Quality control for Vacuum Insulating Glass using Explainable Artificial Intelligence

Authors

Henrik Riedel, Sleheddine Mokdad, Isabell Schulz, Cenk Kocer, Philipp Rosendahl, Jens Schneider, Michael A. Kraus, Michael Drass

Abstract

Vacuum insulated glazing (VIG) defines an energy efficient glazing unit. Their thermal performance units can reach values as good as quadruple insulating glass, with thinner dimensions and less material. Owing to their internal vacuum, VIGs are under the permanent influence of the atmospheric pressure acting on them throughout their entire service life. To withstand the pressure and to ensure sufficient distance between the glass panes, small pillars are positioned in-between. Especially the area around the pillars is prone to damage during the manufacturing of a VIG and during its lifetime. In order to assess this damage efficiently, automatic damage detection is necessary. For this purpose, we use a convolutional neural network. The binary classification model achieves an accuracy of 100 % for clearly recognisable damage and is also able to visualize the detected damage. Through our object recognition model, the input resolution can effectively be increased by cropping the image before the classification. The proposed methods can therefore be used to detect systematic defects even without large amounts of training data. Damage detection and classification can be used for quality control and enables the application of fracture mechanical models for assessing the stability of initial cracks during lifetime.

Introduction

Connecting the domains of deep learning and glass structures is crucial to develop an automated damage evaluation method. Therefore, both areas will be briefly introduced.

Vacuum Insulated Glazing

Vacuum Insulated Glazing (VIG) units, in contrast to traditional insulated glazing units (IGU), are characterised by the fact that there is a vacuum between the glass panes instead of a gas [2]. This allows for thinner dimensions and lower weight compared to IGUs. Due to the internal vacuum, the panes are under constant atmospheric pressure over their service life. To maintain the gap between the glass panes, small pillars are positioned in a regular array between the panes (see figure 1). As a result of the indentation of the pillars on the glass surface, the contact stress near the pillar may damage the glass during manufacturing, transportation, and/or installation, which in turn reduces the lifetime of the VIG unit. Therefore, it is highly desirable to develop an automate damage detection and evaluation method in order to detect and possibly even prevent fractures.

Image Classification

Images can be very complex and difficult to analyse [10]. The most common backbone for detecting objects or classifying images are convolutional neural networks (CNN). CNNs filter the image for many different features [5]. In the first layer these features are very simple, like straight lines or specific colour combinations, but complexity increases with every additional layer. With this method, different large objects can be detected with mostly the same small filters, depending on their combination [4].

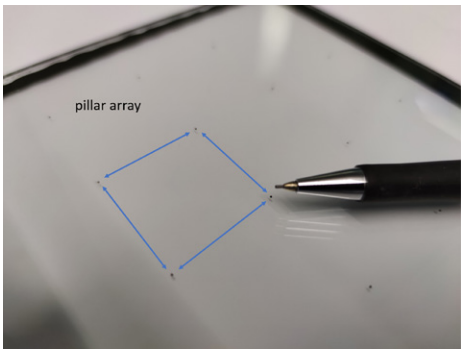

Figure 1: pillar array of a real vig.

The different methods for image analysis can roughly be categorised in classification, localisation and segmentation. Classification has the goal to classify an image into different categories (e.g. residual neural network). Localisation aims to find for example the most accurate box that fits around a detected object and also classifies it (e.g. Region Based Convolutional Neural Networks). Image segmentation is about classifying the image pixel by pixel and is for example used in medical imaging (e.g. Feature Pyramid Network). There are also mixed models like Mask-RCNN solving all of the problems above at once [11].

To understand the advantages and disadvantages of the methods, we must first understand how training works. For training, the model receives our solutions (y) for the respective input values (X) of the training data. This means that the y for all data must be created manually for the training data. The more complex the desired result, the greater the effort required to create the labels for the training data. In this work we will focus on how to maximise the results with a few tricks without the need for a very complex model or for complex labels. This makes it possible to quickly explore the potential of the data and avoid time-consuming mistakes.

Methods

Data Acquisition

To develop the method for quality control of VIGs, pictures of pillars were taken with a microscope. The microscope has a lens with a magnification of 1000 times, a resolution of 1920x1080 pixel and automatic exposure control. The diameter of the examined pillars is about 0.5 mm. For a better and more detailed result, different backgrounds were compared. A black background was selected for this work, because of the enhanced visibility of cracks and compared to a white background. The microscope was manually positioned and triggered, the image quality was manually controlled and finally the images were manually sorted into two categories: damaged and undamaged. As an example, two pillars from each category are shown in figure 2. On average, the processing steps for a single pillar take up to 2 minutes.

Image Preprocessing

In order to maximize the effective resolution for our classifier, we will use a localisation model to optimally crop the image. The second step is augmenting the images to improve our model's performance and make it more robust to new data.

Resolution optimization

Theoretically CNNs can handle very large images as long as there is enough memory and

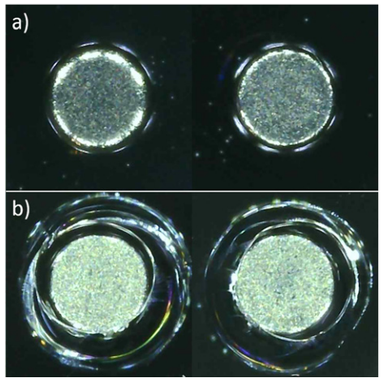

Figure 2: damage classes a) undamaged and b) damaged.

computing power. With the size of the images, not only the size of the input increases, but necessarily also the number of layers. With increasing complexity, a model requires more computing power and training data and often tends to be less stable. Reducing the training data to relevant information only could prevent such problems from the start. Therefore, the images should be cropped as small as possible so that only the pillar and the damage are in the image. Since the microscope was not perfectly centred over the pillar before the picture was taken, we had to locate the pillar first. For this we used the PyTorch implementation of Faster-RCNN with a ResNet-50-FPN backbone [7].

The pillars location is returned as a bounding box by the trained object detector. This information can be used to zoom into images without losing relevant information. In our case, we were capable of reducing the resolution from 1000 • 1000 to 700 • 700, which already roughly halves the number of pixels. After zooming into the image, we can make use of the symmetry of the images. Even though our images contain asymmetric cracks, those images contain either cracks in all of its quadrants or in none of its quadrants and are therefore symmetrical for their label. This allows us to split the image into the four quadrants, without the need for creating new labels. Classification models like the ResNet halve the size of its layers several times to efficiently detect different sized details. We chose 352 • 352 as resolution for the quadrants instead of 350 • 350, so that the resolution can be divided often enough without a remainder. As a result, the images overlap slightly and the size of the model input is further reduced to one quarter. The 3 steps are displayed in figure 3 from left to right: captured image, cropped image and quadrant. The figure shows an example with the pillar very near the image border. Since the resolution of all images needs to be the same and the pillars have to be centred before splitting it into quadrants, the missing parts of the images must be filled with black pixel. In total, the number of input pixels has been reduced to an eighth.

Augmentation

In order to improve the performance of our model and to make it more robust for future applications (e.g. pictures are taken with different microscopes or a VIG from a different manufacturer) we used multiple data augmentation methods [9]. We used a custom script for randomly erasing parts of the image. Our script overwrites 25-40 % of the image with random values by a chance of 50 %. In addition, we have used random rotation, random channel shift, horizontal flip, vertical flip and random brightness.

Classification Model

Assigning an input to a specific class is called classification. In a neural network, there is usually one output neuron per class. The value of this neuron can be regarded as the predicted probability that the input belongs to a certain class. The classification problem in this work consists of two exclusive classes and thus a single value is sufficient to represent both classes. In order to achieve the best possible results, we have developed our own model for this project based on residual neural network (ResNet) [3]. The ResNet is characterised by the so-called skip connections. The input of a convolution block is passed by its skip connection to two different layers: the filtering layer at the start and the output in the end of the block. In a ResNet the next layer then receives the sum of unfiltered and filtered data, while in our model the filtered data gets concatenated to the unfiltered data. Thus, the respective results of each convolution block remain unchanged and can be used together by the last filtering layer. Without skip connections, errors from previous layers can no longer be compensated and sometimes can even be amplified. The final architecture of our model is shown in figure 4.

Training Parameter

161 images each of Pillars with damage and without damage were selected (see figure 5) and 20 % of them were separated before augmentation in the form of a validation set. The maximum number of epochs was capped at 100 epochs. The batch size was set to 6 and the following callbacks were used:

1. Early stopping (patience of 30 epochs)
2. Reduce learning rate on plateau (patience of 8 epochs and reduction factor of 0.1)

Early stopping is used to stop the training, if the model's improvement stales, to save computing time. It regulates how many epochs the model is trained, although it no longer improves. Generally, the performance of a model does not improve in a strictly monotonic manner, but rather fluctuates with an upward trend. This means that the model should continue to train even with temporary performance degradation as long as there is an upward trend [6]. To avoid a problem caused by too high a learning rate, the learning rate is reduced when a plateau in improvement occurs. As a result, the learning rate will be decreased 3 times, before training is stopped, due to lack of performance improvement.

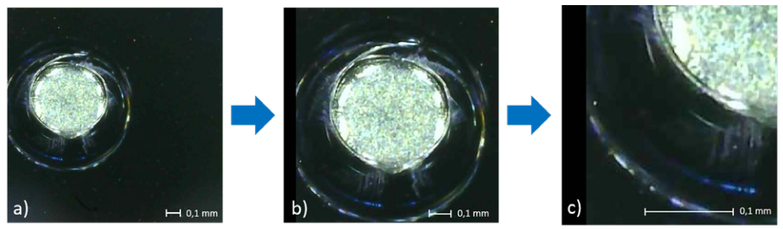

Figure 3: image preprocessing from a) the raw image to b) the cropped image to c) the quadrant.

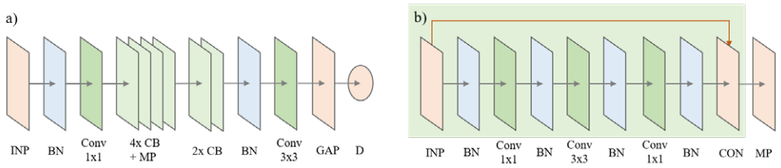

Figure 4: The figure shows a) the general architecture of the custom model and b) the convolution block. The layers of the custom model are as follows: input (INP), batch norm (BN), convolution 2D (Conv), convolution block (CB), max pooling 2D (MP), global average pooling (GAP) and dense (D).

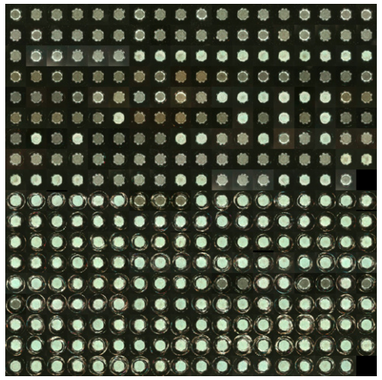

Figure 5: dataset with undamaged pillars on the top and damaged pillars on the bottom.

Gradient-weighted Class Activation Mapping

Gradient-weighted Class Activation Mapping (Grad-CAM) allows to visualize where a model locates an object of a specific class [8]. Thus, more knowledge can be gained about a model and its intermediate results can be used for localisation, even though the model has only been trained for classification. The Grad-CAM returns a value per pixel of the last filtering layer, which can be interpreted as the probability of detection. We visualised these values with a colormap from the OpenCV library [1]. For this colormap, purple means that nothing was detected and turquoise means that some kind of damage was detected.

Results and Discussion

For the evaluation of the models, we have summarised the results in table 1 from the 5 runs. For each run, we only used the weights of the epoch with the lowest loss. We used the following metrics to evaluate the performance of the models: minimum area under the curve (AUC) for the receiver operating characteristic (ROC), maximum number of training epochs, false negatives (FN) with a classification threshold of 95 %, false positives (FP) with a classification threshold of 10 %, minimum accuracy, minimum precision with a threshold selected for 100 % recall and mean loss (for binary crossentropy). The custom model achieves better results than the other models in all metrics. Only in one run did the custom model achieve a non-perfect result in a single metric due to a single FP classification. In most runs, the ResNets also had good results with 100 % in accuracy, AUC for ROC and precision at 100 % recall. In these metrics, ResNet152V2 was imperfect 3 times, ResNet101V2 2 times and ResNet50V2 1 time. In the FP and FN metrics, the ResNets were only error-free once or twice each.
Using the same images as in figure 5, we examined the performance of the custom model and the ResNet50V2 using Grad-CAM. While the custom model mainly detects the cracks themselves and not the pillar (see figure 6), the ResNet50V2 focuses on the pillars and rarely detects the cracks (see figure 7). This supports the result from our validation set and suggests that the custom model would probably also respond very well to new data.

In order for the pillars to be examined efficiently, the recording of the pillars should also be automated. At this stage, microscope positioning, image quality checking and documentation are done manually and can take up to 2 minutes per pillar per side. Since a VIG has hundreds of pillars, it takes tens of hours to examine a single panel. Automating the pillar recordings would not only reduce costs, but could also speed up the process and gather additional information in the process. The orientation and position of the camera would be particularly valuable for investigating systematic errors and would be much more reliable if this was not documented manually. Based on this, a risk analysis could make it possible that not every pillar or even not every VIG needs to be examined to reach the quality goals.

Conclusion

We used a microscope to investigate indentation damage caused by pillars in VIGs and developed a method for automatic damage detection. To do this, we took images of pillars with and without visible fractures. These images were cropped to their relevant information by locating the pillar with an object detection model. By splitting the images into quadrants, we were able to reduce the overall input size of the classifier to one-eighth. For the classification of the damage, we developed our own model and compared it with state-of-the-art ResNet models. The custom model achieved better results on the validation set in every metric compared to the ResNet models. In addition, we tested the generalisation capability of the custom model and a ResNet model using the Grad-CAM method. Here, too, the custom model performed much better.
We were also able to show that the Grad-CAM method can be used to localise damage by using a classification model.

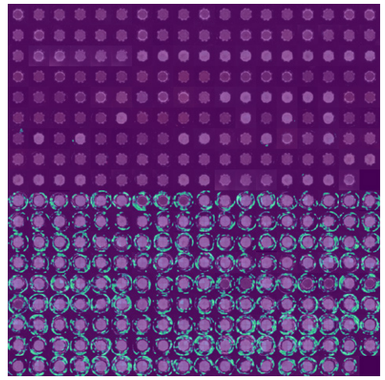

Figure 6: Grad-CAM results of the custom model.

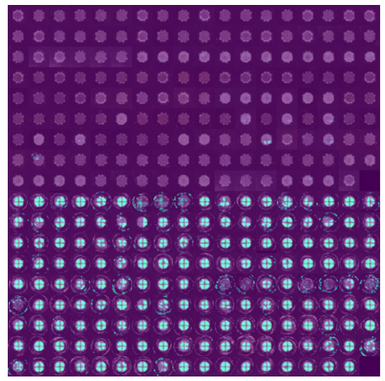

Figure 7: Grad-CAM results of the ResNet50V2 model.

References

1. Bradski, G., 2000. The OpenCV Library. Dr. Dobb's Journal of Software Tools.
2. Collins, R., Asano, O., Misonou, M., Katoh, H., Nagasaka, S., 1999. Vacuum glazing: Design options and performance capability, in: Glass in Buildings Conference, Bath UK.
3. He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep residual learning for image recognition. CoRR abs/1512.03385. URL: http://arxiv.org/

	Max epochs	Min AUC for ROC	FN 95 % mean	max	FP 10 % mean	max	Min precision at 100 % recall	Min accuracy	Mean loss
ResNet152V2	63	0.9995	3.8	6	5.4	15	0.9682	0.9841	0.0286
ResNet101V2	48	0.9997	3.6	7	2.2	4	0.9818	0.9762	0.0232
ResNet50V2	41	0.9960	6.6	15	2.2	5	0.9143	0.9683	0.0313
Custom Model	35	1.0000	0.0	0	0.2	1	1.0000	1.0000	0.0003

Table 1: summary of the training results

Quality Management

abs/1512.03385, arXiv:1512.03385.
4. LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. URL: https://doi.org/10.1038/nature14539, doi:10.1038/ nature14539.
5. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324. doi:10.1109/5.726791.
6. Prechelt, L., 1998. Early Stopping - But When?. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 55–69. URL: https://doi.org/10.1007/ 3-540-49430-8_3, doi:10.1007/3-540-49430-8_3.
7. Ren, S., He, K., Girshick, R.B., Sun, J., 2015. Faster R-CNN: towards real-time object detection with region proposal networks. CoRR abs/1506.01497. URL: http://arxiv.org/abs/1506.01497, arXiv:1506.01497.
8. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2017. Grad-cam: Visual explanations from deep networks via gradient based localization, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV).
9. Shorten, C., Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. Journal of Big Data 6, 60.
10. Szeliski, R., 2010. Computer Vision: Algorithms and Applications. 1st ed., Springer-Verlag, Berlin, Heidelberg.
11. Voulodimos, A., Doulamis, N., Doulamis, A., Protopapadakis, E., 2018. Deep learning for computer vision: A brief review. Computational Intelligence and Neuroscience 2018, 7068349. URL: https://doi.org/ 10.1155/2018/7068349, doi:10.1155/2018/7068349.