IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received February 18, 2021, accepted March 13, 2021, date of publication March 19, 2021, date of current version April 15, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3067302

Improving Build Quality in Laser Powder Bed Fusion Using High Dynamic Range Imaging and Model-Based Reinforcement Learning

CHRISTIAN KNAAK, (Member, IEEE), LUKAS MASSELING, EMIL DUONG, PETER ABELS, AND ARNOLD GILLNER
Fraunhofer-Institute for Laser Technology ILT, 52074 Aachen, Germany

Corresponding author: Christian Knaak (christian.knaak@ilt.fraunhofer.de)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825030. https://ec.europa.eu/research/participants/docs/h2020-funding-guide/grants/grant-management/acknowledge-funding_en.htm.

ABSTRACT In laser-based additive manufacturing (AM) of metal parts from powder bed, information about actual part quality obtained during build is essential for cost-efficient production and high product quality. Reliable and effective monitoring strategies for laser powder bed fusion (LPBF) therefore remain in high demand and are the subject of current research. To address this demand, a novel analysis approach using high dynamic range (HDR) optical imaging in combination with convolutional neural networks (CNN) is proposed for spatially resolved and layer-wise prediction of the surface roughness of LPBF parts. In a further step, the predicted surface roughness maps are used as a feedback signal for a reinforcement learning technique that employs a dynamics model to subsequently identify optimal process parameters under varying and uncertain conditions. The proposed approach ultimately combines the estimation of the local surface roughness based on image texture and model-based reinforcement learning to an in-situ optimization framework for LPBF processes. In addition, the relationship between the layer surface roughness of the part and the overall part density is discussed on the basis of experimental data, which also indicate the applicability of the proposed method in industrial environments. This preliminary study is a first step towards highly adaptive and intelligent machines in the field of automated laser powder bed fusion with the primary goals of reducing production costs and improving the environmental fingerprint as well as print quality.

INDEX TERMS Laser powder bed fusion, convolutional neural networks, model-based reinforcement learning, high dynamic range imaging, surface roughness optimization.

I. INTRODUCTION

A. PROCESS PARAMETER AND SURFACE ROUGHNESS

As a sub-branch based of AM, the laser powder bed fusion (LPBF) technology is frequently used in machine tool and automotive industries [1], in aerospace engineering [2] as well as for medical devices [3]. LPBF is considered one of the key technologies that enables the fabrication of increasingly complex parts and systems with high demands on mechanical properties (e.g., yield strength, ductility, or heat resistance) [4].

However, the lack of process reproducibility and the resulting quality differences between work pieces hinder the transition of the technology to mass production. Hence, a reliable and cost-effective approach for in-situ quality monitoring and process optimization is highly demanded [5]–[7].

A significant quality parameter in LPBF is the increased roughness of the as-built surfaces, which potentially leads to reduced fatigue life of the final part due to the concentration of residual stresses on the surfaces [8]. Additionally, high surface roughness generally leads to poor surface quality and therefore requires long and expensive post-finishing operations. The final part surface is often specified to be in range of the roughness defined by the current application which can require a surface roughness of 0.8 μm or better to prevent mechanical failure of the part due to cracks initiating on its surface [9].

In comparison to the overall part surface roughness, that is particularly difficult to measure during the build process, the local top surface roughness can be estimated for each layer

The associate editor coordinating the review of this manuscript and approving it for publication was Wai-Keung Fung.

after its processing. Further, the layer-wise surface roughness is a key feature to evaluate the results of laser-material interaction during a build process and provides a useful indication of the part quality [10].

The roughness of LPBF surfaces mainly results from the layer-wise build process using overlapping laser tracks, the applied process parameters, and incomplete melted material [8].

The following parameters have been investigated in literature for different materials and indicate a significant correlation to the surface roughness of LPBF parts:

1) Laser power −− [8], [10]
2) Scan velocity + [10]
3) Build orientation ++ [10]–[14]
4) Layer thickness ++ [11], [13]
5) Hatch distance + [15]
6) Scan strategy + [14], [16]–[18]

From the physical point of view, as the laser power increases, the size of the melt pool also increases [19]. During the layer-wise build process, larger melt pools increase the intersection area between different tracks and therefore lead to a smoother surface. However, if the laser power exceeds a certain limit, the increased energy intensity may result in the formation of a high fluctuant keyhole that can introduce additional defects such as subsurface pores or spatters [8]. In addition to increasing surface roughness, the appearance of defects such as pores and lack of fusion can critically reduce the final density of the part [20]–[22].

Moreover, the applied scanning strategy has a significant effect on the microstructure of the component, the mechanical properties, and on the inter-layer surface roughness [14], [16], [17]. For example, DePond *et al.* [18] reported that the lack of rotation of scan vectors between layers leads to an increase in surface roughness. In addition, Snyder and Thole [16] suggest that roughness is strongly correlated with melt pool volume and shape. The authors have also reported that scan strategies leading to layer-wise similar directional scan vectors correlate with large-scale roughness features. Therefore, it is suggested that knowledge of the influence of scan strategies should be incorporated to effectively control surface roughness.

Furthermore, surface roughness, microhardness, and part density can be improved by using laser remelting strategies [17], [23], [24]. Although remelting increases processing time because the same area is scanned twice, Yu *et al.* [24] show that roughness can be effectively reduced by about 50 % for AlSi10Mg parts. Remelting strategies do not necessarily have the same process parameters as the main process and therefore bring into play additional variables with respect to process optimization. However, to limit the scope of this study, scan strategy variations and remelting methods are not considered as optimization variables.

Recent studies have also revealed that thermal load induced by the layer-wise processing significantly impact the quality of the final part [25]. The literature suggests that the occurring thermal gradients can generate residual stresses in the component and cause deformations, which increases the susceptibility to crack formation or layer delamination [26], [27]. Thus, process parameter optimization with respect to low surface roughness and defect-free parts, is limited by physical process boundaries leading to gas pores and high thermal gradients at increased volumetric energy density (VED). At low VED balling may occur which eventually results in high surface roughness and lack of fusion defects [22], [26].

Therefore, a trade-off between increased defect probabilities due to physical process limits and further aspects of the part quality (e.g., surface roughness, part density) must be found by the proposed optimization framework.

B. SURFACE ROUGHNESS MEASUREMENT

In order to find optimal process parameters with respect to high part quality, a measurement technology is required that allows to capture key aspects of the part's properties, which in this work is the surface roughness and the percentage of defective surface areas. Conventional roughness measurement methods such as contact-type surface roughness profile measurements and advanced optical methods (e.g., chromatic confocal microscopy, white light interferometry) provide quantitative information in the vertical direction (i.e., height). In contrast to this, 2D imaging techniques only provide information about the texture and in some cases about spectral reflection properties (e.g., color) of the surface.

However, if the 2D imaging system is combined with further image processing that subsequently maps image features to roughness parameters (e.g., *Ra*, *Sa*, *Rz*), these instruments can be used for quantitative roughness measurement as well [28]–[31].

As opposed to the other measurement technologies, a camera-based roughness measurement system that uses surface texture information enables layer-wise and fast measurement times at low system costs.

For AM parts the most often used areal roughness parameter according to [32] is *Sa*, which represents the average of absolute height deviations within a defined area. The areal *Sa* parameter corresponds to the roughness profile *Ra* already used in many industrial applications.

In this work, the roughness is approximated as [12]:

$$Sa \approx \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\left|f_{ij}\right| \tag{1}$$

where NxM is the number of measured height deviations and f_{ij} states the deviation from the average surface height. In recent works, different image processing methods were used to establish the relation between image texture and surface roughness [33]. Based on a defined reference surface, the texture of a surface can be interpreted as the structured or random deviation from the reference that defines the topography of the surface. The texture of technical surfaces is the result of nano- and micro roughness, macro roughness (i.e., waviness), defects and lay [34].

The models used to predict the surface roughness can be divided into statistical models (regression, classification) based on machine learning (ML) and analytical models [11] that rely on physical relationships. Analytical approaches play an important role when estimating surface roughness based on process parameters. However, in case the model input is represented by an image of the surface, advanced statistical techniques and ML are favorable methods to obtain high prediction accuracies.

For example, Kamguem *et al.* [35] demonstrate that different roughness characteristics (i.e., *Ra*, *Rq*, *Rv*, *Rt* and *Rz*) show a strong correlation to certain surface image features. Therefore, a ML model was successfully developed that requires no information about the process parameters used. Instead, several image characteristics based on the surface texture of turned surfaces have been identified as important: aggregated image gradients, average grey level and average texture frequency. The correlation between the image features and the associated surface roughness was found to be high (93-98%).

In the field of AM, Akhil *et al.* [29] investigated the surface characteristics of Ti-Al-4V LPBF parts based on the image texture. Statistical image features based on first and second order statistics were used to describe relevant image properties. The most relevant features were selected by applying neighborhood component analysis (NCA) and subsequently used as input for several ML models to predict the surface roughness. The authors reported that the Gaussian process regression (GPR) model is able to provide an accurate roughness estimation with R^2 greater than 0.9.

Recently, methods that incorporate the process of feature extraction from images into the ML pipeline, have been developed to predict surface roughness characteristics in industrial [36] and infrastructure engineering [37]. These models are based on recent developments in Deep Learning that led to a new class of neural networks known as convolutional neural networks [38], which allow to analyze high dimensional sensor and further process-related data more efficiently.

To quantitatively estimate the concrete surface roughness from high-resolution images in infrastructure engineering, an approach based on convolutional neural networks was recently developed and implemented [37].

For this purpose, a CNN architecture called ResNet50, originally proposed by Microsoft in 2015 was utilized [38]. ResNet50 is a 50-layer CNN that can be used to automatically extract and classify complex visual features from image data. The authors used a model pre-trained on the ImageNet dataset. Finally, the classification section of the network was redesigned and trained to categorize three different concrete roughness classes. Using the pre-trained ResNet50 model, the surface roughness could be predicted with more than 93% accuracy for new images.

A similar approach has recently been used for image-based LPBF material identification and defect detection by Narayanan *et al.* [39]. The authors compared the performance of established CNN architectures such as AlexNet and ResNet using transfer learning with that of an approach that uses principal component analysis and SVM, as well as with their proposed CNN. It was shown that established architectures can be adopted to the new domain, however, coming with higher computational costs compared to simpler architectures.

Although the recent developments show promising results, further investigations in case of surfaces manufactured via LPBF are required, since the detection of local roughness deviations on highly reflective metal surfaces remains challenging. In addition to our knowledge none of the approaches found in relevant literature provide image texture-based roughness prediction with spatial resolution (i.e., a roughness map), which is a requirement to identify local roughness trends and deviations.

C. EXISTING WORK ON PROCESS OPTIMIZATION AND LIMITATIONS

Process control for LPBF typically requires an in-situ measurement signal as feedback to apply a control action to ultimately adjust the controlled variable as desired. An essential part of the control system is the control algorithm. The proportional-integral-derivative (PID) controller is among the most frequently used techniques used for industrial control systems according to the literature [40]. PID controllers have already been used and investigated in various applications in the field material processing to control parameters such as cladding height, bead geometry or temperature [41]–[43]. More sophisticated techniques for process control such as model predictive control (MPC) [43]–[45], Linear–Quadratic Regulators (LQR) [46] and Iterative Learning Control (ILC) [47], have recently been suggested in the field of industrial manufacturing. However, the proposed control strategies have often been implemented only in the form of single-input single-output (SISO) systems, which do not reflect the full potential of process optimization.

Cao *et al.* [48] proposed a multi-input-multi-output (MIMO) control concept for the laser metal deposition processes. In this approach, the current layer height and the average melt pool temperature serve as input variables. The laser power and the processing speed are used as manipulated variables. As part of the model predictive control, a nonlinear process model was utilized to adjust the deposition height and melt pool temperature with respect to a target value. The performance of the system was validated in a single layer case study. It was shown that further improvements in terms of robustness as well as generalizability are required for deployment in real industrial environments.

In summary, many of the proposed control techniques require extensive experimentation to determine the parameters of the controller, which are then often only valid as long as the model assumptions made do not change. In this context, solutions are needed that enable robust and cost-effective process monitoring and optimization, while complying with high industrial standards. Current solutions are only suitable in a limited number of applications due to the frequent need for expert intervention and time-consuming fine-tuning and should therefore be further improved in terms of usability, autonomy, and flexibility.

With recent advances in Deep Learning and ML in general, data-driven techniques are increasingly being used for various tasks in the area of 3D printing. In various sub-domains such as part design, quality control, process optimization, cloud platforms services, as well as cyber-attack and weapon detection, ML has already shown its potential in a variety of applications [49]–[51]. In the specific field of process optimization and in-situ quality control, artificial neural networks can be considered as the most widely used ML technique [50]. However, Goh *et al.* [52] showed that several other methods such as genetic algorithm, k-means, random forest, reinforcement learning, support vector regression and ensemble methods have also been used to address problems related to process optimization and quality control. In addition to these supervised learning techniques, a self-organizing feature map in the sense of unsupervised learning has recently been used to discover process-structure-properties (PSP) relationships in large and high-dimensional AM process datasets [53]. Overall, ML algorithms seem to be generally superior to conventional optimization methods such as polynomial regression, Taguchi or analysis of variance (ANOVA) due to their ability to establish nonlinear relationships between input and output variables [52].

Given this background and recent advances in ML, the development and implementation of self-learning, adaptive, and intelligently behaving controllers appears to be possible. Although not commonly used in the field of AM so far, reinforcement learning (RL) seems to be an attractive method as it utilizes interactions and rewards to reinforce the use of process parameters which lead to high long-term rewards for a specific process. RL can provide solutions to complex optimization tasks while remaining flexible by constantly interacting and exploring the given process [54].

In the domain of industrial machining, a meta-reinforcement learning approach for learning turning machining parameters with respect to energy efficient process control has been developed by Xiao *et al.* [55].

The authors used an approach based on Meta-Reinforcement Learning (MRL) to determine optimal machining parameters in material processing. The approach allows to identify the similarities of different optimization models which enables a fast adaptation in case of changing task conditions. The continuous parametric optimization task has been addressed by using the Actor-Critic (AC) framework. To improve the generalization abilities of the optimizer, meta-policy training was applied. It was shown that most of the algorithms used for comparison have been outperformed by the proposed approach in terms of optimization performance.

In the domain of laser welding, the authors of [56] used RL to control the laser power based on optical and acoustic measurement signals. Two different RL approaches (i.e., Q-learning and policy gradient algorithms) were investigated. Laboratory experiments have shown that both methods can successfully learn a control strategy for achieving the required weld penetration depth in a reasonable amount of time.

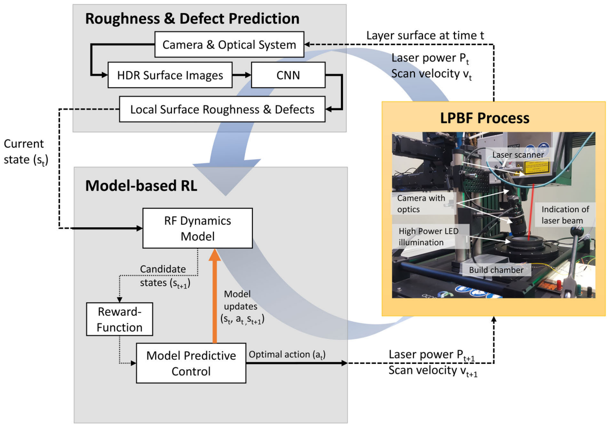


FIGURE 1. Proposed framework for layer-wise monitoring and optimization of LPBF processes.

More recently, a RL approach was used as a feedback control scheme in robotic wire arc additive manufacturing [57]. The authors used RL to approximate the nonlinear effect of process parameters on multilayer multi-bead prints and then correct for geometric deviations that occur based on thermal photodiode signals and 2D-profile measurements. Experimental results showed that the proposed learning framework can be used to reduce thermally induced deformations of the parts.

Although it can be seen that these methods are useful in metal-based AM, more research is required to assure process stability, short training time for fast adaption with respect to multiple goals.

Thus, a new approach for layer-wise optimization of LPBF processes with respect to a defined target (reward) function is proposed. Based on HDR images at high spatial resolution, a CNN-model uses image patches derived from the original images to estimate a roughness map for each surface image. Subsequently, the performance of the proposed CNN-model is compared to state of the art CNNs and to a classical ML approach using statistical features as well as texture descriptors in combination with support vector machines (SVM) for roughness estimation. Furthermore, a model-based reinforcement learning (MBRL) approach with high sample efficiency is used to find an optimal control policy faster, compared to model-free RL methods. The approach allows to find and update an optimal policy, which eventually enables the selection of an optimal action (i.e., a set of process parameters) for the next processing layer, depending on the current process state. Additionally, the MBRL method is compared against a model-free Q-learning RL algorithm as baseline. Finally, the evaluation of the overall framework as shown in FIGURE 1, based on CNNs for spatial resolved surface

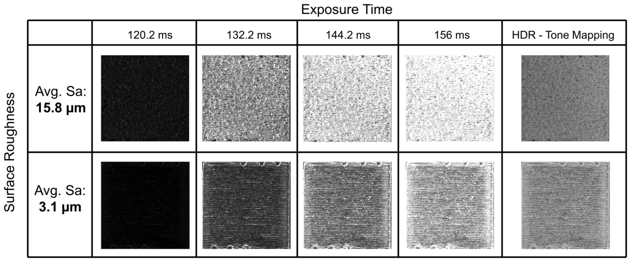


FIGURE 2. **LDR and tone-mapped HDR images of two different LPBF top surfaces.**

roughness estimation and MBRL, is presented in section IV. The individual modules and the applied methodology will be explained more detailed in section II. In section III, a description of the experimental setup and the applied data preprocessing can be found.

II. METHODOLOGY

A. IMAGE-BASED SURFACE ROUGHNESS ESTIMATION

Initially, the roughness estimation module described in FIGURE 1 is engaged in the acquisition of HDR image data of the surface of the current layer. The HDR technique allows to increase the range of luminosity that can be represented within a single image. This is particularly important for scenes that contain high brightness gradients that cannot be adequately represented by the camera's standard dynamic range, which is 71.23 dB for the camera used in this work. Due to the high reflectivity of the investigated metallic surfaces and its strong dependency on the direction of light incidence, some surface areas only lead to a low intensity in the acquired image. Other areas, however, appear highly reflective and show saturation of the affected pixel values. For the metallic as-build surfaces investigated in this work, examples in the form of low dynamic range (LDR) images acquired at different exposure times are shown in FIGURE 2

The HDR images of the specimen's top surfaces are generated by capturing the same scene using different exposure times. Thus, four photos are taken for each HDR image at exposure times ranging from 120.2 ms to 156.2 ms with a step size of 12 ms. It is worth mentioning that all surface pictures are taken and processed as grayscale images. Subsequently, the algorithm of Debevec *et al.* is used to merge the multiple exposure images into a single HDR image [58, 59]. As a result, the HDR image is represented as pixels of type float32 with values rescaled between zero and one. These scaled HDR images are then used as input for the CNN to estimate the local surface roughness.

However, to show the HDR images on conventional displays so that most details are preserved, a linear tone-mapper with bilateral filtering is used [59]. FIGURE 2 also shows that the overall brightness level appears to depend on the average surface roughness, since the relative camera position and the relative position of the illumination to the samples do not vary for all images studied in this work.

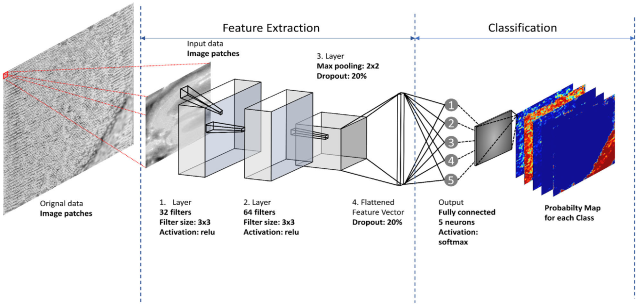


FIGURE 3. **Architecture of the proposed CNN model used to predict surface roughness and defective areas.**

After image capturing, the images are divided in small image patches and subsequentially used to train a convolutional neural network that assesses the surface roughness based on individual image patches. Although the origin of CNNs lays back in the 1980s, huge attention was recently given where high performance GPU implementation enabled to train complex CNNs with a high number of parameters that outperformed many other methods in the most important image recognition contests [38], [60]. CNNs can not only be used for image data, but they bring certain advantages to these applications, such as translation invariance through weight sharing, and local connectivity that takes the spatial structure of images into account [61].

The architecture used for this work is depicted in FIGURE 3 and consists of three types of layers, which are connected consecutively to create a deep neural network model: Convolutional layer, fully connected layer and pooling layer. In the convolutional layer, small filter kernels convolve over the input array to produce class specific filter responses as layer output. The coefficients of each filter kernel defined in a given convolution layer are determined during the training process by backpropagating the actual error between the network prediction and the given training data. The output of a convolutional layer can be denoted as follows [62]:

$$\boldsymbol{X}_d^l = \boldsymbol{f}(\textstyle\sum_{i \in M_d} \boldsymbol{X}_i^{l-1} \times \boldsymbol{K}_{id}^l + \boldsymbol{b}_d^l). \tag{2}$$

where $\boldsymbol{X}_d^l$ is the dth output feature map (image) of the lth convolutional layer. On the right side, the ith output feature map $\boldsymbol{X}_i^{l-1}$ of the previous layer $l-1$ is convolved with the idth kernel $\boldsymbol{K}$ of the current layer. $\boldsymbol{b}_d^l$ denotes the offset (bias), and M_d represent the input feature maps while $\boldsymbol{f}$ represents the activation function.

The convolutional layer is frequently followed by a pooling layer to reduce the input dimensions for the following layers by down sampling feature maps from the previous layer. Typical types of pooling layers are max pooling and average pooling. The output x_d^l can be represented by the following equation:

$$\boldsymbol{X}_d^l = \boldsymbol{f}(\delta_d^l subsample(\boldsymbol{X}_d^{l-1}) + \boldsymbol{b}_d^l \tag{3}$$

where l is the number of the pooling layer, $\boldsymbol{f}$ can be an activation function, δ_d^l denotes the resample factor and

TABLE 1. Performance metrics (F1-Scores & execution times) for different classification methods and input data.

Roughness Estimation Model	**Proposed Convolutional Neural Network**		Statistical features and linear Support Vector Machine	
Image quality	**HDR**	LDR	HDR	LDR
Sa∈[0,3]	**0.81**	0.69	0.77	0.72
Sa∈(3,5]	**0.85**	0.73	0.69	0.69
Sa∈(5,12]	**0.93**	0.79	0.74	0.44
Sa∈(12,20]	**0.98**	0.93	0.82	0.85
Distortion	**0.99**	0.99	0.86	0.79
Average F1-Score	**0.91**	0.83	0.78	0.70
Trainable parameters	**1,512,197**		55	55
Training time	**2034 s**		3551 s	
Inference time per part surface image (49,728 patches)	**4.5 s**		336 s	

TABLE 2. Performance metrics (validation F1-Scores & execution times) for established CNN models from literature.

Roughness Estimation Model	ResNet50 (transfer learning	ResNet50 (fully retrained)	VGG16 (transfer learning)	**VGG16 (fully retrained)**
Image quality	HDR	HDR	HDR	**HDR**
Sa∈[0,3]	0.63	0.86	0.78	**0.90**
Sa∈(3,5]	0.20	0.91	0.72	**0.89**
Sa∈(5,12]	0.77	0.93	0.88	**0.95**
Sa∈(12,20]	0.86	0.99	0.95	**0.98**
Distortion	0.50	0.98	0.98	**0.99**
Average F1-Score	0.60	0.93	0.86	**0.94**
Trainable parameters	2,360,069	25,894,661	590,597	**15,305,285**
Training time	1702 s	4983 s	1815 s	**4407 s**
Inference time per part surface image (49,728 patches)	21.8 s	21.8 s	25.1 s	**25.1 s**

subsample (.) represents the down-sampling function (e.g., mean or max pooling), and $\boldsymbol{b}_d^l$ is the bias (offset). Pooling, especially max pooling, is a convolution-based operation that is applied to reduce overlapping in feature maps and can help to avoid over fitting and may lead to a more generalized model [63]. During network training, random drop out was used after the pooling layer and in the fully connected region of the network (layer 4) as an additional regularization technique that prevents the network from overfitting.

After concatenating the output of the third layer, the flattened feature vector is used as input for the output layer, using fully connected nodes with softmax activation to classify the image patches into five different surface classes.

To generate spatially resolved roughness maps, overlapping image patches are extracted by a sliding window approach that resamples the original image using a step size of 16 pixels. The process of extracting image patches from the original layer surface image of a specimen is parallelized and optimized to run on GPU. The roughness class probabilities for the output layer are obtained by feeding 1024 patches per batch as input for the trained CNN-model. The calculation of the roughness map for a single LPBF part surface image (i.e., 3661 × 3617 pixels relating to 49,728 image patches) takes approximately 4.5 seconds (TABLE 1 & TABLE 2) on a NVIDIA GeForce 1080 Ti GPU.

To evaluate the performance of the proposed CNN-architecture for roughness prediction, a benchmarking scheme as proposed by Narayanan *et al.* [64] is adopted to evaluate F1-Scores as well as training and inference times for different models. For this purpose, state-of-the-art CNN architectures, namely ResNet50 [38] and VGG16 [65], were used for comparison with respect to prediction performance and execution times. First, the pre-trained models have been used in the sense of transfer learning by replacing the original classification heads with a new fully connected head consisting of a hidden layer (i.e. 128 nodes) and the output layer (i.e. 5 nodes). In the case of transfer learning, all other layers were set to be untrainable. In addition, the performance of fully re-trained VGG16 and ResNet50 models with adapted classification head has been investigated.

Based on image characteristics extracted from the individual patches, a support vector machine algorithm [66] with linear kernel, was additionally used for roughness classification representing a classical ML approach. In this regard, four first-order statistical features, namely, "mean", "variance", "skewness" and "kurtosis" were extracted to describe a given image patch. Additionally, image features based on second order statistical methods were used in this work. Related to this, Haralick *et al.* proposed Gray-Level Co-Occurrence Matrix (GLCM) as an effective method for texture characterization [67]. The GLCM describes the histogram of co-occurring gray levels for a given area within a given image. Several features can be extracted from the GLCM for characterizing the texture of the given image patch. In addition to the aforementioned statistical features, the following GLCM characteristics are utilized for surface classification in this study: "dissimilarity", "Angular Second Moment" (ASM), "contrast", "homogeneity", "energy" and "correlation" [29], [67]. It is worth mentioning that GPUs were used for training and inference of the CNN models, while the classical approach was executed on CPUs.

For each CNN training process, the Nesterov-accelerated Adaptive Moment Estimation (Nadam) optimizer was used to minimize the loss function (i.e., categorical cross-entropy). The number of epochs was determined by tracking the training and validation errors. When it was observed that the validation error stagnated or increased, which is a sign of overfitting, the training was stopped. This resulted in 200 epochs for training the proposed CNN and 65 epochs for the reference CNN architectures. The architectures were implemented using TensorFlow 2.3 [68] and Python 3.6. Compared to other deep CNN architectures such as VGG16 or ResNet, a comparatively shallow network architecture is assumed to be more suitable for the given problem as it prevents overfitting and leads to higher computational efficiency and lower memory requirements.

B. MODEL BASED REINFORCEMENT LEARNING FOR PROCESS OPTIMIZATION

Learning to maximize the cumulative feature rewards of an agent through active interaction with a given environment essentially describes the framework of RL [54]. As a branch of machine learning, RL allows an agent to choose an optimal action a_t from a pool of candidate actions A in a given state s_t by selecting those actions that maximize the rewards the agent is expected to receive. To express the problem within the RL framework it is formulized as a Markov Decision Process (MDP) that shows the following characteristics:

1) An action $a_t \in \mathcal{A}$ the agent can choose to interact with its environment.
2) The agent's current state $s_t \in \mathcal{S}$ and the state $s_{t+1} \in \mathcal{S}$ after taking an action a_t.
3) The state-transition probabilities $p(s_{t+1}|s_t, a_t)$ that action a_t in state s_t at the agent's next step will result in state s_{t+1}.
4) The expected immediate reward $r(s_{t+1}, a_t, s_t)$ received after the state transition (i.e., s_t to s_{t+1}) based on action a_t.

The goal of MDP is to approximate a decision policy in form of a function $\pi(s)$ that defines the optimal action to be selected by the agent based on a given state.

In this work, the agent denotes a software implementation of a RL method that choses LPBF process parameter based on a given state and a learnt policy. For that, the agent's current state s_t is defined as a tuple $s_t = (P_t, v_t, Sa_{mean,t}, \delta_t)$ comprised of the applied laser power P_t and scan velocity v_t as well as mean surface roughness $Sa_{mean,t}$ and percentage of defective areas δ_t for a given part surface at time t. The quality metrics $Sa_{mean,t}$ and δ_t are estimated by the CNN-model explained in section 2.1. In each step, the agent can choose from a defined set of action described as follows:

$$\mathcal{A} = \left\{ \begin{matrix} (P_{none}, v_{down}); (P_{none}, v_{up}); (P_{down}, v_{none}) \\ (P_{down}, v_{down}); (P_{down}, v_{none}); (P_{up}, v_{none}) \\ (P_{up}, v_{down}); (P_{up}, v_{up}) \end{matrix} \right\} \quad (4)$$

where each element in the list represents an action tuple that consists of a possible action value for laser power P_{t+1} and scan velocity v_{t+1} to be applied in the next layer. Each process parameter can have the value up, down, or none which represent the action of increasing, decreasing, or applying no change at all with respect to the given process parameter.

Different RL algorithms can be used to estimate the optimal policy for choosing an action in a given state. The policy can be learnt with or without having a model that approximates the environment, referring to model-based and model-free RL respectively. Both approaches face their own challenges, but each offers unique advantages. Generally, model-based approaches show higher data efficiency and faster learning compared to model-free methods, which on the contrary can be used for a variety of applications and avoid the incorporation of model errors into the learnt policy [69], [70].

Since model-free learners require a larger number of training examples, their attractiveness for real-world applications such as laser materials processing is reduced. Without a corresponding simulation in which the agent can interact with the modeled environment to derive an effective control strategy, model-free learning appears to be disadvantageous.

In MBRL, the system dynamics are modeled as a function $\hat{f}_\theta(a_t, s_t)$ to estimate the next state s_{t+1}. The function is parametrized by θ and can be formulated as a regression task within the scope of supervised learning that maps a given state s_t and action a_t at time t to predict the subsequent state s_{t+1} at time $t + 1$. Based on the dynamics function $\hat{f}_\theta(a_t, s_t)$, planning can be used to infer the optimal action for a given state using sampling-based planning. A simple method based on random sampling is used to generate C candidate random actions sequences. The sequences are sampled from a fixed distribution and subsequently evaluated based on $\hat{f}_\theta(a_t, s_t)$, the current state s_t and the expected immediate reward $r(s_{t+1}, a_t, s_t)$ formulated as [54]:

$$(a_t, \ldots, a_{t+T-1}) = \underset{(a_t,\ldots,a_{t+T-1})}{\arg max} \sum_{t'=t}^{t+T-1} r(\hat{s}_{t'}, a_{t'})$$
$$\text{with } \hat{s}_{t'+1} = \hat{s}_{t'} + \hat{f}_\theta(a_{t'}, \hat{s}_{t'}) \quad (5)$$

Random sampling of action sequences avoids solving Eq. 5 numerically, which may lead to instabilities due to non-linear reward and dynamics functions [71].

Since the action space defined for the problem in this work is comparatively small (8 actions), random sampling sequences of consecutive actions $(a_t, \ldots, a_{t+T-1})$ can be a feasible way to solve the optimal control problem. Based on equation (4), the action sequence that promises the highest expected cumulative reward with respect to the predicted future states, $T_{horizon}$ time steps ahead is then selected. Finally, by utilizing the framework of Model Predictive Control (MPC) [72], the first action a_t of the optimized action sequence is chosen for execution. As the agent transits to a new state s_{t+1}, the MPC based policy recalculates the optimal action sequence and selects the first action for the next time step. The steps for model-based RL is denoted as Algorithm 1 based on the work of Nagabandi *et al.* [69].

As reference method a model-free RL approach based on Q-Learning is used for comparison. The algorithm is based on a bellman equation as a simple value iteration update and combining the old and new value using weighted average [54]:

$$Q_{\pi,\text{new}}(s_t, a_t) \leftarrow (1-\alpha) * Q_{\pi,\text{old}}(s_t, a_t) + \alpha * (r_t + \gamma \max_a Q(s_{t+1}, a)) \quad (6)$$

where α represents the weight formally known as the learning rate, r_t denotes the expected immediate reward and $\gamma max_a Q(s_{t+1}, a)$ describes the maximum reward that can

Algorithm 1 Model (Random Forest) Based Reinforcement Learning

```
Gather dataset $\mathcal{D}_{ini}$ of experimental data (e.g., random policy)
Initialization of empty dataset $\mathcal{D}_{real}$ and
Random Forest with n estimators
for layer = 1 to layer_max do:
    Train $\hat{f}_\theta\ (a_t, s_t)$ based on random forest
    algorithm [66] and datasets $\mathcal{D}_{ini}$ and $\mathcal{D}_{real}$
    for t in $T_{horizon}$ do:
        Obtain the current state defined
        as $s_t = (P_t, v_t, Sa_{mean,t}, \delta_t)$
        Estimate optimal action
        sequence $(a_t, \ldots, a_{t+T-1})$
        using $\hat{f}_\theta\ (a_t, s_t)$ and (Eq. 5)
        Apply first action $a_t$ from
        estimated optimal action
        sequence
        Add data point $(a_{t-1}, s_{t-1})$,
        $s_t$ to dataset $\mathcal{D}_{real}$
    end for
end for
```

be obtained from state s_{t+1} based on all available actions a, discounted by γ. The algorithm was implemented in Python 3.6 and further hyperparameters were determined to $\gamma = 0.9$ and $\alpha = 0.3$. For both algorithms' epsilon-greedy ($\varepsilon = 0.1$) action selection was used during training

C. REWARD AND ENVIRONMENT SPECIFICATION

The reward function has major impact on the success of the agent to learn and is defined in a way that enforces the agent to take actions that result in a high cumulative reward and avoid actions, which lead to low or negative reward.

The agent used in this work receives a negative numeric reward for the current state $s_t = (P_t, v_t, Sa_{mean,t}, \delta_t)$ if the percentage of surface defects δ_t detected by the CNN in the current surface image is higher than 10 %. At this point, a human expert would have to decide whether the build process should be stopped and modified or aborted. From the agent's perspective, the optimization episode terminates with a negative reward of -10,000 for the recent action applied.

A positive reward of 2,000 is given for a predicted roughness value $Sa_{mean,t}$ below 4 μm in combination with defective surface areas smaller than 10 %. At this point, the optimization episode is terminated, as the current state is considered optimal. In addition, a continuous reward function is formulated as $r_{Sa}\ (a_t, s_t)$:

$$r_t = \begin{cases} -10,000 & if\, \delta_t > 10\% \\ 2,000 + r_{Sa}\ (a_t, s_t) & if\, Sa_{mean,t} < 4\mu m \\ r_{Sa}\ (a_t, s_t) & else \end{cases}$$

$$\text{With } r_{Sa}\ (a_t, s_t) = w_{Sa} \frac{1}{Sa_{mean,t}}$$

$$\text{and } Sa_{mean,t} = \hat{f}_\theta(a_t, s_{t-1}) \qquad (7)$$

were $Sa_{mean,t}$ represents the continuous mean surface roughness of the part and w_{Sa} ($w_{Sa} = 150$ this work) denotes an experimentally determined weight factor to adjust the overall reward behavior.

To generate continuous roughness values $Sa_{mean,t}$ based on the predicted roughness classes, the following empirical formula is defined:

$$\begin{aligned} Sa_{mean,t} &= \sum_{i=1}^{4} p_i * c_i \\ With c_1 &= 2.75\mu m; \\ c_2 = 4.5\mu m;\ c_3 &= 9\mu m;\ c_4 = 18\mu m; \end{aligned} \qquad (8)$$

where p_i denotes the relative number of pixels belonging to class i within a given image and c_i states an experimentally derived scaling factor for each class.

The reward function above provides rewards exponentially to decreasing surface roughness values, which are normalized between 0 and 1. This additionally encourages the agent to find optimal actions that allow transitions to a state with particularly low surface roughness.

In our setting the agent interacts with a simulated environment based on data from the experiments. Based on an action the agent takes in a certain state, the simulation provides the agent a new state considering the next possible parameter change available within the experimental dataset.

In this work, the upper limit for laser power values is 450 W, the lower limit is 150 W with a step size of 75 W. The lower limit for scan velocity is 667 mm/s, the upper limit is 2000 mm/s. Possible steps for scan velocity are 667, 800, 1000, 1333, 2000 mm/s. In case a transition is not possible due to an invalid action selection (i.e., parameter is out of range), the state will not change, and the agent eventually would have to learn an alternative action to reach the maximum reward. Although the described setup allows building parts using 25 combinations of different laser powers and scan velocities, the experimental data in this preliminary study contain only nine process parameter combinations. However, for each combination a part was built with two different heights and in at least four different positions on the build platform. If more than one data point is available for a new parameter state requested as a result of an action taken (e.g., parts with $P_l = 300$, $v_s = 1000$mm/s were built at four different positions on the build platform for two different layer heights), the simulation randomly chooses a part that fulfills the requested state-action transition. Consequently, based on the part's surface image evaluation via CNN-model, it provides the full state s_t=$(P_t, v_t, Sa_{mean,t}, \delta_t)$ to the agent. Consistent with the high variances in the estimated quality metrics as reported in FIGURE 10, the agent must cope with uncertainty in state-action transitions due to the influence of the part position and layer height within the build chamber and other unknown effects on surface roughness.

It is worth mentioning that the current environment implementation assumes that a parameter change is independent of the surface texture in the previous layer. This might be true for

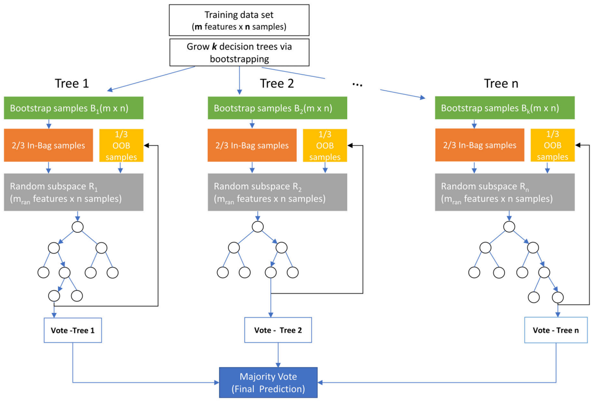

Training data set
(m features x n samples)
Grow k decision trees via bootstrapping
Tree 1
Tree 2
...
Tree n
Bootstrap samples B$_1$(m x n)
Bootstrap samples B$_2$(m x n)
Bootstrap samples B$_k$(m x n)
2/3 In-Bag samples
1/3 OOB samples
Random subspace R$_1$ (m$_{ran}$ features x n samples)
Random subspace R$_2$ (m$_{ran}$ features x n samples)
Random subspace R$_n$ (m$_{ran}$ features x n samples)
Vote -Tree 1
Vote - Tree 2
Vote - Tree n
Majority Vote
(Final Prediction)

FIGURE 4. Schematic overview of the random forest algorithm used to approximate the dynamics function in this study.

small changes of process parameters but it is not necessarily true in case of strong surface distortions and defects which would require several layers to heal or to not a heal at all.

D. RANDOM FOREST FOR DYNAMICS FUNCTION APPROXIMATION

While neural networks [69] and Gaussian Process Regression [57] were recently used for data-driven modeling of process dynamics, in this work an ensemble technique is proposed to approximate the dynamics function $\hat{f}_\theta$. Random Forest Regression is chosen to establish a dynamics model that predicts the next state s_{t+1} based on the current state s_t=(P_t, v_t,$Sa_{mean,t}$, δ_t) and a given action a_t. The values for the next process parameters P_{t+1} and v_{t+1} can directly be calculated by the action given. For the remaining state members, two different RF regression models were trained separately, one to predict the surface roughness $Sa_{mean,t+1}$ and the other to estimate the percentage of surface defects δ_{t+1}. Starting with FIGURE 4, a brief introduction of random forests is given. A random forest (RF), introduced by Breiman *et al.*, can be considered as a combination of a specific number k, typically a few hundred to several thousand of decision trees while each tree represents a single non-parametric regression model. The final regression decision is made by majority votes based on the participating decision trees. RF uses bootstrap aggregating (bagging) and a random feature subspace for each decision tree to increase the level of randomness [73]. This procedure can be interpreted in such way that each tree in the RF focuses on a slightly different aspect of the data set, leading to different expert models each contributing a vote for the final prediction.

Based on this strategy it turns out that RF is a very competitive alternative to many other regressors, including linear regression, support vector regression and neural networks [74]. The advantage of combining randomized decision trees and bagging to a random forest is a potentially more robust model, which achieves a better generalization error [75]. In summary, the main advantages of RFs for dynamic functions are sound resistance against overfitting, insensitiveness to prior feature selection and less outlier sensitivity during training [76].

The number of trees k (i.e., 100 in this work) of which the forest consists is an important parameter for the final regression performance. Additionally, mean squared error is used as a metric that measures the quality of a split during the generation of a decision tree. An efficient RF implementation based on scikit-learn was used in this work [66].

The dynamics function can be learnt using data from previous experiments in the form of tuples $\mathcal{D}_{ini}$ = $\{(a_{t,ini}, s_{t,ini}),s_{t+1,ini}\}_{t=1}^N$ that contain information about the state, action and resulting state relationship in form of an initial data set $\mathcal{D}_{ini}$. Additionally, as the agent gathers new experience while interacting with the real environment, the new data $\mathcal{D}_{real}$= $\{(a_{t,real}, s_{t,real}),s_{t+1,real}\}_{t=1}^N$ can be used to retrain the dynamics function based on the aggregated data set $\mathcal{D}_{ini}$ + $\mathcal{D}_{experience}$ to increase the model accuracy and continuously adapt it towards real word (changing) process dynamics.

III. EXPERIMENTAL SETUP AND DATA PREPROCESSING

According to experimental investigations mentioned in section I, the surface roughness significantly depends on the VED, a measure for the energy input related to a specified material volume. The energy density can be calculated by the following formula [20]:

$$VED = \frac{P_l}{\Delta y_s \cdot v_s \cdot \mathcal{D}_s} \tag{9}$$

where *VED* stands for volumetric energy density [J/mm^3], P_l for laser power [W], Δy_s (90 μm = const.) for distance between two neighboring laser scanning paths [mm], v_s for scan velocity [mm/s] and D$_s$ (40 μm = const.) for layer thickness [mm]. The variation of the energy density in this work is realized by adjusting the scan velocity and laser power. As a reference, laser power and laser scanning speed are used as 100 % at 300 W and 1000 mm/s respectively, resulting in a VED of 75 J/mm^3. For the production of different cubic samples, the laser power and laser scanning speed vary, so that the relative VED changes from 50 % to 150 % by 25 % each step. Based on this, 44 cubic specimens (10 × 10 × 10.96 mm^3) of identical height (274 layers) were built on the first platform using nine different sets of process parameters in at least four different and randomly selected positions, as shown in FIGURE 5. The parts on the second platform were built using the same process parameters and positions on the platform, however the build job was stopped in layer 121, resulting in additional 44 cubic samples of size 10 × 10 × 4.82 mm^3. Overall, 88 cubic samples with the same surface area size were built, representing 18 unique parameter combinations. As shown in FIGURE 10, the experimentally determined LPBF process window for this setup is narrower (i.e., 75+/−10 J/mm3) than the range covered by the combinations (i.e., 35 to 112.5 J/mm3). However, a larger range was chosen so that the optimization algorithm can also explore

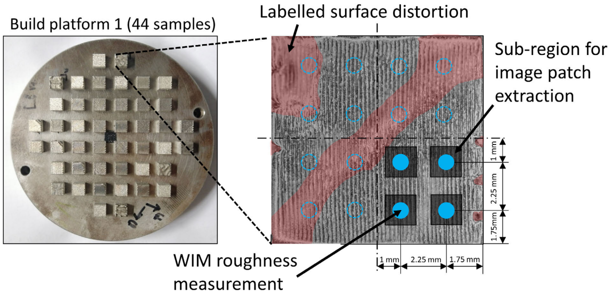

Build platform 1 (44 samples)
Labelled surface distortion
Sub-region for image patch extraction
WIM roughness measurement
1 mm
2.25 mm
1.75 mm
1 mm
2.25 mm
1.75 mm

FIGURE 5. Photography of the build platform 1 (left). Definition of WIM measurement points and sub-region areas (right). The same points are measured in each quadrant to obtain 16 roughness values (Sa) for each cubic sample.

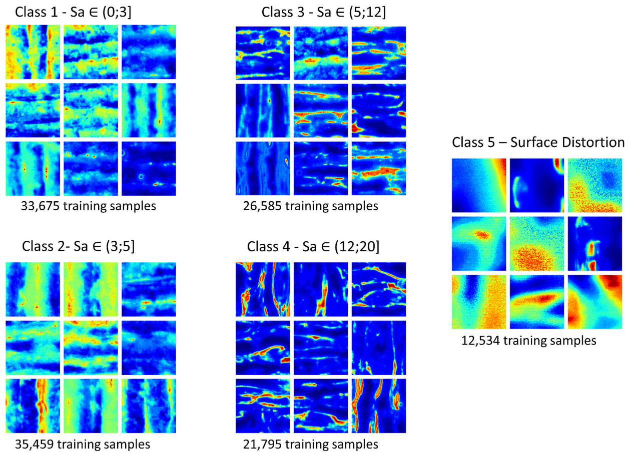


FIGURE 6. Example HDR image patches (96 × 96 pixels) and corresponding class definitions.

critical process situations that can lead to surface deviations and low part density. The material chosen for all samples built for this study is Inconel 718 and a meander-based scan strategy with layer-wise scan vector rotation of 90 degree was applied.

After processing the parts, HDR images were generated as explained in section II-A. For this preliminary study, the decision was made to take the images after removing the parts from the build chamber to investigate the influence of high-resolution images that are difficult to obtain from a setup within the build chamber. The imaging system is comprised by an industrial camera (LUCID vision TRI200S 20 MP -mono) with a pixel size of 2.74 μm and a 105 mm optical lens. After calibration of the lens, the field of view is approximately 12.34 mm × 8.26 mm with a resolution of approximately 2.26 μm/Pixel. The field of depth is determined to $+/-1.1$ mm.

In addition to the HDR image patches, a measurement of the local top surface roughness of the LPBF samples is required as input information for training the CNN. Instead of using the common line-related Ra or Rz values, the surface roughness is characterized in this work by area-related mean arithmetic height Sa (Eq. 1).

For roughness measurement, a white light interferometer (WIM) namely Zygo Newview 7300 was used. The component surface was enlarged ten times which results in a resolution of 1.14 μm and measurement area is 0.73×0.54 mm^2. The roughness measurement was carried out at 16 reference points according to the definition in FIGURE 5. For the class ‘‘surface distortion’’, an expert carefully analyzed the LPBF parts together with the related surface image data. In case of visible surface distortions in form of height deviations greater than 1 mm, the corresponding surface areas were annotated accordingly. To create a database for the CNN model to be trained, each surface image from the total of 88 LPBF cube surfaces is first divided into sixteen sub-regions with the size of 1.7×1.7 mm^2. The center point of each sub-region is the same as it is for the WIM measurement as depicted in FIGURE 5.

Subsequently, small image patches (96 × 96 pixels) are extracted from the sub-regions using a sliding window approach and finally aggregated, as well as labeled separately according to the measured Sa value in each sub-region.

Examples for typical image patches that belong to different roughness classes shown in FIGURE 6 and are defined by the following classes:

1) Class 1 - Sa ∈ (0;3] (very low roughness)
2) Class 2 - Sa ∈ (3;5] (low roughness)
3) Class 3 - Sa ∈ (5;12] (mid roughness)
4) Class 4 - Sa ∈ (12;20] (high roughness)
5) Class 5 - Surface Distortion (undefined roughness)

To prevent the model from learning only features that are highly directional, data augmentation such as mirroring, and 90-degree rotation is used. The CNN for roughness classification is trained on 130,048 annotated HDR image patches extracted from data of 67 specimen layer surface images. The evaluation of the CNN-model is based on 4-fold cross validation and the final evaluation of the overall framework is only performed on data from the remaining 21 parts. In a last step, the density of the manufactured parts is evaluated. For this purpose, the parts are separated from the building platform by means of wire erosion. For evaluation of the relative density, the samples are cut in half for preparation of polished cross sections. For each cross section five microscopic images with an enlargement factor of 50 are acquired and analyzed in view of porosity by applying threshold analysis, that yields a mean value of the samples’ relative density.

IV. RESULTS AND DISCUSSION

A. ROUGHNESS PREDICTION

The following section first shows the model’s performance assessed during the training procedure. Subsequently, the roughness prediction model is validated based on unseen LPBF surfaces. In FIGURE 7, the accuracy of the proposed CNN-model during training of 200 epochs is depicted. The graphs on the left side show the 4-fold cross-validated training accuracy for different input patch sizes. As expected, the accuracy increases with increasing input patch size, which maximizes at high overall classification accuracy ($>$90 %) for

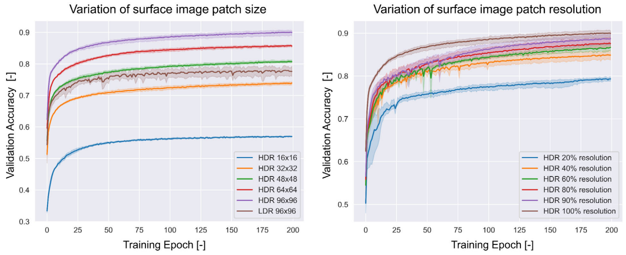


FIGURE 7. Performance of the proposed CNN during training using different image patch sizes and image resolutions based on 4-fold cross validation.

input patches at size 96 × 96. Although the patch size could be further increased, memory consumption reaches hardware limit (without further optimization), while the performance benefit is marginal. When using the trained CNN-model in real-world scenarios, large input patches would also require larger surface areas with constant roughness for correct classification, which does not meet real-world requirements. Therefore, the input size is set to 96 × 96 pixel in this work. The same graph shows the training performance based on HDR image patches (purple line) and LDR patches (brown line) for input. The HDR version finally outperforms the LDR version by more than 12 %.

The graphs on the right part of FIGURE 7 show the dependency of the image resolution on model accuracy. For each graph, the original input patches of size 96 × 96 pixels and 2.26 μm/pixel are downsampled by the given percentage factor and subsequently upsampled to meet the required CNN target input shape of 96 × 96 pixels. The combination of down and upsampling approximates the influence of optical and digital image resolution on model accuracy. When downsampling the bilinear sampling filter kernel is scaled to properly anti-alias the input image signal [68]. The results show that image downsampling to 20 % (approx. 11.32 μm/pixel) of the original resolution is possible while the classification performance is still moderate (>78 %). This supports the thesis that low frequency image information (i.e., the surface waviness due to scan vector overlapping) is important for surface roughness classification. According to FIGURE 7, a spatial image resolution of at least 5.66 μm/pixel, representing 40 % of the original resolution, is recommended for high roughness classification accuracies (>80 %).

For different input image types (i.e., HDR and LDR), the comparison of the proposed CNN-based surface classification with the reference SVM approach is described in TABLE 1. HDR imaging shows its benefits for both roughness estimation methods. Moreover, the proposed CNN architecture clearly outperforms the classical ML approach that uses statistical and texture feature extraction in combination with support vector machines for classification.

When the proposed CNN is compared with established CNN models, as depicted in TABLE 2, it is shown that the highest F1 score of 0.94 can be obtained with a fully re-trained VGG16 with an adapted classification head. The performance of the VGG16 and ResNet50 models using weight transfer is also shown in TABLE 2. It is assumed that the comparatively lower prediction performance is due to large differences in the shape of the images on which the models were originally trained (i.e., 224 × 224 ×3 pixels) and the image patches used in this work (i.e., 96 × 96 pixels). Additionally, the domain gap between the original dataset (i.e., ImageNet) and the dataset used in this work could play an important role. Furthermore, due to their higher complexity, the established models require more time for training and inference. In particular, the high inference time of the VGG16 of 25.1 s per surface image is too high for real-time inter-layer process optimization. Since powder recoating is required after each layer is processed, the time to move the slider (i.e., recoating time) can be used to derive quality metrics and calculate new process parameters.

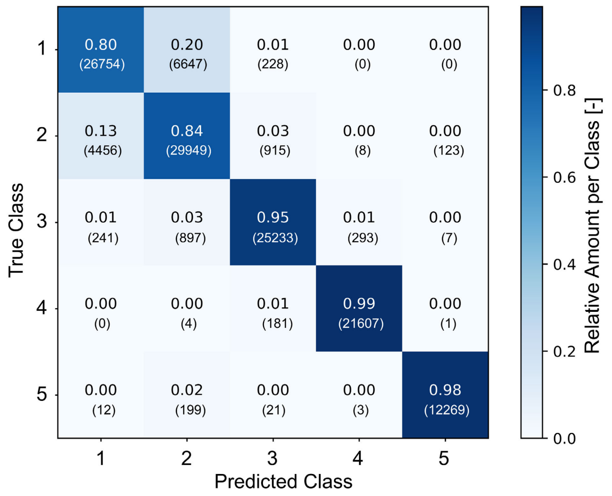
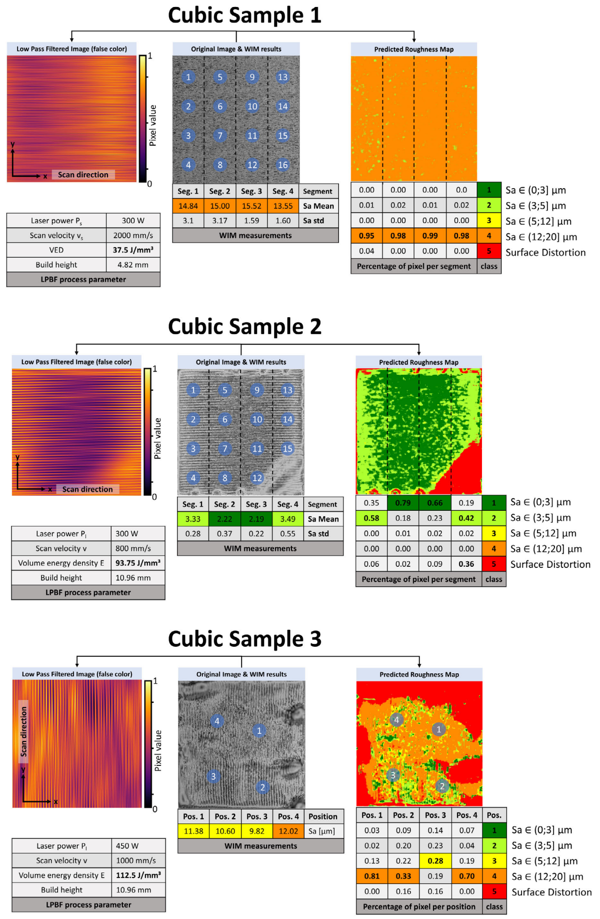


FIGURE 8. Normalized and cumulative confusion matrix (with absolute values in brackets) for the proposed CNN-model using 4-fold cross-validation.

For the LPBF machine used in this study, the recoating time is determined to 5.3 s. In this comparison, it is shown that the proposed CNN allows the estimation of surface roughness within 5.3 seconds and is therefore suitable with respect to the time constraints in this application. As depicted in FIGURE 8 the cumulative confusion matrix for the proposed CNN based on four different cross-validation test folds is shown for surface classes 3, 4 and 5. All presented methods have difficulties to distinguish class 1 and class 2 from each other, which also appear to be visually similar according to sample patches in FIGURE 6.

Moreover, since the more complex reference CNN models have higher F1-Scores, this may indicate that the proposed CNN slightly underfits the data. However, due to further timing constraints (e.g., recoating time between layers), possible underfitting caused by reduced model complexity is tolerated to achieve low inference times. Nevertheless, an overall accuracy of 91 % and a comparatively fast inference time of 4.5 s



Cubic Sample 1

Laser power P_s	300 W
Scan velocity v_s	2000 mm/s
VED	**37.5 J/mm³**
Build height	4.82 mm

LPBF process parameter

Seg. 1	Seg. 2	Seg. 3	Seg. 4	Segment
14.84	15.00	15.52	13.55	**Sa Mean**
3.1	3.17	1.59	1.60	**Sa std**

WIM measurements

0.00	0.00	0.00	0.0	1	Sa ∈ (0;3] µm
0.01	0.02	0.01	0.02	2	Sa ∈ (3;5] µm
0.00	0.00	0.00	0.00	3	Sa ∈ (5;12] µm
0.95	**0.98**	**0.99**	**0.98**	4	Sa ∈ (12;20] µm
0.04	0.00	0.00	0.00	5	Surface Distortion
Percentage of pixel per segment				**class**	

Cubic Sample 2

Laser power P_l	300 W
Scan velocity v	800 mm/s
Volume energy density E	**93.75 J/mm³**
Build height	10.96 mm

LPBF process parameter

Seg. 1	Seg. 2	Seg. 3	Seg. 4	Segment
3.33	2.22	2.19	3.49	**Sa Mean**
0.28	0.37	0.22	0.55	**Sa std**

WIM measurements

0.35	**0.79**	**0.66**	0.19	1	Sa ∈ (0;3] µm
0.58	0.18	0.23	**0.42**	2	Sa ∈ (3;5] µm
0.00	0.01	0.02	0.02	3	Sa ∈ (5;12] µm
0.00	0.00	0.00	0.00	4	Sa ∈ (12;20] µm
0.06	0.02	0.09	**0.36**	5	Surface Distortion
Percentage of pixel per segment				**class**	

Cubic Sample 3

Laser power P_l	450 W
Scan velocity v	1000 mm/s
Volume energy density E	**112.5 J/mm³**
Build height	10.96 mm

LPBF process parameter

Pos. 1	Pos. 2	Pos. 3	Pos. 4	Position
11.38	10.60	9.82	12.02	Sa [µm]

WIM measurements

Pos. 1	Pos. 2	Pos. 3	Pos. 4	Pos.	
0.03	0.09	0.14	0.07	1	Sa ∈ (0;3] µm
0.02	0.20	0.23	0.04	2	Sa ∈ (3;5] µm
0.13	0.22	**0.28**	0.19	3	Sa ∈ (5;12] µm
0.81	**0.33**	0.19	**0.70**	4	Sa ∈ (12;20] µm
0.00	0.16	0.16	0.00	5	Surface Distortion
Percentage of pixel per position				**class**	

FIGURE 9. **Surface roughness measurements and roughness classification results based on HDR image data and the proposed CNN.**

appear promising for the proposed CNN-based roughness prediction model, which is therefore used for further analysis.

After training the neural network the obtained predictive model is used to classify roughness patches extracted from unseen part surface images into the defined categories to generate a surface roughness map. Examples for the surface classification results are depicted in FIGURE 9. Sample number one was build using 375 W laser power and a scan velocity of 2000 mm/s. The positions where the roughness was determined via WIM are shown as circles in the original surface image. Due to high directional dependencies, the surface is divided in four equal-sized segments and the mean roughness measurements are shown for each segment. Right to the original image, the roughness map derived from the CNN predictions is shown. The CNN-model classifies most (>97 %) of the surface into category four, which represents roughness values within a range of 12 to 20 μm. Although this potentially is a large range for roughness classification, the result is consistent with the roughness values measured by WIM. Regarding surface texture, the original image and the predicted roughness map show high texture homogeneity for the given surface, which also matches the WIM measurement results for each segment. Left to the original image of sample one, a low pass filtered false color version of the original image is shown. As high image frequencies (i.e., sharp edges and corners due to pixel values that are changing fast over space) are suppressed, the surface waviness due to the meander-based scan strategy (i.e., horizontal track overlapping) becomes visible.

In case the same filtering procedure is applied to the second sample, the scan vector related waviness appears more clearly, especially at the horizontal edges of the cube's surface. Compared to the first part, sample two was built at increased VED (93.75 J/mm^3), which leads to a completely different image texture with less overall surface texture homogeneity. The predicted roughness values for segment 2 and segment 3 (center region) are categorized into class 1, which matches the measured roughness values for those sections. For segment 1 and 4, the border region, most pixels are categorized into class 2, which represents roughness values of 3-4.99 μm (*Sa*).

The WIM measurement for these sections confirms the prediction and shows increased roughness values for the border regions. In the right bottom corner of the surface image, a surface distortion leads to the partially categorization of pixels in section 4 into class 5. The surface distortion can be described as a strong deviation in height (i.e., −/+1–5 mm) from the normal surface and leads to a blurring effect due to a limited depth of focus that is also visible in the original image. In the low pass filtered image of part five, the defective area shows less waviness and higher pixel values, which is highlighted by a higher proportion of low image frequencies in this area. It is assumed that a positive correlation exists between the frequency of occurrence of this type of surface distortion and increasing volumetric energy density.

When evaluating sample three, which was built with increased VED (i.e., 112.5 J/mm^3), strong deviations in surface height can be obtained by visual inspection that are also visible in the original HDR surface image. The blurred surface areas lead to a categorization by the CNN-model into class 5 for most of the border regions of the image.

Due to the high surface deviations, roughness measurements could only be performed in the center region. In this case, the roughness values were obtained at four different positions. The results range between 9.82 -12.02 μm (*Sa*), which is only partially consistent with the prediction by the CNN. Instead, the roughness prediction for these positions results in an increased number of pixels belonging to the classes 2, 3, and 4. Compared to the other parts, the low pass filtered image reveals a high degree of inhomogeneity, which corresponds to the occurring surface deviations.

In addition to the evaluation of single samples, the overall results for all parts are shown in FIGURE 10. The measured and predicted relative amount of defective surface areas of samples within different groups of volumetric energy densities is depicted in FIGURE 10 - a). Compared to the ground truth that is based on the defect annotations of a human expert, the prediction results for each VED group are in good align-

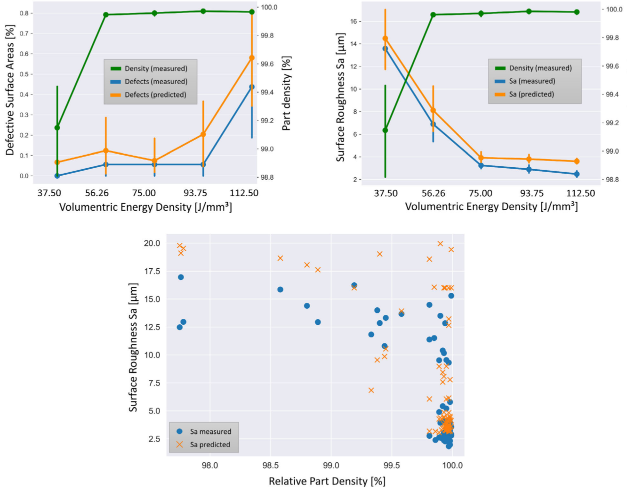


FIGURE 10. Correlation between surface roughness, surface defects, part density and energy input based on 88 LPBF cube surfaces.

ment. Both graphs show increased probabilities for surface defects at high VEDs. At the same time, no clear correlation can be established between the predicted amount of defective surface areas within a given surface image and the measured overall part densities.

In FIGURE 10 - b), the relationship between the surface roughness, the part density and the VED is shown.

For easier comparison, continuous roughness values are obtained from the predicted roughness classes by using Eq. 9.

It is shown that measured roughness values correspond to the predicted values, which, however, reveal higher standard deviations. The measured part density shows a negative correlation with the surface roughness especially for parts with lower VED. This is also supported by FIGURE 10 - c), which reveals the surface roughness over part density. It is shown that low surface roughness correlates with high part density. According to this figure, if the part density were to be above 99.75%, the surface roughness would have to be kept at 7.5 μm. It must be mentioned that both quality indicators (i.e., roughness and part density) are strongly influenced by VED. Overall, it can be seen from the experiments that high surface roughness values indicate decreased part density, which needs to be avoided due to decreasing mechanical properties. The VED can be increased up to a certain level, whereby surface roughness decreases, and part density remains at a high level. However, if the VED exceeds a threshold value, in this case 93.75 J/mm^3, the probability of surface defects due to deformations caused by overheating effects increases rapidly. Furthermore, the maximum VED is particularly dependent on the material used and the component geometry.

B. PROCESS OPTIMIZATION

Using MBRL, the predicted surface roughness and the predicted amount of defective surface areas are leveraged to optimize part quality. For that, the MBRL agent is trained based on Algorithm 1. The dataset $\mathcal{D}_{ini}$ for initializing the dynamics function is based on ten different surfaces from parts build in previous experiments. In a next step, 50 % of the experimental data was used to train the agent to choose optimal actions (i.e., process parameter for the next layer). FIGURE 11 shows the results of the training process where each step represents an interaction of the agents with the environment. Training was stopped after 1000 steps since the MBRL algorithm showed significant convergence as the moving average of 100 steps, which was chosen due to high variance in the reward signal, stopped increasing. The reward received by the MBRL agent starts at a higher value and converges to a higher total reward faster than the reference implementation (Q-learning). Since the dynamics function is initialized with data from previous experiments, the MBRL agent is able to choose actions better than by random choice at the beginning of the training process. After each step, the dynamics function is updated with new data generated through interaction with the environment, which further improves the agent's performance. Opposite to that, the Q-learning algorithm starts with randomly choosing actions at the beginning and updates its policy after each step.

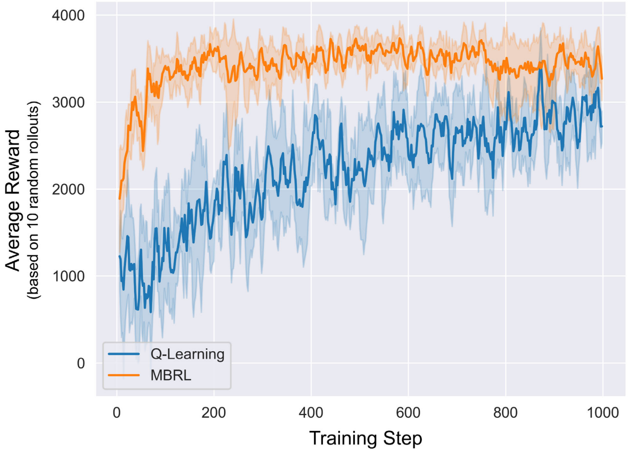


FIGURE 11. Rewards received by the MBRL and Q-learning agents in each step during the training process. For a more robust evaluation, the rewards are calculated as the average of 10 random rollouts on the training data.

It should be mentioned that the Q-learning agent could also be initialized with data from previous experiments. However, the effort to build a suitable environment for the agent to interact with, including the necessary rewards, makes Q-learning approach less efficient than MBRL. Creating an initial dataset for training the dynamics function is comparatively inexpensive and comes with the additional benefit not being biased towards the used reward function. Finally, both agents can improve the surface roughness and reduce the number of defective areas, however, the MBRL approach reaches the defined goal significantly faster as shown in FIGURE 11.

After the training procedure, the MBRL approach is evaluated on unknown top surfaces of LPBF cubic samples. The environment starts with a randomly selected surface image from a list of 21 LPBF surfaces that shows either a measured

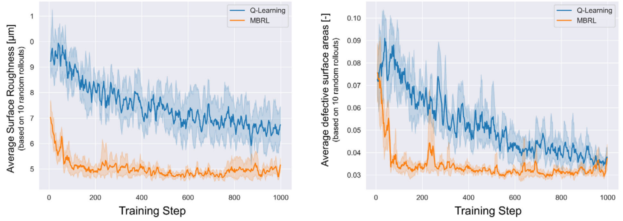


FIGURE 12. Predicted mean surface roughness and percentage of defective areas after each training step. For a more robust evaluation, the metrics are calculated as the average of 10 random rollouts on the training data.

TABLE 3. Overall results for 21 sample surfaces evaluated using the trained CNN in combination with the MBRL agent.

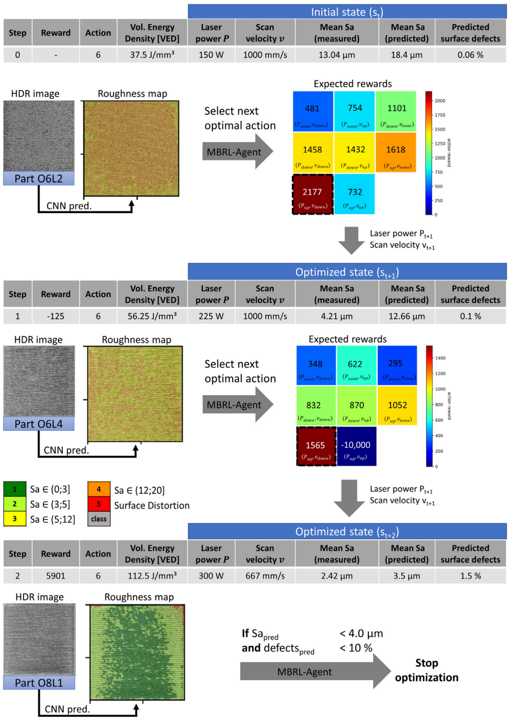
Criteria	Average (21 parts)	Standard deviation (21 parts)
Initial surface roughness $Sa_{mean,ini}$	10.40 µm	5.44 µm
Post optimization surface roughness $Sa_{mean,opt}$	3.38 µm	0.28 µm
Post optimization surface defects δ_{opt}	2.9 %	1.2 %
Required optimization steps	1.97	0.89
Reward after optimization	5851	823

mean surface roughness value above 4.0 μm or is labeled as "surface distortion". After the mean surface roughness $Sa_{mean,t}$ and the percentage of defective surface areas δ_t are predicted by the CNN, the agent must choose an optimal action or action sequence to achieve the optimization goal. An example for an optimizations sequence is given in IV-C. The trained MBRL agent starts with a state representing low laser power (150 W), medium scan velocity (1000 mm/s) and a measured mean surface roughness of 13.04 μm. After assessing the current state $s_t = (P_t, v_t, Sa_{mean,t}, \delta_t)$, the agent choses the action with highest expected rewards, estimated by using the dynamics function (Eq. 5) in combination with the reward function (Eq. 7).

The MBRL approach is able to reduce the average surface roughness during two optimization steps to 2.42 μm by increasing the laser power to 300 W and decreasing the scan velocity to 667 mm/s. The roughness prediction suggests slightly higher roughness values in both cases. The predicted amount of defective surface areas increases from 0.06 % to 1.5 %, which is probably due to the uncertainty of the CNN-model regarding the prediction at the surface edge. In both optimization steps, the optimal action is the one that maximally increases the VED (i.e., (P_{down}, v_{up}) which agrees with the relationships depicted in FIGURE 10.

An overall evaluation of the 21 LPBF surfaces optimized by the approach presented in this work is given in TABLE 3. The mean surface roughness Sa of components was 10.40 +/− 5.44 μm.

Five out of 21 components are labeled as "surface distortion". Each surface was set as starting point for an individual optimization sequence based on the combination of roughness evaluation through CNN-model and parameter optimization using the trained MBRL agent. It is shown that the mean surface roughness $\mathrm{Sa}_{\mathrm{mean,t}}$ can be reduced to an average value of 3.38 μm. The percentage of defective surface areas can be lowered to an average of 3.2 %. Moreover, it takes 1.97 optimization steps on average to reach the optimum state in this setup.


Initial state (s_t)

Step	Reward	Action	Vol. Energy Density [VED]	Laser power P	Scan velocity v	Mean Sa (measured)	Mean Sa (predicted)	Predicted surface defects
0	-	6	37.5 J/mm³	150 W	1000 mm/s	13.04 µm	18.4 µm	0.06 %

Optimized state (s_{t+1})

Step	Reward	Action	Vol. Energy Density [VED]	Laser power P	Scan velocity v	Mean Sa (measured)	Mean Sa (predicted)	Predicted surface defects
1	-125	6	56.25 J/mm³	225 W	1000 mm/s	4.21 µm	12.66 µm	0.1 %

Optimized state (s_{t+2})

Step	Reward	Action	Vol. Energy Density [VED]	Laser power P	Scan velocity v	Mean Sa (measured)	Mean Sa (predicted)	Predicted surface defects
2	5901	6	112.5 J/mm³	300 W	667 mm/s	2.42 µm	3.5 µm	1.5 %

FIGURE 13. Example of an inter-layer roughness optimization sequence using MBRL and CNN-based image processing.

C. DISCUSSION

Since the raw HDR image of each surface must be interpreted in terms of the quality metrics to be optimized, the performance of CNN-based surface classification is of particular importance. Although the CNN architecture can be used for regression tasks and would therefore allow to predict the roughness values directly, the roughness values were first binned into specific roughness classes. There are two reasons for this. First, during development and testing it was shown that the regression models turned out to be unreliable compared to the classification models. This might be due to uncertainty in the roughness measurements and lack of precise alignment of the individual patches and the measurement position. The second reason is the integration of the class "surface distortion" which does not represent certain roughness values that would be required for building a regression model. Therefore, in this work, the roughness estimation problem was formulated as a classification task, which allows a rather coarse categorization of the investigated surfaces.

FIGURE 10 shows that the overall measurement trends are reproduced by the image-based roughness estimation. However, the increased standard deviations of the predicted variables show significant uncertainty in classification performance, which could be further improved by incorporating more training data from different layers. Nevertheless, the predicted roughness maps can be used by the MBRL algorithm to improve the previously mentioned surface quality metrics.

The dynamics function, which estimates the future state based on the current state and a given action, plays a key role. Although a Random Forest regressor is finally used in this work, different function approximators such as multi-layer perceptron, Gaussian Process regression and Support Vector Regression were tried during the development. Based on the results it is assumed that RF is especially suited to model the process dynamics due to the increased robustness based on the individual decision trees and the ability to handle non-linear parameters efficiently. The output of the RF regressor is used together with the reward function to select the most promising action sequences. An important parameter for estimating the expected rewards is the prediction horizon $T_{horizon}$. Experiments have shown that a planning horizon longer than two steps leads to decreased optimization performances. It is assumed that the uncertainty introduced by RF-based state prediction, combined with the uncertainty in roughness estimation via CNN, leads to unreliable estimates of expected future rewards that increase substantially with the length of the prediction horizon (i.e., propagation of uncertainty).

In this work, the agent's action space is restricted to seven actions and the state space is indirectly restricted as it depends on the experimental database. These restrictions might help to achieve faster convergence during training since fewer combinations from the state-action space need to be visited and learned. However, in real world scenarios, such restrictions could also be useful at the beginning of a LPBF build process when training data is rare, and a rough estimation of optimal process parameters would already be advantageous. As the amount of data increases, the state and action spaces should be dynamically increased to refine the outcome of the optimizer.

By optimizing the surface roughness, the proposed framework indirectly optimizes the component density as high surface roughness correlates with low component density and low VED according to FIGURE 10. This relationship makes sense from a physical point of view, since a low VED leads to insufficient fusion between the current and the previous component layer. The resulting inter-layer lack of fusion can lead to reduced component density due to resulting gas or material inclusions. In addition, increased fluctuations within the laser-powder interaction zone, which increasingly occurs at low VED, can simultaneously lead to high surface roughness.

Regarding the generalizability of the approach, the authors assume that the proposed technique can be applied to many other 3D printing devices and materials where layer-wise surface observation by optical imaging is possible. In some applications, the execution times could be more critical and therefore further optimization of the inference time should be considered. It is also assumed that the CNN model for roughness estimation and the RL agent, which is responsible for optimizing the process parameters, need to be updated based on new process data.

V. CONCLUSION

In the presented work, a new approach based on HDR imaging combined with CNNs and model-based RL for inter-layer quality optimization of LPBF processes is proposed. The targeted quality metrics in this work are the current layer's surface roughness as well as the percentage of distortion on the layer's surface. Although this is a preliminary study, the intermediate results indicate that the framework has the potential to be successfully applied in industrial LPBF processes. The following experimental results are encouraging to continue and improve the demonstrated concept:

1) Surface roughness classification based on optical imaging and deep neural networks outperforms a classical ML approach using statistical texture features under the same image resolution and dynamic range conditions by more than 20 % in F1-Score.
2) HDR imaging increases the classification accuracy by more than 12 % compared to its LDR counterpart. Experiments indicate that an image resolution of at least 5.66 μm/pixel is required for roughness classification accuracies greater than 80 %.
3) Based on measured surface roughness data, the negative correlation between surface roughness and volumetric energy density could be reproduced using the image-based roughness predictions.
4) Moreover, the experimental evaluation also supports the assumption that a low surface roughness correlates with high component densities.
5) For 21 unknown LPBF surfaces, the proposed MBRL approach finds optimal process parameters resulting in high rewards and a low surface roughness of 3.38 μm (average for 21 parts) obtained faster than with the Q-Learning reference implementation. At the same time, the MBRL effectively avoids actions that would result in a high percentage ($>$10 %) of predicted defective surfaces and thus be penalized by negative rewards.

While the outcome of this work appears promising, future work should address the real-time implementation of the proposed framework that enables the quality assessment and process optimization during the build of more complex components. To improve in-situ quality assessment, additional sensors, such as high-speed coaxial pyrometers or thermographic images, can be integrated to provide the RL algorithm with additional information for deriving optimal control decisions at each layer. In addition, further improvements to the framework can be expected if viable physical simulations are provided as training environments. With the help of transfer learning, the knowledge from physical simulation can be used to create high-performance optimization strategies that require less experimental data.

REFERENCES

[1] C. Y. Yap, C. K. Chua, Z. L. Dong, Z. H. Liu, D. Q. Zhang, L. E. Loh, and S. L. Sing, ''Review of selective laser melting: Materials and applications,'' *Appl. Phys. Rev.*, vol. 2, no. 4, p. 41101, 2015, doi: 10.1063/1.4935926.

[2] D. Türk, R. Kussmaul, M. Zogg, C. Klahn, A. B. Spierings, H. Könen, P. Ermanni, and M. Meboldt, ''Additive manufacturing with composites for integrated aircraft structures,'' in *Proc. Int. SAMPE Tech. Conf.*, 2016, pp. 1404–1418.

[3] W. W. Wits, S. J. Weitkamp, and J. van Es, ''Metal additive manufacturing of a high-pressure micro-pump,'' *Procedia CIRP*, vol. 7, pp. 252–257, Jan. 2013, doi: 10.1016/j.procir.2013.05.043.

[4] Y. Zhai, D. A. Lados, and J. L. LaGoy, ''Additive manufacturing: Making imagination the major limitation,'' *JOM*, vol. 66, no. 5, pp. 808–816, May 2014, doi: 10.1007/s11837-014-0886-2.

[5] G. Tapia and A. Elwany, ''A review on process monitoring and control in metal-based additive manufacturing,'' *J. Manuf. Sci. Eng.*, vol. 136, no. 6, pp. 060801-1–060801-10, Dec. 2014, doi: 10.1115/1.4028540.

[6] P. I. Stavroulakis and R. K. Leach, ''Invited review article: Review of post-process optical form metrology for industrial-grade metal additive manufactured parts,'' *Rev. Sci. Instrum.*, vol. 87, no. 4, p. 41101, 2016, doi: 10.1063/1.4944983.

[7] S. K. Everton, M. Hirsch, P. Stravroulakis, R. K. Leach, and A. T. Clare, ''Review of *in-situ* process monitoring and *in-situ* metrology for metal additive manufacturing,'' *Mater. Des.*, vol. 95, pp. 431–445, Apr. 2016, doi: 10.1016/j.matdes.2016.01.099.

[8] B. Whip, L. Sheridan, and J. Gockel, ''The effect of primary processing parameters on surface roughness in laser powder bed additive manufacturing,'' *Int. J. Adv. Manuf. Technol.*, vol. 103, nos. 9–12, pp. 4411–4422, Aug. 2019, doi: 10.1007/s00170-019-03716-z.

[9] E. O. Olakanmi, R. F. Cochrane, and K. W. Dalgarno, ''A review on selective laser sintering/melting (SLS/SLM) of aluminium alloy powders: Processing, microstructure, and properties,'' *Prog. Mater. Sci.*, vol. 74, pp. 401–477, Oct. 2015, doi: 10.1016/j.pmatsci.2015.03.002.

[10] K. Alrbaey, D. Wimpenny, R. Tosi, W. Manning, and A. Moroz, ''On optimization of surface roughness of selective laser melted stainless steel parts: A statistical study,'' *J. Mater. Eng. Perform.*, vol. 23, no. 6, pp. 2139–2148, Jun. 2014, doi: 10.1007/s11665-014-0993-9.

[11] D. Ahn, H. Kim, and S. Lee, ''Surface roughness prediction using measured data and interpolation in layered manufacturing,'' *J. Mater. Process. Technol.*, vol. 209, no. 2, pp. 664–671, Jan. 2009, doi: 10.1016/j.jmatprotec.2008.02.050.

[12] G. Strano, L. Hao, R. M. Everson, and K. E. Evans, ''Surface roughness analysis, modelling and prediction in selective laser melting,'' *J. Mater. Process. Technol.*, vol. 213, no. 4, pp. 589–597, Apr. 2013, doi: 10.1016/j.jmatprotec.2012.11.011.

[13] P. B. Bacchewar, S. K. Singhal, and P. M. Pandey, ''Statistical modelling and optimization of surface roughness in the selective laser sintering process,'' *Proc. Inst. Mech. Eng. B, J. Eng. Manuf.*, vol. 221, no. 1, pp. 35–52, Jan. 2007, doi: 10.1243/09544054JEM670.

[14] F. Cabanettes, A. Joubert, G. Chardon, V. Dumas, J. Rech, C. Grosjean, and Z. Dimkovski, ''Topography of as built surfaces generated in metal additive manufacturing: A multi scale analysis from form to roughness,'' *Precis. Eng.*, vol. 52, pp. 249–265, Apr. 2018, doi: 10.1016/j.precisioneng.2018.01.002.

[15] A. Majeed, A. Ahmed, A. Salam, and M. Z. Sheikh, ''Surface quality improvement by parameters analysis, optimization and heat treatment of AlSi10Mg parts manufactured by SLM additive manufacturing,'' *Int. J. Lightweight Mater. Manuf.*, vol. 2, no. 4, pp. 288–295, Dec. 2019, doi: 10.1016/j.ijlmm.2019.08.001.

[16] J. C. Snyder and K. A. Thole, ''Understanding laser powder bed fusion surface roughness,'' *J. Manuf. Sci. Eng.*, vol. 142, no. 7, pp. 071003-1–071003-15, Jul. 2020, doi: 10.1115/1.4046504.

[17] S. L. Sing and W. Y. Yeong, ''Laser powder bed fusion for metal additive manufacturing: Perspectives on recent developments,'' *Virtual Phys. Prototyping*, vol. 15, no. 3, pp. 359–370, Jul. 2020, doi: 10.1080/17452759.2020.1779999.

[18] P. J. DePond, G. Guss, S. Ly, N. P. Calta, D. Deane, S. Khairallah, and M. J. Matthews, ''*In situ* measurements of layer roughness during laser powder bed fusion additive manufacturing using low coherence scanning interferometry,'' *Mater. Des.*, vol. 154, pp. 347–359, Sep. 2018, doi: 10.1016/j.matdes.2018.05.050.

[19] J. Gockel, J. Beuth, and K. Taminger, ''Integrated control of solidification microstructure and melt pool dimensions in electron beam wire feed additive manufacturing of Ti-6Al-4 V,'' *Additive Manuf.*, vols. 1–4, pp. 119–126, Oct. 2014, doi: 10.1016/j.addma.2014.09.004.

[20] F. Caiazzo, V. Alfieri, and G. Casalino, ''On the relevance of volumetric energy density in the investigation of inconel 718 laser powder bed fusion,'' *Materials*, vol. 13, no. 3, p. 538, Jan. 2020, doi: 10.3390/ma13030538.

[21] A. Simchi and H. Pohl, ''Effects of laser sintering processing parameters on the microstructure and densification of iron powder,'' *Mater. Sci. Eng. A*, vol. 359, nos. 1–2, pp. 119–128, 2003, doi: 10.1016/S0921-5093(03)00341-1.

[22] U. S. Bertoli, A. J. Wolfer, M. J. Matthews, J.-P.-R. Delplanque, and J. M. Schoenung, ''On the limitations of volumetric energy density as a design parameter for selective laser melting,'' *Mater. Design*, vol. 113, pp. 331–340, Jan. 2017, doi: 10.1016/j.matdes.2016.10.037.

[23] E. Yasa, J. Deckers, and J. Kruth, ''The investigation of the influence of laser re-melting on density, surface quality and microstructure of selective laser melting parts,'' *Rapid Prototyping J.*, vol. 17, no. 5, pp. 312–327, Aug. 2011, doi: 10.1108/13552541111156450.

[24] W. Yu, S. L. Sing, C. K. Chua, and X. Tian, ''Influence of re-melting on surface roughness and porosity of AlSi10Mg parts fabricated by selective laser melting,'' *J. Alloys Compounds*, vol. 792, pp. 574–581, Jul. 2019, doi: 10.1016/j.jallcom.2019.04.017.

[25] A. J. Dunbar, E. R. Denlinger, J. Heigel, P. Michaleris, P. Guerrier, R. Martukanitz, and T. W. Simpson, ''Development of experimental method for *in situ* distortion and temperature measurements during the laser powder bed fusion additive manufacturing process,'' *Additive Manuf.*, vol. 12, pp. 25–30, Oct. 2016, doi: 10.1016/j.addma.2016.04.007.

[26] Ö. Poyraz and M. C. Kuşhan, ''Residual stress-induced distortions in laser powder bed additive manufacturing of nickel-based superalloys,'' *Strojniski Vestnik J. Mech. Eng.*, vol. 65, no. 6, pp. 343–350, Jun. 2019, doi: 10.5545/sv-jme.2019.6004.

[27] P. Mercelis and J. Kruth, ''Residual stresses in selective laser sintering and selective laser melting,'' *Rapid Prototyping J.*, vol. 12, no. 5, pp. 254–265, Oct. 2006, doi: 10.1108/13552540610707013.

[28] A. A. Akbari, A. M. Fard, and A. G. Chegini, ''An effective image based surface roughness estimation approach using neural network,'' in *Proc. World Autom. Congr.*, Budapest, Hungary, Jul. 2006, pp. 1–6.

[29] V. Akhil, G. Raghav, N. Arunachalam, and D. S. Srinivas, ''Image data-based surface texture characterization and prediction using machine learning approaches for additive manufacturing,'' *J. Comput. Inf. Sci. Eng.*, vol. 20, no. 2, pp. 021010-1–021010-16, Apr. 2020, doi: 10.1115/1.4045719.

[30] G. Samtaş, ''Measurement and evaluation of surface roughness based on optic system using image processing and artificial neural network,'' *Int. J. Adv. Manuf. Technol.*, vol. 73, nos. 1–4, pp. 353–364, Jul. 2014, doi: 10.1007/s00170-014-5828-1.

[31] H. Pan, P. Gao, H. Zhou, R. Ma, J. Yang, and X. Zhang, ''Roughness analysis of sea surface from visible images by texture,'' *IEEE Access*, vol. 8, pp. 46448–46458, 2020, doi: 10.1109/ACCESS.2020.2978638.

[32] A. Townsend, N. Senin, L. Blunt, R. K. Leach, and J. S. Taylor, ''Surface texture metrology for metal additive manufacturing: A review,'' *Precis. Eng.*, vol. 46, pp. 34–47, Oct. 2016, doi: 10.1016/j.precisioneng.2016.06.001.

[33] I. O. Aver'Yanova, D. Y. Bogomolov, and V. V. Poroshin, ''ISO 25178 standard for three-dimensional parametric assessment of surface texture,'' *Russian Eng. Res.*, vol. 37, no. 6, pp. 513–516, Jun. 2017, doi: 10.3103/S1068798X17060053.

[34] B. Bhushan, ''Surface roughness analysis and measurement techniques,'' in *Modern Tribology Handbook, Two Volume Set*, 1st ed. Boca Raton, FL, USA: CRC Press, Dec. 2000, p. 1760. [Online]. Available: https://www.taylorfrancis.com/books/edit/10.1201/9780849377877/modern-tribology-handbook-two-volume-set-bharat- bhushan?refId=2810b65d-664e-44fa-bbf9-bc1f131bbb50, doi: 10.1201/9780849377877.

[35] R. Kamguem, S. A. Tahan, and V. Songmene, ''Evaluation of machined part surface roughness using image texture gradient factor,'' *Int. J. Precis. Eng. Manuf.*, vol. 14, no. 2, pp. 183–190, Feb. 2013, doi: 10.1007/s12541-013-0026-x.

[36] W. Sun, B. Yao, B. Chen, Y. He, X. Cao, T. Zhou, and H. Liu, ''Noncontact surface roughness estimation using 2D complex wavelet enhanced ResNet for intelligent evaluation of milled metal surface quality,'' *Appl. Sci.*, vol. 8, no. 3, p. 381, Mar. 2018, doi: 10.3390/app8030381.

[37] A. Valikhani, A. J. Jahromi, S. Pouyanfar, I. M. Mantawy, and A. Azizinamini, ''Machine learning and image processing approaches for estimating concrete surface roughness using basic cameras,'' *Comput.-Aided Civil Infrastruct. Eng.*, vol. 36, no. 2, pp. 213–226, Feb. 2021, doi: 10.1111/mice.12605.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Oct. 2015, *arXiv:1512.03385*. [Online]. Available: http://arxiv.org/abs/1512.03385

[39] B. N. Narayanan, K. Beigh, S. Duning, and D. Erdahl, "Material identification and segmentation using deep learning for laser powder bed fusion," *Appl. Mach. Learn., Int. Soc. Opt. Photonics*, vol. 11511, 2020, Art. no. 115110P, doi: 10.1117/12.2567007.

[40] R. P. Borase *et al.*, "A review of PID control, tuning methods and applications," *Int. J. Dyn. Control*, 2020, 10.1007/s40435-020-00665-4.

[41] A. Fathi, A. Khajepour, E. Toyserkani, and M. Durali, "Clad height control in laser solid freeform fabrication using a feedforward PID controller," *Int. J. Adv. Manuf. Technol.*, vol. 35, nos. 3–4, pp. 280–292, Dec. 2007, doi: 10.1007/s00170-006-0721-1.

[42] C. Doumanidis and Y.-M. Kwak, "Geometry modeling and control by infrared and laser sensing in thermal manufacturing with material deposition," *J. Manuf. Sci. Eng.*, vol. 123, no. 1, pp. 45–52, Feb. 2001, doi: 10.1115/1.1344898.

[43] B. S. Taysom, C. D. Sorensen, and J. D. Hedengren, "A comparison of model predictive control and PID temperature control in friction stir welding," *J. Manuf. Processes*, vol. 29, pp. 232–241, Oct. 2017, doi: 10.1016/j.jmapro.2017.07.015.

[44] Y. K. Liu and Y. M. Zhang, "Model-based predictive control of weld penetration in gas tungsten arc welding," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 3, pp. 955–966, May 2014, doi: 10.1109/TCST.2013.2266662.

[45] A. Bollig, D. Abel, C. Kratzsch, and S. Kaierle, "Identification and predictive control of laser beam welding using neural networks," in *Proc. Eur. Control Conf. (ECC)*, Cambridge, U.K., Sep. 2003, pp. 2457–2462.

[46] P. Escárate, J. C. Agüero, S. Zúñiga, M. Castro, and J. Garcés, "Linear quadratic regulator for laser beam shaping," *Opt. Lasers Eng.*, vol. 94, pp. 90–96, Jul. 2017, doi: 10.1016/j.optlaseng.2017.02.009.

[47] L. Tang, J. Ruan, T. E. Sparks, R. G. Landers, and F. Liou, "Layer-to-layer height control of laser metal deposition processes," in *Proc. Amer. Control Conf.*, St. Louis, MO, USA, Jun. 2009, pp. 5582–5587.

[48] X. Cao and B. Ayalew, "Multivariable predictive control of laser-aided powder deposition processes," in *Proc. Amer. Control Conf. (ACC)*, Chicago, IL, USA, Jul. 2015, pp. 3625–3630.

[49] C. Yu and J. Jiang, "A perspective on using machine learning in 3D bioprinting," *Int. J. Bioprinting*, vol. 6, no. 1, p. 253, Jan. 2020, doi: 10.18063/ijb.v6i1.253.

[50] F. W. Baumann, A. Sekulla, M. Hassler, B. Himpel, and M. Pfeil, "Trends of machine learning in additive manufacturing," *Int. J. Rapid Manuf.*, vol. 7, no. 4, p. 310, 2018, doi: 10.1504/IJRAPIDM.2018.095788.

[51] C. Wang, X. P. Tan, S. B. Tor, and C. S. Lim, "Machine learning in additive manufacturing: State-of-the-art and perspectives," *Additive Manuf.*, vol. 36, Dec. 2020, Art. no. 101538, doi: 10.1016/j.addma.2020.101538.

[52] G. D. Goh, S. L. Sing, and W. Y. Yeong, "A review on machine learning in 3D printing: Applications, potential, and challenges," *Artif. Intell. Rev.*, vol. 54, no. 1, pp. 63–94, Jan. 2021, doi: 10.1007/s10462-020-09876-9.

[53] Z. Gan, H. Li, S. J. Wolff, J. L. Bennett, G. Hyatt, G. J. Wagner, J. Cao, and W. K. Liu, "Data-driven microstructure and microhardness design in additive manufacturing using a self-organizing map," *Engineering*, vol. 5, no. 4, pp. 730–735, Aug. 2019, doi: 10.1016/j.eng.2019.03.014.

[54] R. S. Sutton, F. Bach, and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[55] Q. Xiao, C. Li, Y. Tang, and L. Li, "Meta-reinforcement learning of machining parameters for energy-efficient process control of flexible turning operations," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 1, pp. 5–18, Jan. 2019, doi: 10.1109/TASE.2019.2924444.

[56] G. Masinelli, T. Le-Quang, S. Zanoli, K. Wasmer, and S. A. Shevchik, "Adaptive laser welding control: A reinforcement learning approach," *IEEE Access*, vol. 8, pp. 103803–103814, 2020, doi: 10.1109/ACCESS.2020.2998052.

[57] A. G. Dharmawan, Y. Xiong, S. Foong, and G. S. Soh, "A model-based reinforcement learning and correction framework for process control of robotic wire arc additive manufacturing," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2020, pp. 4030–4036.

[58] P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," in *Proc. 24th Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, 1997, pp. 369–378.

[59] G. Bradski, "The OpenCV library," *Dr. Dobb's J. Softw. Tools*, 2000.

[60] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 3, pp. 328–339, Mar. 1989, doi: 10.1109/29.21701.

[61] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, vol. 2, 2011, pp. 1237–1242.

[62] K. B. Lee, S. Cheon, and C. O. Kim, "A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes," *IEEE Trans. Semicond. Manuf.*, vol. 30, no. 2, pp. 135–142, May 2017, doi: 10.1109/TSM.2017.2676245.

[63] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015, doi: 10.1016/j.neunet.2014.09.003.

[64] B. N. Narayanan, K. Beigh, G. Loughnane, and N. U. Powar, "Support vector machine and convolutional neural network based approaches for defect detection in fused filament fabrication," *Appl. Mach. Learn., Int. Soc. Opt. Photon.*, vol. 11139, Sep. 2019, Art. no. 1113913.

[65] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[66] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[67] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[68] A. Martín. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Accessed: Feb. 15, 2021. [Online]. Available: https://www.tensorflow.org/

[69] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 7559–7566.

[70] A. S. Polydoros and L. Nalpantidis, "Survey of model-based reinforcement learning: Applications on robotics," *J. Intell. Robotic Syst.*, vol. 86, no. 2, pp. 153–173, May 2017, doi: 10.1007/s10846-017-0468-y.

[71] A. Rao, "A survey of numerical methods for optimal control," *Adv. Astron. Sci.*, vol. 135, pp. 497–528, Aug. 2010.

[72] C. E. García, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice—A survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989, doi: 10.1016/0005-1098(89)90002-2.

[73] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[74] A. Liaw and M. Wiener, "Classification and regression by randomForest," *Forest*, vol. 23, pp. 18–22, Dec. 2001.

[75] D. Wu, C. Jennings, J. Terpenny, R. X. Gao, and S. Kumara, "A comparative study on machine learning algorithms for smart manufacturing: Tool wear prediction using random forests," *J. Manuf. Sci. Eng.*, vol. 139, no. 7, pp. 071018-1–071018-9, 2017.

[76] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random forests and decision trees," *Int. J. Comput. Sci. Issues*, vol. 9, p. 272, Sep. 2012.

CHRISTIAN KNAAK (Member, IEEE) received the B.Eng. degree in mechanical engineering from the University of Applied Sciences, Aachen, Germany, in 2013, and the M.Sc. degree in mechanical engineering from RWTH Aachen University, Germany, in 2016. He is currently pursuing the Ph.D. degree in mechanical engineering.

Since 2016, he has been working as a Research Assistant with the Process Control and Systems Technology Group, Fraunhofer Institute for Laser Technology ILT, Aachen. His research interests include image processing and machine learning for process control, with a special focus on autonomous laser processing systems. His competencies include real-time process monitoring using camera systems and the application of advanced data analytics in laser processing.

LUKAS MASSELING received the M.Sc. degree in mechanical engineering from Ruhr-Universität Bochum, Germany, in 2015. He is currently pursuing the Ph.D. degree in mechanical engineering with the Fraunhofer Institute for Laser Technology ILT, Aachen, Germany.

From 2015 to 2021, he was a Research Assistant with the Fraunhofer Institute for Laser Technology ILT. Since 2019, he has been the Co-Founder and the CTO of Aixway 3D, Aachen. His research interest includes the development for LPBF, in particular the development of the μ-LPBF process for the production of particularly small and high-resolution components.

EMIL DUONG received the master's degree in physics from RWTH Aachen University, Germany, in 2016. He is currently pursuing the Ph.D. degree in mechanical engineering with the RWTH Aachen Chair for Laser Technology LLT. Since 2016, he has been a Research Assistant with the Fraunhofer Institute for Laser Technology ILT, Aachen. His research interests include the development of process monitoring systems for laser powder bed fusion process to detect defects using a sensor fusion approach and the development of inline-topography sensors based on the principle of conoscopic holography.

PETER ABELS studied mechanical engineering at the University of Applied Sciences, Aachen. In 1986, he started as a Researcher with the Fraunhofer Institute for Laser Technology ILT, where he is currently the Leader of the Process Control and System Technology Group. He worked initially with the Laser Development Department, later on he changed to the Department of Laser Material Processing, especially in the section of cutting with CO_2 lasers. He co-founded the initiative for process monitoring and control in the Fraunhofer ILT. He has more than 20 years' experience in this field and is the Leader of the Interdisciplinary Research Group. His group is active in industrial contract research and in national and international public funded projects.

ARNOLD GILLNER studied physics at the University of Darmstadt. He received the Ph.D. degree in mechanical engineering from RWTH Aachen, in 1994.

Since 1985, he has been working as the Scientist of the Fraunhofer-Institute for Laser Technology. Starting in 1992, he developed the Department for Micro Technology, ILT, where he has been the Head of the Department of Ablation and Joining, since 2010. Together with more than 55 scientists, he is developing industrial laser processes for macro and micro joining, packaging, micro and nano structuring, polymer applications, and life science applications. He is the Head of the Board of the Aachen Competence Center for Medical Technology and the Head of the Advisory Board of LifeTec Aachen Jü lich. He has published more than 150 articles and book chapters in scientific journals and other scientific contributions and holds more than 20 patents on laser processes.