DCGAN-Based Data Augmentation for Enhanced Performance of Convolution Neural Networks

Christian Reser and Christoph Reich
Institute for Data Science, Cloud Computing and IT Security
Furtwangen University of Applied Science
Furtwangen, Germany
Email:{christian.reser, christoph.reich}@hs-furtwangen.de

***Abstract*—The quality of steel is essential for many products. Unfortunately, during the production process of steel, surface defects (scratches, inclusions, etc.) occur, resulting in financial losses for steel producers. Therefore, to find and classify surface damage at the earliest stage of the steel production process to take actions for mitigating quality is preferred. Recently, neural networks have shown the usefulness of image classification. Prerequisite is a large data set. But to collect a large data set often takes too long and is too expensive. This paper investigates how to handle smaller data sets, generate artificial data by augmentation and evaluate their efficiency. Of special interest is the augmentation of images by Deep Convolutional Generative Adversarial Networks (DCGANs). A detailed evaluation and comparision with other augmention techniques show that DCGAN augmentation outperformed other augmentations in accuracy and loss, but it is no replacement for a large data set.**

***Keywords*—Convolutional Generative Adversarial Network; Steel Surface Damage; Augmentation; Image Classification; Neural Network; Industry 4.0.**

I. INTRODUCTION

In the field of machine learning, image classification using convolutional neural networks is nowadays one of the most common approaches. Convolutional neural networks gained popularity because of their success in many image classification problems and the acceptable processing time through the availability of fair GPU (Graphics Processing Unit) prizes. Further, the training time has been cut down, by pre-trained deep convolutional neural networks, such as ResNet [1], which can classify thousands of images from the public available imagenet data set [2]. Mostly essential for a good performance of high accuracy and low loss of neural network results is a huge data set for the training. If there is no such huge data set, because of difficulties to collect (e.g., particle collisions), high expenses (e.g., deep water pictures), or high time consumption (e.g., seldom events), image augmentation can be the solution. This is often the case in the steel industry, as companies often do not find the time to collect enough images to create a good data set. This may require processes to be interrupted, which can lead to financial loss. Surface inspection would be so important for the industry, because it allows material defects to be detected early and sorted out before further processing. Typically, through augmentation, a data set can be expanded by flipping images (horizontally or vertically), apply random zoom, random rotation or random shear of images, for example. This method can make a machine learning model more robust, more accurate, and prevent it from overfitting, but only, if the data set itself has enough variations. Variations of images are: intra-class variation, scale variation, view-point variation, occlusion, illumination, background Clutter [3]. But often, the data set is too small and a model can become overfitted easily. To improve such small data set, a new approach is taken. The augmentation by Deep Convolutional Generative Adversarial Network (DCGANs) introduced by Radford et al. in [4]. DCGANs are a variation of the Generative Adversarial Networks introduced by Goodfellow in [5], especially for images.

In the next section (Section: II), related work is discussed. In Section III, the used steel image data is described and how the data is prepared for our experiments. Section IV will give information about the used augmentation methods of this work. Section V describes the used neural network architecture, the training method, and the evaluation method. The results of the experiments are shown in Section VI and in the last Section VII, a conclusion and an outlook are given.

II. RELATED WORK

The work from Shorten and Khoshgoftaar in [6] deals with the problem of limited data in data sets. They focus on data augmentation to enhance the size and quality of image data sets to get better training results and prevent overfitting at the same time. They provide an overview of all the different augmentation techniques. In general, there are two main branches of augmentation techniques. Basic image manipulations and deep learning approaches. The basic image manipulations takes one original image and performs different manipulations on it, such as geometric transformations or color space transformations. With these techniques, multiple images can be generated out of one original image to enhance the data set. Advanced techniques, based on Deep learning augmentation make use of Generative Adversarial Networks (GANs). These augmentation techniques will be used in this work, especially for generating new images for the NEU-DET data set [7].

He et al. in [8] developed a defect detection system to precisely classify and locate the damage on a steel plate surface. They used the NEU-DET data set [7] which is used in this work too. A detailed explanation of the data set will be given in Section III. He et al. used deep learning methods and gained a very high classification accuracy of almost 99%. The difference to our work is the usage of a lightweight neural network for faster training and predictions. Furthermore, we only want to use a small part of the data set to simulate a small data set and evaluate the trained models on the whole. Our small data set will be enhanced by augmentation techniques explained by Tschuchnig in [9]. He describes the process of augmenting images of online accessible celebrity faces data set with DCGAN networks to improve the training results. Tschuchnig DCGAN network generates images of 64x64 pixels, which is not sufficient enough for the data set used in this paper. In this work the size of 128x128 images will be generated.

In [10], Perez and Wang compared traditional augmentation techniques with GAN augmentation on the tiny-imagenet-200 data set [2]. The difference to this work is that Perez and Wang used a GAN to do style transformations instead of generating new images. They came to the conclusion that it is not worth, because traditional augmentations performed better and had three times less computing time than the GAN style transformation.

In [11], Li et al. data set of six different steel surface damages is used. They also used a specialized You Only Look Once (YOLO) network. Their YOLO model can classify and localize the damage in the steel surface images. In this work, we try to classify similar defects with a smaller data set and with a smaller CNN (Convolutional Neural Network) architecture.

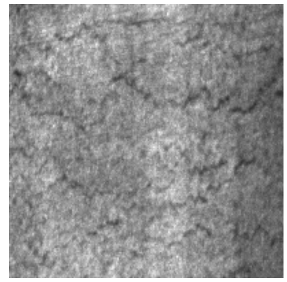
Other works of surface inspection deal not only with steel, but also with textile processing like Stübl et al. in his work [12] or with transparent materials like Satorres et al. in [13]. Zamuner and Jacot did it even with watch parts in [14].

III. Northeastern University Data Set

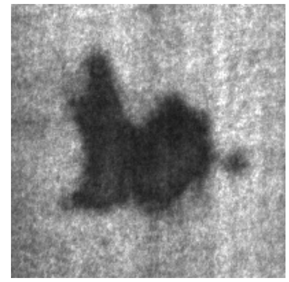
The data set we use for the augmentation experiments and evaluation is provided from the Northeastern University (NEU) and public available at [7]. The data set contains images with six different steel surface damages: Crazing, inclusion, patches, pitted surface, rolled inscale and scratches. Each of the six classes contain 300 samples, 1800 images in total. For doing the augmentation experiments we decreased the data set and removed three classes of the data set. So we only had to deal with the three classes, crazing, patches, inclusion, shown in Figure 1. Further, the data set of 300 images per steel surface image class is reduced to the range of 10 to 50 (3.3 to 15% of the original data set). This allows to generate 250 to 290 images by augmentation and evaluate the achievable classification accuracy, either by using the small data set complemented by augmented images or by using the original data set. Further detailed explanation about the augmentation of steel surface images can be found in Section IV.

IV. Augmentations

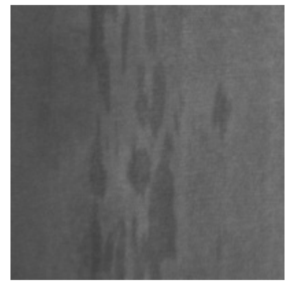
A common problem in machine learning with deep neural networks are data sets containing too few data samples. The success of deep learning models are highly dependent on the



(a) Crazing



(b) Patches



(c) Inclusion

Fig. 1. Steel Surface Image Classes

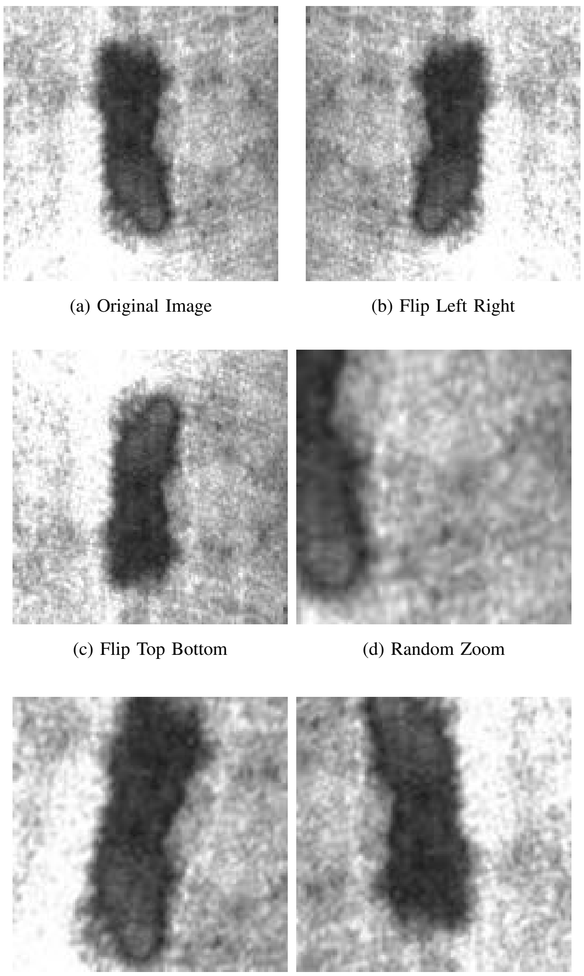
underlying data. Consistency, accuracy, and completeness of data sets are essential for the achievement of good classification results by neural networks. There are a couple of challenges to collect image samples for specific domains. In Roh et al. [15], the challenges are divided into data improvement of existing data, manual or weak labeling of data, and the data acquisition. All these challenges result very often in a weak data set, with too few numbers of samples. One approach to extend the data set is by using augmentation. Image augmentation is the technique to increase the size of the training set without acquiring new images. The basic augmentation technique is duplicating images with some kind of variation (e.g., flipping) so the model can learn from more examples. Ideally, we can augment the image in a way that the features of an original data set are preserved, but the changes within the image are enough to add some variation. Usually, images from the data set are inverted vertically or horizontally, randomly zoomed, stretched, rotated or noise is added to them. A newer method to extend a data set is by using DCGAN Networks to generate new samples. By giving the network many samples of one class, the network learns class-specific patterns in images and generate new images out of a noise vector to expand a given data set. When used correctly in industry, a lot of time can be saved when filling a huge data set, thus preventing financial losses.

A. Common Basic Augmentation

As explained in the introduction, a data set with little data can be expanded by augmentations. The most commonly used augmentations are image transformations. To do so, we used the Augmentor Python package available from [16]. With this technique, we can generate multiple images from every

 ISBN: 978-1-61208-816-7

image in the original data set. For example, Figure 2 shows an original image from the class patches with the different augmentations we used in this work. We use random horizontal and vertical flipping, random zoom 0-20% and random rotation in a range from -180 degree to +180 degree. All of these augmentations are applied with a certain probability to every image, which means that sometimes all augmentations are applied and sometimes only a few. With basic augmentations, such as flipping and rotation, features from the original image still remain. With augmentation like random zoom, features get scaled. This is an obstructing factor when analysing the size of any preexisting damage. But, since it is not relevant in this case, random zoom is used in this work.



(a) Original Image (b) Flip Left Right

(c) Flip Top Bottom (d) Random Zoom

(e) Random Rotate (f) All Together

Fig. 2. Common Image Augmentations

B. Augmentation with DCGANs

This technique of augmentation can be used to generate real-looking samples for the data set preserving the features of the original images. The general architecture of a DCGAN network can be seen in Figure 3. A GAN consists of two concatenated models, the Generator and the Discriminator network. The Generator creates fake images out of an N-dimensional noise vector. The Discriminator gets real and fake images as input and determines whether an image is real or not. The adversarial loss is provided by the Discriminator to the Generator which then creates images, that are as close as possible to real images. [17]. The assumption is, that these new images are expected to be variations of the original image, preserving the features of the original image, but that has not been proven yet.

V. Experimental Setup Using DCGANs for Augmentation

In this section, we explain the neural network architecture, the training method and the evaluation method.

A. Preparation of the Data Sets

We want to show that a small data set enhanced with generated samples from a trained DCGAN model performs better than a model trained without the generated samples. To do so we took subsets of 10, 20, 30, 40, 50 samples per class from the full data set of 300 samples per class and trained DCGAN models for each subset. Each subset requires three models, one for each class. All models were trained for 3000 epochs. Checkpoints of the generator were saved after 600, 1200, 1800, 2400 epochs and the last. That makes a total number of 15 models, each on five different checkpoints. In Figure 4, we can see a generated image for each class from the best performing generator model. The full result of the models can be seen in Section VI.

B. DCGAN Network Architecture

A DCGAN consists of a generator and a discriminator. For both, the architecture of Shrestha was taken from his blog article in [18]. The dimension of the noise vector which is the input for the generator is 100 and it generates an image of 128x128 pixels. The generator has 24 layers. The discriminator has 22 layers and takes 128x128 pixel images as input. The output of the discriminator is a binary decision if the input image is a real image or a generated fake image from the generator.

C. Convolutional Neural Network Description

Since the used data set has been trained successfully on a deep convolutional neural network by He et al. in [8], the approach in this work is to train it on a lightweight convolutional neural network shown in Table I, to measure only the improvements with the different augmentation techniques.

TABLE I. Neural Network Architecture

LAYER	FILTERS	OUTPUT SHAPE	ACTIVATION
Input Layer	-	(128, 128, 3)	-
Conv2D	64	(128, 128, 64)	relu
MaxPooling2D	-	(64, 64, 64)	-
Flatten	-	262144	-
Dense	64	(64, 64, 64)	relu
Output Layer	-	3	softmax

It consists of an input layer that takes images of 128x128 pixels as input, only one convolutional layer with 64 filters

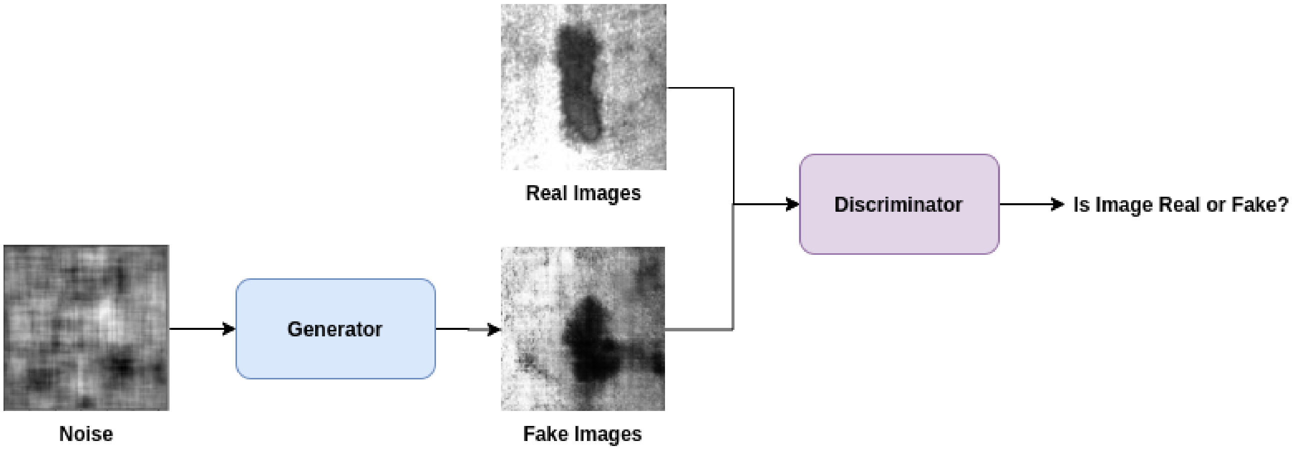
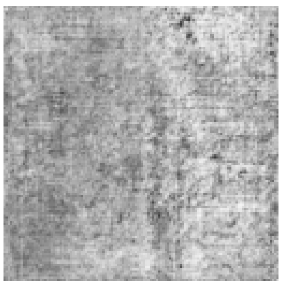
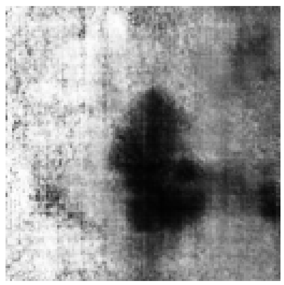


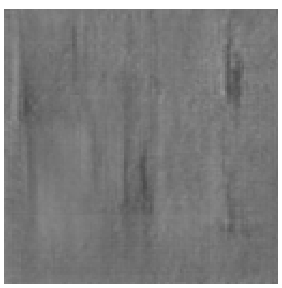
Fig. 3. Generative Adversarial Network



(a) Crazing



(b) Patches



(c) Inclusion

Fig. 4. Augmentation by DCGANs

and Rectified Linear Unit (ReLU) activation function, one maxpooling layer, one flattening layer, one dense layer with 64 units and ReLU activation function and the output layer with 3 neurons for class probabilities provided by the softmax activation function.

D. Training Setup

Every model was trained with the same hyperparameters as described in the Table I above to compare the results. The networks were trained with the architecture from I for 100 epochs on the different data sets. As optimizer we used Adam (short for Adaptive Moment Estimation) which has a learning rate of 0.001 initially. The loss function is categorical cross-entropy because multiclass classification is used. Each epoch took 120 images from the image generator for training and 30 for validation. While training, the models weights got saved every time the validation loss improved. If the model did not improve in at least every 7 epochs, the learning rate got decreased by the factor of 0.1 to enable fine-tuning of the weights.

E. Evaluation Setup

After training, the trained models loaded the weights of the best performing epoch and got evaluated on the whole original data set, consisting of 300 images per class. The results will be given in the next section.

VI. Experimental Results

A. Original Data Set

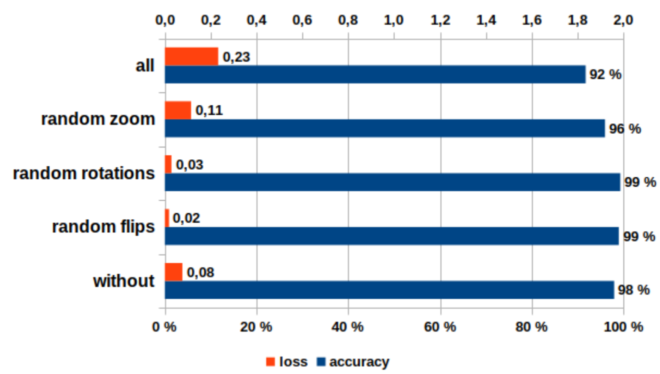


Fig. 5. Original data set

The results from the models trained on the original 300 samples per class data set are given in Figure 5. It can be see seen that the trained model performed quite well with an accuracy of 98.7% even without augmentations . Additional data samples of random flips and rotations improved the accuracy by 1% and the loss from 0.08 to 0.02 with flips and 0.03 with rotations. The model with random zoom augmentation did not improve. Most likely the features of the original image are not preserved by this augmentation method. The model that used all augmentations together wasn't as successful, adding image variations that are too far from the original images.

B. Neural Networks Trained with Reduced Data Set

The results from the neural network models trained on the reduced data sets are illustrated at Figures 6 to 10. As expected, it shows that these models perform much worse than those from the original data set. The models from 10 and 20 samples per class almost never reached a validation loss below 1. The different augmentations showed us that random flips and rotations always improve the validation accuracy but not the validation loss significantly. The best performing model from the reduced data set in terms of accuracy was the one with 40 samples per class, augmented with random rotations. It reached 90% validation accuracy but with a validation loss of 0.72. The best performing model in terms of validation loss was the one trained with 30 samples per class augmented with random rotations. It reached 0.37 validation loss and 85% validation accuracy. Combined it is a drawback to the original data set of 9% validation accuracy and 0.96 validation loss. Surprisingly, it was not the models trained with 50 samples per class. This might be connected to the images being selected randomly out of the original data set for every reduced data set, therefore the quality of the 40 was better than the quantity of the 50 images per class.

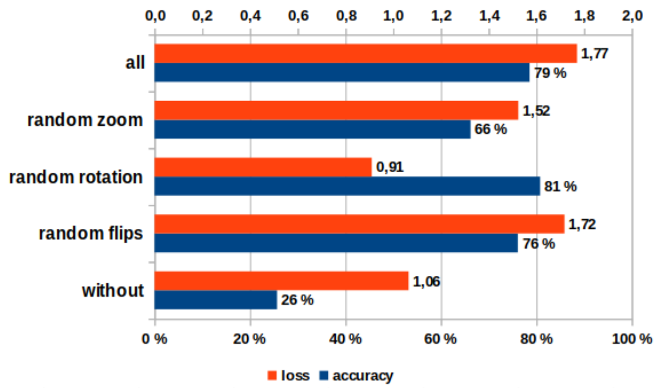


Fig. 6. 10 Samples per Image Class

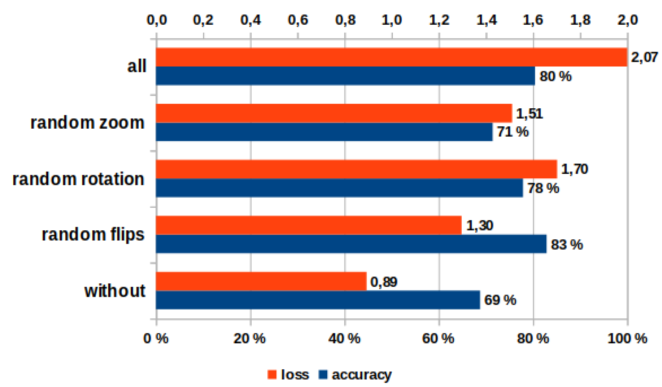


Fig. 7. 20 Samples per Image Class

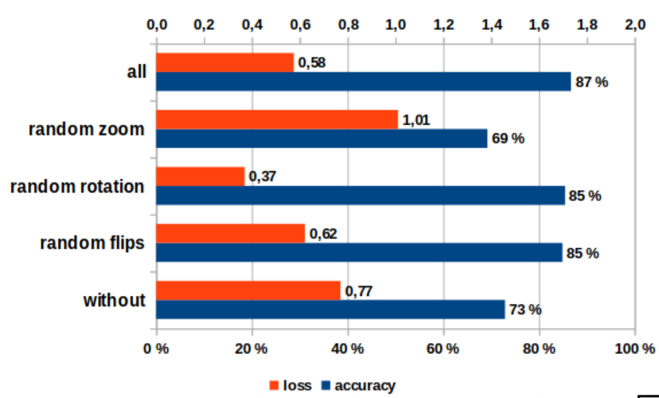


Fig. 8. 30 Samples per Image Class

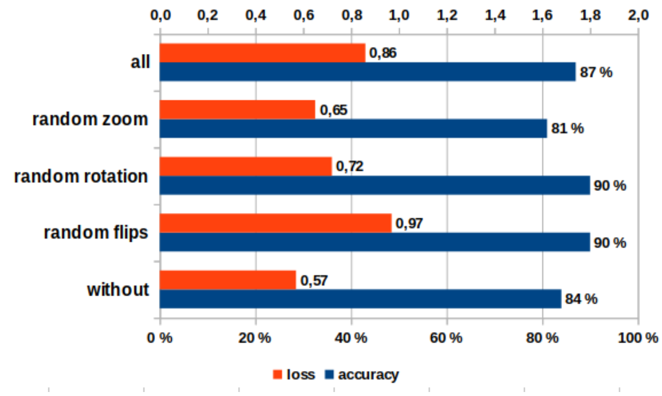


Fig. 9. 40 Samples per Image Class

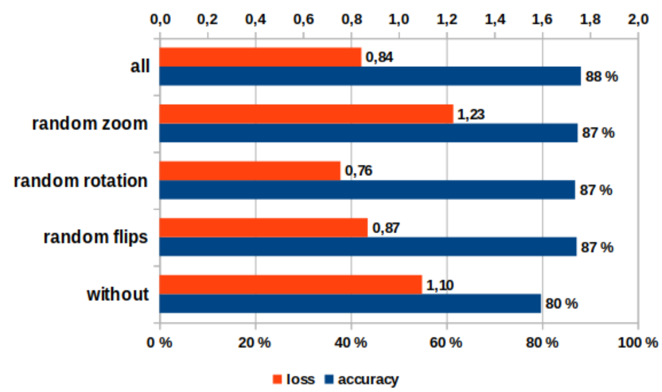


Fig. 10. 50 Samples per Image Class

C. Neural Networks Trained with DCGAN Generated Data Set

As described in Section V, the DCGAN models provided generated data sets for each subset. They are all listed in Table II. On these generated data sets, models were trained the same way as the original and reduced data sets were trained.

The results show that data sets from 10 and 20 samples per class lead to a very significant loss. Using 30 samples per class is more efficient, but the best results were achieved with 40 and 50 samples per class. These DCGAN augmented data sets outperformed other augmentations on the reduced data sets in loss and accuracy. The best model was trained on

 ISBN: 978-1-61208-816-7

TABLE II. RESULTS WITH GAN GENERATED DATA SETS

SAMPLES PER CLASS + 300 AUGMENTED	EPOCH	ACCURACY	LOSS
10	600	80%	0.87
10	1200	73%	1.13
10	1800	67%	2.13
10	2400	75%	1.12
10	3000	71%	2.19
20	600	78%	0.62
20	1200	74%	0.72
20	1800	61%	1.17
20	2400	74%	0.77
20	3000	71%	1.03
30	600	88%	0.49
30	1200	86%	0.38
30	1800	86%	0.47
30	2400	82%	0.61
30	3000	84%	0.56
40	600	88%	0.32
40	**1200**	**92%**	**0.23**
40	1800	88%	0.31
40	2400	89%	0.32
40	3000	88%	0.30
50	600	87%	0.47
50	1200	87%	0.55
50	1800	85%	0.35
50	2400	87%	0.40
50	3000	84%	0.56

a data set generated out of 40 samples per class after 1200 epochs. It reaches a validation accuracy of 92% and a loss of 0,23 which is in terms of accuracy 2% better and in terms of loss 0.14 better than the best models from the reduced data sets together. A full comparison of the best models with different techniques can be seen in Figure 11.

As expected the overall best model was trained on the original data set with a validation accuracy of 99% and a validation loss of 0.02. From the reduced data sets, the DCGAN augmented data set outperformed every other augmentation used in this work with an accuracy of 92% and a loss of 0.23. The best models with common augmentations reached an accuracy of only 90% and a loss of 0.37.

Through DCGAN augmentation we reached improvements of 2% accuracy and 0.14 loss towards common augmentation techniques. The trade-offs to the original data set were 8% accuracy and 0.21 loss.

VII. CONCLUSION AND OUTLOOK

The goal of this work was to enhance a shortened data set with DCGAN generated images and to train a model that performs better on the original data set than models from the shortened data set with common augmentation techniques. In the end, our results show that a well trained DCGAN network can generate images to improve a data set with limited image samples for such a use case in steel surface damages.

One drawback of this method is that all DCGAN models generated images from the same checkpoint. In this work, it is basically the average best models for each class. For further research DCGANs from different checkpoint epochs and classes could be used to improve the quality of the data set quality even more. One observation was that a good variety of generated samples is needed. To do so, the model should not train too few epochs and not too many. If the model trains too little, it likely generates more noise and if it trains too much, the model always generates the same image.

One other observation from the common augmentation techniques was that if you put all augmentations together, the model performs worse than with only one or without any augmentation. We assume that in this case there are too many variations possible within one picture, which weakens the model. This will be part of further investigation.

We can conclude that this method can improve the quality of a small data set, but it cannot replace the quality of a large data set.

ACKNOWLEDGEMENT

This work has received funding from EFRE (European Regional Development Fund) and the Ministries for Research of Baden-Wuerttemberg from the program: Innovation and Energy Transition.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[2] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[3] R. Poppe, "A survey on vision-based human action recognition. image and vision computing 28(6), 976-990," *Image Vision Comput.*, vol. 28, pp. 976–990, 06 2010.

[4] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015.

[5] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[6] C. Shorten and T. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, 12 2019.

[7] "Neu-det data set," available: http://faculty.neu.edu.cn/yunhyan/NEU_surface_defect_database.html.

[8] Y. He, K. Song, Q. Meng, and Y. Yan, "An end-to-end steel surface defect detection approach via fusing multiple hierarchical features," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 4, pp. 1493–1504, 2020.

[9] M. E. Tschuchnig, "Adversarial networks — a technology for image augmentation," in *Data Science – Analytics and Applications*, P. Haber, T. Lampoltshammer, and M. Mayr, Eds. Wiesbaden: Springer Fachmedien Wiesbaden, 2019, pp. 97–98.

[10] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," 2017.

[11] J. Li, Z. Su, J. Geng, and Y. Yin, "Real-time detection of steel strip surface defects based on improved yolo detection network," *IFAC-PapersOnLine*, vol. 51, no. 21, pp. 76 – 81, 2018, 5th IFAC Workshop on Mining, Mineral and Metal Processing MMM 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896318321001

[12] G. Stübl, B. Moser, and J. Scharinger, "On approximate nearest neighbour field algorithms in template matching for surface quality inspection," in *Computer Aided Systems Theory - EUROCAST 2013*, R. Moreno-Díaz, F. Pichler, and A. Quesada-Arencibia, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 79–86.

[13] S. Satorres, J. Ortega, J. Garcia, A. García, and E. Estevez, "An industrial vision system for surface quality inspection of transparent parts," *The International Journal of Advanced Manufacturing Technology*, vol. 68, pp. 1123–1136, 09 2013.

[14] G. Zamuner and J. Jacot, "A system for the quality inspection of surfaces of watch parts," in *Precision Assembly Technologies and Systems*, S. Ratchev, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 134–143.

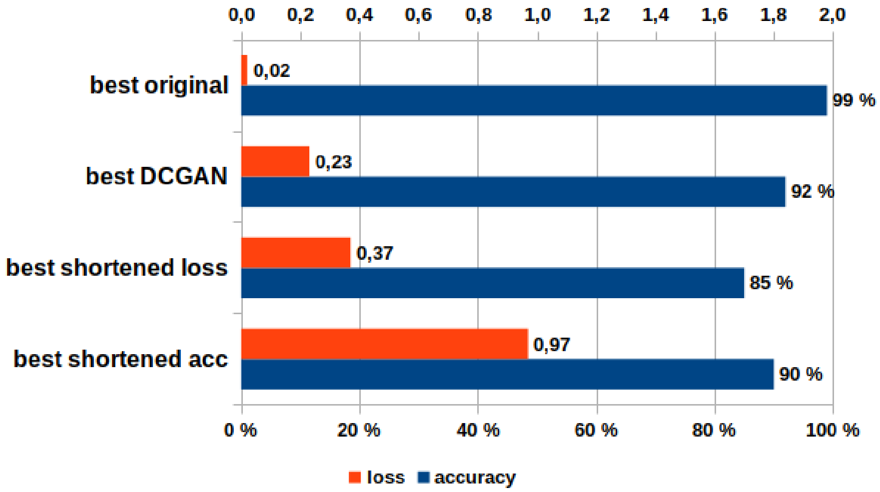


Fig. 11. Overview of the Best Results

[15] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: a big data - AI integration perspective," *CoRR*, vol. abs/1811.03402, 2018. [Online]. Available: http://arxiv.org/abs/1811.03402

[16] Mdbloice, "mdbloice/augmentor," Mar 2020. [Online]. Available: https://github.com/mdbloice/Augmentor

[17] M. Salvaris, D. Dean, and W. H. Tok, *Generative Adversarial Networks*. Berkeley, CA: Apress, 2018, pp. 187–208, available: https://doi.org/10.1007/978-1-4842-3679-6_8.

[18] A. Shrestha, "Generating modern art using generative adversarial network(gan) on spell," Jul 2020. [Online]. Available: https://towardsdatascience.com/generating-modern-arts-using-generative-adversarial-network-gan-on-spell-39f67f83c7b4

 ISBN: 978-1-61208-816-7