Deep learning-based automated characterization of crosscut tests for coatings via image segmentation

Gaoyuan Zhang, Christian Schmitz, Matthias Fimmers, Christoph Quix, Sayed Hoseini

Abstract A manual scratch test to measure the scratch resistance of coatings applied to a certain substrate is usually used to test the adhesion of a coating. Despite its significant amount of subjectivity, the crosscut test is widely considered to be the most practical measuring method for adhesion strength with a good reliability. Intelligent software tools help to improve and optimize systems combining chemistry, engineering based on high-throughput formulation screening (HTFS) technologies and machine learning algorithms to open up novel solutions in material sciences. Nevertheless, automated testing often misses the link to quality control by the human eye that is sensitive in spotting and evaluating defects as it is the case in the crosscut test. In this paper, we present a method for the automated and objective characterization of coatings to drive and support Chemistry 4.0 solutions via semantic image segmentation using deep convolutional networks. The algorithm evaluated the adhesion strength based on the images of the crosscuts recognizing the delaminated area and the results were compared with the traditional classification rated by the human expert.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11998-021-00557-y.

G. Zhang, C. Quix, S. Hoseini (✉)
HS Niederrhein University of Applied Sciences, Krefeld, Germany
e-mail: sayed.hoseini@hs-niederrhein.de

C. Schmitz
Institute for Coatings and Surface Chemistry, HS Niederrhein University of Applied Sciences, Krefeld, Germany

M. Fimmers
HIT-Institute of Surface Technology, HS Niederrhein University of Applied Sciences, Krefeld, Germany

C. Quix
Fraunhofer Institute for Applied Information Technology FIT, St. Augustin, Germany

Introduction

Organic coatings are commonly applied to protect working surfaces from corrosion, abrasion, and erosion[1] or impart an aesthetic appearance.[2] These coatings act as barriers covering the underlying substrate from the environmental impact and promise long-life durability. The most critical failures are caused by delamination or other surface defects and result in malfunction of the workpiece or disfigurement. Thus, adhesion is the minimum requirement for the development of new lacquers or paints. Moreover, durability is expected through weathering, hot–cold temperature cycles, mechanical strain and chemical exposure over a long period.[3,4]

The simplest manner for rating the destruction of the coating is the visual control by humans during lab tests under conditions being required for the application. Many characterization methods have moved to more empirical values, e.g., indentation testing[5] or gloss reduction after abrasion,[6] in an attempt to overcome the subjective estimation. However, other procedures described by DIN EN ISO or ASTM standards still rely on evaluation by the operator.[7–10] Even though the significance of the rating might differ between experts, fine distinctions can rarely be recognized excluding the human control. For some techniques, the implementation of automated processes requires a quicker response without user interference.

This automation is applied in the quality control as in-line methods[11] of the manufacturing and also more and more in the lab development of new coating materials.[12]

With high-throughput equipment (HTE) the coating preparation and testing is fully automated for faster screening of formulations.[13] Robots are introduced to execute repetitive tasks like dosing, mixing, application on the substrate, and characterization of the hardened surface with a high accuracy for more than hundred samples a day. Because surface control by imaging is an established part on HTE, a reliable recognition of any kind of surface defect and delamination would be beneficial for such innovative trends in material development.

One important procedure is the crosscut test[9] evaluating the delamination of coatings by comparing the pattern of the damage being closely related to the delaminated area. Another method is the the pulloff test,[13] but since it requires gluing a dolly onto the surface which highly increases the difficulty of the test with automated sampling, the crosscut test is more common. Previous work has targeted a quantitative evaluation of the test,[14] however automating the suggested method is not feasible and computational approaches for image segmentation are potentially more efficient. Online imaging is a fast quality control that is already used in post-processing of industrial manufacturing.[15,16] An automated image recognition valuing the defect of the crosscutting would be desirable for a faster throughput with no interference of the human operator. The differences between intact surface and the damage need to be recognized even in case of small optical deviations between some combinations of substrates and coatings (e.g., clear coating on glass substrate).

Computer algorithms operating state-of-the-art machine learning are established for several applications in image detection and classification.[17] Machine learning enables computers to address problems by learning from data. Deep learning (DL) is a type of machine learning that uses a hierarchical recombination of features to extract relevant information and then learn the patterns represented in the data. For over a decade, DL has been increasingly applied to a wide variety of chemical challenges, like improving computational chemistry to drug and materials design and synthesis planning.

High-throughput formulation screening (HTFS) represents one feasible opportunity for systematic study of chemical systems using advanced robotic systems.[18] Here, many experiments can be carried out under controlled and repeatable conditions providing valuable, good data. A more sophisticated approach is the use of machine learning algorithms running the analysis right on time, directly linked to the responses (measured output data) and suggesting new experiments based on the digital model uncertainty created from the actual data.[19] In the future, machine learning algorithms will automatically control the HTFS equipment and conduct experimental processes independently and automatically and propose new experiments on the basis of the results of previous runs. An important step toward the goal to fully automate complete chemical processes is the automated characterization of samples as part of the comprehensive machine learning process. The combination of data collection from HTFS and analysis with machine learning fulfills requirements of Chemistry 4.0. The term Chemistry 4.0 describes the digital transformation of the chemical industry[20] and separates its historical development into four periods with the perspective of a higher degree of automation in the future by applying methods of big data and self-monitoring systems.

In this work, we utilize convolutional neural networks (CNNs) to evaluate samples based on visual data. The development of CNNs has led to a series of scientific breakthroughs to promote artificial intelligence; they were successfully applied to image recognition tasks for example on the ImageNet dataset[21–23] and for face recognition,[24] image segmentation tasks that are very important in autonomous driving[25] and biological applications[26] and generative adversarial networks[27] that can generate new random samples with the same statistics as the training data. We focused on the automated evaluation of delamination by crosscutting. The cutting itself was performed by an automated system handling incoming test panels with several kinds of soft and hard material as the substrate. The pattern of damage was imaged and evaluated by a deep learning algorithm segmenting delaminated and intact area. As a result, the delamination could be traditionally rated as described by standard procedures or be determined based on the exact ratio between delaminated and intact area, which is a more rigorous and objective approach. Since other kinds of defects like scratch or abrasion tests are usually optically rated by the area of the damage, this technique can be easily introduced into these testing procedures giving scalable numbers rather than applying user-defined classifications.

Background

Neural networks

Feed forward neural networks or multilayer perceptrons (MLPs)[17] belong to the family of supervised learning algorithms and can approximate any function with arbitrary precision given there is enough hidden layers and units.[28] They are inspired by biological neurons and in their simplest form they belong to the class of generalized linear discriminants where the equation of the output of the network is calculated from the input x (Fig. 1):
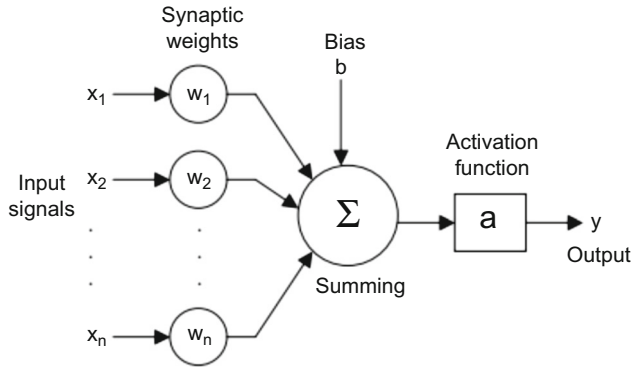
Fig. 1: Single neuron MLP, image adopted from reference (28, p. 6)

$$\hat{y}_{MLP}(\boldsymbol{x}; w, b) = a\left(\sum_{i=0}^{n} w_i x_i + b\right), \text{ where } a(x)$$
$$= \frac{1}{1 + \exp(-x)} \tag{1}$$

where w_i, the corresponding weight factor, and b, the bias term, are learnable parameters. $a(x)$ is called activation function and adds nonlinearity to the network to increase the expressibility of the model; in equation (1) the *sigmoid* function is used, which is the most common activation function. In this form, a MLP can easily be extended to perform multidimensional linear regression and multiclass classification. However, the expressiveness of neural networks is based on the extension to multiple hidden layers. Addition of hidden layers and extension of the number of classes to k change the equation for the prediction to:

$$\hat{y}_k(\boldsymbol{x}; w, b) = a^{(2)}$$
$$\left(\sum_{j=0}^{n_2} w_{kj}^{(2)} a^{(2)} \left(\sum_{i=0}^{n_1} w_{ij}^{(1)} x_i + b_1\right) + b_2\right) \tag{2}$$

The number of hidden neurons in each layer can be selected arbitrarily, such that the corresponding weight matrices and bias vectors $\mathbf{W}, \mathbf{b} = \left(W^{(1)}, b^{(1)}, \ldots, W^{n_l} b^{n_l}\right)$ are obtained. The number of hidden layers and the size of each layer represent typical hyperparameters for the architecture selection of the model. Learning occurs via a loss function $\boldsymbol{J}$ that calculates the deviation of the model's prediction $\hat{y}$ from the ground truth vector t. A commonly used loss function is represented by the binary crossentropy loss (BCE), which optimizes the model for binary classification (i.e., foreground vs background segmentation of a pixel in images):

$$\boldsymbol{J}_{BCE}(t, \hat{y}) = -\frac{1}{n} \sum_{i=0}^{N} t_i \log(\hat{y}_i) + (1 - t_i) \log(1 - \hat{y}_i) \tag{3}$$

where n represents the number of data points and t_i represents the i-th element of the ground truth vector. The selection of a loss function is crucial and depends heavily on the task to be solved. A series of loss functions was implemented in this work which will be covered later. In order to minimize this loss function, one utilizes the backpropagation algorithm[29] which calculates partial derivatives with respect to the weights. The most basic approach for optimization is stochastic gradient descent (SGD), where one iteration of the algorithm updates the learnable parameters w and b in the following way:

$$w_{ij}^{(\tau+1)} = w_{ij}^{(\tau)} - \alpha \frac{\partial J(w, b)}{\partial w_{ij}} \quad b^{(\tau+1)} = b^{(\tau)} - \alpha \frac{\partial J(w, b)}{\partial b} \tag{4}$$

The hyperparameter α is called *learning rate* and controls the learning by scaling the obtained gradients, which is an important factor for effective learning. Training a neural network often leads to local minima, which can be avoided by choosing an appropriate value for α. Other optimization algorithms, like Adam,[30] have more sophisticated mechanisms to deal with the learning rate; thus, we applied Adam for our application.

Convolutional neural networks and image segmentation tasks

Convolutional neural networks (CNN) have attracted a lot of interest recently. Compared to conventional supervised machine learning methods, deep learning methods do not depend on hand-crafted features, but automatically learn a hierarchy of increasingly complex features directly from data.[17] Because images are large, often with several hundred variables (pixels), a fully connected first layer with hundreds of hidden units would already contain several tens of thousands of weights. Such a large number of parameters increases the cognitive capacity of the system, and not only requires a larger training set, but rules out certain memory limitations to store so many weights. However, the main deficiency of plain MLPs for images is the lack of built-in invariance with respect to translations or local distortions of distinctive features in input objects. In CNNs, as described below, shift invariance is automatically obtained by forcing the replication of weight configurations across space. Secondly, a deficiency of fully connected architectures is the ignorance towards topology of the input, because visual data has a strong 2D local structure: pixels that are spatially or temporally nearby are highly correlated (e.g., edges, corners, etc.). CNNs force the extraction of

local features by restricting the receptive fields of hidden units to be local.[31]

The linear mathematical 2D convolution operation addresses these problems and together with the pooling operation, they form the basic building blocks of CNNs and leverage neural networks by three important ideas: sparse interaction, parameter sharing and equivariant representations.[17] A first CNN called *LeNet* was presented by LeCun;[31] important contributions where made through *AlexNet*[22] and the *VGG* architecture.[21] A nice visualization of the effect of convolutions along the hierarchy of the network architecture was given by Zeiler and Fergus[32] (see Fig. 2b). The figure displays from left to right the increasing granularity of the feature extraction, from coarse edges to fine-grained shapes to complete subparts of the target patterns. These inital CNN architectures outperformed all previously existing methods for image classifiation by a large margin.[22]

Another important contribution was the implementations of residual connections by He et al.[23] A residual block can be thought of as a shortcut in the network that allows the effective propagation of gradients to the earliest layers during backpropagation (Fig. 3a). While one could have formulated the evaluation of the crosscut test as a classification task, the novelty and effectiveness of this work is that the problem is approached as a segmentation task. In semantic segmentation one aims to label each image pixel with predefined category labels (Fig. 3b), which in this case includes delamination and intact coating areas.

Solution approach and experimental setup

Sample preparation

For the preparation of samples several coatings and substrates have been combined creating a data set covering all possible categories of the crosscut test, where the coating was applied to the substrates via scrambling. In order for the model to better generalize to unseen data, it is advantageous to create a diverse dataset that contains a high degree of variance in the patterns it aims to learn. We used four types of coatings with diverse colors and degrees of gloss as well as four types of substrates (steel, electrogalvanized, phosphated, and glass) to create samples with differing appearances. For details of the coatings used and the various combinations of coating and substrate, please refer to supplementary information (see Supplemental Material S II). While the steel, electrogalvanized, and the glass substrate would be first cleaned with n-butyl
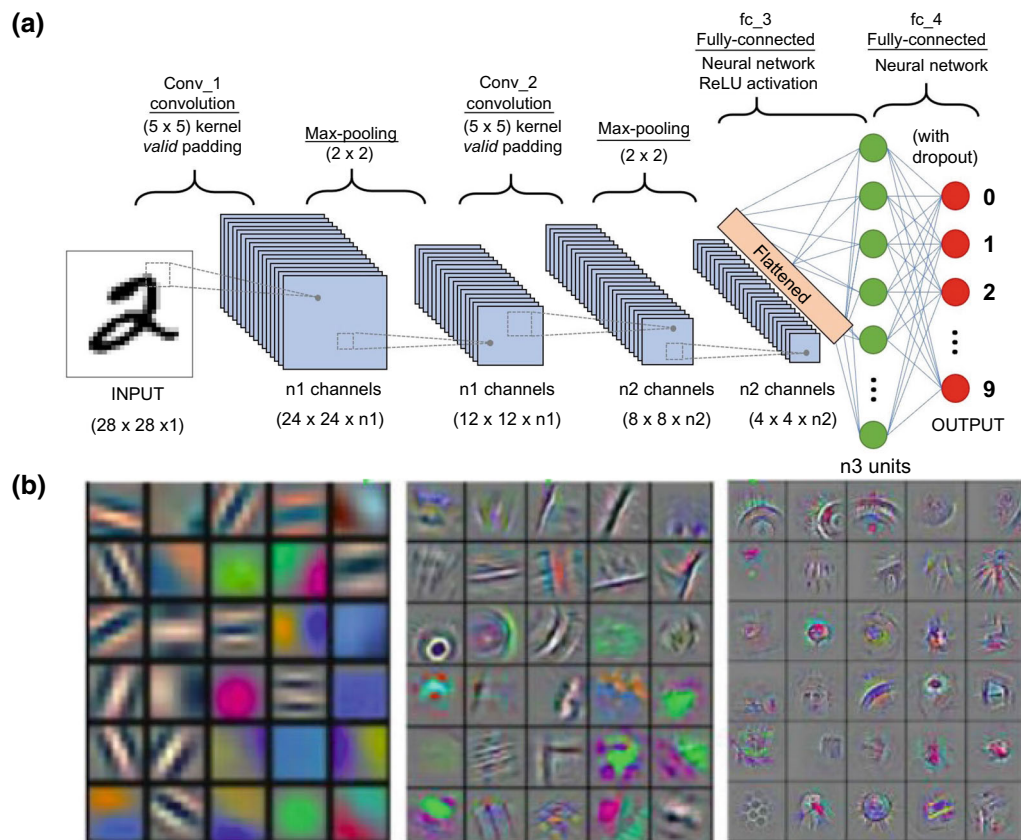

Fig. 2: (a) Architecture of CNNs [Source: Sumit Saha, via http://towardsdatascience.com (CC0)]; **(b)** effect of multiple convolutional layers, taken from reference (32)
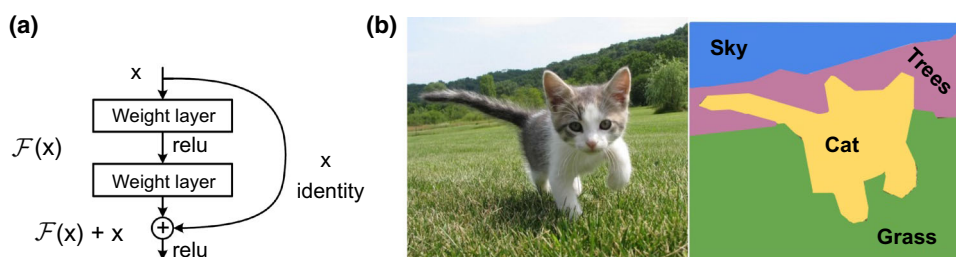
Fig. 3: (a) Residual block taken from reference (23); **(b)** Semantic segmentation [Source: Fei-Fei et al., via http://www.stanford.edu/ (CC0)]

acetate, the phosphated substrates were directly used for coating application. The resin and the hardener of the two-component polyurethane system were mixed at 4:1, 2:1 and 1:1 according to the functional groups to achieve different crosslinking ratios. In this way, we created samples with not only good, but also bad adhesion between the coating and the substrate. Then, the mixed coatings were drawn down to 80 μm for each type of substrate. The polyurethane coated substrates as well as the one-component waterborne alkyd resin were allowed to dry at room temperature with a relative humidity of $(50\% \pm 5)$ for two weeks. For all samples the achieved dry file thickness lies between 40 and 60 μm based on different types of coatings. According to ISO 2409:2020, the crosscut test was carried out with a 1 mm pattern on the coated samples. The gloss of the coating varies from matt to glossy according to the different coatings types.

Automated crosscutting

The crosscut and imaging have been carried out with an automated high-throughput equipment, that handles coated test panels and cuts, scratches or brushes the surface, controlled by an embedded computer. The device was set up comparable to other automated platforms processing such destructive mechanical testing being available off-the-shelf.[33] In addition to the referenced commercial devices, it possesses an industrial camera system and a linear motor table, which serves as an interchange position for coated panels. This allows integration of the mechanical testing into large scale high-throughput equipment, that formulates lacquers, applies them by spraying or draw-down, and stores the samples under controlled atmospheric conditions for hardening and further testing.[34] The automatic rating of the pictures by the computer algorithm and the exchange of coated samples facilitates a full cycle of sample generation and characterization in an experimental screening design or the analysis with adaptive learning strategies by the computer-assisted system.[34] A rail-system moves the panels from the storage location to the testing and imaging station. Two z_1, z_2-linear servo drives with clamping systems can be mounted with tools like a Wolfram needle for cutting,

and a brush to remove delaminated material. The pattern of the cut and reciprocal strokes with the brush removing delaminated coat were accessible due to the maneuverability of the automation. Furthermore, the force resulting in a pressure of the two tools on the surface was controlled during the movement and achieved comparable destruction regardless of the hardness of the coated surface. The controlled force of the tools also allows mechanical tests like the scratch resistance[35] that require different loads identifying the point of material failure. All samples for this study were first cut with a force of 5 Newton and then brushed with a force of 8 Newton. Three crosscuts were carried out on each panel for the evaluation of the delamination. The imaging was performed with an optical setup for a balanced illumination of the sample avoiding artifacts from reflections or other side effects (Fig. 4).

Data preprocessing

The dataset contains 217 images, which have been taken with a camera under constant conditions and with no further changes to the camera and crosscut test parameters, to ensure objectivity of the process. The images have then been cropped to contain only the 'cutted-on' area, where the grid has been applied to the substrate. Subsequently, a graphics editor was used to define the ground truth as a binary mask, i.e., manually labeling each pixel location with a value of 255 for areas in which there is no coating left and 0 for pixel locations that still show a coating on the surface. In addition, the image is transformed from RGB color space into grayscale. The manual segmentations have been used as the ground truth in both segmentation model training and final segmentation performance evaluation. This process is illustrated in Fig. 5, where the center and right picture form one of the input data points for the training of the algorithm.

The goal of the model is to reproduce the existing and nonexisting coating locations for each pixel when given a new, previously unseen sample image. To relate to DIN EN ISO 2409,[9] the percentage of the remaining coating was calculated to characterize the adhesion strength of a certain sample in an automated
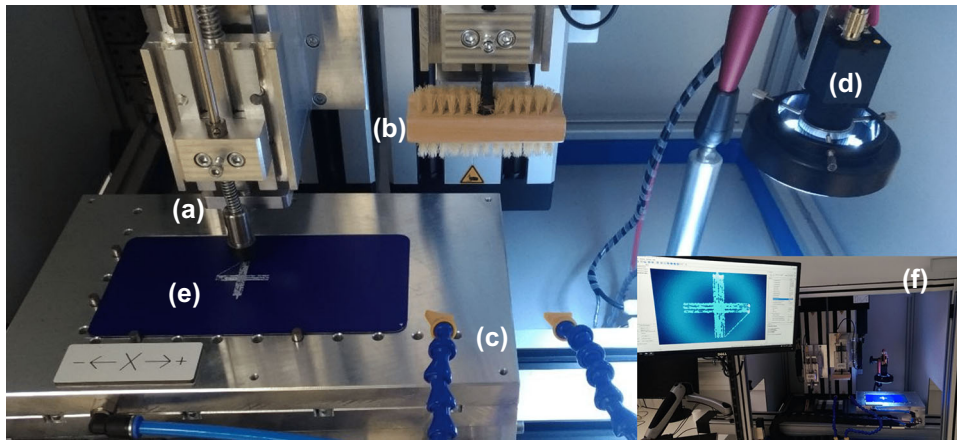
Fig. 4: Images of the automated crosscut testing equipment. The tools for cutting (a) and brushing (b) are connected to the *x, y, z*-linear motors. Pressure air (c) removes delaminated coating shreds before imaging under ring light (d). A linear motorized table moves the substrate from the transfer point to the automated tools. Inset picture (f): View of the complete setup with a computer screen for the imaging and analysis
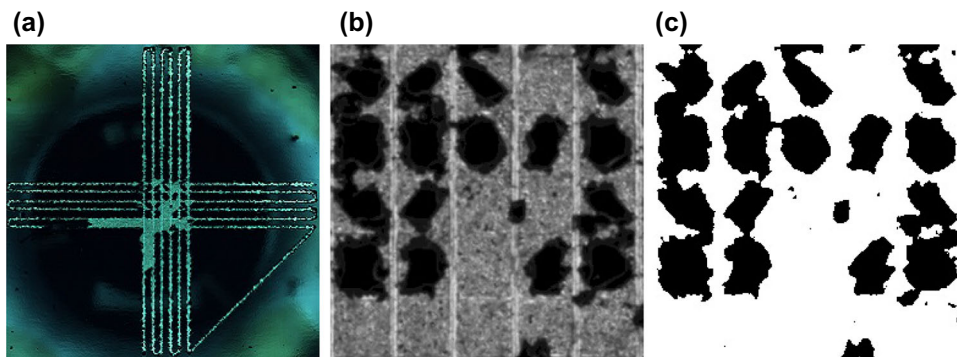

Fig. 5: (a) original image; (b) cropped image; (c) manually labeled image

manner. There is large consent that successful training of deep networks requires many thousand annotated training samples.[17] While this may be true for the traditional classification tasks, where you have one target for an entire image, it is different for segmentation tasks. Here, each pixel location of the image serves as a target (either 0 or 255), which is why the network can adapt much easier as there is more ground truth available to learn from reference (26). The dataset is divided into a training, validation, and testing set:

- 174 training points, where each data point consists of a grayscale image with a resolution of 256 by 256 pixels and a binary mask of the same resolution,
- 35 data points for validation during training, and
- The remaining 8 data points were kept to form a test set combined with 23 selected points from the validation set to contain 31 points for evaluation.

Data augmentation

Data augmentation can help to improve the network performance by intentionally producing more training images from the original images using linear transformations.[21,22] In this study, a set of data augmentation methods summarized in Table 1 has been applied. Simple transformation such as flipping, rotation, shift, and zoom can result in displacement fields to images, but will not create training samples with very different shapes. Shear and zoom operations can slightly distort the global shape. When an image is shifted by 20%, there is some free space which is filled as in fill mode.

Architectures

The basic idea of *U-net*-like architectures follows an encoder-decoder style, where the model first contracts the input by convolution and pooling operations as in usual CNNs. The feature maps are then upsampled again using the deconvolution operation resulting in an

Table 1: Summary of the applied data augmentation methods

Method	Parameter
Rotation	$\pm\ 20°$
Shift	Both horizontal and vertical with 5% probability
Shear	5° shear angle in counter-clockwise direction
Zoom	5% range for random zoom in any direction
Flip	Random horizontal and vertical flip inputs with 50% probability
Fill-mode	'Nearest' fill the area with the nearest pixel and stretching it
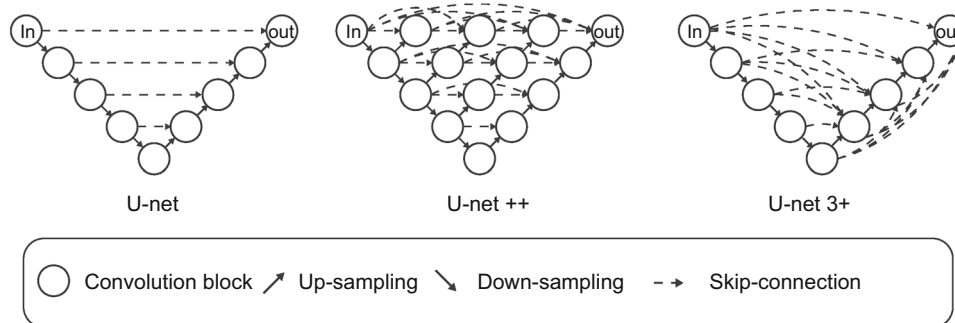
Fig. 6: UNET Architectures (a) Modified U-net; (b) U-net++; (c) U-net 3+ (Color figure online)

output with the same shape as the input. The output is of the same shape as the input, because ultimately the goal of the model is to reproduce a correct binary label for each pixel (see Fig. 5c) of a previously unseen new image (see Fig. 5b). While a convolution goes from higher to lower resolution, the deconvolution operation (also called transposed convolution) goes in the opposite direction. The other important feature is the usage of skip connections (see Fig. 3a) to combine the high-level semantic feature maps from the decoder and corresponding low-level detailed feature maps from the encoder (dashed lines in Fig. 6). *U-net++* and *U-net 3+* differ from the original *U-net* by a reorganization of these skip connections. *U-net++* also implemented a series of nested dense convolutional blocks to bridge the semantic gap between the feature maps of the encoder and decoder. *U-net 3+* differs from this version by omitting these dense convolutional blocks and just reorganizing skip connections yielding a model that captures fine-grained details and coarse-grained semantics in full scales. These three different model architectures have been used in our study and are illustrated in Fig. 6. Each model architecture has been slightly modified from the original versions in references (26, 36, 37). First, the implementation was transformed from PyTorch to TensorFlow/Keras, as the latter is currently the more popular and more efficient machine-learning framework. For the *U-net++* and *U-net 3+* models, we have removed 'Deep Supervision' which basically enables supervision at each decoder stage. Our model corresponds to what the authors call 'U-net w/o DS' in their papers. No

additional adaptations to *U-net++* and *U-net 3+* have been made, but the training was performed with different loss functions (see next subsection). The original *U-net* architecture is the oldest model (2015) and several modifications were performed to improve the performance. The implementation details can be found in supplemental material (S-I).

Loss functions

Each of the presented models (Fig. 6) can be paired with a different loss function. As mentioned earlier, the choice of a loss function is crucial as it influences the training heavily. The loss function defines the optimization criteria which is evaluated for each batch of data and thereby computing gradients, which are backpropagated for each individual weight in the network. In recent years, several optimization criterion (also called metrics) have been introduced to evaluate the accuracy of segmentation approaches leading to a list of multiple different loss functions.[38] The best metric in terms of interpretability is the Intersection-over-Union (IoU),[39] also called Jaccard distance (JD), which basically computes the ratio of correctly predicted pixel locations, i.e., True Positives*, over the sum of all target locations and incorrectly predicted locations. During the training usually no 'firm' pixels, but the output probabilities ($\hat{y}_i$) for each location are assigned.

---

* TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives

In order to use the IoU, one introduces a smoothing factor s to avoid numeric issues, which is omitted here as it is an implementation detail. The Jaccard distance loss is defined by:

$$IoU = \frac{TP}{TP + FP + FN} \quad IoU(t, \hat{y}) = \frac{\sum_{i=1}^{n}(\hat{y}_i t_i)}{\sum_{i=1}^{n} \hat{y}_i + t_i - \hat{y}_i t_i} \tag{5}$$

$$\boldsymbol{J}_{JD}(t, \hat{y}) = (1 - IoU(t, \hat{y})) \tag{6}$$

where $\hat{y}$ denotes the vectorized prediction, t the target vector and n the number of pixels. Another important metric is called the Dice coefficient,[40] which is equivalent to the F1-Score (harmonic mean of recall and precision):

$$Dice = \frac{2TP}{2TP + FP + FN} \quad Dice(t, \hat{y}) = \frac{\sum_{i=1}^{n} \hat{y}_i t_i}{\sum_{i=1}^{n} \hat{y}_i + t_i} \tag{7}$$

$$\boldsymbol{J}_{Dice}(t, \hat{y}) = 1 - Dice(t, \hat{y}) \tag{8}$$

The binary crossentropy (BCE) Loss and the Dice loss can be paired to form a logarithmic version $\boldsymbol{J}_{Hybrid}$:

$$\boldsymbol{J}_{Hybrid}(t, \hat{y}) = \boldsymbol{J}_{BCE}(t, \hat{y}) - \log(1 - \boldsymbol{J}_{Dice}(t, \hat{y})) \tag{9}$$

In reference (41), the authors propose a novel loss function that addresses problems of the Dice coefficient such as the equal weighting of FN and FP. It generalizes the Dice loss by introducing hyperparameters which can fine-tune the model. The Tversky Index (TI) is defined by:

$$TI = \frac{TP}{TP + \alpha FP + (1 - \alpha)FN}$$

$$TI(t, \hat{y}) = \frac{\sum_{i=1}^{n} t_i \hat{y}_i}{\sum_{i=1}^{n} t_i \hat{y}_i + \alpha \sum_{i=1}^{n}(1 - t_i)\hat{y}_i + (1 - \alpha)\sum_{i=1}^{n} t_i(1 - \hat{y}_i)} \tag{10}$$

where α ranges from 0 to 1, which in our approach was set to 0.7. The Focal Tversky Loss is then defined as:

$$\boldsymbol{J}_{FT}(t, \hat{y}) = (1 - TI)^{\gamma} \tag{11}$$

where the hyperparameter γ was set to 0.75 in our implementation. In this setting, the model can optimize itself even when the loss is low encouraging it to learn more details about the images.

Evaluation

Training

The deep convolutional network's training and testing were performed on a system running Ubuntu 20.04 with a AMD EPYC 7502P CPU @ 2.50 GHz (32 cores) with 256 GB of RAM and NVIDIA Quadro RTX 8000 (4608 Cuda cores) with a graphics memory of 48 GB. On the software side Python 3.8.5, TensorFlow 2.3.0, CUDA 10.1 and libcudnn 7.6 were used. The Adam optimizer with initial learning rate of 0.0001 and a batch size of 4 were chosen and each architecture was trained with the four losses introduced in the last section using the dice coefficient as the metric to evaluate the performance on the validation data during training and trained every model for 100 epochs.

Results

As mentioned before, we have set up a test set with 31 data points, where some of these data points have been used for validation during training, however the information of these points was not back-propagated and thus can be regarded as 'unseen'. In the following, first a table is presented containing the mean dice coefficient (mDC) and its standard deviation evaluated on the test set for the three models paired with each loss function as well as the inference time for the output and the number of parameters for each model. A confusion matrix aggregated overall 31 test images (see Table 2), i.e., $31 \times 256^2 = 2031616$ pixels in total are presented. The segmentation approach is further visualized by highlighting true positive, true negative, false positive or false negative (see Fig. 7) on a specific sample, where the visualization was created with the *U-net* architecture trained on the BCE loss function for only 10 epochs. It is evident that the segmentation already performed very well, and errors occur only at the boundaries between fore- and background, which are not of significant importance for the final percental classification.

Finally, the image classification was directly compared with the results of human ratings by a group of lab technicians (see Table 3), because traditionally the human user estimates the delamination and classifies the defect manually. The fraction of pixels accounting for the sole crosscut was determined to be approximately 25%. The ratio of fore- and background was rescaled and converted to a categorical value via the ranges defined in Table S-III (Supplemental Material) in accordance to the standard[9] applying the aforementioned heuristic. Table 3 illustrates one sample for each level, respectively, and the comparison for the overall test samples is shown in the supplemental information (S-IV).

Table 2: Confusion matrices of the various models being applied for the segmentation into delaminated and intact area

Model	TP	TN	FP	FN	MC (%)
U-net BCE-Loss	1038613	826275	119540	47188	8.2
U-net Hybrid-Loss	1015222	833869	111946	70579	9
U-net JD-Loss	1044455	824061	121754	41346	8
U-net FT-Loss	1067902	796155	149660	17899	8.2
U-net++ BCE-Loss	1078702	743299	202516	7099	10.3
U-net++ Hybrid-Loss	1071194	757005	188810	14607	10
U-net++ JD-Loss	1069129	808316	137499	16672	7.6
U-net++ FT-Loss	1076982	798863	146952	8819	7.7
U-net 3+ BCE-Loss	1059228	929381	16434	26573	**2.1**
U-net 3+ Hybrid-Loss	1060248	928031	17784	25553	**2.1**
U-net 3+ JD-Loss	1064279	906687	39128	21522	3
U-net 3+ FT-Loss	1079253	816668	129147	6548	6.7

Each pixel in the output of the segmentation was examined to be either true positive (TP), true negative (TN), false positive (FP) or false negative (FN). The overall fraction of false predictions is given through the misclassification rate (MC)

Bold indicates the model with lowest misclassification rate
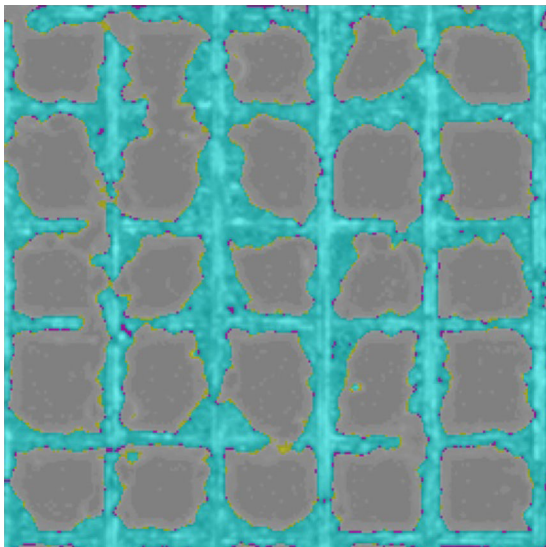
Fig. 7: Image of a crosscut and analysis of the segmentation in analogy to the confusion matrix (Table 2). The colored area depicts true and false segmented pixels of the image into the two labeled intact (positive) and delaminated (negative) areas. Cyan—True Negative, gray—True Positive, yellow—False Positive, magenta—False Negative
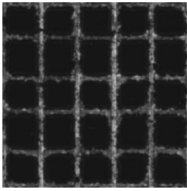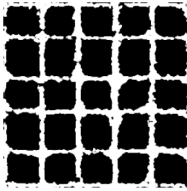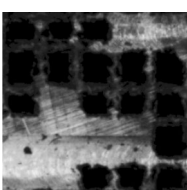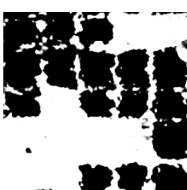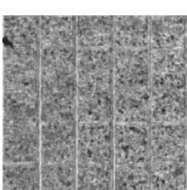
Discussion
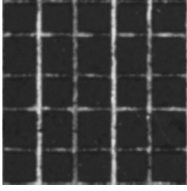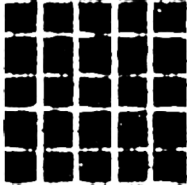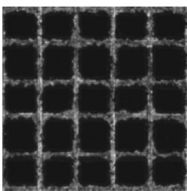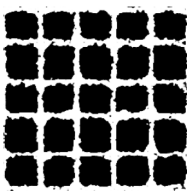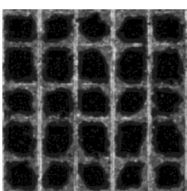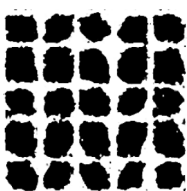
It is evident that the *U-net 3+ BCE* and the *U-net 3+* Hybrid model outperform all other implementations. With a misclassification rate of less than 3%, they can evaluate samples for the crosscut test with more precision than required by the DIN ISO Norm.[9] The *U-Net++* and *U-net 3+* models can solve the problem to a sufficient degree even without 'Deep Supervision'.

The standard deviations for the mDCs are rather high (see Table 4). This can be traced back to a few test images where the model performs poorly. Those samples consist of a bright coating applied to a metal substrate, the bad model performance is due to the fact that these types of samples were under-represented in the training set. The manual labeling to create a binary mask for each data point is tedious and needs to be done with much care and may have led to a slight loss of performance due to lack of detail in labeling. Some of the errors of the model can be traced back to ambiguous regions in the original image.

In the future, we will use methods such as laser-scanning microscopy with a higher resolution that will help to achieve a higher accuracy in labeling. Additionally, the image size, which was chosen to be 256 × 256 due to hardware limitations, is low and can be increased. We have not used any noise reduction in the dataset, which is a fairly easy task and will be implemented in a future work. We will also apply a histogram equalization to the images in order to eliminate the influence of illumination inspiration. Furthermore, other model architectures can and should be taken into consideration. Table 2 shows the separation in the two domains of delaminated and intact area. The separation and its ratio allows the computational evaluation of the crosscut, which can be the continuous percental ratio or a discrete classification in accordance to the evaluation scale of the standard procedure.[9] The continuous value should be favored for the purpose of a subsequent computational data analysis of an experimental design, but the confidence of such a scale requires further statistical analysis since its use is so far uncommon.

The legitimacy of the image recognition by the *U-net 3+* model trained with the Hybrid loss function was examined via comparison of the computational with

Table 3: Comparison between human and algorithmic rating for samples selected across all six levels defined in the norm[9]

Input image	Prediction	Intact area (%)	Algorithmic rating	Human rating
		76.5	0	0 ± 0
		72.23	1	1 ± 0.51
		69.68	2	1 ± 0.45
		60.3	3	2 ± 0.29
		46.43	4	5 ± 0
		0.1	5	5 ± 0

The first two columns show the input and the predictions calculated by the *U-net 3+* architecture trained on the Hybrid Loss function. The fraction of pixels corresponding to the undamaged coating is obtained by the prediction and classified with respect to the standard procedure

the human rating of the 31 test images (see Table S-IV). The model does not calculate an average value, but outputs a definite value for adhesion of the sample and therefore there is no associated standard deviation available, because there is only a single value for each sample. However, in the evaluation we discuss the misclassification rate of individuals pixels in detail, which one may regard as a general uncertainty of the method. The user ratings deviate analogously to the precision of the procedure given by the standard,[6] but the standard does not differ between the error originated by the sample handling and the subjective rating.

Table 4: Mean Dice coefficient in % calculated over all 31 test images with standard deviation, inference time for the test set and the number of parameters for each model paired with each of the different losses

mDC (%)	BCE	Hybrid	JD	FT	Inference time (ms)	# param.
U-net	91.3 ± 18.1	89.8 ± 18.7	91.6 ± 17.9	91.5 ± 17.3	6	8.641.697
U-net++	89.2 ± 18	89 ± 18.7	92.4 ± 17.6	92.4 ± 17.4	8	9.041.601
U-net 3+	$\mathbf{96.3 \pm 6.8}$	$\mathbf{96.2 \pm 6.8}$	95.5 ± 9.7	93.2 ± 17	22	14.400.577

Bold indicates the model with lowest misclassification rate

The standard deviation is low for values at the upper and lower bound of the scale since the rating is unambiguous if the coating is visibly fully delaminated or intact. If the coating is partly delaminated, the evaluation almost always includes a standard deviation between 0.29 and 0.95, which is in line with the conclusion of the standard, that recommends a difference not bigger than one or two levels for repeatability and reproducibility.[6] Thus, the benchmark for the image classification by the algorithm should be at least within this recommended limitation of the method. The comparison of the results in Table S-IV between the human and computational rating fulfills these demands. Therefore, the fully automated process including the computational evaluation also provides valid results regarding the traditional crosscut classification. Again, the continuous value of the delaminated area should be favored for the evaluation of the adhesion strength since the statistical error of the traditional rating is high and misses precision in research studies due to the assumed deviation of ± 1 class.

Conclusions

In this paper, we have shown that the crosscut test can be performed by an automated system being able to control the force of the cutting tool depending on the hardness and bending properties of the substrate. With fully 3-dimensional control of the system, any mechanical stress (e.g., scratch or abrasion) can be operated with this tool including automated imaging of the area of interest. The deep learning algorithm is able to recognize the difference between intact surface and ablations and segments the picture transferring only the information of importance for evaluation of the defect and thus increases objectivity of the whole process. The adhesion strength can be evaluated by the percentage of delaminated area being recognized and also classified according to the norm for the crosscut test.[9]

The corresponding evaluation by human operators is presented and compared. The major advantage of the segmentation approach over classification is the introduction of a continuous value, which is a crucial feature to compute gradients for data analysis with computational methods. Moreover, this segmentation method can be expanded to other applications in which optical inspections are common. For instance, this could be brittle fracture of a scratch or blistering and pinholes,[6,7] which are usually evaluated by the average size of one defect or the overall affected area. Furthermore, similar applications of automated testing and image recognition can be applied in the quality control of manufacturing processes improving the data quality of a various number of processes related to Chemistry 4.0.

References

1. Streitberger, HJ, Goldschmidt, A, *BASF Handbook Basics of Coating Technology, European Coatings* (2018)

2. McKnight, ME, Martin, JW, "Advanced Methods and Models for Describing Coating Appearance." *Prog. Org. Coat.*, **34** (1–4) 152–159 (1998)

3. Pappas, SP, "Weathering of Coatings-Formulation and Evaluation." *Prog. Org. Coat.*, **17** (2) 107–114 (1989)

4. Fotovvati, B, Namdari, N, Dehghanghadikolaei, A, "On Coating Techniques for Surface Protection: A Review." *J. Manuf. Mater. Process.*, **3** (1) 28 (2019)

5. Chalker, P, Bull, S, Rickerby, D, "A Review of the Methods for the Evaluation of Coating-Substrate Adhesion." *Mater. Sci. Eng. A*, **140** 583–592 (1991)

6. ASTM D6037-21, "Standard Test Methods for Dry Abrasion Mar Resistance of High Gloss Coatings." ASTM International, West Conshohocken, PA, 2021, https://doi.org/10.1520/D6037-21

7. ISO, "Paints and Varnishes-Evaluation of Degradation of Coatings-Designation of Quantity and Size of Defects, and of Intensity of Uniform Changes in Appearance-Part 8: Assessment of Degree of Delamination and Corrosion Around a Scribe or Other Artificial Defect." ISO 4628-8

8. Emmler, R, Perez, R, Caon, C, Roux, M, Bogelund, J, Calver, S, van de Velde, B, Adamczak, Z, "Testing Furniture Surfaces." *Eur. Coat. J.*, **9** 30 (2005)

9. ISO2409:2020, Paints and Varnishes, Cross-Cut Test

10. ASTM Standard, "Standard Test Method for Film Hardness by Pencil Test." D3363–20, (2020). https://doi.org/10.1520/D3363-20.

11. Parker, JM, Cheong, YL, Gnanaprakasam, P, Hou, Z, Istre, J, "Inspection Technology to Facilitate Automated Quality Control of Highly Specular, Smooth Coated Surfaces." In: *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, Vol. 3, IEEE, pp. 2567–2574 (2002)

12. Chisholm, BJ, Webster, DC, Bennett, JC, Berry, M, Christianson, D, Kim, J, Mayo, B, Gubbins, N, "Combinatorial Materials Research Applied to the Development of New Surface Coatings VII: An Automated System for Adhesion Testing." *Rev. Sci. Instrum.* **78** (7) 072213 (2007)

13. Rickerby, D, "A Review of the Methods for the Measurement of Coating-Substrate Adhesion." *Surf. Coat. Technol.*, **36** (1–2) 541–557 (1988)

14. Zhao, Y, Wang, J, Cui, X, Wang, H, "The Use of Photoshop Software to Estimate the Adhesion and Rust-Resistant Properties of Coating Film." *Surf. Interface Anal.*, **43** (5) 913–917 (2011)

15. Liao, KW, Lee, YT, "Detection of Rust Defects on Steel Bridge Coatings via Digital Image Recognition." *Autom. Constr.*, **71** 294–306 (2016)

16. Bongardt, Q, Lorek, M, Wilde, E, "Method for the Recognition and Evaluation of Defects in Reflective Surface Coatings." US Patent 5,887,077 (Mar. 23 1999)

17. Goodfellow, I, Bengio, Y, Courville, A, Bengio, Y, *Deep Learning*, Vol. 1. MIT press, Cambridge (2016)

18. Strehmel, B, Schmitz, C, Cremanns, K, Göttert, J, "Photochemistry with Cyanines in the Near Infrared: A Step to Chemistry 40 Technologies." *Chem. Eur. J.*, **25** (56) 12855–12864, https://doi.org/10.1002/chem.201901746.(2019)

19. Cremanns, K, Roos, D, "Deep Gaussian Covariance Network." CoRR abs/1710.06202. arXiv:1710.06202

20. Meincke, H, Nickel, JP, Westerheide, P, "Practitioner's Section." *J. Bus. Chem.* **15** (1) 42 (2018)

21. Simonyan, K, Zisserman, A, "Very Deep Convolutional Networks for Large-Scale Image Recognition." In: *International Conference on Learning Representations*, (2015)

22. Krizhevsky, A, Sutskever, I, Hinton, GE, "Imagenet Classification with Deep Convolutional Neural Networks." *Commun. ACM*, **60** (6) 84–90 (2017)

23. He, K, Zhang, X, Ren, S, Sun, J, "Deep Residual Learning for Image Recognition." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)

24. Taigman, Y, Yang, M, Ranzato, M, Wolf, L, "Deepface: Closing the Gap to Human-Level Performance in Face Verification." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708 (2014)

25. Fridman, L, Brown, DE, Glazer, M, Angell, W, Dodd, S, Jenik, B, Terwilliger, J, Kindelsberger, J, Ding, L, Seaman, S, et al., "Mit Autonomous Vehicle Technology Study: Large-Scale Deep Learning Based Analysis of Driver Behavior and Interaction with Automation." arXiv preprint arXiv:1711.06976

26. Ronneberger, O, Fischer, P, Brox, T, "U-net: Convolutional Networks for Biomedical Image Segmentation." In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, pp. 234–241 (2015). https://doi.org/10.1007/978-3-319-24574-4_28

27. Goodfellow, I, Pouget-Abadie, J, Mirza, M, Xu, B, Warde-Farley, D, Ozair, S, Courville, A, Bengio, Y, "Generative Adversarial Nets." *Adv. Neural Inf. Process. Syst.*, **27** 2672–2680 (2014)

28. Csáji, BC, et al., "Approximation with Artificial Neural Networks, Faculty of Sciences, Etvs Lornd University." *Hungary*, **24** (48) 7 (2001)

29. Rumelhart, DE, Hinton, GE, Williams, RJ, "Learning Representations by Back-Propagating Errors." *Nature* **323** (6088) 533–536 (1986)

30. Kingma, DP, Ba, J, *Adam: A Method for Stochastic Optimization*, (2017). arXiv:1412.6980

31. LeCun, Y, Bottou, L, Bengio, Y, Haffner, P, "Gradient-Based Learning Applied to Document Recognition." *Proc. IEEE* **86** (11) 2278–2324 (1998). https://doi.org/10.1109/5.726791

32. Zeiler, MD, Fergus, R, "Visualizing and Understanding Convolutional Networks." In: *European Conference on Computer Vision*, Springer, pp. 818–833 (2014)

33. Cross Hatch Cutting and Adhesion Testing, Scratch Hardness Tester 430 p-iv-smart, Accessed: 09/2021. www.erichsen.de/en-gb/products/surface-testing/adhesion-impact-and-elasticity-testing/cross-hatch-cutting-and-adhesion-testing/cross-hatch-cutting-and-adhesion-testing-6/scratch-hardness-tester-430-p-smart-1?set_language=en-gb

34. Christian Schmitz, GB, Cremanns, K, "Application of Machine Learning Algorithms for Use in Material Chemistry." Comput. Data-Driven Chem. Using Artif. Intell., ISBN: 9780128222492

35. BSI Standards, BS EN ISO 1518-1: 2011-Paints and Varnishes-Determination of Scratch Resistance Part 1-Constant-Loading Method

36. Zhou, Z, Siddiquee, MMR, Tajbakhsh, N, Liang, J, "Unet++: A Nested u-net Architecture for Medical Image Segmentation." In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, Springer, pp. 3–11 (2018)

37. Huang, H, Lin, L, Tong, R, Hu, H, Zhang, Q, Iwamoto, Y, Han, X, Chen, Y-W, Wu, J, "Unet 3+: A Full-Scale Connected Unet for Medical Image Segmentation." In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 1055–1059 (2020)

38. Jun, M, "Segmentation Loss Odyssey." arXiv preprint arXiv: 2005.13449

39. Rahman, MA, Wang, Y, "Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation." In: *International Symposium on Visual Computing*, Springer, pp. 234–244, (2016)

40. Drozdzal, M, Vorontsov, E, Chartrand, G, Kadoury, S, Pal, C, "The Importance of Skip Connections in Biomedical Image Segmentation." In: *Deep Learning and Data Labeling for Medical Applications*, Springer, pp. 179–187 (2016)

41. Abraham, N, Khan, NM, "A Novel Focal Tversky Loss Function with Improved Attention U-net for Lesion Segmentation." In: *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, IEEE, pp. 683–687 (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.