Proceedings of the 24th International Conference on Harbor, Maritime and Multimodal Logistic Modeling & Simulation (HMS), 003
19th International Multidisciplinary Modeling & Simulation Multiconference

2724-0339 © 2022 The Authors.
doi: 10.46354/i3m.2022.hms.003

Automated damage detection of trailers at intermodal terminals using deep learning

Pavel Cimili[1,*], Jana Voegl[1], Patrick Hirsch[1] and Manfred Gronalt[1]

[1]University of Natural Resources and Life Sciences, Vienna, Institute of Production and Logistics,
Feistmantelstraße 4, 1180 Vienna, Austria

[*]Corresponding author. Email address: pavel.cimili@boku.ac.at

Abstract

Intermodal transport plays a crucial role for sustainable transport in Europe. Inefficiencies at the interface of road and terminal reduce the acceptance of this transport mode. Increasing the efficiency of the gate-in process is therefore vital. An essential part of this process is the damage detection of trailers entering the terminals due to safety and liability issues. It is usually performed by trained personnel. Deep learning promises to assist human resources reliably in this step. While automated damage detection has been discussed in various fields, including container deliveries, trailers stayed out of the research scope. Thus, this work focuses on automatic damage detection of trailers using deep learning. We observe two approaches: transfer and semi-supervised learning. While the first one is based on the MobileNetV2 network, the latter uses convolutional autoencoders and might be helpful not only for detection but also for damage segmentation and visualization. One of the significant difficulties is the trailer detection on images and its partitioning, which is necessary due to the large recording resolution. That is why we also observe a pre-processing algorithm for the real-world images received from an intermodal terminal in Austria.

Keywords: Intermodal transport; trailer; damage detection; deep learning; transfer learning

1. Introduction

Inland terminals represent the gateway to shifting goods from road to rail or waterways. However, especially at the interface of road and terminal, a lack of interoperability and integrated solutions is evident, both in terms of physical infrastructure and information and communication technology. All of these make the change of transport mode time-consuming and cost-intensive.

Especially the gate-in process involves a great deal of time and effort. Here Intermodal Transport Units (ITUs) such as trailers and containers must be registered, assigned to their waiting positions, and checked for damages before entering the terminal (Posset et al., 2020).

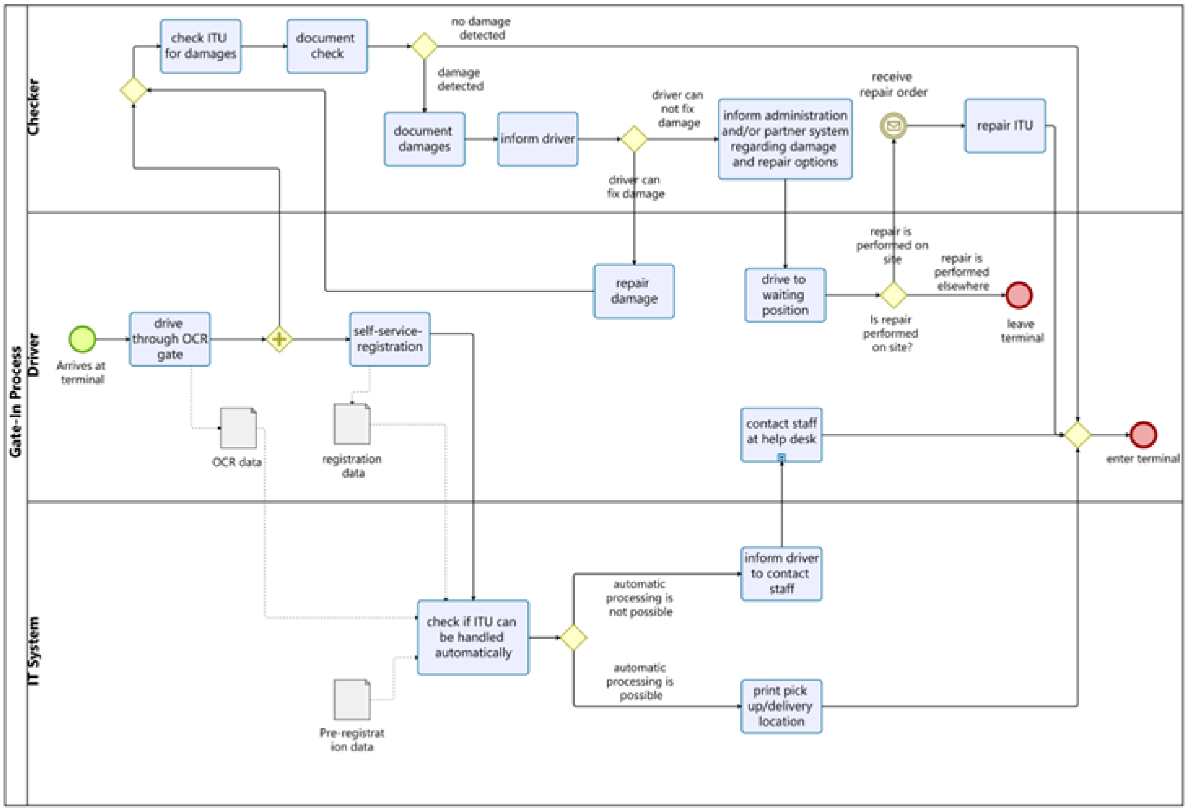
Gate-In Process
Checker
Driver
IT System
check ITU for damages
document check
no damage detected
damage detected
document damages
inform driver
driver can not fix damage
driver can fix damage
inform administration and/or partner system regarding damage and repair options
receive repair order
repair ITU
repair damage
drive to waiting position
repair is performed on site
repair is performed elsewhere
Is repair performed on site?
leave terminal
Arrives at terminal
drive through OCR gate
self-service-registration
OCR data
registration data
contact staff at help desk
enter terminal
inform driver to contact staff
automatic processing is not possible
check if ITU can be handled automatically
automatic processing is possible
print pick up/delivery location
Pre-registrat ion data

Figure 1. Gate-In process at an intermodal terminal

Figure 1 shows the gate-in processes. First, a vehicle arrives and drives through the optical character recognition (OCR) gate. The OCR gate retrieves data, such as the plate number and images of the ITU. The terminal's operation system (TOS) then processes the data. The ITU is checked for damages by qualified staff (checkers) in the following. During the check, the driver performs self-registration. The check is a particularly delicate process for the terminal due to possible safety and liability issues in case of insufficient documentation. If damage is detected, this is documented. The driver gets informed, and repair decisions are made depending on the extent of the damage.

Inefficiencies in the gate-in process can lead to longer waiting times for carriers and higher costs for terminal operators, thus, inhibiting the increase in the share of combined freight transport. Compared with 2009, the number of goods in tons transported with the intermodal rail freight in Europe grew by 49.9%, while the share of trailers among other transportation units in the European Combined Transport was 21% in 2018 (Posset et al., 2020). Inland terminals may differentiate themselves from others by offering value added services (VAS) (Protic et al., 2020). As the implementation of new VAS has risks, possible new VAS such as improved damage detection need to be analyzed thoroughly. Improving the gate-in process might lead to reduced waiting times and an even higher acceptance of intermodal transport.

That is why, we focus on the automated damage detection of trailers in this ongoing research. Automatic recognition aims to accelerate the process and relieve the checkers, who could also be assigned to other tasks. As shown in Section 2, containers' automated damage detection (ADD) was first discussed almost 30 years ago. However, while there has been an increasing number of publications in the last years on this topic, trailers were not part of the research until now.

For this reason, we develop an algorithm for the ADD of trailers using two deep learning approaches: transfer and semi-supervised. The first one allows to perform classification with a high detection rate. The second approach uses convolutional autoencoders and might be useful not only for damage detection but also for damage visualization. Such a method may also be helpful in similar studies if the dataset is unbalanced, with most images representing negative (non-damaged) cases.

After introducing the state of the art in Section 2, the data, processing steps, and structure of the ADD algorithms are presented in Section 3. In Section 4, we provide the results of the ADD algorithms after the training. Then we present conclusions and future work in Section 5.

2. State of the art

Over 25 years ago, Nakazawa et al. (1995) discussed the need for an automatic detection system for container damage. They mainly focused on holes, which may cause damage of container loads due to leaking water. The authors propose a combination of the reflection of light (which does not exist in the case of holes), and a more sophisticated method based on the photometric stereo for reefer containers. Since that time, many new technologies have been invented and improved.

İmamoğlu (2019) shows an example of the successful application of transfer learning with the VGG16 network for damage detection for containers. The author additionally discusses the importance of applying data augmentation, right parameter choice (layers and their numbers, activation functions, or learning rate), and early stopping for automatized container damage detection. Wang et al. (2021) applied a similar transfer learning approach for container damage detection with a MobileNetV2 network developed by Google.

As the last two research items showed good performance in container damage detection, we decided to transfer this idea to our problem. As MobileNetV2 demonstrated better performance than VGG16 or InceptionNetV3, we focused on experiments with this network using a method similar to the one proposed by Wang et al. (2021), considering all trailer construction specialties compared with containers.

Mujeeb et al. (2018) discuss the methods used to identify manufacturing defects when there are few defective images available. They use image partitioning, data augmentation, and convolutional autoencoders. The authors additionally show how to evaluate the difference between reproduced images and reference instances. Mao et al. (2016) and Sarafijanovic-Djukic and Davis (2019) proposed their versions of CAE (RED-Net and inception-like CAE) that were used in our research for semi-supervised learning.

Wang et al. (2019) observe the topic of anomaly detection in wind turbines and focus on analyzing images with large resolutions. The authors suggest partitioning the image into smaller patches. This method appeared helpful for our study, though we chose another partitioning schema. For the CAE we used the Structural Similarity Index (SSIM) for the residual mapping, accuracy metric and loss function. Such an approach was proposed by Bergmann et al. (2019) and showed better performance than the traditional L2-distance metric. We also used an extension to this approach with additional finetuning step after the training of CAE proposed by Boumessouer (2020).

Many authors worked on the detection of anomalies and damages. Some entered the area of multimodal transport by discussing container damages. However, the specific needs of inland terminals with higher volumes of trailers have not yet been considered. While there are obvious similarities to damage detection of containers, trailers have specific features. For instance, these are complex structures at the lower part of the trailer side with multiple straps. That is where the most damages are typically located, requiring special attention with separate data preparation and individually trained neural models. Thus, further research on ADD at inland terminals is needed. In this work, we therefore focus on the ADD for trailers. We develop deep learning algorithms to identify and rate the detected damages. The ADD results will be compared in the future with the manual inspection. If the ADD will be able to reach a certain fitness level, the staff involved in the checking process can be supported and the whole gate-in process is presumably getting much faster.

3. Materials and Methods

Figure 2 demonstrates the workflow of this study, including steps like project understanding and planning, data collection and preparation, model training, evaluation, and approach comparison.

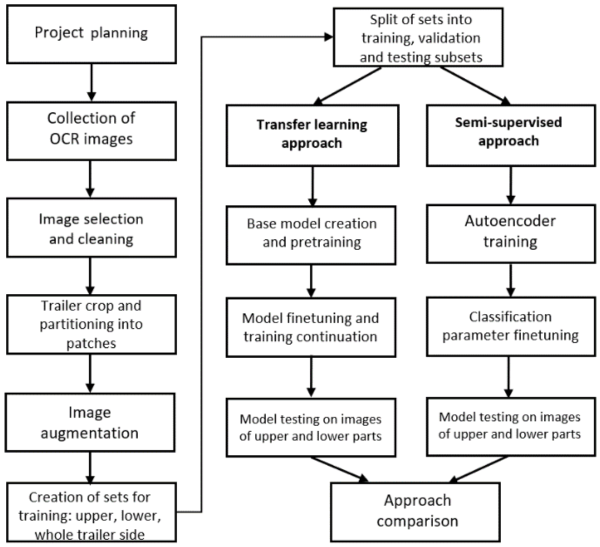
Project planning
Collection of OCR images
Image selection and cleaning
Trailer crop and partitioning into patches
Image augmentation
Creation of sets for training: upper, lower, whole trailer side
Split of sets into training, validation and testing subsets
Transfer learning approach
Semi-supervised approach
Base model creation and pretraining
Autoencoder training
Model finetuning and training continuation
Classification parameter finetuning
Model testing on images of upper and lower parts
Model testing on images of upper and lower parts
Approach comparison

Figure 2. Workflow of the study

3.1. Dataset and training preparation

For both transfer and semi-supervised learning approaches, we used real-world images of trucks. The cameras took them at the OCR gate at the entrance of an intermodal inland terminal in Europe. Each folder with photos for every passing vehicle contains images for different trailer parts: left, right, rear, and upper part. The images show not only the trailer itself but also the driver's cabin, wheels, and surroundings. The trailer's position on the images differs as it might be tilted. In addition, the length of the images varies due to the different speeds of vehicles driving through the OCR gate. All observed damage cases are on the trailer's left or right sides. That is why in this research, we focus only on these sides. The percentage of images of damaged trailers is small compared to the number of non-damaged cases: less than 5%. The two most frequent problems are cracked tarpaulin (34%) and improper tarpaulin patches (38%).

For the ADD, we first processed the images, identified the trailer, and cut it using RetinaNet (Lin et al., 2017) network with ResNet (He et al., 2016) backbone. We first rotate the image using the difference between corner points' positions in degrees or radians to offset possible tilting. This step is essential as training with surfaces that do not belong to the trailer might result in poor performance of the ADD in later steps. Next, we crop the images. The model detects several points in the corners and at the bottom of the trailers which are characteristic for all instances.

The size of the cropped trailer is still relatively large — around 2000×9000 pixels. Hence, it would be inefficient to train the detection model for the whole trailer surface. Additionally, training with images of such size is limited by the available hardware. An example of the partition is shown in Figure 2. It has the following structure: 125 patches of size 400 ×400 pixels, while the first 100 patches (0-99) display the upper part and the last 25 parts (100-124) display the lower part of the trailer side, respectively. In the following, we call these parts created after the partitioning “images” and use them for our models.

Because of the entirely different physical structures of the upper and lower parts of the trailer side, we propose two different approaches for ADD training:

- train single anomaly detection model on all the patches together.
- train two models separately: one for the upper part and another one for the lower part.

In the case of two model training, we guarantee that the network for the lower part does not learn the unnecessary patterns typical for the upper part and vice versa.

3.2. Transfer learning

With transfer learning we can effectively use the knowledge collected while training the existing neural networks (MobileNetV2 in our case) developed for other similar purposes.

We prepared a dataset of 1000 images (created at the partitioning step) of both classes (damaged and non-damaged) for training on lower and upper trailer parts. For better accuracy, the minority class of cases representing trailer defects was oversampled via data augmentation through rotation with a maximal degree of 20 grades, horizontal flip, brightness adjustment, zooming, and shifting along the X and Y axes. For training the network for the whole trailer side, we used 2000 images of both classes. Before each training, 10% of the whole dataset was reserved for testing (around 100 images). At the same time, 10% of the remaining set was used for validation during training.

At the beginning of the training by transfer learning, we excluded the top classification layers of the pre-trained MobileNetV2 network for further feature extraction. Afterward, the whole convolutional base of the network got frozen to avoid weight change. We applied the global average pooling to convert features to a single feature vector and added the dense layer for the binary classification. At this point, we compiled the model for the first training phase using a learning rate = 0.0001 with “Adam” optimizer and binary cross entropy loss function. We made preliminary training of 10 epochs and moved to the finetuning step. The finetuning was made via unfreezing all the layers except those at the bottom starting from the 101st inclusively. All other layers till 101 stayed frozen. To achieve better training performance, we had to decrease the learning rate to 0.00001 to avoid overfitting and switch to “RMSprop” optimizer. Then we proceeded with the training, whereas the maximal number of training epochs was set to 50.

Additionally, we used an early stopping callback with 10 epochs patience to avoid unnecessary training in case of no improvement. The reduction of the learning rate on the plateau was used to improve training performance before early stopping occurred. Finally, we saved the best model version with minimal validation loss. When the training is finished, the trained model is stored in the file with the Hierarchical Data Format version 5 (HDF5), which is suitable for storing complex data.

3.3. Semi-supervised learning

The semi-supervised approach was used at the early stage of the project when we had very few images of the trailer defects and lots of images of non-damaged instances.

The basis for the semi-supervised model is the CAE, trained using only damage-free instances. An autoencoder aims to transform the input image into the output avoiding corruption (Baldi & Lu, 2012). The output of a CAE for damaged cases differs significantly from the initial input image, which allows for classifying it as an anomaly. As we do not use images of the damages for training, the neural network will not be able to represent them correctly.

We used around 70000 images in case of training the models for the upper part, around 17000 for the lower trailer part, and around 85000 images for the model trained for the whole trailer side. All images represent negative cases with no damage. The test sets for the upper and lower parts of the trailer side consisted of around 100 images for each class.

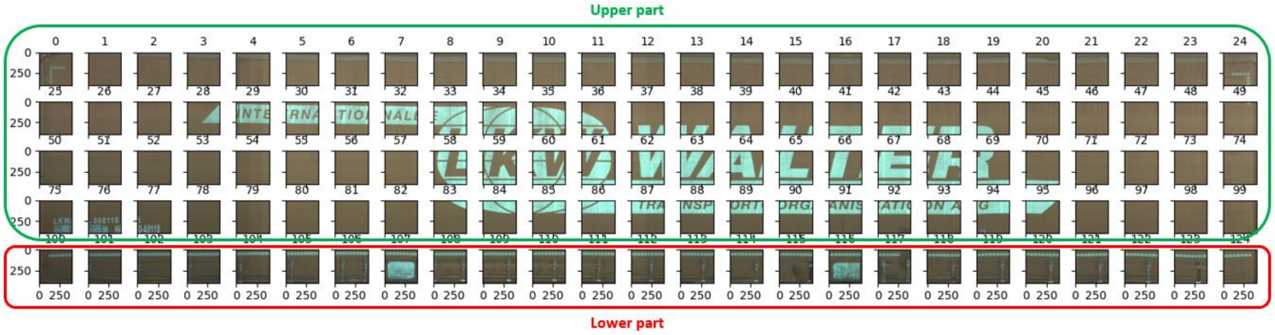

Figure 3. Left side of a trailer — partitioned into 125 parts

During the training, the learning rate is additionally adjusted – if there are no improvements for eight epochs, the learning rate decreases. If this procedure brings no results, early stopping occurs after 11 epochs. Early stopping reduces the probability of overfitting when the neural model tends to recognize patterns only typical for the training dataset and will show weak performance while being used on a different dataset.

At the fine-tuning step, the algorithm tries different thresholds for the size of the "anomaly area" from the observed minimal to maximal residual values with incrementations. It selects one that guarantees the highest rate of true positive and true negative predictions. Images where the size of the area with anomalies is larger than the fine-tuned threshold should be classified as damage cases. For this procedure, we used 20% of images from the test set.

For testing, residual maps with a pixel-wise comparison of the initial and generated images are created. Additionally, it is helpful to perform a binary segmentation to make the difference between damaged and damage-free cases more visible. The generated image's pixels, with a loss (measured in 1-SSIM) higher than the threshold found in the fine-tuning step, are represented in white color, while "non-defect" pixels are in black, as shown in Figure 12.

3.4. Software and Hardware

We used the following software environment to develop and test of the ADD: Python 3.7, Keras 2.8.0, Tensorflow 2.8.0, Scikit-learn 0.24.2., Skimage 0.17.2. All computations were performed on a computer with AMD Ryzen 5 5600G, Nvidia GeForce RTX 3060 12 GB and DDR4 RAM 32 GB. For performance boost Tensorflow ran on the graphics card.

4. Results and Discussion

In the following, we first present the results of the approach based on transfer learning as it seems to be more promising. Then we show the summary of the experiments using the semi-supervised technique.

4.1. Transfer learning approach

The accuracy and loss histories for the training and validation datasets of the model for the upper trailer part are presented in Figure 4 and Figure 5. The complete training ran for 50 epochs while the early stopping did not occur due to the minor but constant improvements in the validation results.

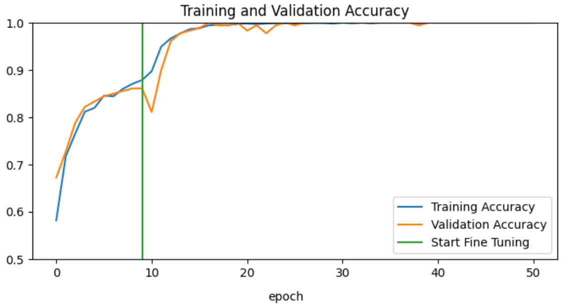

Figure 4. Training and validation accuracy curves of the transfer model for the upper trailer part

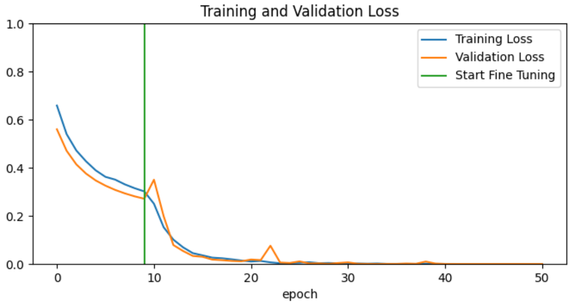

Figure 5. Training and validation loss curves of the transfer model for the upper trailer part

Table 1 and Table 2 demonstrate the performance comparison between the model trained solely for the lower/upper trailer parts and the common model for the whole trailer side.

Table 1. Performance comparison of the model for the lower part and the model for the whole trailer side applied to the testing dataset for the lower part

Metrics	Model for the lower part	Model for the whole side
Precision	0.8859	0.8803
Recall	0.8982	0.8956
Accuracy	0.8896	0.8869
F1 Score	0.8821	0.8879

Table 2. Performance comparison of the model for the upper part and the model for the whole trailer side applied to the testing dataset for the upper part

Metrics	Model for the upper part	Model for the whole side
Precision	0.8913	0.8235
Recall	0.82	0.7
Accuracy	0.86	0.775
F1 Score	0.8542	0.7567

The model trained just for the upper part performs significantly better than the common one achieving an average accuracy of 86 % versus 77.5% on the test set. For the lower part, however, the difference is not so big. Almost all the performance parameters of the model trained exclusively for the lower part of the trailer side slightly outperform the results of the common model, except the F1 Score.

Receiver operating characteristic (ROC) curves in Figure 6 and Figure 7 additionally visualize this performance comparison for different thresholds.

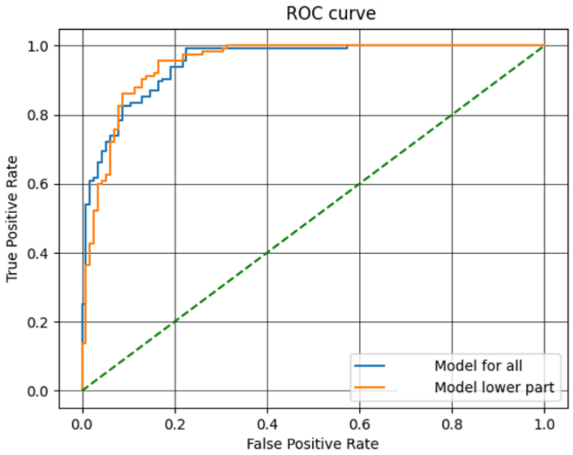

Figure 6. ROC-curves of the model for the lower part and the model for the whole trailer side applied to the testing dataset for the lower part

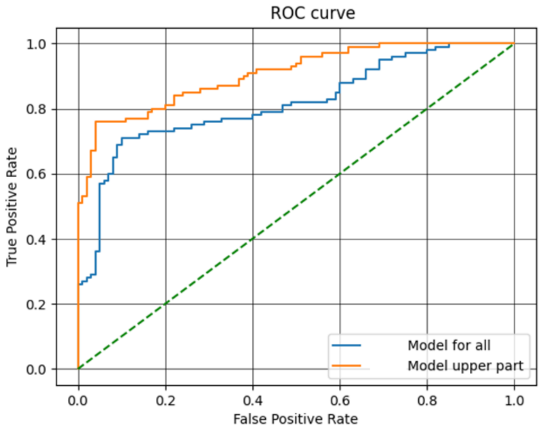

Figure 7. ROC-curves of the model for the upper part and the model for the whole trailer side applied to the testing dataset for the upper part

Extra information on the model's performance trained only for the lower trailer part is presented with the confusion matrices in Figure 8 (percentage values) and Figure 9 (absolute values).

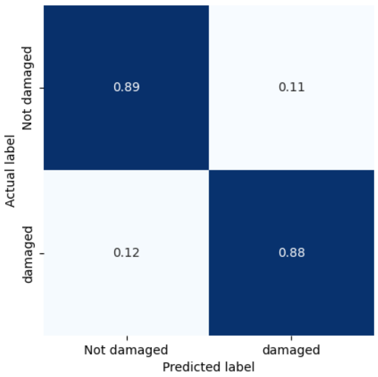

Figure 8. Confusion matrix for the classification result (in %) of the model trained specially for the lower trailer part applied to the testing dataset for the lower part

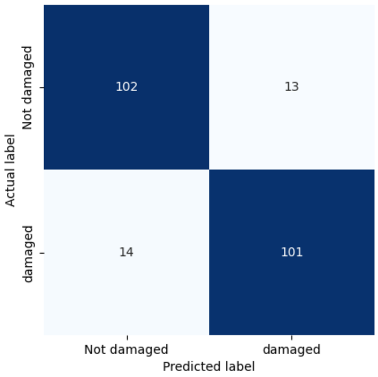

Figure 9. Confusion matrix for the classification result (absolute values) of the model trained specially for the lower trailer part applied to the testing dataset for the lower part

4.2. Semi-supervised approach

We did preliminary testing with the half-reduced image set to select the best CAE for the semi-supervised training. The best performance was achieved with the RED-Net and inception-like CAE variations. We additionally found that RED-Net worked appropriately both with colored and grayscale images, while inception-like CAE performed better in grayscale mode. However, training on the large image set is not doable for inception-CAE due to technical limitations. Thus, in the following, we work with RED-Net.

Figure 10 shows the training history of the RED-Net for the lower trailer part. The green and red lines represent accuracy curves for the training and validation sets of images (accuracy is measured in SSIM). The blue and orange lines show the loss history for both image sets measured as 1-SSIM. Early stopping occurred after around 100 epochs of training, when both training and loss curves converged, flattened out, and no further improvements could be achieved.

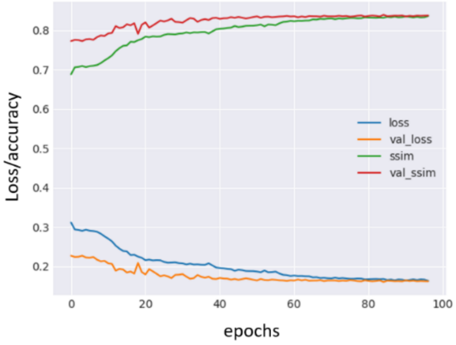

Figure 10. Training history for the upper part

The learning rate scheduler demonstrates changes in the learning rate value during the training, as shown in Figure 11. When the model could achieve considerable improvements quickly, the learning rate had greater values in the beginning. Nevertheless, after around 15000 iterations, when no significant improvements in validation loss were detected, the learning rate had to be decreased from the initial value of 0.00065.

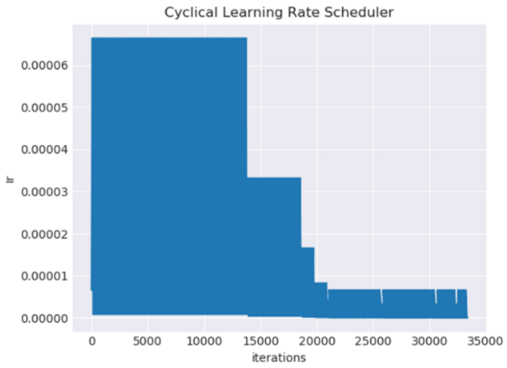

Figure 11. Cyclical Learning Rate Scheduler

The contrast in segmentation for damaged and anomaly-free cases is presented in Figure 12. The difference in white area size between the two cases is visible. Some patterns, e.g., parts of straps, were still recognized as anomalies.

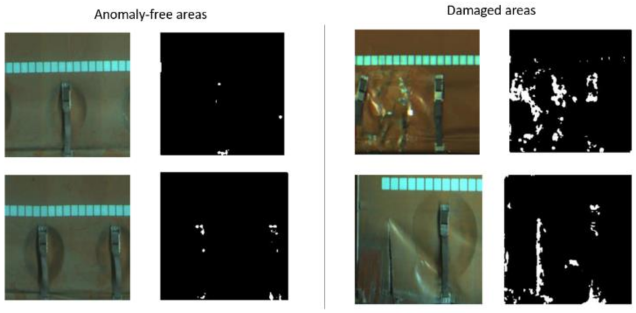

Figure 12. Contrast for anomaly-free areas

The testing results (Table 3 and Table 4) of the semi-supervised model demonstrate worse results than the results of the transfer learning model in all the cases.

Table 3. Performance comparison of the CAE model for the lower part and the CAE model for the whole trailer side applied to the testing dataset for the lower part

Metrics	Model for the upper part	Model for the whole side
Precision	0.6	0.8
Recall	0.685	0.5882
Accuracy	0.6621	0.6200
F1 Score	0.6397	0.6780

Table 4. Performance comparison of the CAE model for the upper part and the CAE model for the whole trailer side applied to the testing dataset for the upper part

Metrics	Model for the upper part	Model for the whole side
Precision	0.77	0.6551
Recall	0.6063	0.6172
Accuracy	0.6350	0.6244
F1 Score	0.6784	0.6356

Howbeit, we still see the tendency earlier observed during the transfer learning — models trained exclusively for the upper or lower trailer provide better accuracy than the model for the whole trailer side.

5. Conclusions

The aim of the methods discussed in this paper is the automatic damage detection on images of trailers at intermodal inland terminals and thus, increase the efficiency of the gate-in process. This process is important as long dwell times of vehicles may result in lower attractiveness of intermodal terminals preventing the use of more sustainable modes of transport.

Due to the different structures and surfaces of the upper and the lower part of trailers, for both transfer and semi-supervised approaches, we developed and tested two separate models (for the lower and the upper part).

Transfer learning with the MobileNetV2 neural network showed good performance for the problem of damage detection on the trailer surface. Due to the structure of the trailers compared to containers, it is highly recommended to train two separated models for the upper and lower trailer parts to achieve better performance in damage detection.

Semi-supervised training could not outperform transfer learning model due to the complex structure of the trailer surface and its defects. One of its main weaknesses is the impossibility of differentiating between cases when damage and non-damage examples look similar. This is the case for improper and proper patches, for instance. However, the model based on transfer learning is capable of such identification if there is enough training data.

In the future, we plan to improve accuracy of the transfer learning model after receiving more data for damaged cases. This model still has the potential for increasing complexity and can be applied not just for binary classification but also for the classification of multiple damage classes.

Funding

Financial support by Grant No 877682 of the program Mobility of the Future provided by the Austrian Research Promotion Agency and the Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology, is gratefully acknowledged.

The funding body did not have a part in the design of the study, the collection, analysis, or interpretation of data or in writing the manuscript.

The Principal Investigator of the grant is Manfred Gronalt.

References

Baldi, P., & Lu, Z. (2012). Complex-valued autoencoders. *Neural Networks, 33*, 136-147. doi:10.1016/j.neunet.2012.04.011.

Bergmann, P., Löwe, S., Fauser, M., Sattlegger, D., & Steger, C. (2019). Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications 5,* 372-380. doi:10.5220/0007364503720380.

Boumessouer, A. (2020, September 20). *MVTec-Anomaly-Detection.* Retrieved from https://github.com/AdneneBoumessouer/MVTec-Anomaly-Detection.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 770-778). doi:10.1109/CVPR.2016.90.

İmamoğlu, Z. (2019). *Container damage detection and classification using container images.* Unpublished master's thesis, İzmir Institute of Technology, İzmir, Turkey.

Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2017). Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision* (pp. 2980-2988). doi:10.1109/ICCV.2017.324.

Mao, X.-J., Shen, C., & Yang, Y.-B. (2016). *Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections*. arXiv preprint arXiv:1606.08921.

Mujeeb, A., Dai, W., Erdt, M., & Sourin, A. (2018). Unsupervised surface defect detection using deep autoencoders and data augmentation. In *2018 International Conference on Cyberworlds (CW)* (pp. 391-398). IEEE. doi:10.1109/CW.2018.00076.

Nakazawa, K., Iwasaki, I., & Yamashita, I. (1995). Development of damage detection system for container. *Proceedings of IECON'95-21st Annual Conference on IEEE Industrial Electronics* (Vol. 2, pp. 1160-1163). IEEE.

Posset, M., Gronalt, M., Peherstorfer, H., Schultze, R.-C., & Starkl, F. (2020). *Intermodal Transport Europe*. Universität f. Bodenkultur Wien.

Protic, S. M., Fikar, C., Voegl, J., & Gronalt, M. (2020). Analysing the impact of value added services at intermodal inland terminals. *International Journal of Logistics Research and Applications, 23*(2), 159-177. doi:10.1080/13675567.2019.1657386.

Sarafijanovic-Djukic, N., & Davis, J. (2019). Fast distance-based anomaly detection in images using an inception-like autoencoder. *International Conference on Discovery Science* (pp. 493-508). Springer, Cham. doi:10.1007/978-3-030-33778-0_37.

Wang, Y., Yoshihashi, R., Kawakami, R., You, S., Harano, T., Ito, M., . . . Naemura, T. (2019). Unsupervised anomaly detection with compact deep features for wind turbine blade images taken by a drone. *IPSJ Transactions on Computer Vision and Applications, 11*(1). doi:10.1186/s41074-019-0056-0.

Wang, Z., Gao, J., Zeng, Q., & Sun, Y. (2021). Multitype Damage Detection of Container Using CNN Based on Transfer Learning. *Mathematical Problems in Engineering, 2021.* doi:10.1155/2021/5395494.