applied sciences

MDPI

Article

Camera-Based in-Process Quality Measurement of Hairpin Welding

Julia Hartung [1,2,*], Andreas Jahn [1], Oliver Bocksrocker [3] and Michael Heizmann [2]

[1] TRUMPF Laser GmbH, Aichhalder Str. 39, 78713 Schramberg, Germany; andreas.jahn@trumpf.com
[2] Institute of Industrial Information Technology, Karlsruhe Institute of Technology, Hertzstraße 16, 76187 Karlsruhe, Germany; michael.heizmann@kit.edu
[3] TRUMPF GmbH + Co. KG, Johann-Maus-Str. 2, 71254 Ditzingen, Germany; oliver.bocksrocker@trumpf.com
* Correspondence: julia.hartung@trumpf.com

Abstract: The technology of hairpin welding, which is frequently used in the automotive industry, entails high-quality requirements in the welding process. It can be difficult to trace the defect back to the affected weld if a non-functioning stator is detected during the final inspection. Often, a visual assessment of a cooled weld seam does not provide any information about its strength. However, based on the behavior during welding, especially about spattering, conclusions can be made about the quality of the weld. In addition, spatter on the component can have serious consequences. In this paper, we present in-process monitoring of laser-based hairpin welding. Using an in-process image analyzed by a neural network, we present a spatter detection method that allows conclusions to be drawn about the quality of the weld. In this way, faults caused by spattering can be detected at an early stage and the affected components sorted out. The implementation is based on a small data set and under consideration of a fast process time on hardware with limited computing power. With a network architecture that uses dilated convolutions, we obtain a large receptive field and can therefore consider feature interrelation in the image. As a result, we obtain a pixel-wise classifier, which allows us to infer the spatter areas directly on the production lines.

Keywords: hairpin; laser welding; semantic segmentation; dilated convolution; sdu-net; spatter detection; quality assurance; fast prediction time

check for updates

Citation: Hartung, J.; Jahn, A.; Bocksrocker, O.; Heizmann, M. Camera-Based in-Process Quality Measurement of Hairpin Welding. *Appl. Sci.* **2021**, *11*, 10375. https://doi.org/10.3390/app112110375

Academic Editor: Andreas Fischer

Received: 13 September 2021
Accepted: 19 October 2021
Published: 4 November 2021

1. Introduction

In the production of electric motors, the automotive industry relies on a new technique known as hairpin. Instead of twisted copper coils, single copper pins that are bent like hairpins are used, which give the technology its name. These copper pins are inserted into the sheet metal stacks of a stator and afterward welded together in pairs. As with conventional winding, the result is a coil that generates the necessary magnetic field for the electric motor. This method replaces complex bending operations and enables a more compact motor design while saving copper material [1,2]. Depending on the motor design, between 160 and 220 pairs of hairpins per stator are welded. If at least one weld is defective, the entire component may be rejected. Therefore, continuous quality control is necessary and the weld of every hairpin pair should be monitored [1].

In most cases, a laser is used to weld the pins. The laser welding process enables a very specific and focused energy input, which ensures that the insulation layer is not damaged during the process. In addition, unlike electron beam welding, no vacuum is required and laser welding is a flexible process that can easily be automated in a short cycle time [1]. The lower-cost laser sources that are scalable in the power range emit in the infrared wavelength range, which is comparatively difficult for working with copper. At this wavelength of about 1030 nm or 1070 nm copper, is highly reflective at room temperature, so very little incoming laser light is absorbed [1,3]. Just before reaching the melting temperature the absorption level rises from 5% to 15% and reaches almost 100%

when the so-called keyhole is formed. Based on this dynamic, the process is prone to defects and spattering [4]. A spatter occurs when the keyhole closes briefly and the steam pressure causes the material to leak out of the keyhole. If the ejected material gets into the stator, it may cause short circuits or other defects [1]. In addition, less material will be used to form the weld, which often leads to a loss of stability. For these reasons, it is extremely important to prevent spatter as much as possible. Various processes can improve the welding result on copper. Three approaches are briefly touched upon below. By moving the laser spot fast and simultaneously during forwarding motion (wobbling), stable dynamics can be created in the weld pool. This can improve the process quality when welding with an infrared laser. Another approach is welding with different strengths of inner and outer fiber core. This means that the inner fiber core is used to create the desired welding depth with high intensity, while the molten pool is stabilized by an outer fiber core—the fiber ring. In addition, there is the possibility of using a visible wavelength of a green laser, which results in higher absorption of the laser light and thus higher process reliability [5–7]. Furthermore, there may also be external causes that lead to spattering. These include, for example, contamination, gaps, misalignment or an oxidized surface.

The correct setting of the laser welding parameters such as laser power, speed and focus size is very important in copper welding. In addition, the process may not drift or this must be detected at an early stage. The presence of spatter on the component can be used as an indicator of an unstable situation in the welding process, as its occurrence is closely related to the quality of the weld seam [8,9]. Due to the briefly mentioned reasons, it is essential to monitor the welding process while focusing particularly on spattering. This allows a conclusion about the quality of individual welds, the occurrence of defects, as well as the overall quality of the stator. An important requirement is also fast process time which is a prerequisite for a system to be used in large-scale production. The welding of an entire engine takes just a bit more than one minute and quality monitoring should not slow down the process [1,2].

Currently, there are only a few machine learning applications that are used for quality assessment in laser welding [10]. Some approaches are presented by Mayr et al. [11], including an approach for posterior quality assessment based on images using a convolutional neural network (CNN). They use three images in the front, back, and top view of a hairpin, to detect quality deviations [12]. In [13] a weld seam defect classification with help of a CNN is shown. They achieve a validation accuracy of 95.85% in classifying images of four different weld defects, demonstrating the suitability of CNNs for defect detection. Nevertheless, some defects cannot be seen visually on the cooled weld seam. For example, pores in the weld seam or a weld bead that is too small due to material ejection can not be visually distinguished from a good weld seam.

That is why imaging during the process of hairpin welding offers more far-reaching potential for machine learning than subsequent image-based inspection of the weld. Important criteria are the mechanical strength of the pin and the material loss [12]. Both criteria are in correlation with a stable welding process and the occurrence of spatter. For spatter detection, a downstream visual inspection of the component is also possible [14]. However, this approach is problematic for hairpins since there is little material around the hairpins that can be verified for spattering.

Therefore, this paper presents an approach that enables spatter detection during hairpin welding. One of the main challenges of spatter detection directly during the welding process is the fast execution time on hardware with low computing power. The algorithm should be executed directly in the production line, where the installed hardware is often fanless and only passively cooled due to ingress protection. Another important issue is the amount of training data. Since this is an application in an industrial environment, training should only be done on a small data set so that the labeling effort is low and the algorithm can be quickly adapted to new processes. These two aspects are considered in the following.

In Section 2 the data basis and the analysis methods are presented in detail. On the one hand, the network architecture is discussed, but also comparative algorithms, such as morphological filters with their configurations, are presented. Subsequently, in the result section, the training parameters and the results are shown. Finally, the results are discussed and summarized in Section 4.

2. Materials and Methods

To obtain a comprehensive data basis, images were recorded from two different perspectives while welding hairpins. The captured images were then used to perform different approaches to monitor the occurrence of spatter. An automated solution for spatter detection is recommended because it ensures consistent evaluation and only in this automated way can spatter be detected reliably. The image-based approach contributes to the detection of spatter caused by external factors. By using artificial intelligence, a high feature variance in the data is covered. Additionally, the required properties of the categories do not need to be defined in detail. Images were continuously recorded in the process to implement in-process quality monitoring. Since a visual inspection of the hairpins after the welding process is not always clear, a post-weld inspection is not precise enough.

2.1. Data Basis

We use three different approaches for generating data sets for the observation of the welding process using an industrial camera. These differ firstly in the perspective from which the data were recorded and secondly in the type of preprocessing.

In the first approach, the images taken during the welding process were summed up to a single image per process. The superimposition of the individual images provides information about the spatter occurrence during the entire process [15] and can thus be used to evaluate the welding behavior. When summing up the images, those taken at the time the laser beam pierced the material must be removed. In this process step, a white glow occurs, which takes up so much space on the image, that it would hide possible spatters. An example is shown in Figure 1.

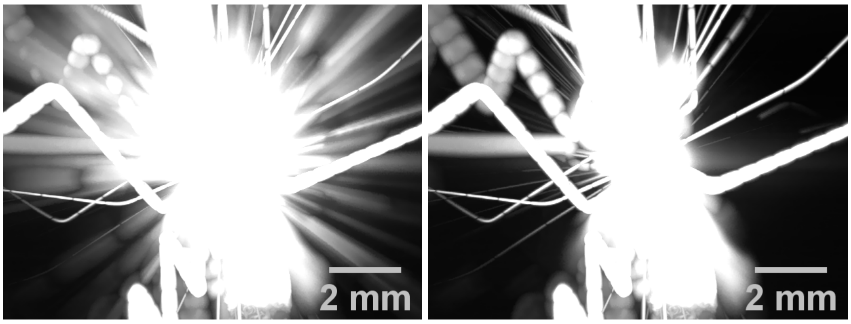


Figure 1. Summed image with and without piercing.

On the one hand, the images were taken with a high-speed camera mounted on the side of the welding process. This provides a lateral view in which the welding process and spatter are visible. The data set containing the summed images of a welding process is referred to as *laterally complete* in the following.

The second view of the images is coaxial through the laser optics. Often, a camera is already installed in a laser welding system that captures images through laser optics. These images are used for example to determine the component position before the welding process. To achieve more accurate results in this process, the magnification optics are often selected to provide a good imaging ratio for the component. In contrast, when monitoring the welding process for spatter, the lowest possible magnification would be optimal. Since the camera is usually part of an existing system, the magnification cannot be adjusted specifically, which often means that a higher magnification must be used. For our test setup, we use a gray-scale camera with a recording frequency of 2 kHz. The summed images, which were generated based on the coaxial view, are called *coaxial complete*.

For a third approach, we evaluate the individual images acquired by the coaxial view. The third data set is called *coaxial single*. While spatters are shown as lines on the summed image, they are usually visible as dots on the single images, depending on the exposure time.

With semantic segmentation, labeling is time-consuming and error-prone. Many developments in the industry are carried out specifically for a customer project on customer data. The labeling effort, which means time resources and therefore costs, is recurring for each customer project. In addition, there are often confidentiality agreements and data flow for a large database is difficult. Therefore, especially in the industry, an attempt should be made to work with a small amount of training data. To teach the network the desired invariance and robustness properties even in training with only a few training data sets, data augmentation is essential [16]. Various network architectures, such as the U-Net architecture, are designed for strong data augmentation. We enlarge our training data set using rotation, vertical and horizontal shift, vertical and horizontal flip, adjustment of the brightness range, zoom and shear. Through the strong use of data augmentation we also have the advantage of avoiding overfitting. Dosovitskiy et al. [17] have shown in this context the importance of data augmentation to learn a certain invariance. Despite a small data set, the network never gets the same image with an identical setting presented multiple times. Thus, it cannot learn the image by memory.

The segmentation masks could not be exact for each pixel. Especially in the summed images where the spatter is shown as a line, the labels are not accurate for each pixel. As shown by Tabernik et al. [18] inaccurate labeling is sufficient for predictions aimed at a quality assurance based on defect detection. The paper processes the topic of surface-defect detection. They used a segmentation-based approach to detect cracks on an image and transferred the problem afterwards to binary image classification with the classes *defect is present* and *defect is not present*. In an experiment, it was shown that the segmentation results are better if larger areas around the crack are marked in the annotation mask. Although pixels that are not part of the crack are marked as defect class, they achieve better results than with an accurate annotation mask. Our use case is very similar to the use case of Tabernik et al., which is why their result can be applied to our task. Instead of surface defects, we detect spatter in our application.

We train separate models, one for each data generation approach. For the two models based on the summed images, we use 14 images each in the training process. Each of these images represents the complete welding process, whereas with the *coaxial single* data set we have about 700 images per welding process. There, we use 500 images from different processes for training. We categorize the pixel-wise class assignment into the background, process lights, and spatter.

2.2. Network Architecture

The images are evaluated using a neural network. We use a segmentation network to localize the process light and the spatter pixel by pixel. Compared to the object detection, which could be done for example with YOLO [19] or SSD [20], this has the advantage that through the pixel-based loss function each pixel can be considered as an individual training instance [18]. This increases the effective number of training data massively and thus also counteracts overfitting during training.

Unlike many other network architectures, the U-Net architecture is very well suited for small data sets, as shown by Ronneberger et al. [16]. Therefore a neural network whose architecture is based on the stacked dilated U-Net (SDU-Net) architecture presented by Wang et al. [21] is used to evaluate our images. The abbreviation SD is short for stacked dilated. To avoid confusion considering the SDU-Net, it should be mentioned that various other U-Net modifications exist, which are also called SDU-Net. However, in these nets the abbreviation SD describes other concepts. For example, there is the spherical deformable

U-Net (SDU-Net) from Zhao et al., which was developed for medical imaging in the inherent spherical space [22]. Because there is no consistent neighborhood definition in the cortical surface data, they developed another type of convolution and pooling operations, especially for this data. Another U-net modification, which is also called SDU-Net, is the structured dropout U-Net presented by Guo et al. [23]. Instead of the traditional dropout for convolutional layers, they propose a structured dropout to regularize the U-Net. Gadosey et al. present the stripping down U-Net, with the same abbreviation, for segmentation images on a platform with low computational budgets. By use of depth-wise separable convolutions, they design a lightweight deep convolutional neural network architecture inspired by the U-Net model [24].

As mentioned before, we used a SDU-Net modification with stack dilated convolutional layers. This U-Net variant adopts the architecture of the vanilla U-Net but uses stacked dilated convolutions. Instead of using two standard convolutions in each encoding and decoding operation, the SDU-Net uses one standard convolution followed by multiple dilated convolutions which are concatenated as input for the next operation. Thus the SDU-Net is deeper than a comparable U-Net architecture and has a larger receptive field [21]. In Figure 2 our architecture is shown in detail. We used a gray-scale image of the size 256 × 256 pixels as input. In designing the network architecture we set the concatenate output channel numbers to $n_1 = 16$, $n_2 = 32$, $n_3 = 64$ and $n_4 = 128$ instead of $n_1 = 64$, $n_2 = 128$, $n_3 = 256$ and $n_4 = 512$ like the original implementation of the paper. Since our images are far less complex compared to the medical images used in the paper from Wang et al., this number is sufficient. So all in all we have 162,474 trainable parameters instead of 6,028,833. With this comparatively small amount, we achieve a fast inference time, which is about 20 ms on CPU. This is important to be able to run the network prediction on an industrial computer directly at the production line, where often no GPU is available.

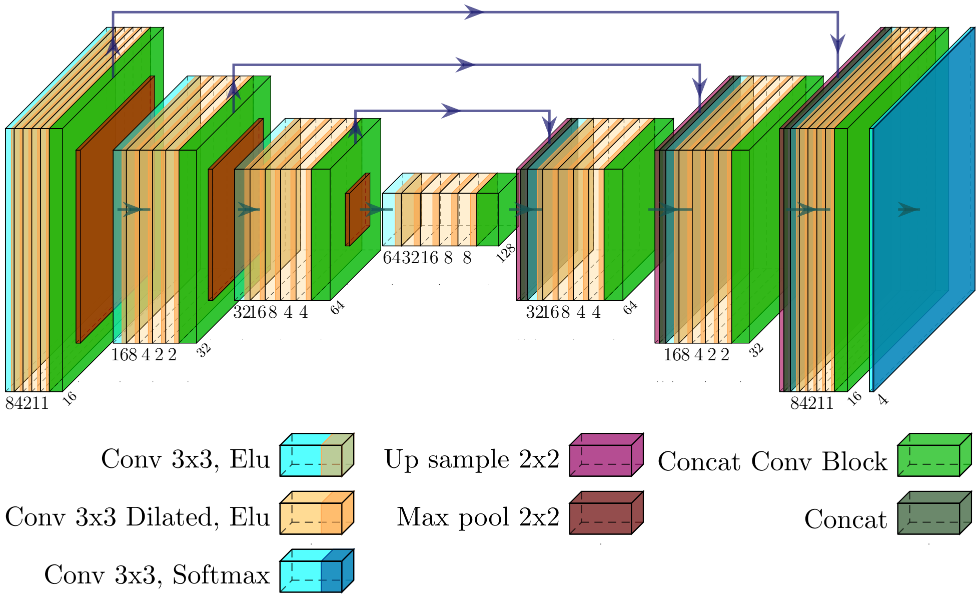


Figure 2. Our optimized small SDU-Net architecture. The input size is 256 × 256 pixels and we have 32 × 32 pixels at the lowest resolution. The number of channels is denoted at the bottom of the boxes.

2.3. Evaluation

An alternative approach is to implement spatter detection with a morphological filter. We choose the opening operation, which involves an erosion of the data set I followed by dilation, both with the same structural element H:

$$I \circ H = (I \ominus H) \oplus H. \tag{1}$$

In the first step, the erosion, the opening process eliminates all foreground structures that are smaller than the defined structural element. Subsequently, the remaining structures are smoothed by the dilation and thus grow back to approximately their original size. With the opening filter, we identify the process light in the images based on the structure element H. In Figure 3 the process is visualized. An input image is shown in Figure 3a and the corresponding filtered image in Figure 3b. Afterwards, we define the spatters by subtracting the filtered image from the original image. The remaining image elements represent the spatters, shown in Figure 3c. Figure 3d shows an overlaid image, where the process light is painted in green and the spatters in red. For this algorithm we use the original image size with 480 × 640 pixels for the coaxial images and 840 × 640 pixels for the lateral images. We choose the structure elements as ellipse with $H = 45 \times 45$ pixels for the *coaxial single*, $H = 90 \times 90$ pixels for *coaxial complete*, and $H = 60 \times 60$ pixels for *lateral complete* images. The definition of the structural element is based on the average size of the process light, which is estimated in the images. The spatters usually represent smaller elements and can thus be distinguished from the elements found by the filter.

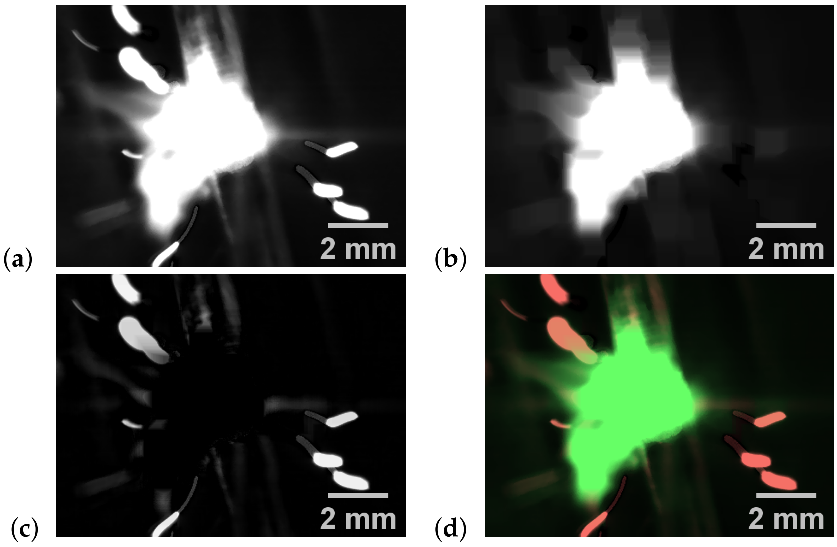


Figure 3. Gray-scale opening, (**a**) gray-scale image (**b**) process light (**c**) spatter (**d**) overlaid image with spatters colored in red and the process light in green.

We have also used other network models, such as a comparable small version of the U-Net model according to Ronneberger et al. [16]. We trained this model equivalent to the SDU-Net architecture with the same input images and the same parameters, shown in the next section.

3. Results

We trained different models of the small SDU-Net architecture for each input data generation approach, *coaxial single*, *coaxial complete*, and *lateral complete*. All models were trained with a batch size of 6, an input of gray-scale images in the size of 256 × 256 pixels, and 500 steps per epoch. We used the Adam Optimizer and started the training process with a learning rate of 0.001. The learning rate was reduced by 5% after 3 epochs without any improvement until a learning rate of 0.000005 is reached. The training process was stopped when no further improvement has occurred in 20 consecutive epochs. This results in different long training times for the different models. The loss value and the accuracy of the different models can be seen in Table 1. To verify the results during training, we used validation data sets. These contained 3 images each for *coaxial complete* and *lateral complete* and 18 images for *coaxial single*, according to the small database. The validation data sets were also enlarged with strong use of data augmentation. After the training, we used a separate test data set, each containing 50 images with the corresponding ground truth image.

Because the number of pixels per class is very unbalanced, and especially the less important background class contains the most pixels, we used the loss functions weighted dice coefficient loss (DL) and the categorical focal loss (FL) [25]. The network results are shown in Figure 4 and in Table 2.

The advantage of focal loss is that no class weights have to be defined. The loss function, which is a dynamically scaled cross entropy (CE) loss, down-weights the contribution of easy examples and focuses on learning hard examples:

$$FL(p_t) = -\alpha(1 - p_t)^{\gamma} log(p_t) \tag{2}$$

The two parameters α and γ have to be defined. The parameter α represents the balancing factor, while γ is the focusing parameter. The CE loss is multiplied by the factor $(1 - p_t)^{\gamma}$. This means that with the value $\gamma = 2$ and a prediction probability of 0.9, the multiplier would be 0.1^2, i.e., 0.01, making the FL in this case 100 times lower than the comparable CE loss. With a prediction probability of 0.968, the multiplier would be 0.001, making the FL already 1000 times lower. This gives less weight to the easier examples and creates a focus on the misclassified data sets. With $\gamma = 0$ the FL works analogously to the cross entropy. Here the values $\alpha = 0.25$ and $\gamma = 2$ were chosen.

For comparison, we used the weighted dice coefficient loss, where the loss value is calculated for each class, weighted with the respective class weighting, and then added up. The class weights were calculated based on the pixel ratio of the respective class in the training images. The classes that contain only a few pixel values, such as the spatter class, must be weighted more heavily so they are considered appropriately during training. Since the values are calculated based on the number of pixels in the training data, these weights vary between the different input data sets.

Besides training individual models for each input data approach, we also trained one model for the prediction of all data, the coaxial and lateral view, summed, and single images. Since the different data sets have the same classes and a similar appearance, one model approach is also possible. The advantage of this approach is that a higher variance of data can be covered in one model and therefore we do not need to define new models or parameters for each data type. To train the global model we used 14 images of each *coaxial* and *lateral complete* data set and 34 *coaxial single* images.

Another advantage is that additional classes can be added to the model. We have introduced a new class, which includes the cooling process of the welding bead. From the moment the process light turns off, the weld is assigned to the cool-down class. This class cannot be identified via the previously described structure recognition with the subsequent exclusion procedure using the morphological filter. Only image elements of different sizes can be detected and distinguished from each other. For elements of similar size with different properties, the method reaches its limits.

The result of training the small SDU-Net as a single model for all data is also shown in Table 1. All four classes were considered in the training process. In the summed images, the cooling process is not visible. For comparison, a SDU-Net model with twice the number of filters was trained. This net has more trainable parameters, but in our test, no significantly better results in loss, accuracy as well as in evaluation could be obtained. In addition, the results of the comparatively small U-Net model are shown.

The classification results are compared using the Intersection over Union metric (*IoU*). The metric compares the similarity between the predicted classification mask and a drawn annotation mask as ground truth by dividing the size of the intersection by the size of the union:

$$IoU(A, B) = \frac{A \cap B}{A \cup B}. \tag{3}$$

Table 1. Training results of the neural network models based on the SDU-Net architecture. Three different models are trained for each input data generation approach. The fourth model was trained for the prediction of all data. In comparison the results of an SDU-Net architecture with twice the number of filters and of the small U-Net are shown.

	Coaxial Single	Coaxial Complete	Lateral Complete	All Data	SDU-Net Double Filter Size, All Data	U-Net, All Data
Steps per epoch	500					
Batch size	6					
Input size [pixels]	256 × 256					
DL	0.173	0.196	0.198	0.156	0.146	0.163
Accuracy	0.993	0.966	0.995	0.980	0.980	0.975
Validation DL	0.488	0.347	0.489	0.361	0.339	0.428
Validation Accuracy	0.976	0.831	0.953	0.963	0.963	0.951

In Table 2 the evaluation results of the different approaches are shown. The second value in the rows shows the IoU for all pixels in the entire image. The dark area around the weld is most correctly classified as background. Especially for the coaxial single dataset, the background takes up the main part of the image, so the IoU over the entire image is very high for all methods. Therefore, the specific class pixels are again considered separately. This consideration can be seen in the first values in the table. This represents the IoU based on the pixels assigned to a specific class, except for the background class. This value gives more information about the actual result than the total IoU. However, the larger the area of the background, the fewer pixels are included in the calculation of the class-specific IoU. As result, the value is more influenced by individual misclassified pixels.

In Figures 4 and 5 the first value, without consideration of the background pixels, is used. As shown in Figure 4 the two weighted loss functions, FL and DL, result in comparable distributions in which the DL performs only marginally better.

Using a single model trained on all three data sets, an outlier with IoU close to 0 can be seen in each of our test sets in Figure 4. There, a shot during the cooling process with spatter was misclassified as process light. When using different models per data set, this error case did not occur. On the one hand, the error can be attributed to an underrepresentation of the cooling class in the overall data set, since this only occurs in the *coaxial single* images. On the other hand, the occurrence of spatter on images at this point in the welding process is very rare, which is why the case was not sufficiently present in the training. In productive use, it is assumed that the data is taken only from one perspective. Nevertheless, this experiment can show that the model generalizes well even on different input data with only very few training data and thus covers a high data variance.

Figure 5 shows the IoU without considering the background pixels of the different data sets all trained with the dice coefficient loss. This graph shows that the largest deviations are contained in the *coaxial single* data set. In these images, the background occupies the largest image area, which makes small deviations of the other classes more significant, as shown in Table 3.

In all three input data sets, *coaxial single*, *coaxial complete*, and *lateral complete*, the SDU-Net provides the best results compared to the other methods. The disadvantages of the U-net architecture arise from the fact that only simple and small receptive fields are used, which leads to a loss of information about the image context. In our use case, it leads to the fact that the classes cannot always be clearly distinguished from each other. The SDU-Net processes feature maps of each resolution using multiple dilated convolutions successively and concatenates all the convolution outputs as input to the next resolution. This increases the receptive field and both, smaller and larger receptive fields are considered in the result.

Table 2. Evaluation results of the different approaches. The first value shows the average IoU value of the pixel which was assigned to a specific class, excluding the background class and the second value shows the average IoU value for the entire image.

	Morph. Binary	Morph. Gray	SDU-Net, Different Models, DL	SDU-Net, Different Models, FL	SDU-Net, One Model, DL	SDU-Net, One Model, FL	U-Net, One Model, DL
Coaxial Single	0.406, 0.962	0.421, 0.958	0.819, 0.980	0.821, 0.990	0.746, 0.990	0.739, 0.989	0.709, 0.965
Coaxial Complete	0.731, 0.847	0.722, 0.568	0.830, 0.909	0.810, 0.937	0.850, 0.953	0.840, 0.952	0.828, 0.848
Lateral Complete	0.497, 0.860	0.482, 0.920	0.644, 0.935	0.611, 0.968	0.681, 0.967	0.688, 0.984	0.659, 0.932
Average	0.544, 0.889	0.542, 0.789	0.764, 0.943	0.747, 0.965	0.759, 0.972	0.756, 0.975	0.732, 0.915

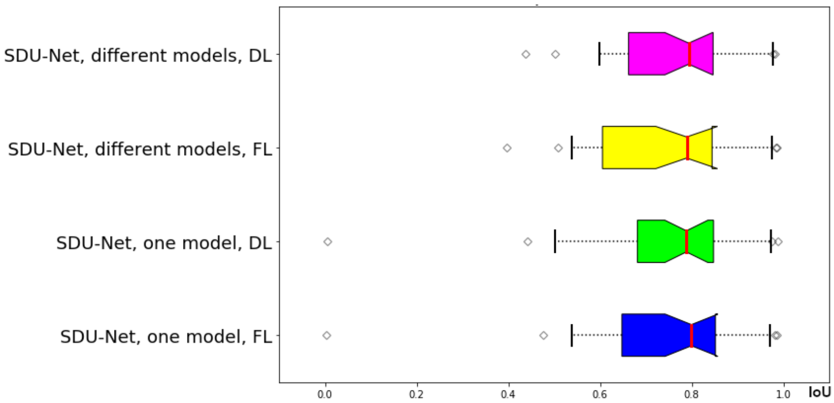


Figure 4. Comparison of the IoU of the classes process light, cool down, and spatter for the different loss functions dice coefficient loss and focal loss, as well as the approach with three different models and one model for all data.

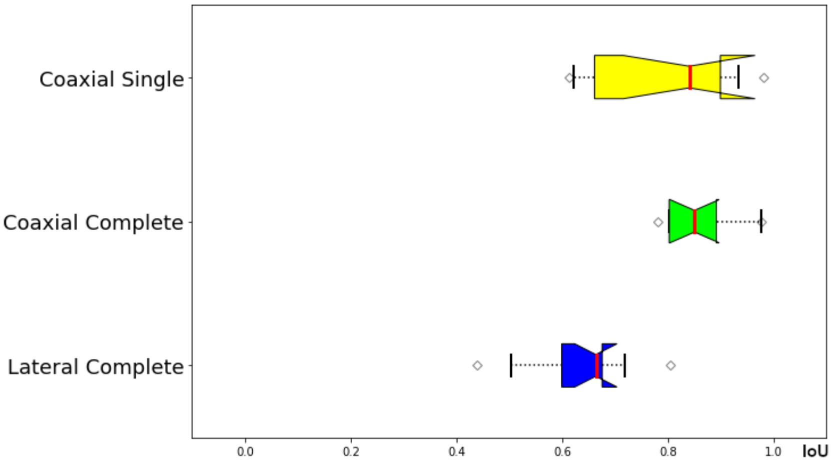


Figure 5. Comparison of the IoU of the different input data sets. All three models are trained with the weighted dice coefficient loss.

Visualized results of the different methods are shown in Table 3. In comparison to the small SDU-Net, results of small U-Net, binary opening, and gray-scale opening are shown. The models of the SDU-Net and the U-Net are trained on all data and with four classes, while the morphological filter on the one hand requires a structural element of different sizes per data set and also cannot distinguish between process light and cooling process. For better visualization, the pixel-by-pixel classification of the neural networks is displayed in different colors and superimposed on the input image. The class of the process light is shown in green color, the spatters in red, and the cooling process in blue. The resulting images of the morphological filter are displayed analogously to Figure 3.

With the morphological filtering, small regions always remain at the edge of the area of the process light, since the structural element can never fill the shape exactly. As a result, the exclusion method would always recognize some pixels as spatter, which must be filtered in post-processing. Even small reflections, which occur mainly in the lateral images, are detected as spatter by the exclusion procedure. On the other hand, spatter that is larger than the defined structural element is detected as process light and not as spatter, which also leads to a wrong result. Compared to the binary opening, the steam generated during welding, which is mainly visible on the lateral images, is usually detected as process light in the gray-scale opening. With the binary opening, the steam area is usually already eliminated during binarization.

Without runtime optimizers, our classification time of the small SDU-Net model on the CPU is about 20 ms. In comparison, the binary and gray-scale opening reached 12 ms for coaxial single, 1.61 ms for coaxial complete, and 40 ms for the lateral complete images. The deviating process times of the opening operation are caused by the different image sizes and the different sizes of the structural element. By using a larger or differently shaped structural element, the process times can be further improved, but the resulting detection quality suffers.

The production time of a stator is about one minute for all 160 to 220 pairs of hairpins. In the best case, 270 ms are needed for welding one hairpin. The quality assurance with the SDU-Net needs 20 ms which is not even 10% of the welding time and 0.33 per mill of the whole welding process. With a time-delayed evaluation of the images, the previous pin can be evaluated each time the next pin is welded. Thus the time sequence of the welding of a stator remains unaffected by this setup due to the fast prediction time. The evaluation was deliberately calculated on the CPU since the model is to be executed directly at the production plant on an industrial PC, where GPUs are not always available. Thus, a strong spatter formation, which can indicate a drifting process or contaminated material, can be reported directly to the user. This allows the user to react directly and stop or adjust the process.

Table 3. Overlaid result images of the different methods. The green color marks the process light, red the spatters, and blue the cooling process. Average IoU value of the pixel which was assigned to process light, spatters, or cooling process.

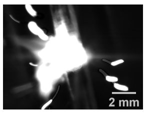
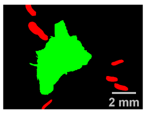
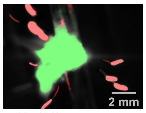
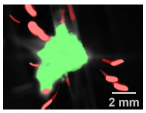
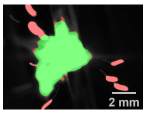
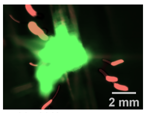
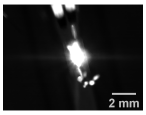
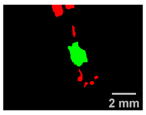
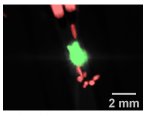
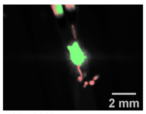
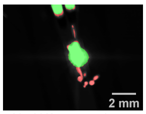
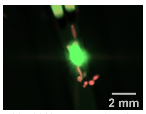
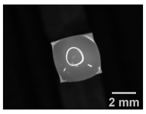
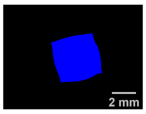
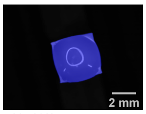
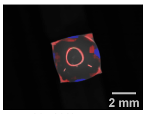
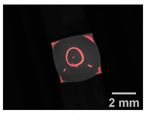
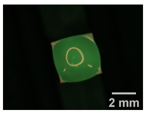
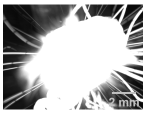
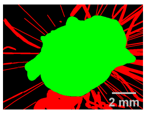
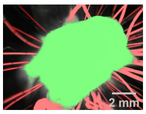
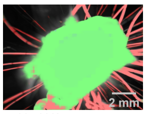
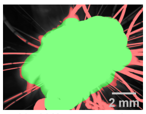
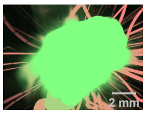
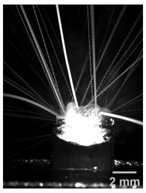
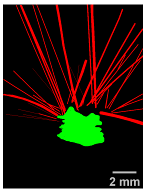
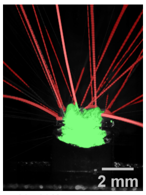
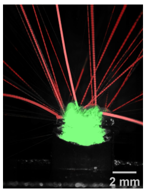
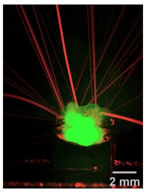
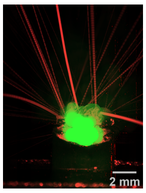
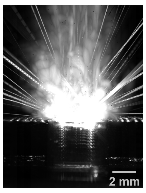
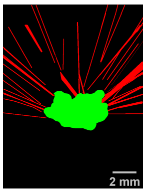
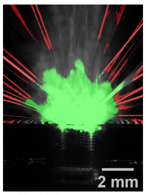
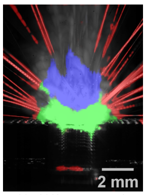
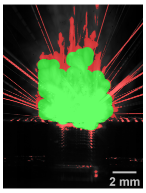
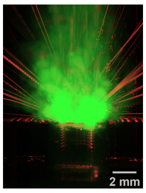
Image	Ground Truth	SDU-Net	U-Net	Binary Opening	Gray-Scale Opening
		Coaxial Single			
		 IoU = 0.88	 IoU = 0.86	 IoU = 0.82	 IoU = 0.82
		 IoU = 0.79	 IoU = 0.522	 IoU = 0.482	 IoU = 0.471
		 IoU = 0.986	 IoU = 0	 IoU = 0	 IoU = 0
		Coaxial Complete			
		 IoU = 0.79	 IoU = 0.79	 IoU = 0.71	 IoU = 0.70

Table 3. *Cont.*

Image	Ground Truth	SDU-Net	U-Net	Binary Opening	Gray-Scale Opening
			Lateral Complete		
		 IoU = 0.68	 IoU = 0.65	 IoU = 0.37	 IoU = 0.37
		 IoU = 0.50	 IoU = 0.34	 IoU = 0.32	 IoU = 0.31

4. Discussion and Outlook

By training one model on all three input data sets, it could be shown that a high variance of data can be covered with this approach. The data varies greatly in terms of both recording position and spatter optics. This high level of data variance will not occur in a production line, where a position of data acquisition, as well as a type of pre-processing, will be determined. However, this experiment suggests that it will be possible to use one model for different applications and with slightly different recording mechanisms. We obtain an average IoU of the specific classes without background class of 0.759 in this approach. In comparison, the IoU values of the morphological filter are 0.544 and 0.542, even though these methods were parameterized specifically for the particular data set. This generalization opens the possibility of using one model for different optics without having to make adjustments. With the execution time of 20 ms we are also in a similar range as the execution times of the morphological filter. This requires 12 ms, 1.61 ms, or 40 ms depending on the input data. For some input data, especially *coaxial complete* with 1.6 ms, this process takes less time, but for other input data it takes even longer.

The spatter prediction works on the summed images as well as on the single images. These show the average IoU value for the specific class with 0.821 for *coaxial single*, compared to 0.810 for *coaxial complete*. To record single images, where every spatter is shown punctually, a high recording frequency is required. Cheaper hardware usually has a lower capture frequency, which means that individual spatters would be missed. To counteract this, the exposure time can be increased so that the spatters are visible as lines on the images, similar to the cumulative images we used. The tests showed that even in this application, the spatter can be detected in the image and thus cheaper hardware can be used for quality monitoring.

By using a segmentation approach and a model architecture which works well with strong data augmentation, it is possible to work with very small training data sets. This makes the labeling effort for new processes, e.g., to new customer data, manageable, and thereby saves time and costs. By using a small network architecture with few parameters, both the training time as well as the prediction time are short. Thanks to the short prediction time, the application can be run directly on the production line on a conventional industrial computer. By analyzing the data during production, it is possible to react interactively, which is more efficient than a completely downstream analysis. The algorithm can also be continuously optimized by feeding new data into the neural network under defined monitoring conditions and then training it further. Further knowledge can be generated through the proper application of data feedback. However, in this application, it is important to ensure that the application is not retrained by a drifting process. In addition, an online learning approach for the laser parameters would also be conceivable. The algorithm can be used to check whether spatter occurs with a certain configuration and thus readjust the laser settings.

The data can be recorded coaxially through the laser optics or laterally to the welding process. Spatter detection works well with both recording methods. In the average IoU of the process light and the spatter class, we achieve 0.850 for the coaxial view, while we only achieve 0.688 for the lateral images. It should be noted that the input images in both cases look very different and the relevant image area has different sizes. In the lateral images, a larger area is covered. In both cases, care must be taken to ensure that the distance to the weld seam is large enough to ensure that the spatter is still within the camera's field of view. The coaxial camera setup is often already available on production lines and can therefore be integrated more easily. In this case, the spatter detection could be upgraded in a production line with the help of a software update.

When considering the entire welding process, welding monitoring with a focus on spatter can be seen as just one part of a desired automated 100% inspection of the welding result. This step could be integrated into a three-stage quality monitoring system: in the first step, a deviating position of the hairpins can be detected in the process preparation and thus the welding position can be corrected. In addition, it makes sense to integrate a check of the presence of both hairpins and their correct parallel position. In the second step, spatter monitoring can be carried out directly in the process. This provides information on whether the welding process is unstable and enables rapid response. In the third step, subsequent quality control of the welding results can be carried out. Due to the in-process monitoring, random samples are sufficient in this step.

However, if 100% monitoring for spatter occurrence is to be implemented, additional hardware is required. As mentioned before, an industrial camera installed at production lines usually does not have such a high frame rate that images can be recorded without short times were spatters can be missed. This can be counteracted with the help of an extended exposure time and a larger field of view in which the spatter can be detected, but a 100% view during the process is unrealistic. In this case, an event-based camera or other sensor technology would have to be used. The approach presented in this paper focuses on quick and easy integration into an existing production system without the need for investment in additional hardware. This is often very costly and can lead to additional calibration effort.

Further on, the presented approach can be extended by an additional consideration of laser parameters or other sensor technology, which is already installed in the system. With the help of the information fusion in which the camera-based in-process monitoring for spatter is integrated, it is also possible to control the process even more comprehensively with existing hardware.

Author Contributions: Conceptualization, O.B.; methodology, J.H., A.J. and M.H.; software, J.H.; validation, J.H., A.J. and M.H.; formal analysis, J.H.; data curation, O.B., J.H. and A.J.; writing—original draft preparation, J.H.; writing—review and editing, all authors. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sievi, P.; Käfer, S. Hairpins für E-Mobility Schweißen (German). 2021. Available online: https://www.maschinenmarkt.vogel.de/hairpins-fuer-e-mobility-schweissen-a-1003621/ (accessed on 28 June 2021).
2. Kaliudis, A. It's Heading This Way. 2018. Available online: https://www.trumpf.com/en_INT/presse/online-magazine/its-heading-this-way/ (accessed on 28 June 2021).
3. Kaliudis, A. Green Light for Welding Copper. 2017. Available online: https://www.trumpf.com/en_INT/presse/online-magazine/green-light-for-welding-copper/ (accessed on 28 June 2021).
4. Schmidt, P.A.; Zaeh, M.F. Laser beam welding of electrical contacts of lithium-ion batteries for electric- and hybrid-electric vehicles. *Prod. Eng.* **2015**, *9*, 593–599. [CrossRef]
5. Welding Copper—One Application, a Few Challenges. 2021. Available online: https://www.trumpf.com/en_INT/solutions/applications/laser-welding/welding-copper/ (accessed on 28 June 2021).
6. Dold, E.M.; Willmes, A.; Kaiser, E.; Pricking, S.; Killi, A.; Zaske, S. Qualitativ hochwertige Kupferschweißungen durch grüne Hochleistungsdauerstrichlaser (German). *Metall* **2018**, *72*, 457–459.
7. Franco, D.; Oliveira, J.; Santos, T.G.; Miranda, R. Analysis of copper sheets welded by fiber laser with beam oscillation. *Opt. Laser Technol.* **2021**, *133*, 106563. [CrossRef]
8. Kaplan, A.F.; Powell, J. Laser welding: The spatter map. In Proceedings of the 29th International Congress on Applications of Lasers and Electro-Optics, ICALEO 2010—Congress Proceedings, Anaheim, CA, USA, 26–30 September 2010; Volume 103. [CrossRef]
9. Zhang, M.; Chen, G.; Zhou, Y.; Li, S.; Deng, H. Observation of spatter formation mechanisms in high-power fiber laser welding of thick plate. *Appl. Surf. Sci.* **2013**, *280*, 868–875. [CrossRef]
10. Weigelt, M.; Mayr, A.; Seefried, J.; Heisler, P.; Franke, J. Conceptual design of an intelligent ultrasonic crimping process using machine learning algorithms. *Procedia Manuf.* **2018**, *17*, 78–85. [CrossRef]
11. Mayr, A.; Weigelt, M.; Lindenfels, J.V.; Seefried, J.; Ziegler, M.; Mahr, A.; Urban, N.; Kuhl, A.; Huttel, F.; Franke, J. Electric Motor Production 4.0—Application Potentials of Industry 4.0 Technologies in the Manufacturing of Electric Motors. In Proceedings of the 2018 8th International Electric Drives Production Conference, EDPC 2018—Proceedings, Schweinfurt, Germany, 4–5 December 2018. [CrossRef]
12. Mayr, A.; Lutz, B.; Weigelt, M.; Glabel, T.; Kibkalt, D.; Masuch, M.; Riedel, A.; Franke, J. Evaluation of Machine Learning for Quality Monitoring of Laser Welding Using the Example of the Contacting of Hairpin Windings. In Proceedings of the 2018 8th International Electric Drives Production Conference, EDPC 2018—Proceedings, Schweinfurt, Germany, 4–5 December 2018. [CrossRef]
13. Khumaidi, A.; Yuniarno, E.M.; Purnomo, M.H. Welding defect classification based on convolution neural network (CNN) and Gaussian Kernel. In Proceedings of the 2017 International Seminar on Intelligent Technology and Its Application: Strengthening the Link Between University Research and Industry to Support ASEAN Energy Sector, ISITIA 2017—Proceeding, Surabaya, Indonesia, 28–29 August 2017; Volume 2017-January. [CrossRef]
14. Hartung, J.; Jahn, A.; Stambke, M.; Wehner, O.; Thieringer, R.; Heizmann, M. Camera-based spatter detection in lasser welding with a deep learning approach. In *Forum Bildverarbeitung 2020*; KIT Scientific Publishing: Karlsruhe, Germany, 2020. [CrossRef]
15. Leitz, A. Laserstrahlschweißen von Kupferund Aluminiumwerkstoffen in Mischverbindung (German). Ph.D. Thesis, Universität Stuttgart, Stuttgart, Germany, 2015.
16. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Cham, Switzerland, 2015; Volume 9351. [CrossRef]
17. Dosovitskiy, A.; Fischer, P.; Springenberg, J.T.; Riedmiller, M.; Brox, T. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 1734–1747. [CrossRef]
18. Tabernik, D.; Šela, S.; Skvarč, J.; Skočaj, D. Segmentation-based deep-learning approach for surface-defect detection. *J. Intell. Manuf.* **2020**, *31*, 759–776. [CrossRef]
19. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; Volume 2016-December. [CrossRef]
20. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Volumee 9905 LNCS. [CrossRef]
21. Wang, S.; Hu, S.Y.; Cheah, E.; Wang, X.; Wang, J.; Chen, L.; Baikpour, M.; Ozturk, A.; Li, Q.; Chou, S.H.; et al. U-Net using stacked dilated convolutions for medical image segmentation. *arXiv* **2020**, arXiv:2004.03466.
22. Zhao, F.; Wu, Z.; Wang, L.; Lin, W.; Gilmore, J.H.; Xia, S.; Shen, D.; Li, G. Spherical Deformable U-Net: Application to Cortical Surface Parcellation and Development Prediction. *IEEE Trans. Med. Imaging* **2021**, *40*, 1217–1228. [CrossRef] [PubMed]
23. Guo, C.; Szemenyei, M.; Pei, Y.; Yi, Y.; Zhou, W. SD-Unet: A Structured Dropout U-Net for Retinal Vessel Segmentation. In Proceedings of the 2019 IEEE 19th International Conference on Bioinformatics and Bioengineering, BIBE 2019, Athens, Greece, 28–30 October 2019. [CrossRef]

24. Gadosey, P.K.; Li, Y.; Agyekum, E.A.; Zhang, T.; Liu, Z.; Yamak, P.T.; Essaf, F. SD-UNET: Stripping down U-net for segmentation of biomedical images on platforms with low computational budgets. *Diagnostics* **2020**, *10*, 110. [CrossRef] [PubMed]
25. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2999–3007. [CrossRef] [PubMed]