defects can be provided. The focus is on the detection of small and weakly contrasted defects, which only cover a few pixels in the surface image. The photographed surfaces shows a regular pattern that varies in its alignment and distortion from part-to-part.

The proposed artificial defect synthetization algorithm makes use of a stochastic process to generate a wide variety of defect skeletons. A texture of the skeleton is randomly generated and widened afterwards. In the final step the defect texture is added to a fault-free surface image patch and fed to a supervised machine learning model together with unmodified surface image patches. In this paper a deep convolutional neural network with 7-layer was successfully trained with an accuracy of 99,7% on images with artificially synthetized defects. In comparison to that 33 of 35 real defects could be detected in the test set.

ACKNOWLEDGMENT

The research work of this paper was performed at the Polymer Competence Center Leoben GmbH (PCCL, Austria) within the framework of the COMET-program of the Federal Ministry for Transport, Innovation and Technology and the Federal Ministry of Economy, Family and Youth with contributions by Schöfer GmbH. The PCCL is funded by the Austrian Government and the State Governments of Styria and Upper Austria.

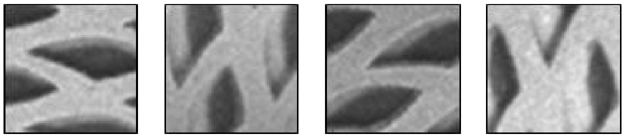

Figure 7. Exemplary fault-free image patches that were erroneously detected as defective.

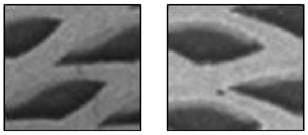

Figure 8. The two defects in the test set that were not properly detected by the DCNN.

REFERENCES

[1] C. Demant, B. Streicher-Abel and P. Waszkewitz, "Industrial Image Processing: Visual Quality Control in Manufacturing," Springer: Berlin, Heidelberg, 1999

[2] S. Ravikumara, K.I. Ramachandran and V. Sugumaran, "Machine learning approach for automated visual inspection of machine components," Expert Syst. Appl. 38(4), 2011, pp. 3260-3266

[3] M.A.F. Pimentel, D.A. Clifton, L. Clifton and L. Tarassenko, "A review of novelty detection," Signal Processing 99, 2014, pp. 215-249

[4] E. Weigl, W. Heidl, E. Lughofer, T. Radauer and C. Eitzinger, "On improving performance of surface inspection systems by online active learning and flexible classifier updates, Machine Vision and Applications, vol. 27, issue 1, pp. 103-127

[5] F. Pereira, P. Norvig and A. Halevy, "The Unreasonable Effectiveness of Data," IEEE Intelligent Systems, Issue No. 02, vol. 24, March/April 2009, pp. 8-12

[6] C. Sun, A. Shrivastava, S. Singh and A. Gupta, "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era", arXiv:1707.02968v1, 10 Jul 2017

[7] A. Martinez, J. Castany and J. Aisa, "Characterization of In-Mold Decoration Process and Influence of the Fabric Characteristics in This Process," Materials and Manufacturing Processes, Issue 9, vol. 26, 2011, pp. 1164-1172

[8] M. Haselmann and D.P. Gruber, "Anomaly detection on arbitrarily distorted 2D patterns by computation of a virtual golden sample", Image Processing (ICIP), 2016, pp. 4398-4402

[9] M. Earl, https://github.com/matthewearl/deep-anpr, 28.02.2017

[10] M. Haselmann and D.P. Gruber, "Machine Learning im Bereich der automatisierten Qualitätsinspektion von Dekormuster", 26. LKK, 2017

[11] V. Nair and G.E. Hinton, „Rectified linear units improve restricted boltzmann machines,", In Proc. 27th International Conference on Machine Learning, 2010

[12] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," arXiv:1502.03167, 2015

[13] D.P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization", arXiv:1412.6980, 2014

[14] M. Abadi et. al., "TensorFlow Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org

[15] D.P. Gruber, J. Macher, D. Haba, G.R. Berger, G. Pacher and W. Friesenbichler, „Measurement of the visual perceptibility of sink marks on injection modling parts by a new fast processing model," Polymer Testing, vol. 33, 2014, pp. 7-12