The International Journal of Advanced Manufacturing Technology
https://doi.org/10.1007/s00170-023-11784-5

ORIGINAL ARTICLE

Neural networks for inline segmentation of image data in punching processes

Maximilian Lorenz[1,2] · Robert J. Martin[3] · Thomas Brueckmayr[1] · Christian Donhauser[2] · Bernd R. Pinzer[1]

Received: 3 March 2023 / Accepted: 10 June 2023

Abstract

Punching is a process that is sensitive to a multitude of parameters. The estimation of part and punch quality is often based on expert knowledge and trial-and-error methods, mostly carried out as a separate offline process analysis. In a previous study, we developed an optical inline monitoring system with subsequent image processing which showed promising results in terms of capturing every manufactured part, but was limited by slow image processing. Here, we present a more efficient image processing technique based on neural networks. For our approach, we manually identify the burnish parts in images based on criteria established via an expert survey in order to generate a training dataset. We then employ a combination of region-based and boundary-based losses to optimize the neural network towards a segmentation of the burnish surface which allows for an accurate measurement of the burnish height. The hyperparameter optimization is based on custom evaluation metrics that reflect the requirements of the burnish surface identification problem as well. After comparing different neural network architectures, we focus on optimizing the backbone of the UNet++ structure for our task. The promising results demonstrate that neural networks are indeed capable of an inline segmentation that can be used for measuring the burnish surface of punching parts.

Keywords Image processing · Semantic segmentation · Convolutional neural networks · Deep learning · Punching

Punching is a wide-spread production process that is applied when massive amounts of identical cheap parts are needed [1]. One important quality indicator of parts manufactured by punching is the *burnish height* or *burnish surface area*, which is particularly important for electrical connectors or parts with sealing purposes. A burnish surface area that is as large and continuous as possible is desirable. Note that, as shown in Fig. 1, the burnish height is currently defined in the profile section, not in the surface view. This is the case for many other quality indicators as well [2].

In contrast to the highly economical production process, the evaluation of the cutting surface is still cost intensive and time consuming. Currently, used evaluation methods include metallography, confocal microscopy, tactile measuring systems, or motorized measurement devices [3], which require parts to be taken out of the production process and analyzed separately. This also means that a continuous quality control cannot be guaranteed. To counteract this, we have developed an inline monitoring system which is capable of acquiring an image of each punching surface directly after it emerges from the punching tool [4]. We also developed an automated image processing for the segmentation of the burnish height by an active contours algorithm. This approach showed promising results in terms of accuracy and prescriptive recognition of tool wear. However, the processing of an image (cf. Figure 2) takes 40-60 seconds, which is not acceptable with a cycle time of 80–240 ms; note that stroke rates of up to 1000 strokes per minute are possible [1, 3]. Furthermore, this algorithm cannot recognize *multiple* burnish surface regions, which can occur in many random forms, depending on factors such as material combination of punch and sheet metal, fluctuations within the tensile strength of the sheet metal or the location and geometry of punch edge failures. The exact causal relationship between these factors and the occurrence of disturbances in

✉ Maximilian Lorenz
maximilian.lorenz@hs-kempten.de

[1] Laboratory for Optical 3D Metrology and Computer Vision, University of Applied Sciences Kempten, Bahnhofstr. 61, Kempten 87435, Germany

[2] Laboratory for Machine Tools and Production Engineering, University of Applied Sciences Kempten, Bahnhofstr. 61, Kempten 87435, Germany

[3] Faculty of Engineering, University of Duisburg-Essen, Friedrich-Ebert-Str. 12, Duisburg 47119, Germany

Published online: 12 July 2023

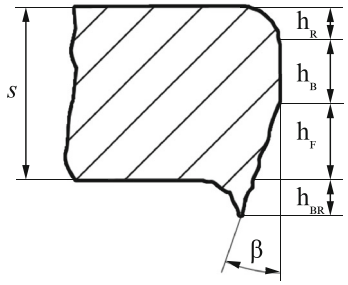


Fig. 1 Definition of the cutting surface parameters [2]

the burnish surface—such as tear sections or holes, as shown later in Fig. 4—has not yet been investigated. However, since these disturbances do occur in real burnish surfaces, they have to be addressed by an image processing algorithm.

Here, we therefore consider a neural-network-based approach for the image segmentation problem. Most generally, machine learning techniques for such tasks can be divided into *instance segmentation* and *semantic segmentation* methods: instance segmentation treats multiple objects within the same class as separated instances, whereas semantic segmentation converts every pixel in the input image to a category class within one instance. Since an algorithm for identifying the burnish surface should classify all pixels as *the* burnish part, we focus on semantic representation networks in the following.

Neural networks have already shown promising results for segmentation tasks in terms of accuracy and processing time,

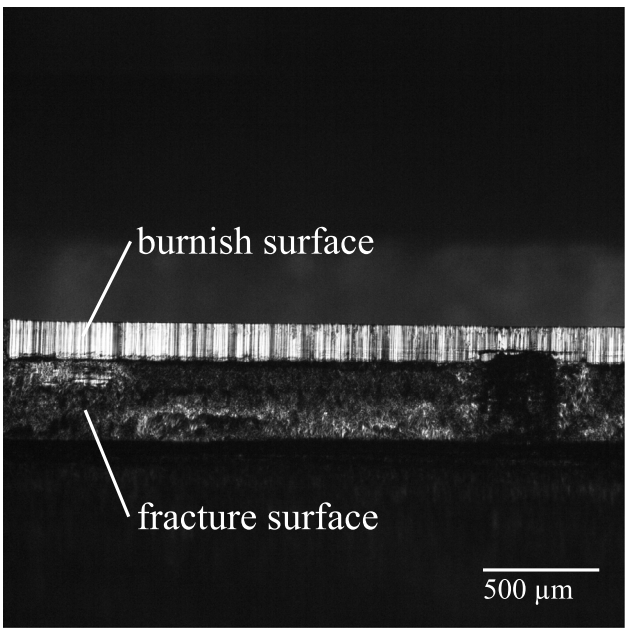


Fig. 2 Image of a produced punching part, captured by an inline monitoring system. The image corresponds to phase 1 as described in the text

for instance, in biomedical image processing [5–7]. Although medical imaging is often subject to greater noise compared to other areas involving image processing, it is possible to recognize and segment tumours even in the most diverse organs. Currently, segmentation networks are attracting interest for the monitoring of manufacturing processes, although collecting and preparing data for the training remains a time-consuming and cost-intensive process [8]. Recent examples for the use of segmentation networks in manufacturing are the measurement of the strip position in a rolling mill production [9] and the detection of surface defects in a steel mill production [8, 10]. Lin et al. [11] and Bergs et al. [12] also developed a method for wear detection of milling tools, while Scime et al. [13] showcased the monitoring of additive manufacturing processes.

The goal of our study is to analyze and optimize networks for the processing of image data to segment the burnish surface of punching parts. The segmentation needs to be accurate even in the presence of multiple disconnected burnish parts and should be realizable within an inference time below 80 ms to be suitable for inline-quality control. To this end, we adapt a network architecture that was originally developed for medical image processing.

1 Materials and methods

1.1 Burnish surface

For our purpose, i.e. online quality control, an accurate *measurement* of the burnish surface—in particular its height—is of primary importance. Therefore, the segmentation must not only be accurate regarding the covered area, but the *shape* of the burnish surface as well. In particular, the *boundary*, i.e. the transition between burnish part and fracture, needs to be identified accurately.

However, it is important to note that determining the burnish height in the surface view is not only a technical difficulty, but rather a conceptual one, since there is no standardized *definition* of the burnish surface. To demonstrate this lack of a commonly accepted definition, we carried out a survey, asking 12 industry experts to mark the transition between the burnish surface and the fracture surface, according to their understanding, in the surface view of a punching part. The results, which are shown in Fig. 3, suggest that there is no clear consensus; rather, the individual definitions of the burnish part are highly dependent on the component produced and its application. However, by investigating the overlapping main characteristics of the different experts' segmentations, we conclude that in a surface-view image, the burnish part

- is brightly illuminated,

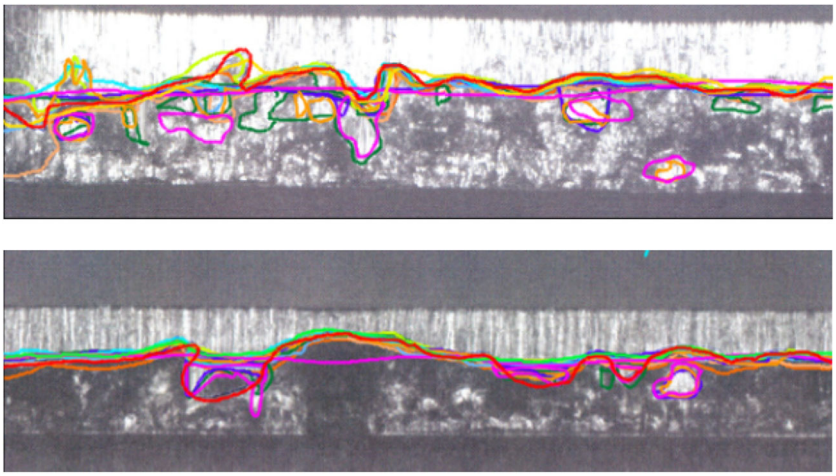


Fig. 3 In a survey, 12 experts were asked to draw the transition line between the burnish and the fracture surface. The scattering of the lines shows that there is no consensus, but rather different application-related approaches

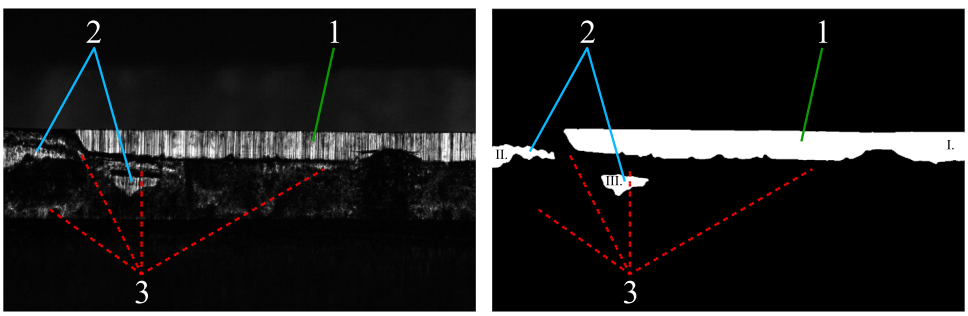
- is fluctuating over its length,
- has a structure with vertical grooves,
- can have holes,
- can have multiple tear sections,
- can increase or decrease in height over the width of the part.

Our manual labelling of the image dataset, as described below (cf. Figure 4), is therefore based on these criteria and the definitions given in [2]. We also note that generally, fewer holes in the burnish part (or none) are favourable for most produced parts, as are fewer tear sections. In particular, an increasing number of holes or tear sections and a decreasing burnish height are a sign of wear on the punching tool.

1.2 Dataset

Images for training, validation, and testing were captured with a monitoring system [4] within the punching process with a resolution of 1280 x 1024 pixels in greyscale. They were taken during a material test where a punch failure occurred on the left-hand side of the images. Overall, 17000 images were captured during this test. In the images, the burnish surface is brightly illuminated, textured with vertical grooves and an inhomogeneous transition to other cutting surface parts. Tear sections of the burnish surface occurred as well, caused by parameter fluctuation or punch failure.

Disjoint subsets of these images were chosen as the training and test data. In order to represent the ongoing wear within a punch lifetime, images were taken from different phases within the dataset: phase one contains images with uniform wear rate and consistent burnish height, apart from natural fluctuation; phase two contains images with progressed wear rate and therefore de-/increasing burnish height. Finally, phase three contains images of parts produced with a damaged punch and show tear-off within the burnish height. In total, 415 images were selected for the dataset. A ground truth mask image was created for each image of dataset by manually segmenting every section of the burnish surface according to the criteria specified above. In particular, the labels provide a per-pixel partition of each image into the classes *burnish surface* and *background* based on expert knowledge. Since all parts – and thereby all images – were produced with the same tool and the same parameters, there is of course a high risk of overfitting to features from this particular process. Since the segmentation should ideally be applicable to images from different processes without re-training (cf. Sect. 4), we try to avoid this effect by extending the dataset via augmentation methods: Each image and corresponding ground truth mask was duplicated and



(a) Original image (b) Manually segmented ground truth mask

Fig. 4 During manual labelling, we distinguished between the main section (1), tear sections (2), and the background (3). Roman numbers denote different components used for the metric evaluation

altered with different operations. These consist of changing brightness values to represent different material combinations, vertical mirroring to change the location of the tears or defects, and scaling of the images to represent different material thicknesses; for simulating thinner materials, the images were compressed along the height axis and inserted into an image with the same background noise to preserve dimensions, while for thicker materials, the images were scaled by a ratio of 1.5 and 3 and clipped randomly along the cutting surface such that they would have the same ratio of pixels below and above, as would be expected from the images of the monitoring system. The full data augmentation structure can be seen in Table 1. Note that since each augmentation technique simulates a difference in material properties, we will consider each of these subcategories individually for our evaluation.

Overall, the image augmentation expanded the dataset to 10086 images, divided into training (6052 images), validation (2017 images) and test (2017 images). Finally, to decrease training time, all images were rescaled to 256×256 pixels. Although a higher resolution might be more suitable for precise measurement tasks, the segmentation functionality can still be analyzed with this reduced image size.

The ratio between background (BG) and foreground (FG) in the image dataset, which is important for the choice of a network and loss function, shows a mild imbalance with a ratio of 9:1, which could increase to 20:1 in applications depending on the specifications of the monitoring system.

1.3 Evaluation metrics and loss functions

1.3.1 Evaluation metric

In order to assess the quality of our neural network based image processing approach, it is crucial to select an appropriate evaluation metric to measure the *accuracy* of the area identified as the burnish surface by the neural network regarding the ground truth labels (i.e. the actual burnish surface in the image according to expert knowledge).

To this end, a combined metric (CM) has been created to evaluate the predictions according to our definition. As indicated above, the total size and, in particular, the height of the burnish part is an important quality indicator. For quantification of the burnish height, however, it is important to obtain a precise segmentation of the boundary. Furthermore, the metric should allow for weighting based on the size and the number of tear sections found, which also play an important role for assessing the part quality.

For a region-based metric, we selected the *Dice similarity coefficient*

$$\mathrm{DSC}(G, S) = \frac{2|G \cap S|}{|G| + |S|}; \tag{1}$$

here and in the following, S and G represent the burnish surface according to the segmentation algorithm and the ground truth, respectively, with $|X|$ denoting the number of pixels in a subset X of the image. Note that, $0 \leq \mathrm{DSC}(G, S) \leq 1$ and that the maximum value 1 is attained if and only if the predicted area S and the ground truth region G are identical.

For the boundary-based part, the *normalized surface distance*

$$\mathrm{NSD}(G, S, \tau) = \frac{|\partial G \cap \partial S^{(\tau)}| + |\partial S \cap \partial G^{(\tau)}|}{|\partial G| + |\partial S|} \tag{2}$$

was used, where $\partial G, \partial S$ denote the boundaries of the segmentation surface and the ground truth, and $\partial S^{(\tau)}, \partial G^{(\tau)}$ represent the border regions at tolerance τ, i.e. the set of pixels whose distance from the boundary is less or equal τ. Note that for $\tau = 0$, this metric only accounts for the predicted boundary pixels which match the ground truth boundary *exactly*, whereas higher tolerance values do not distinguish between an approximate and an exact boundary match.

Finally, the *combined metric*

$$\begin{aligned}\mathrm{CM}(G, S, \tau) = \alpha\ \mathrm{DSC}(G, S) + \beta\ \mathrm{NSD}(G, S, \tau_1) \\ + \gamma\ NSD(G, S, \tau_2)\end{aligned} \tag{3}$$

considers both the region-based DSC and the boundary-based NSD. By selecting the weight factors α, β, γ and the tolerances τ_1, τ_2, this metric prioritizes either the overlap

Table 1 Data augmentation structure of training and evaluation data; the listed colours are used in Fig. 7

Scale	scale 1.0 (blue)	scale 0.5 (orange)	scale 1.5 (red)	scale 2.0 (green)	scale 3.0 (purple)
Augmentation	brighter	brighter	brighter	brighter	brighter
	darker	darker	darker	darker	darker
	plain	plain	plain	plain	plain
	brighter mirrored	brighter mirrored	brighter mirrored	brighter mirrored	brighter mirrored
	darker mirrored	darker mirrored	darker mirrored	darker mirrored	darker mirror
	plain mirrored	plain mirrored	plain mirrored	plain mirrored	plain mirrored

between the identified area and the ground truth burnish surface (for higher values of α) or the accuracy of the predicted outlines of the area.

In the following, we choose the tolerances $\tau_1 = 0$ and $\tau_2 = 1$, which leads to both a positive evaluation for predicted outlines close to the actual boundary and an additional distinction between an approximate and an exact boundary match. Using the weights $\alpha = 0.5$, $\beta = 0.45$ and $\gamma = 0.05$, we put equal emphasis on the area overlap measured by DSC and the boundary matching via NSD. Figure 5 shows the behaviour of the combined metric for different degrees of deviation from the ground truth image.

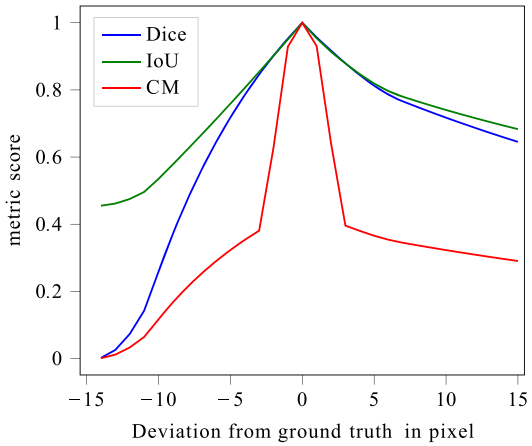
Note that the number of tear sections is not explicitly taken into account by the combined metric, which needs to be calculated for each section individually. Here and in the following, sections are defined as four-way connected area of pixels, with one component for each tear section [14]. Overlapping tear sections in the prediction and ground truth mask are combined into one component. The combined metric is then calculated separately for each of the found components. The metric scores of each component are then weighted in relation to the area of the respective components and summed up; thus larger tear sections have a greater influence on the overall metric score than smaller ones.

While this metric already allows for a general assessment of the accuracy of the prediction, some topological information (e.g. tear sections that are either missing or newly added in the prediction) is not taken into account. To address this problem, we consider the following additional metrics:

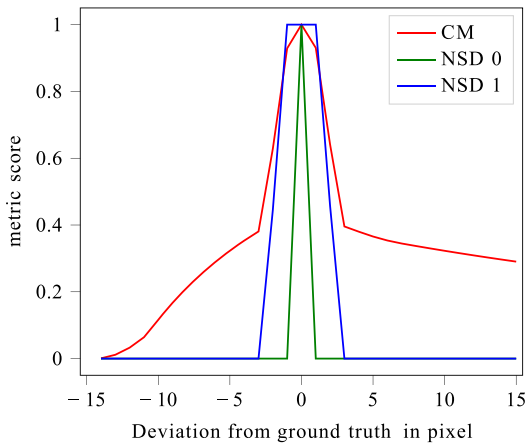
- the ratio between the predicted burnish surface to the ground truth area,
- the percentage of tear sections that could be mapped to components of the ground truth,
- the ratio between predicted and true tear sections,
- the ratio between predicted and true holes.

Here, the term "hole" is defined as an eight-way-connected area which is surrounded by pixels that belong to a different class. These four expansions are well-suited for this work to represent the different properties of the predictions.

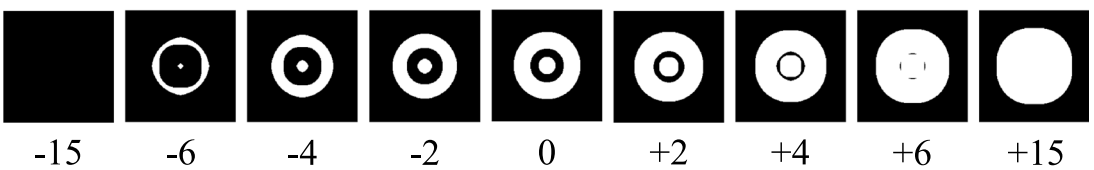
Of course, it is possible for an end user evaluating a segmentation method (such as a neural network) for a specific task to decide based on all the above criteria. In this case, depending on which score is more significant for the task at hand, higher importance can be assigned to particular metrics. For fully automated hyperparameter optimization,



(a) Dice similarity coefficient, IoU and Combined metric



(b) Normalized surface distance with $\tau = 0$ and $\tau = 1$



(c) Over-segmentation and infra-segmation by different numbers of pixels compared to the ground truth (deviation by 0 pixels).

Fig. 5 Different metric scores for images of over-segmentation and infra-segmation

however, it would be necessary to aggregate the individual scores into a single metric, e.g., via a weighted sum.

1.3.2 Loss function

During the actual training of the neural network for given hyperparameters, the parameters of the network are modified to minimize a *loss function* over the set of training data. Choosing a suitable loss function is therefore of major importance to ensure that the prediction by the neural network accurately corresponds to the ground truth. In a previous analysis, [15] compared multiple loss functions on four segmentation tasks. For a dataset containing liver and liver tumour images, which can be considered similar to our dataset based on the BG:FG ratio, a combination loss with a Dice-related compound proved suitable for segmentation tasks. The Dice loss is a region-based loss function that penalizes the mismatched regions between ground truth and prediction, similar to the Dice similarity coefficient. For the general case of images with N pixels and C distinct classes, the Dice loss can be defined by [15, 16]

$$L_{\text{Dice}} = 1 - \frac{2\sum_{c=1}^{C}\sum_{i=1}^{N} g_i^c s_i^c}{\sum_{c=1}^{C}\sum_{i=1}^{N} g_i^c + \sum_{c=1}^{C}\sum_{i=1}^{N} s_i^c}\,, \tag{4}$$

where g_i^c denotes the ground truth binary indicator of class c for pixel i and the s_i^c is the corresponding output confidence of the neural network. Note that if only the burnish surface class with ground truth indicator g is considered, and if the output s is binary, then the Dice loss can be simplified to

$$\begin{aligned} L_{\text{Dice}} &= 1 - \frac{2\sum_{i=1}^{N} g_i s_i}{\sum_{i=1}^{N} g_i + \sum_{i=1}^{N} s_i} \\ &= 1 - \frac{2|G \cap S|}{|G| + |S|} \;=\; 1 - \text{DSC}(G, S)\,, \end{aligned}$$

where DSC denotes the Dice similarity coefficient as defined in Eq. (1).

While the DiceTopK-loss showed particularly promising results in the study by [15], the burnish surface identification problem requires a different approach due to the importance of the transition between burnish and fractured part. In order to emphasize the boundary of the burnish surface over its area distribution, we therefore selected the *DiceBD loss* [15, 17]

$$L_{\text{DiceBD}} \;=\; L_{Dice} + L_{BD}\,, \tag{5}$$

which combines the Dice loss with the *BD loss*

$$L_{\text{BD}} = \sum_{i=1}^{N} \phi_i\, s_i\,. \tag{6}$$

Here, ϕ_i denotes the level set representation of the boundary ∂G of the ground truth region, defined by

$$\phi_i = \begin{cases} -\,\text{dist}(i, \partial G) & \text{if } i \in G, \\ \;\;\text{dist}(i, \partial G) & \text{if } i \notin G\,, \end{cases} \tag{7}$$

where $\text{dist}(i, \partial G)$ is the distance between a pixel i and the boundary ∂G [15, 18].

1.4 Network architecture

For our purpose, it seems reasonable to use a neural network architecture that was specifically developed for processing monochrome images. In particular, we consider several network structures that have been previously employed – or even originally developed – for medical image segmentation tasks. First, neural networks from three selected types of architecture are trained, analyzed and compared on the given dataset. Afterwards, the network that provides the best performance is analyzed and developed further. The chosen architectures are *SegNet* [19], *UNet++* [6], *MedT* [20] and *nnU-Net* [7].

SegNet was originally developed for road scenes, with focus on low memory consumption and efficient computational time [19]. Therefore, this architecture contains fewer trainable parameters than UNet++ or MedT. SegNet's main novelty is the decoder upsampling, i.e. the pooling of indices computed in the max-pooling step of the corresponding encoder to perform non-linear upsampling.

UNet++ [6] is an extension of U-Net [5], which is built for including data augmentation to effectively learn from datasets with very few labelled images. The classical UNet++ network consists of five layers. The encoder is called *backbone* and, compared to U-Net, contains additional skip-connections to the decoder in the form of a pyramid structure, which is supposed to overcome the problem that the outputs of the simple skip-connections in U-Net are too different in kind. In addition, *deep supervision* is introduced into the learning process (cf. Fig. 6).

MedT consists of a global subnetwork with two layers and a local subnetwork with five layers. The global subnetwork processes the complete input, whereas for the local subnetwork, the input images are divided into 16 parts which are processed individually and then reassembled. We selected MedT explicitly as an alternative to classical CNN approaches, since this architecture does not consist solely of convolutions, but includes gated axial-attention layers to act as the main processing units. In addition, the composition of a global and a local branch ensures that the local subnetwork is effectively trained with more images, which can be advantageous for smaller datasets such as the ones considered here. Furthermore, due to the splitting of the input in the local subnetwork, the positional variance of the image

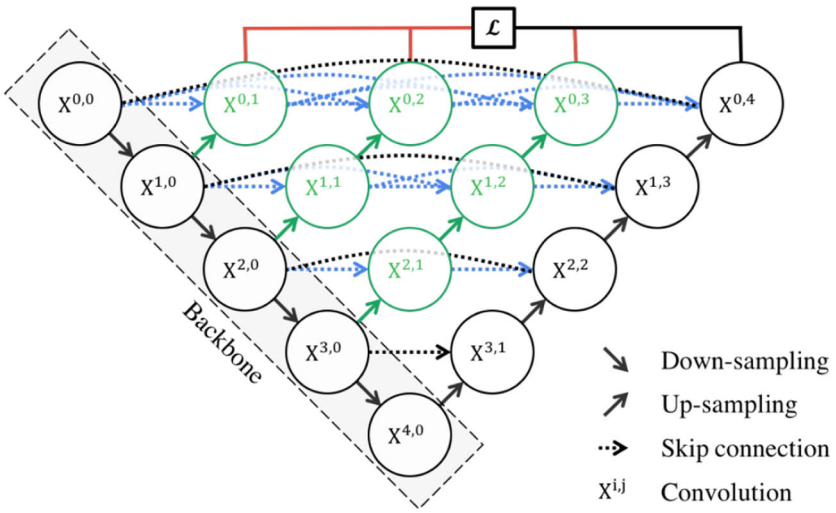


Fig. 6 Structure of UNet++ [6]

content is automatically included in the training process, and the network is confronted with images with different brightness gradients [20].

Finally, nnU-Net is a self-configuring method for medical image segmentation which automatically generates an architecture layout, with training and post-processing based on interdependent rules and empirical descension. It is publicly available and scored best in multiple biomedical segmentation competitions [7].

1.5 Hardware and training

Training and evaluation were implemented in PyTorch [21] with mixed precision and performed on an NVIDIA Quadro RTX 5000. Because of the differences in memory consumption between the networks, different batch sizes had to be used. Every network was trained for 100 epochs. The learning rate started at 3e-4 and was multiplied by 0.2 whenever the moving average of the training had stagnated for 20 epochs until a minimum learning rate of 1e-6 had been reached. The training of nnU-Net was performed in its own framework [7].

2 Network analysis

2.1 Comparison of different architectures

After training an instance of each architecture type, the metric scores were calculated for each augmentation subcategory (cf. Table 1) of the test dataset. The course of the loss function during training shows a successful training (see Fig. 7a). Since real-time segmentation is crucial for the process, the inference time is also taken into consideration. As shown in Table 2, with respect to the combined metric, UNet++ performed 17.94 percentage points better than SegNet, 5.67 percentage points better than MedT and 3.25 percentage points better than nnU-Net. For the other scores, UNet++ also performed comparatively well. Furthermore, the analysis of the metric for each subcategory (see Fig. 7b) shows that UNet++ responds best to three-times enlarged images in comparison with nnU-Net, MedT and SegNet; the latter two, in particular, falsely tend to recognize multiple tear sections instead of a single main section, as shown in Fig. 8. Based on these results, UNet++ is chosen as the most suitable network architecture for identifying the burnish surface.

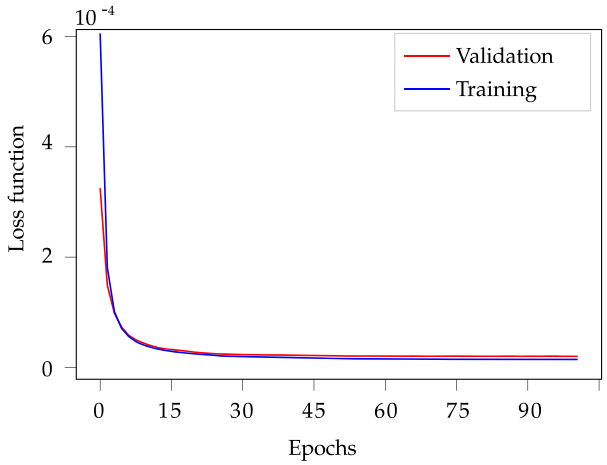
2.2 UNet++ optimisation

For further investigating the properties and hyperparameters of UNet++, we first established a reference score by training the network a total number of five times with default parameters. The mean metric values and their standard deviations are given in Table 3. We then trained the model with different hyperparameter settings and compared the individual metric scores to these reference values.

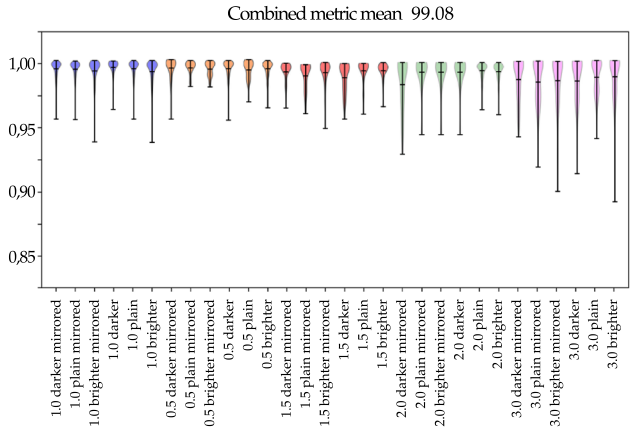
As shown in Table 4, only scores which differ from the reference value by more than a standard deviation are considered significant changes. Note that for some metrics (e.g., the ratio of tear sections), an improvement (△) is indicated by a lower score, while for others (e.g., the combined metric CM), higher scores correspond to a more accurate prediction.

2.3 Hyperparameter variations

The hyperparameters analyzed in the following are the numbers of network layers, feature maps per layer and convolutional layers per block (the *block depth*). Our reference



(a) Course of loss function during training of the UNet++



(b) Evaluation of each image augmentation class

Fig. 7 Course of loss function during training and evaluation of each image augmentation class (cf. Table 1) with the combined metric

UNet++ uses 5 layers and block depth 2 with 32 feature maps in the first layer; this number is doubled with each layer, so that the last layer uses 512 feature maps.

Feature maps To analyze the relationship between the number of feature maps and the prediction, networks with 8, 16 and 64 feature maps in the first layer were compared. The duplication per layer is retained.

The results, as shown in Table 4, indicate a minor, but significant improvement by 0.67 percentage points in terms of the combined metric after increasing the number of first-layer feature maps to 64. Moreover, all other metrics improve with this configuration as well. As expected, less detail is extracted from the image with fewer feature maps. As a result, these networks are less sensitive to changes in the image structure and tend to achieve worse results with a wider distribution.

Table 2 Comparison of architectures with default hyperparameters

	SegNet	UNet++	MedT	nnU-Net
Training time	30 h	28 h	288 h	25 h
Inference time	5.84 ms	4.29 ms	91.24 ms	4.52 ms
Combined metric	81.14	99.08	93.41	95.83
Assignable area [%]	97.32	99.64	96.68	92.51
Assignable tear sections [%]	95.32	99.21	94.42	89.16
Tear section ratio	2.20	2.19	2.86	2.57
Hole ratio	1.70	1.06	1.45	0.84

This is confirmed with images that have been magnified three times.

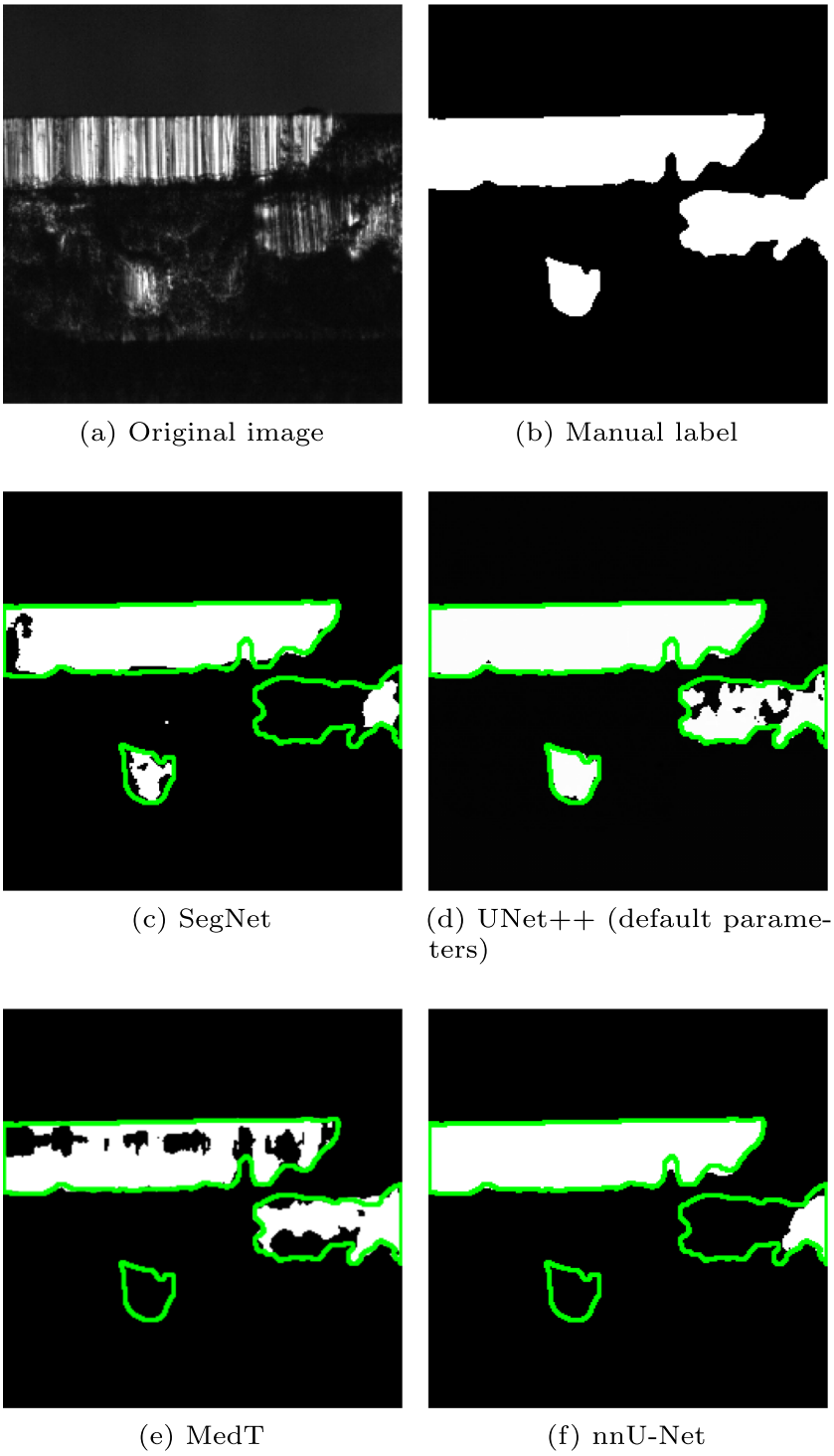
Depth We also considered networks with one and three layers per block ("Depth 1/3" in Table 4). Due to the resulting changes to the amount of data processed per layer, the network with depth 1 shows a decreased inference time, but scores slightly worse overall. A higher number of layers per block, on the other hand, results in minor improvements: more tear sections can be assigned, and the combined metric score is slightly higher (0.08 percentage points above the standard deviation) compared to reference architecture. These advantages, however, come with an increased inference time.

Number of layers Table 4 also shows the comparison between networks with different numbers of layers. While fewer layers deliver worse results in terms of the combined metric, the scores for assigned tear sections and hole ratio is clearly improved by increasing the number of layers, which suggests that the deep layers can assist in processing a more complex feature (such as holes).

Synergy between hyperparameters Considering the results, we selected a network with 64 feature maps, three layers per block and 6 layers for further comparison. This model can be seen as the synthesis between the best-performing hyperparameters and, as shown in Table 5, provides an improvement of the default model by 0.66 percentage points in terms of the combined metric while performing better in every other metric except the hole ratio. We note, however, that the inference time increased significantly as a result of the more complex structure.

2.4 Backbone modification

The previous analysis has shown that competitive predictions can be achieved with only one layer per block. However, during the evaluation, we observed that in this case, the segmentation tends to fail for images with more involved details,



(a) Original image (b) Manual label

(c) SegNet (d) UNet++ (default parameters)

(e) MedT (f) nnU-Net

Fig. 8 Comparison between the predicted burnish surfaces for different network architectures

such as a higher number of tear sections. Increasing the number of layers beyond 5 lead to significant improvements, albeit at the cost of an increased processing time.

As a compromise, we therefore consider an architecture with an increasing block depth per layer and a total of number of 6 layers. More specifically, we modify the UNet++ structure with an incremental block depth such that the blocks in the first (top) layer contain one convolutional layer, the blocks in the second layer contain two layers etc.

The underlying assumption behind this architecture modification is that the processing of the simple properties (e.g. basic positioning and brightness) takes place within the upper layers, whereas the lower layer process more complex features – for example, whether a pixel lies within a larger

Table 3 Mean and standard deviation after fivefold training of the UNet++ architecture

	mean	standard deviation
Combined metric	98.72	± 0.38
Assignable area [%]	99.56	± 0.07
Assignable tear sections [%]	98.75	± 0.42
Tear section ratio	2.2	± 0.01
Hole ratio	1.14	± 0.05

group of bright pixels, how large this group is, how the edge of this group is shaped or whether the group contains a corresponding structure.

However, the results shown in Table 5 suggest that increasing the block depth in lower layers does not lead to better results regarding the different metric scores, whereas the inference time is more than doubled as a result of the more complex structure.

Next, we consider a replacement of the backbone in the UNet++ structure by *DenseNet* [22], similar to work by [23, 24] but extended to the UNet++ structure. Following the underlying assumption that this modification enriches the information about complex features in the deeper layers by connecting each layer with the previous layer via dense connections (see Fig. 9), this should lead to an overall improvement of the boundary details due to recurring influence of features.

Table 5 shows that the dense-backbone architecture indeed leads to comparable or better results regarding the different metric scores with a minor increase in inference time. As a result, the dense backbone is still outperformed by the hyperparameter-optimized network according to the combined metric.

3 Discussion

The above analysis of the hyperparameters and different backbones demonstrates that:

- architectures with fewer than 16 feature maps achieve worse results, but require a shorter inference time;
- architectures with more feature maps achieve better results and require a longer inference time;
- architectures with a lower block depth achieve comparable results and require a shorter inference time;
- architectures with a higher block depth achieve slightly better results and require a longer inference time;
- architectures with fewer layers can achieve worse results and require a shorter inference time;
- architectures with more layers achieve better results and are slightly slower to process;
- architectures with optimized hyperparameters achieve better results, but increase the inference time;
- architectures with increasing block depth achieve worse results and double the inference time;
- architectures with a dense encoder achieve comparable or better results results and require a longer inference time;

Based on these findings, we propose the *UNet++ with 64 feature maps*, as the most suitable configuration. Even if the 6 Layer configuration performs better in some metric scores, the shorter inference time of the selected architecture should be prioritized as it is highly beneficial for the intended purpose of inline segmentation.

4 Transfer evaluation

Currently, to the best of the authors' knowledge, no other monitoring system comparable to the one considered here is currently in use – and thus no other extensive dataset of cutting surface images from punched parts is available for validation purposes. To evaluate the overall performance and transferability of the proposed neural network structure for further applications, we therefore collected a small transfer dataset of 60 images with corresponding mask images. This dataset consists of 40 images of burnish parts from a copper material with varying material thicknesses of 0.5 mm and 0.64 mm as well as images of a steel material with thickness 0.5 mm. All images were collected with the original

Table 4 Comparison of the modified architectures, showing significantly better (△) or worse (▽) results compared to the reference architecture

	8 Features	16 Features	64 Features	Depth 1	Depth 3	3 Layers	4 Layers	6 Layers
Inference time	4.02 ms	4.11 ms	4.43 ms	2.78 ms	5.71 ms	1.75 ms	2.90 ms	6.14 ms
Combined metric	97.87 (▽)	98.69 (=)	99.39 (△)	98.73 (=)	99.19 (△)	97.70 (▽)	98.02 (▽)	99.26(△)
Assignable area [%]	98.63 (▽)	99.84 (△)	99.78 (△)	99.30 (▽)	99.98 (△)	99.59 (=)	100.16 (△)	99.92 (△)
Assignable tear sections [%]	96.24 (▽)	99.12 (=)	99.33 (△)	98.01 (▽)	99.59 (△)	99.01 (=)	99.92 (△)	99.68 (△)
Tear section ratio	2.20 (=)	2.19 (△)	2.20 (=)	2.19 (△)	2.19 (△)	2.20 (=)	2.21 (=)	2.19 (△)
Hole ratio	1.10 (=)	1.03 (=)	1.05 (△)	1.07 (=)	1.06 (=)	1.21 (▽)	1.21 (▽)	1.06 (△)

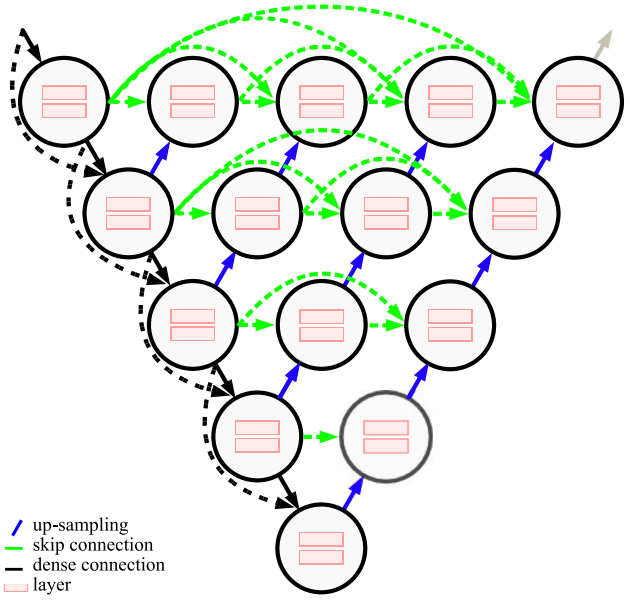


Fig. 9 Architecture of UNet++ with dense backbone

monitoring system. We additionally collected images with an oil film applied to the burnish part, as would be expected in a real production process. Furthermore, images were taken with a Keyence confocal microscope with a different FG:BG ratio and image characteristics and added to the dataset.

We compared the results for UNet++ with 64 feature maps, the reference UNet++, the hyperparameter-optimized variant and 6 layer variant. The results are shown in Table 6. In summary, the best performance is achieved by the *UNet++ with 64 feature maps*. The metric scores are also confirmed by directly observing the images (examples are shown in Fig. 10). All networks tend towards an increased number of predicted tear sections on the transfer dataset, especially for images with an oil film. Considering that the networks were applied to images with formerly unknown characteristics, the performance is generally acceptable, even when a different device acquires the images. It is likely that the results can be improved considerably if images from multiple devices and different punching tools or process parameters are integrated into the training dataset.

5 Conclusion

Fast and accurate segmentation of images is essential for in-cycle processing of quality parameters during the punching process. With prior methods for the segmentation of the burnish surface being too slow for real-time applications, machine learning provides a promising alternative approach.

Since related tasks are well known to be solvable by neural networks in a biomedical environment, we compared the network architectures SegNet, UNet++, MedT and nnU-Net

Table 5 Results for the combination of best-performing hyperparameters (6 Layers, Depth 3, 64 Features) and for a modified backbone with incremental block depth

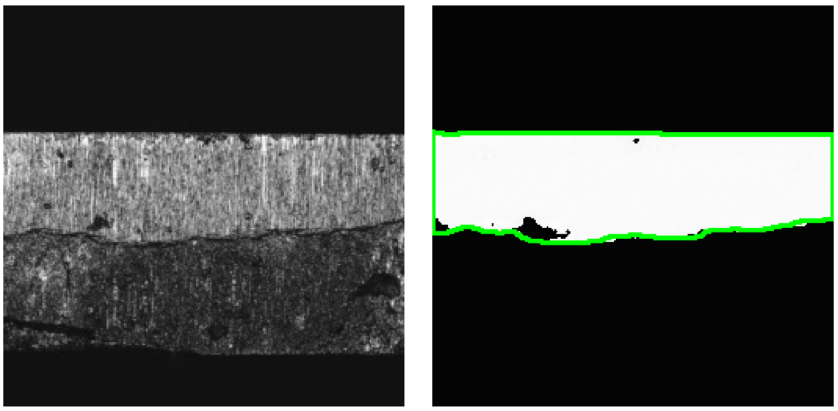
	Hyperparameter combination	Incremental Block Depth	dense-Backbone
Inference time	10.97 ms	9.24 ms	4.95 ms
Combined metric	99.14 (△)	98.62 (=)	98.39 (=)
Assignable area [%]	99.87 (△)	99.16 (▽)	100.98 (△)
Assignable tear sections [%]	99.66 (△)	98.15 (▽)	100.34 (△)
Tear section ratio	2.16 (△)	2.20 (=)	2.20 (=)
Hole ratio	1.07 (=)	1.11 (=)	1.08 (=)

for segmentation of the burnish part. The evaluation is carried out by a newly developed metric, which allows for a simultaneous assessment of the segmentation accuracy in terms of both the boundary and area overlap. The same targets are considered by the loss function that is used for optimizing the networks' parameters. Thereby, it is possible to prioritize characteristics both during training and evaluation. A modular selection of additional metric scores allows for an even more specific assessment of the results; for example, the ratio of tear sections or holes between prediction and ground truth might be considered especially important, depending on the application and the further use of the segmentation.

Moreover, we analyzed the hyperparameters of the UNet++ structure. In our comparison, a UNet++ architecture with 64 feature maps in the first layer achieved the best results, with an inference time of 4.43 ms. In particular, using this segmentation method, it is possible to reliably identify the burnish surface of a produced punching part within the process cycle time. We also tested the developed architecture on a transfer dataset consisting of images with different characteristics from different devices. Although the prediction scores are (expectedly) worse, the proposed modified

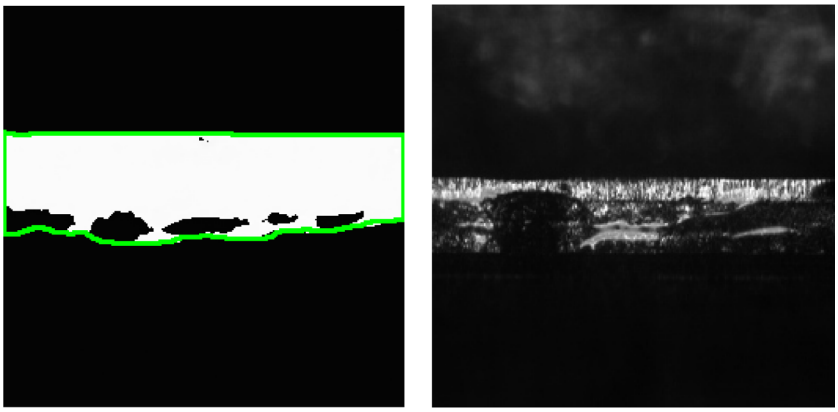
Table 6 Comparison of modified architectures on the transfer dataset

	64 Features	Layer 6	Hyperparameter combination	default UNet++
Inference time	5.23 ms	7.21 ms	10.28 ms	5.67 ms
Combined metric	82.49	81.19	79.24	80.67
Assignable area [%]	106.14	108.58	98.73	106.43
Assignable tear sections [%]	102.06	104.27	93.73	100.66
Tear section ratio	7.75	5.92	5.72	7.28
Hole ratio	4.38	3.28	3.62	5.00



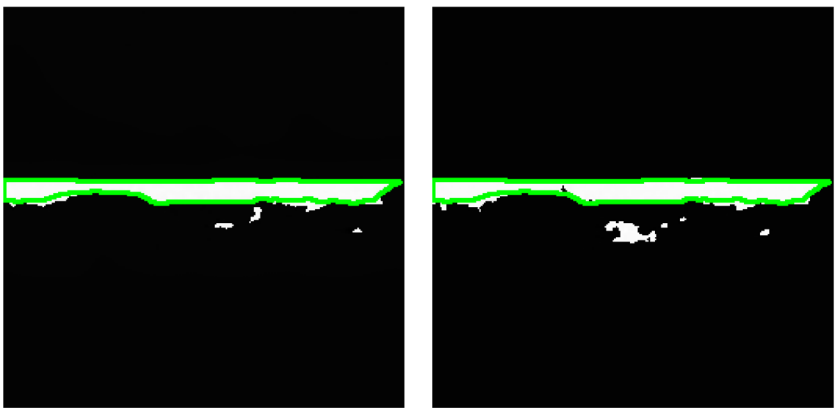
(a) Microscope image

(b) 64 feature maps – result on image a



(c) Hyperparameter combination – result on image a

(d) Oil film



(e) 64 feature maps – result on image d

(f) 6 Layer – result on image d

Fig. 10 Comparison of segmentation results on the transfer dataset, with the ground truth contour in green

UNet++ architecture still performed best. In addition, the results indicate that segmentation does indeed work across different devices and demonstrate that networks for biomedical image segmentation are suitable for manufacturing tasks. In terms of quality monitoring, further research will focus on the performance of the developed architecture and metric with an increased image size; here, we used a resolution of only 256 × 256 pixels to decrease development time. In terms of applications towards predictive maintenance, further research should focus on classifying the image into categories after segmentation – for example, an automated identification of rejects or the distinction between phases of the wear diagram such as running-in, steady state and increasing wear could be considered. Furthermore, the training data should be expanded with additional image data from punching processes with different parameters for thickness and material to avoid an overfitting to specific features.

Author Contributions All authors contributed equally to this work.

Funding Open Access funding enabled and organized by Projekt DEAL. The authors gratefully acknowledge financial support from the interdisciplinary technology network Efficient Production Technology (EffPro), co-funded by the European Union and the Free State of Bavaria from the European Fund for regional development (ERDF) and supported by the University of Applied Sciences Kempten and the Bavarian Ministry of Science and Art.

Availability of data The datasets generated and analysed during the current study are not publicly available due to the fact that they constitute an excerpt of research in progress, but are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Doege E, Behrens BA (2007) Handbuch Umformtechnik: Grundlagen. Technologien, Maschinen, Springer-Verlag, Berlin Heidelberg. https://doi.org/10.1007/978-3-540-48924-5
2. Verein Deutscher Ingenieure (1994) Schnittflächenqualität beim Schneiden, Beschneiden und Lochen von Werkstücken aus Metall Scherschneiden: VDI2906
3. Behrens BA, Krimm R, Nguyen QT, et al (2017) Motorized measurement device for automatic registration of cutting edges. Engineering for a Changing World: Proceedings; 59th IWK, Ilmenau Scientific Colloquium, Technische Universität Ilmenau, September 11–15, 2017 59, 2017(1.3.02)
4. Lorenz M, Menzl M, Donhauser C et al (2022) Optical inline monitoring of the burnish surface in the punching process. Int J Adv Manuf Technol 118:3585–3600. https://doi.org/10.1007/s00170-021-07922-6
5. Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells WM et al (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Springer International Publishing, Cham, pp 234–241, https://doi.org/10.1007/978-3-319-24574-4_28
6. Zhou Z, Siddiquee MMR, Tajbakhsh N, et al (2018) Unet++: A nested u-net architecture for medical image segmentation. Lecture Notes in Computer Science 11045 LNCS:3–11. https://doi.org/10.1007/978-3-030-00889-5_1
7. Isensee F, Jaeger PF, Kohl SAA et al (2021) NNU-Net: a self-configuring method for deep learning-based biomedical image segmentation. Nat Methods 18:203–211. https://doi.org/10.1038/s41592-020-01008-z
8. Damacharla P, V. ARM, Ringenberg J, et al (2021) TLU-net: A deep learning approach for automatic steel surface defect detection. In: 2021 International Conference on Applied Artificial Intelligence (ICAPAI). IEEE, pp 1–6. https://doi.org/10.1109/ICAPAI49758.2021.9462060
9. Lemos A, da Silva L, Nagy B (2020) Automatic monitoring of steel strip positioning error based on semantic segmentation. Int J Adv Manuf Technol 110:2847–2860. https://doi.org/10.1007/s00170-020-05859-w
10. Qian K (2020) Automated detection of steel defects via machine learning based on real-time semantic segmentation. In: Proceedings of the 3rd International Conference on Video and Image Processing. Association for Computing Machinery, New York, NY, USA, ICVIP 2019. p 42-46. https://doi.org/10.1145/3376067.3376113
11. Lin WJ, Chen JW, Jhuang JP et al (2021) Integrating object detection and image segmentation for detecting the tool wear area on stitched image. Sci Rep 11(19):938. https://doi.org/10.1038/s41598-021-97610-y
12. Bergs T, Holst C, Gupta P et al (2020) Digital image processing with deep learning for automated cutting tool wear detection. Procedia Manuf 48:947–958. https://doi.org/10.1016/j.promfg.2020.05.134
13. Scime L, Siddel D, Baird S et al (2020) Layer-wise anomaly detection and classification for powder bed additive manufacturing processes: a machine-agnostic algorithm for real-time pixel-wise semantic segmentation. Addit Manuf 36(101):453. https://doi.org/10.1016/j.addma.2020.101453
14. Gonzalez RC, Woods RE (2018) Digital image processing, 4th edn. Pearson Education
15. Ma J, Chen J, Ng M et al (2021) Loss odyssey in medical image segmentation. Med Image Anal 71(102):035. https://doi.org/10.1016/j.media.2021.102035
16. Drozdzal M, Vorontsov E, Chartrand G, et al (2016) The importance of skip connections in biomedical image segmentation. Lecture Notes in Computer Science 10008 LNCS:179–187. https://doi.org/10.1007/978-3-319-46976-8_19
17. Wu Z, Shen C, van den Hengel A (2016) Bridging category-level and instance-level semantic image segmentation. CoRR abs/1605.06885. Preprint at http://arxiv.org/abs/1605.06885

18. Kervadec H, Bouchtiba J, Desrosiers C et al (2021) Boundary loss for highly unbalanced segmentation. Med Image Anal 67(101):851. https://doi.org/10.1016/j.media.2020.101851
19. Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell 39:2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615
20. Valanarasu JMJ, Oza P, Hacihaliloglu I, et al (2021) Medical transformer: gated axial-attention for medical image segmentation. In: de Bruijne M, Cattin PC, Cotin S, et al (eds) Medical Image Computing and Computer Assisted Intervention – MICCAI 2021. Springer International Publishing, Cham, pp 36–46, https://doi.org/10.1007/978-3-030-87193-2_4
21. Paszke A, Gross S, Massa F, et al (2019) Pytorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, vol 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf
22. Huang G, Liu Z, Maaten LVD et al (2017) Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol 2017-Januar. IEEE, pp 2261–2269. https://doi.org/10.1109/CVPR.2017.243
23. Chen P, Su X, Liu M et al (2020) Lensless computational imaging technology using deep convolutional network. Sensors 20:2661. https://doi.org/10.3390/s20092661
24. Cai S, Tian Y, Lui H et al (2020) Dense-unet: a novel multiphoton in vivo cellular image segmentation model based on a convolutional neural network. Quant Imaging Med Surg 10(6). https://qims.amegroups.com/article/view/43519