Deep representation learning for process variation management in laser powder bed fusion

Sepehr Fathizadan\textsuperscript{a}, Feng Ju\textsuperscript{a,*,b}, Yan Lu\textsuperscript{b,1}

\textsuperscript{a} School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85281 USA
\textsuperscript{b} National Institute of Standard and Technology, Engineering Laboratory, Gaithersburg, MD 20899, USA

\textbf{A R T I C L E I N F O}

\textbf{Keywords:}
Laser powder bed fusion
Additive manufacturing
Melt pool image
Deep learning
Anomaly detection

\textbf{A B S T R A C T}

Laser Powder Bed Fusion (LPBF) is an additive manufacturing process where laser power is applied to fuse the spread powder and fabricate industrial parts in a layer by layer fashion. Despite its great promise in fabrication flexibility, print quality has long been a major barrier for its widespread implementation. Traditional offline post-manufacturing inspections to detect the defects in finished products are expensive and time-consuming and thus cannot be applied in real-time monitoring and control. In-situ monitoring methods by relying on the in-process sensor data, on the other hand, can provide viable alternatives to aid with the online detection of anomalies during the process. Given the crucial importance of melt pool characteristics to the quality of final products, this paper provides a framework to process the melt pool images by a configuration of Convolutional Auto-Encoder (CAE) neural networks. The network’s corresponding bottleneck layer learns a deep yet low-dimensional representation from melt pools while preserving the spatial correlation and complex features intrinsic in the images. As opposed to the manual annotation of data by X-ray imaging or destructive tests, an agglomerative clustering algorithm is applied to these representations to automatically extract the anomalies and annotate the data accordingly. A control charting scheme based on Hotelling’s $T^2$ and $S^2$ statistics is then developed to monitor the process’s stability by keeping track of the learned representations and residuals obtained from the reconstruction of original images. Testing the proposed methodology on the collected data from an experimental build demonstrates that the method can extract a set of complex features that are inextricable otherwise by using hand-crafted feature engineering methods. Moreover, through extensive numerical studies, it is shown that the proposed feature extraction and statistical process monitoring scheme is capable of detecting the anomalies in real-time with accuracy and $F_1$ score of about 95% and 82%, respectively.

\section{Introduction}

Additive manufacturing consists of successive fabrication of materials layer upon layer to manufacture three-dimensional items. There are diverse categories of additive manufacturing processes, including Sheet Lamination Processes (SLP), Binder Jetting Processes (BJP), Extrusion-Based processes (EBP), Material Jetting Processes (MJP), Photo-polymer Vat Processes (PVP), Powder Bed Fusion (PBF), and Directed Energy Deposition (DED) \cite{1-3}. In a variation of PBF called Laser Powder Bed Fusion (LPBF), laser power is utilized to melt and fuse the spread powder on a build plate. An LPBF machine can precisely scan thin layers of powder to form a designed geometry. The laser delivers thermal energy to the powder to create melt pools once it reaches the melting temperature. Developments of porosity and keyholing due to lack of fusion and over melting phenomena, respectively, are two key factors that significantly deteriorate the material properties of finished products \cite{4-6}. Moreover, it has been proven through extensive studies that the different morphological characteristics of the melt pool region are closely correlated to the formation of such defects on parts \cite{7-9}. The formation of melt pool regions can get affected by several exogenous factors such as fluid dynamics, thermal conduction, and preheating temperature. Nonetheless, the process parameter settings play a primary role in controlling the process and forestalling defects and anomalies. Laser power, scan velocity, and hatching distance are the primary energy density related variables that can be leveraged to alter the formation of the melt pool \cite{10,11}.

\textsuperscript{1} Corresponding author.
\textsuperscript{*} E-mail addresses: Sepehr.Fathi@asu.edu (S. Fathizadan), Feng.Ju@asu.edu (F. Ju), yan.lu@nist.gov (Y. Lu).
\textsuperscript{1} This work is supported by Department of Commerce under grant 70NANB20H194.

https://doi.org/10.1016/j.addma.2021.101961

Received 11 November 2020; Received in revised form 29 January 2021; Accepted 15 March 2021

Available online 5 April 2021

2214-8604/© 2021 Elsevier B.V. All rights reserved.
The primary source of investigating the melt pool is captured through acquisition of thermal or optical melt pool images taken in-situ from the laser-powder interface when the laser beam is applied on the powder bed. In general, two post-build approaches correlate a melt pool image to either a normal or defective location/spot on the part: (1) non-destructive inspection by CT-scan X-ray imaging, and (2) destructive inspection [12,13]. Due to the large number of images, obtaining these labels for melt pool images with either approach comes at a great price. The results, in addition, suffer significantly from low accuracy. One can utilize the fact that oversized, small, splashed, or deformed melt pools indicate morphological signatures of defects in parts. Therefore, the objective could be redirected toward detecting the occurrence of such anomalies in the melt pool images. This gives rise to a cheaper yet effective alternative to the manual annotation of data by striving to annotate the data based on the proximity in their corresponding structural properties. The structural properties of the data are captured through a set of high-quality features from samples.

The state-of-the-art methods often rely on oversimplified and hand-crafted features of the melt pool to analyze the images [10,14]. These methods fail to preserve the spatial correlation within images and lose the valuable and intricate information inside each sample. This paper provides a framework to process the melt pool images by a configuration of Convolutional Auto-Encoder (CAE) neural networks. The network’s corresponding bottleneck layer learns a deep low-dimensional representation from melt pools while preserving the spatial correlation and complex features intrinsic in the images. As opposed to the manual annotation of data by X-ray imaging or destructive tests, an agglomerative clustering algorithm is applied to these representations to automatically extract the anomalies and annotate the data accordingly. A control charting scheme based on Hotelling’s $T^2$ and $S^2$ statistics is then developed to monitor the process’s stability by keeping track of the learned representations and residuals obtained from the reconstruction of original images. An experimental build using a spiral concentrating scan pattern with varying laser power was conducted on the Additive Manufacturing Metrology Test bed at the National Institute of Standards and Technology (NIST) [15]. Testing the proposed methodology on the collected data demonstrates that the method can extract a set of complex features that are inextricable by using hand-crafted feature engineering methods. Finally, through extensive numerical studies, it is shown that the proposed feature extraction and statistical process monitoring scheme is capable of detecting the anomalies in real-time with relatively high accuracy.

In what follows, we first discuss an overview of relevant studies in Section 2. The proposed pipeline to process and analyze the melt pool images, including convolutional auto-encoder for deep representation learning, agglomerative clustering for data annotation, and statistical process control for anomaly detection, will be explained in Section 3. The proposed methods will be evaluated by an experimental build. The experiment settings and the numerical results will be provided in Section 4. Finally, conclusions and future directions are outlined in Section 5.

2. Literature review

Overall, the literature in characterizing and analyzing melt pool images is divided into three different categories: parametric modeling, non-parametric modeling, and deep learning. Concerning parametric analysis, Yang et al. predict the melt pool area by considering laser power, scan velocity, and tool-path on multiple track geometry as the predictor variables [10]. First, different rectangular parts with different path strategies are printed. Using different thresholding values on melt pool images, the area of the fitted ellipse is extracted. Finally, taking the spatio-temporal effects of neighboring melt locations into account, a polynomial regression model named “neighboring-effect modeling method” (NBEM) is developed and fitted on the data. In a later study, Yang et al. develop a new model called Layer-wise NBEM (L-NBEM) by further including the scan and exposure settings data from the previous layers [10]. Unlike previous work, a fully-connected neural network is trained on the data with a new set of features to predict the melt pool area. Given the spatio-temporal interdependence of data, Guo et al. develop a framework to model and monitor the melt pool [16]. Following the general linear mixed models, a Spatio-Temporal Conditional Auto-Regressive (STCAR) model is deployed to account for both spatial correlations among pixels in an image and temporal correlations across images. A Statistical Process Control (SPC) scheme based on Hotelling $T^2$ statistics is then used to monitor the estimated parameters from the model over time.

Several studies characterize the data by non-parametric modeling approaches. Khanzadeh et al. propose a framework to predict (detect) porosity based on thermal images from the melt pool [17]. They characterize the melt pool images by transforming the data into a spherical coordinate system and applying bi-harmonic surface interpolation afterward to smoother and extract the vector of features. These vectors are later put into Self-Organizing Maps (SOM) to cluster each image to decide whether it is normal or porous. A CT-scan X-ray imaging system capable of accurately identifying the size and locations of pores is used to benchmark the model’s performance. A very similar work of study is done by Khanzadeh et al. [18], where instead of thermal, the morphological characteristics of the melt pool image are extracted and used for analysis. Specifically, the data points belonging to the melt pool boundary are transformed into the polar coordinate system, and cubic spline is used to serve the purpose of smoothing and feature extraction. Another similar example of research conducted by Khanzadeh et al., uses Functional Principal Component Analysis (FPCA) to obtain the morphological features from melt pool images and classifies each image as normal or porous using different classification algorithms such as Support Vector Machines (SVM) and Decision Trees (DT) [19]. In another effort to preserve the spatio-temporal characteristics of the melt pool images, Khanzadeh et al. develop an SPC approach using Hotelling $T^2$ and $Q$ charts, that monitors the extracted features from the decomposition of tensor data [20]. The obtained accuracy and $F_1$ score measures from this approach are highly volatile due to the large standard deviations of the results.

Following the same stream of research, Grasso et al. unfold the tensor of streams of optical melt pool images to obtain a vectorized form for each pixel across the layers [21]. Using Vectorized Principal Component Analysis (VPCA), a set of weights and Hotelling’s $T^2$ distances are assigned to each frame and each pixel, respectively. Finally, the vanilla $k$-means clustering is utilized to monitor the process. Grasso et al. perform several image processing tasks, including image segmentation by Otsu and contour detection, to extract the melt pools from infrared images [22]. The statistical descriptors to be controlled are the average thermal intensity of all retrieved pixels and the melt pool area defined based on the number of pixels in the detected contour. A bi-variate control charting scheme based on Hotelling $T^2$ is then developed to monitor the melt pool’s stability over time. Gobert et al. conduct another study with the aim of binary classification of voxels from layerwise optical images collected during build-time [12]. Feature extraction is done by applying 3D convolution filters on the data from multiple modalities. On the other hand, the $x$, $y$, $z$ locations of porosity are detected by the convolution of a Gaussian kernel with CT images. Finally, the outputs from single classifiers trained on each modality are concatenated to vote for the porosity decision. The best resultant approximate accuracy and $F_1$ score are reported to be about 85% and 62%, respectively.

As discussed above, the state-of-the-art methods often rely on handcrafted feature engineering methods to characterize the melt pool area. For instance, a few morphological or thermal attributes can be derived and considered as features. A few methods make an effort to take the spatio-temporal characteristics of the process into account. After the feature engineering phase, statistical monitoring methods or different machine learning tasks such as clustering or classifications can be defined to monitor the process’s stability and detect/predict...
anomalies. Recently, another field of data-driven modeling has emerged, which can potentially surpass the previous methods that rely on humans to extract the features. This is realized by interfacing the feature extraction with learning mechanism [23]. Thus, more smart models are developed that can automatically extract and incorporate high-quality features and therefore gain superior performance over machine learning methods.

The employment of deep learning methods within the additive manufacturing domain is still in a very preliminary stage. Among the handful of studies, Kwon et al. utilize the Convolutional Neural Networks (CNN) to build a regression model to predict the laser power values from melt pool images [24]. This method accounts for spatial correlation within the images but fails to address the temporal correlation from melt pool images [24]. This method accounts for spatial matching to process the cross-sectioning and CT images to extract the set of image processing steps such as Otsu thresholding and feature extractions [23] to build a regression model to predict the laser power values. Thus, more smart features and therefore gain superior performance over machine learning methods.

Recently, another field of data-driven modeling has emerged, which can potentially surpass the previous methods that rely on humans to extract the features. This is realized by interfacing the feature extraction with learning a deep representation mechanism. In doing so, this work offers much broader flexibility in further analysis of the data and anomaly detection than crude classification or regression tasks pursued in the existing studies.

3. Methodology

The framework of the proposed methodology is illustrated in Fig. 1. The pre-processed tensor of melt pool images are fed to the CAE, and the bottleneck tensor from latent space is extracted. This encoded deep representation is then flattened and deployed by a method of agglomerative clustering to annotate the data as either normal or anomaly. In the next step, the CAE is retrained on the normal samples, and the new set of extracted encoded vectors are used to construct the phase-I of the proposed multivariate control charting method. The deep representations of the set of previously discovered anomalies are then employed by the phase-II control charting to evaluate the method’s performance in terms of anomaly detection. The three constituents of the framework, including CAE neural networks to process the melt pool images, agglomerative clustering algorithm to annotate the data, and the control charting scheme to monitor the process’s stability and perform anomaly detection, are delineated by the next subsections, respectively.

3.1. Deep representation learning for feature extraction from melt pool images

Assume a set of \( N \) gray-scale images each with a dimension of \( d_1 \times d_2 \) and let’s denote the flattened vector of \( i^{th} \) image by \( \tilde{x}_i \in \mathbb{R}^{1 \times d} \), for \( i = 1, 2, \ldots, N \), where \( P = d_1 \times d_2 \). The vector of weights associated with each neuron \( j \) in the hidden layer is represented by \( \tilde{a}_j = [a_{j1}, a_{j2}, \ldots, a_{jP}] \), for \( j = 1, 2, \ldots, M \), where \( M \) is the total number of neurons in the hidden layer. Furthermore, let \( \tilde{\beta}_k = [\beta_{k1}, \beta_{k2}, \ldots, \beta_{kM}] \) be the vector of weights associated with each neuron \( k \) in the final output layer, for \( k = 1, 2, \ldots, K \), where \( K \) is the total number of neurons in the hidden layer or equivalently the number of classes. The output of each output unit \( k \) for the input \( \tilde{x}_i \), \( y_{ik} \), is the probability of assigning each input vector \( \tilde{x}_i \) to class \( k \). The model utilizes the outputs \( y_{ik} \) to benchmark its performance against \( y_{ik} \), the ground truth for the sample inputs. The criteria for benchmarking is given by a loss function, which is pursued to minimize the probability of misclassifying the samples. Table 1 provides the expression forms of the minimization of two common loss functions, mean squared error and cross entropy.

Table 1

<table>
<thead>
<tr>
<th>Loss function</th>
<th>Mean squared error</th>
<th>Cross entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expression form</td>
<td>( \min_{x,y} \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} (y_{ik} - \hat{y}_{ik})^2 )</td>
<td>( \min_{x,y} \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \ln(y_{ik}) )</td>
</tr>
</tbody>
</table>

![Fig. 1. Framework of proposed deep representation learning for anomaly detection methodology.](image-url)
Stochastic gradient descent is used to train the neural network on large data sets. Without the loss of generality, consider the loss function after dropping the $1/N$. Denoted by $\mathcal{L}(x; a, \beta)$, it is comprised of sum of errors for each sample, i.e. $\mathcal{L}(x; a, \beta) = \sum_{i=1}^{N} \mathcal{L}(x_i; a, \beta)$, where $\mathcal{L}(x_i; a, \beta)$ is equal to $\frac{1}{2}(y_i - y_{\hat{i}})^2$ or $\sum_{k=0}^{K} y_{ik}\ln(y_{ik})$ in case of mean squared error or cross entropy, respectively. Then, given the weights at time $t$, the updated weights at time $t + 1$ can be found by moving in the direction of the negative gradient. The size of the movement is a small non-negative number called learning rate, and is denoted by $\gamma$. Specifically, $\theta_t$ and $\theta\hat{t}_k$ updates are weights as follows:

$$
\begin{align*}
\frac{\partial \mathcal{L}(x_i; a, \beta)}{\partial \theta_1} & = -\gamma \frac{\partial \mathcal{L}(x_i; a, \beta)}{\partial \theta_1} \\
\frac{\partial \mathcal{L}(x_i; a, \beta)}{\partial \theta_2} & = -\gamma \frac{\partial \mathcal{L}(x_i; a, \beta)}{\partial \theta_2} \\
\frac{\partial \mathcal{L}(x_i; a, \beta)}{\partial \theta_3} & = -\gamma \frac{\partial \mathcal{L}(x_i; a, \beta)}{\partial \theta_3}
\end{align*}
$$

The value of partial derivatives of loss of each sample with respect to $a_\theta$ and $b_\theta$ is computed by the chain rule for derivatives and error backpropagation. The process can be iterated until convergence. Adaptive learning rate and momentum such as “Adam” [25], “Adadelta” [26], and “RMSprop” [27] is used to avoid a local minima. Finally, as the number of samples is relatively large in practice, a batch of samples is used to update the parameters as opposed to one sample at a time procedure in Eq. (1).

The explosion in the number of weights as image data gets large in dimension can lead to excessive computational overheads. Furthermore, there is also a loss in spatial information when images are flattened for the feed forwarding process. As a remedy, Convolution Neural Networks are developed to locally connect the features on the input image to the nodes inside the hidden layers. CNNs consist primarily of three different types of layers: convolutions, pooling, and fully connected layers. Fully connected layers are discussed above, so the details for convolutional and pooling layers are given next. A matrix known as a kernel is slid over the input matrix in the convolutional layers to create a feature map by performing convolution operation [23]. Let $K$ and $S$ be the $f_1 \times f_2$ kernel and feature map matrix, respectively. The feature map can be obtained following the 2-D convolution operation denoted by $\ast$ on the two-dimensional image $X$ as follows:

$$
S(i,j) = (X \ast K)(i,j) = \sum_{m=0}^{l-1} \sum_{n=0}^{l-1} X(i+m, j+n) K(m,n).
$$

Then, a non-linear activation function $h$ such as sigmoid function (σ), hyperbolic tangent (tanh), or Rectified Linear Unit (ReLU) [28] is applied to the feature map to obtain the final output, i.e., $h(S)$. In practice, more than one kernel is applied to the images so as to extract different types of features. The number of kernels is called filter or kernel depth. Therefore, the final feature map forms a 3D tensor composed of several feature map matrices.

Next, a pooling layer is applied to the feature map produced by the previous layer to downsample the data while preserving spatial invariance. The notion behind pooling layers is relatively straightforward. Max pooling, the most common type of pooling, is to simply take the maximum value from a given array or matrix. In this case, the feature map matrix is partitioned into $w \times u$ submatrices, and the maximum values inside each submatrix are taken to form a new matrix denoted by $D$:

$$
D(i,j) = \max\{S(i+m, j+n)|m \in [0, w], n \in [0, u]\}.
$$

It is worth noting that the convolutions and max-pooling layers are stacked on top of each other indefinitely to gain better results in practice.

Finally, the structure behind the proposed CAE can be demystified as follows. Specifically, Auto-Encoders are a method of unsupervised learning that learn a compact representation of the unlabeled input data. As the name suggests, it is composed of two components: an encoder and a decoder. The encoder $f$ learns a mapping from input data $X$ to a compact latent space $H$, i.e. $f: X \rightarrow H$, and the decoder $g$ that learns a mapping back from latent space to a reconstructed version of original input data $\hat{X}$, i.e. $g: H \rightarrow \hat{X}$. Remember from above that the ground truth for each sample is represented by $Y = [y_1, y_2, \ldots, y_K]$, where $y_k = 1$, if sample $i$ belongs to the class $k$ and the rest are simply zero. The Auto-Encoders have exactly the same structure and training process as fully connected neural networks with the difference that $\hat{Y}_i$ is simply replaced with $\hat{X}_i$. In the case of CAEs, just as similar to CNNs, the data are not flattened, and therefore, the input and output matrices are mapped to each other while the spatial correlation is preserved. The layer which holds the data in the latent space is called the bottleneck layer. The bottleneck layer outputs an encoded tensor of customized dimensions for each $X_i$ denoted by $H_i$. The model also has several hyper-parameters, including the number of layers, kernel’s and pooling’s size and depth, that need to be tuned based on cross-validation or test data, as will be discussed in Subsection 4.2.

3.2. Anomaly detection in melt pool images by clustering

The final learned representation vector can be fed to other methods of unsupervised or supervised learning for further analysis. As discussed before, manual annotation of melt pool images is time-consuming and expensive. A variation of clustering algorithms is employed to automatically annotate the melt pool images as either normal or anomaly to tackle this challenge. Anomalies are usually defined as individuals or groups of samples isolated from normal samples by their distinguishable characteristics. Agglomerative clustering tends to merge the samples into clusters in a hierarchical fashion. At the beginning of the process, each sample is a cluster of its own. The children clusters are then sequentially combined by the shortest cophenetic distance (linkage) into larger clusters until all elements end up being in the same parent cluster. An immediate intuition from this procedure is that normal samples tend to merge faster and with shorter cophenetic distances while appending the anomaly samples are postponed until the end of the process and when cophenetic distances between clusters are large. Furthermore, anomalies tend to form singleton clusters of their own, while normal samples are grouped together in clusters with large populations. These characteristics make the agglomerative clustering a perfect candidate for the anomaly detection, and thus automatic annotation of melt pool deep representation flattened vectors.

The definition of linkage is what differentiates between the different agglomerative clustering methods. The most widely used linkage functions include single, complete, average, and Ward. Denote the vectors of flattened encoded tensors by $\vec{h}_i \in \mathbb{R}^{1 \times Q}$, for $i = 1, 2, \ldots, N$, where $Q$ is the product of dimensions of the corresponding tensors. Then, the following linkage functions can be defined:

$$
l_{\text{single}}(C_i, C_j) = \min\{|\text{dist}(\vec{h}_i, \vec{h}_j)| \mid \vec{h}_i \in C_i; \vec{h}_j \in C_j\},
$$

$$
l_{\text{complete}}(C_i, C_j) = \max\{|\text{dist}(\vec{h}_i, \vec{h}_j)| \mid \vec{h}_i \in C_i; \vec{h}_j \in C_j\},
$$

$$
l_{\text{average}}(C_i, C_j) = \frac{1}{|C_i| |C_j|} \sum_{\vec{h}_i \in C_i} \sum_{\vec{h}_j \in C_j} \text{dist}(\vec{h}_i, \vec{h}_j),
$$

where $\text{dist}(\vec{h}_i, \vec{h}_j)$ is the distance between two vectors $\vec{h}_i$ and $\vec{h}_j$.
\[ l_{\text{ward}}(C_i, C_j) = \sqrt{\frac{2|C_i||C_j|}{|C_i| + |C_j|}} \text{dist}(C_i, C_j), \]  

where \( l_{\text{ward}}(C_i, C_j) \) denotes the type \( k \) linkage between two cluster \( C_i \) and \( C_j \), \(|C_i|\) is the cardinality of cluster \( C_i \), \( C_i \) is the centroid of cluster \( C_i \) and is computed as \( 1/|C_i| \sum_{h_i \in C_i} h_i \). The measure of proximity or similarity between two flattened encoded vectors \( \vec{h}_i \) and \( \vec{h}_j \) is denoted by \( \text{dist}(\vec{h}_i, \vec{h}_j) \). The proximity measure also varies in type and include cosine similarity, Manhattan or Euclidean distance. For the sake of consistency in computations between different linkage types, Euclidean distance is used as the primary measure of proximity in this study, i.e. \( \text{dist}(h_i, h_j) = \| h_i - h_j \|_2 \). The steps of the agglomerative clustering algorithm are summarized in Algorithm 1. The outputs of the process are summarized by means of a visualization tool called dendrogram, where the hierarchical structure of parent and child clusters are demonstrated by u-shape links. The height of the links in the dendrogram determine the cophenetic distance between two clusters, which can be used to decide for the best number of clusters and linkage types visually.

**Algorithm 1. Hierarchical agglomerative for clustering melt pool images.**

1. **Input:** The set of flattened encoded vectors \( \{ \vec{h}_1, \vec{h}_2, \ldots, \vec{h}_n \} \)
2. Type \( k \) linkage function \( l_{\text{ward}}(C_i, C_j); \)
3. For \( i = 1, \ldots, N \) do // loop over the data
4. \( C_i = \{ \vec{h}_i \}; \)
   // each object is the cluster of its own
5. End for
6. \( C = \{ C_1, C_2, \ldots, C_i \}; \)
   // super set includes all clusters
7. While \( |C| > 1 \) do // loop until super set has only one object
8. \( C_{x+y} = \arg \min |l_{\text{ward}}(C_x, C_y)| C_x \subseteq C \subseteq C; \)
   // pick best clusters to merge
9. \( C = (C_x \cup C_y); \)
10. \( C = C \cup (C_{x+y}); \)
   // add their union to the super set
11. End while

### 3.3. Statistical process monitoring for anomaly detection

As the last step, a multivariate profile monitoring scheme is deployed to perform control charting on the flattened encoded vectors. To this aim, the anomaly images detected by the discussed methods are labeled as out-of-control samples and are discarded out of the phase-I statistical process monitoring where the parameters are estimated, and control limits are built based on normal samples. The anomaly vectors are later employed to evaluate the out-of-control performance of the control charting method. The multivariate profile monitoring scheme consists of a Hotelling’s \( T^2 \) control chart to monitor the deep representation vectors and a Shewhart control chart for monitoring error variance the of residuals \( \epsilon \) denoted by \( \sigma^2 \). The first step toward designing the control chart method in phase-I analysis, involves estimating the mean vector, variance-covariance matrix and upper control limits. Denoting the \( (N_1 - 1) \times Q \) successive difference matrix by \( V \), it can be calculated as follows:

\[ V = \begin{bmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vdots \\ \vec{v}_{N_1 - 1} \end{bmatrix}, \]  

where \( \vec{v}_i = \vec{h}_{i+1} - \vec{h}_i, \) \( i = 1, 2, \ldots, N_1 - 1, \) and \( N_1 \) is total number of normal samples. Then, the variance-covariance matrix \( \Sigma \) and the mean \( \mu \) are estimated as follows:

\[ \mu = \frac{\sum_{i=1}^{N_1} \vec{h}_i}{N_1}, \]  

\[ \Sigma = \frac{1}{2(N_1 - 1)} \sum_{i=1}^{N_1} (\vec{h}_i - \mu)(\vec{h}_i - \mu)^T. \]  

The computational relationships for \( k \)th sample are defined as follows [29]:

\[ T^2_i = [\vec{h}_i - \mu]^T \Sigma^{-1} [\vec{h}_i - \mu], \quad i = 1, 2, \ldots, N_1 \]  

The upper control limit for the above statistics given a normal assumption, is approximated using \( UCL = F_{\alpha/Q} \), where \( x^2_{\alpha/Q} \) is the 100 \((1 - \alpha)\)th percentile of chi-squared distribution with \( Q \) degrees of freedom. However, as the normal assumption is usually violated in practice, the non-parametric kernel density estimator is fitted on the data to estimate the empirical probability density functions [30,31]. Denote the Gaussian kernel function as \( K(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2} \) and its bandwidth by \( b \), then the estimated density at point \( T^2_i \) is measured by:

\[ \hat{f}(T^2_i; b) = \frac{1}{n} \sum_{i=1}^{n} K\left( \frac{T^2_i - T^2}{b} \right), \]  

where the optimal value for \( b \) is determined by a \( K \)-fold cross validation that maximizes the log likelihood of training data. Denote the hold out set at the \( k \)th fold by \( S_k \), then the optimal value for \( b \) is obtained as follows:

\[ b = \arg \max \left( \frac{\sum_{i=1}^{K} \log \hat{f}(T^2_i; b)}{K} \right) \]  

The 100 \((1 - \alpha)\)th percentile of the estimated probability density function determines the upper control limit of the control chart.

Note that the flattened encoded vectors store a low-dimensional representation that preserves only the main patterns in the original images. The residual variance of the process can be further investigated by u-shape links. The height of the links in the dendrogram determine the cophenetic distance between two clusters, which can be used to decide for the best number of clusters and linkage types visually.
determined by \( \alpha' = 1 - \sqrt{1 - \alpha} \).

4. Case study and results

4.1. Experiment setting

An experiment is conducted on an open-architecture laser powder bed fusion system - the Additive Manufacturing Metrology Testbed (AMMT) at the National Institute of Standards and Technology (NIST) [32]. Fig. 2 illustrates a schematic set up of the testbed with five major subsystems: a build platform, a powder recoating system, a laser positioning optical system, and two process monitoring modules [14]. One of the process monitoring systems is a high-speed camera-based coaxial melt pool monitoring system. The system is designed to allow full control of laser scans as well as synchronized data acquisitions. It uses a real-time controller to set galvo position and laser power as well as trigger the acquisition of melt pool images [33].

In this experiment, a total of 3 × 4 Inconel 625 parts are manufactured on a 100 mm × 100 mm build platform [15]. Each part has the same nominal 10 mm × 10 mm × 5 mm geometry but built with a unique scan strategy - a combination of various scan paths, laser power, and scan velocity. An in-house developed AM software is used to program the scan strategy into a series of position commands with a time interval of 10 \( \mu \)s (100 KHz). The high-speed coaxial melt pool camera is triggered at 2 kHz (500 \( \mu \)s of sample interval). However, the monitoring system only captures melt pool images for one of the 12 parts at each layer due to the camera memory limit. Hence, every part has around 3000 melt pool images every other 12 layers.

In this study, part 8 is selected because of its richness of the melt pool characteristics. Fig. 3 shows the scan strategy of this part, named “island spiral concentrating scan strategy”. Each layer on the part is composed of four adjacent islands that are printed sequentially. The laser scans each island, spirally following the numbering order shown in the figure. The part has 250 layers, each with 20 \( \mu \)m thickness. The machine reduces the scan speed when the laser turns direction. The machine also reduces the laser power for scan speed to avoid high energy input. Moreover, to avoid high heat concentration and introduce island shape variance, the machine would rotate the centroid angle at each layer. The range of laser power and scan speed is from 0 to 234.83 W and 0–900 mm/s, respectively. Part 8 has melt images captured at layer 8, 20, and 32, etc. We chose the images from layer 8 for the deep representation learning and control chart method validation.

4.2. Deep representation learning of melt pool images

The melt pool images from layer 8 of the process have been collected. The original tensor of the data-set has shape 3769 × 128 × 120, which means there are 3766 total images each with a resolution of 128 × 120. An example of an original image is shown in Fig. 4a. The original images are cropped to the largest common frame that includes the heated area zone’s information. The cropped images now have a resolution of 60 × 60, as shown in Fig. 4b. As the last step toward image preprocessing and following the convolution operation in Equation 2, a 2D sharpening kernel matrix is slid twice on each image to represent the local features in each image. The example image after the sharpening process is illustrated in Fig. 4c.

Next, the images are randomly split into training and testing data by a proportion of 75% and 25%, respectively. Thus a total of 2824 images are used for training the CAE, and the rest 942 images are utilized to validate the model and prevent potential overfitting. The images are also normalized to have pixel intensities in [0,1] interval before feeding to the CAE. The proposed CAE architecture is comprised of 14 layers, out of which the first and last layers contain the input and output images, respectively. The summary of the model layers, excluding the input and output layers, is shown in Table 2. Layers 2-7 and layers 8-13 in the model form the encoder and decoder, and are represented as ‘operation e#' and ‘operation d#’, respectively in the table. The compact deep representation from each image is stored in the feature map tensor from the encoder’s last layer, i.e., ‘MaxPool_66’. This layer is called the bottleneck layer. The architecture of the resulting feature map tensors is illustrated in Fig. 5. In the figure, the number of slices and the corresponding dimension for each slice in each layer is shown before and after @ sign, respectively. The deep representation tensor has a dimension of 8 × 5 × 5.

The CAE model is compiled using the following settings: optimizer ‘Adadelta’ with a batch of size 32, binary cross-entropy as the loss function, and training on 200 epochs. Fig. 6 demonstrates the learning trend for both train and test data resultant from the training process. The y-axis in the figure displays the loss function and x-axis is the number of epochs. Ten arbitrary learned encoded deep representation of melt pool images are obtained from the bottleneck layer, and each is reshaped to a matrix of shape 40 × 5 so that it could be displayed as an image. The third row in Fig. 7 depicts these vectors. Moreover, the first and second row provides the corresponding original images, and decoded images reconstructed using information from the bottleneck layer, respectively. As shown by the figure, the encoded vectors are capable of down-sampling the original image while preserving the melt pool’s local characteristics. Furthermore, the encoded vectors provide a compact deep representation that can be fed to other methods of supervised or unsupervised learning models for further analysis. For this aim, the encoded tensors of the shape 8 × 5 × 5 are flattened to form 200 × 1 vectors. Therefore the original data set is downsized to a total of 3766 samples, each with 200 features.
4.3. Anomaly detection by clustering

This step aims to annotate the melt pool images as either normal or anomaly. The hierarchical-based agglomerative clustering with single linkage given by Algorithm 1 was performed on the flattened encoded vectors, and the resulting dendrogram is illustrated in Fig. 8. The x-axis shows the samples, and the y-axis provides the cophenetic distance to merge the clusters sequentially. The dendrogram’s visual inspection reveals several parent clusters with their approximate partitionings specified by the vertical dashed lines in the figure. Investigating the images inside each cluster also reveals that they share common characteristics. For instance, the angle of the tail of the melt pools is the same inside each cluster. However, the first and the second clusters display quite different behaviors in the dendrogram. First, the number of samples in the first cluster is low. Second, the samples in this cluster tend to merge at a relatively large cophenetic distance compared to the rest of the samples in the data set, meaning they tend to form the singleton clusters of their own. Lastly, and on the contrary, the samples inside the second cluster are merged at a relatively small cophenetic distance, and their numbers are also large. These points suggest that the data set can be partitioned into three major distinctive clusters: cluster $C_1$ composed of a few singleton clusters, cluster $C_2$, and cluster $C_3$ composed of remaining clusters.

Truncating the dendrogram at the cophenetic distance of 0.70, results in a total of 94 singleton clusters, two doubleton clusters, one triplet cluster, one cluster with 530 samples, and one cluster with 3138 samples. One can confirm that the 97 singleton and near singleton clusters belong to $C_1$, the cluster with 530 samples is $C_2$, and the cluster with 3138 samples is, in fact, $C_3$. Three arbitrary images inside each of these clusters are selected and displayed in Fig. 9. The left-hand side, middle, and right-hand side panel contain images in $C_1$, $C_2$, and $C_3$ clusters, respectively. From now on, clusters $C_1$, $C_2$, and $C_3$ will be referred to as “anomaly”, “noisy”, and “normal” clusters, respectively. This also allows for the construction of multi-variate control charting method as described before, after removing the samples inside anomaly and noisy clusters.

4.4. Statistical process monitoring and anomaly detection performance

Given a set of normal melt pool image obtained by the methodologies described in the last subsections, one can proceed to perform a control charting scheme to monitor the stability of the process and also detect

---

**Table 2**

<table>
<thead>
<tr>
<th>Type</th>
<th>Layer name</th>
<th>Operation</th>
<th>Number of filters</th>
<th>Filter size</th>
<th>Padding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoder</td>
<td>conv2D e1</td>
<td>Convolution + ReLU</td>
<td>16</td>
<td>3 x 3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>MaxPool e2</td>
<td>Maxpooling</td>
<td>1</td>
<td>2 x 2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>conv2D e3</td>
<td>Convolution + ReLU</td>
<td>8</td>
<td>3 x 3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>MaxPool e4</td>
<td>Maxpooling</td>
<td>1</td>
<td>2 x 2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>conv2D e5</td>
<td>Convolution + ReLU</td>
<td>8</td>
<td>3 x 3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>MaxPool e6</td>
<td>Maxpooling</td>
<td>1</td>
<td>3 x 3</td>
<td>0</td>
</tr>
<tr>
<td>Decoder</td>
<td>conv2D d1</td>
<td>Convolution + ReLU</td>
<td>8</td>
<td>3 x 3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Upsamp d2</td>
<td>Upsampling</td>
<td>1</td>
<td>2 x 2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>conv2D d3</td>
<td>Convolution + ReLU</td>
<td>8</td>
<td>3 x 3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Upsamp d4</td>
<td>Upsampling</td>
<td>1</td>
<td>2 x 2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>conv2D d5</td>
<td>Convolution + ReLU</td>
<td>16</td>
<td>3 x 3</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Upsamp d6</td>
<td>Upsampling</td>
<td>1</td>
<td>3 x 3</td>
<td>0</td>
</tr>
</tbody>
</table>

---

**Fig. 4.** Image pre-processing on melt pool images.

**Fig. 5.** The architecture of input images, feature map matrices and output images.
To this aim, first, the normal data set is randomly split into training and testing sets by a proportion of 75% and 25%, respectively. Then, the same configuration of CAE proposed in Table 2 is retrained and validated this time on the new training and testing data, respectively. Next, the flattened encoded tensors of melt pool images belonging to the training data are used to construct the phase-I control charting scheme proposed in Subsection 3.3. In this phase and in regards to Hotelling’s $T^2$ control chart, the mean vector $\mu$ and variance-covariance matrix $\Sigma$ given by Eqs. (9) and (10) and the value of $UCL$ are estimated all based on the training data. $LCL$, $CL$, and $UCL$ values pertaining to residual variance monitoring are also estimated using the original and decoded training images.

In phase-II monitoring, the estimated parameters are used to calculate the statistics for monitoring data and decide if the process is either in control or out of control. Basically, a decision for out of control situation is made when either of $T^2$ or $S^2$ control charts issue an out of control alarm. This happens when either monitoring statistics exceeds their corresponding control limits. On the other hand, if both statistics...
The process can be considered in control. In order to acquire the monitoring data, the melt pool images in the noisy cluster are all discarded. The testing data and images in anomaly clusters are labeled as "normal" and "anomaly" data, respectively, and their encoded and decoded tensors are obtained by running them through the trained CAE. The former and the latter are then employed to evaluate the control charting method’s performance in terms of triggering false alarms and true anomaly detection, respectively. The common practice to this aim involves forming a confusion matrix that stores the values for True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). In the context of the control chart monitoring, a sample can come under one of the following categories:

- **TP**: if it is anomaly and detected as out of control
- **FP**: if it is normal but detected as out of control
- **TN**: if it is normal and detected as in control
- **FN**: if it is anomaly but detected as in control

Using the elements of confusion matrix, a set of criteria are defined as follows to assess the quality of process monitoring:

- **Accuracy** = \( \frac{\sum (TP + TN)}{\sum (TP + FP + TN + FN)} \)
- **Specificity** = \( \frac{\sum (TN)}{\sum (TN + FP)} \)
- **Sensitivity** = \( \frac{\sum (TP)}{\sum (TP + FN)} \)
- **Precision** = \( \frac{\sum (TP)}{\sum (TP + FP)} \)
- **F_1 score** = \( \frac{2 \times Sensitivity \times Precision}{Sensitivity + Precision} \)

As it was discussed by Section 3.3, the 100(1 – α)th percentile of the estimated probability density function for measured statistics will be used to determine the control limits. This means there is a 100α probability that the statistics go above the control limits and get detected as out-of-control (rejected) while they are in fact in control (true null hypothesis). The larger the α, the smaller the control limits and thus more alarms will be issued by the charts and vice versa. This will also impact the out-of-control detection power of control charts that is represented by \( 1 - \beta \) error, where \( \beta \) is the probability of accepting a sample while it is out-of-control. It is common to choose the type-I error as a number between 0.001 and 0.05, which means the corresponding control charts will issue false alarm warnings 0.1% and 5% of the time. 10 equally spaced points inside this interval are chosen to test and find the best control limits and therefore the best trade-off between chart’s false alarm rate and out-of-control detection power. By selecting 10 different
Table 3

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Accuracy</th>
<th>Specificity</th>
<th>Sensitivity</th>
<th>Precision</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0010</td>
<td>0.0064</td>
<td>0.0119</td>
<td>0.0173</td>
<td>0.0282</td>
</tr>
<tr>
<td></td>
<td>(0.0067)</td>
<td>(0.0040)</td>
<td>(0.0032)</td>
<td>(0.0025)</td>
<td>(0.0035)</td>
</tr>
<tr>
<td>Type-I error probability of $\alpha^*$</td>
<td>(0.0061)</td>
<td>(0.0066)</td>
<td>(0.0060)</td>
<td>(0.0064)</td>
<td>(0.0069)</td>
</tr>
<tr>
<td></td>
<td>0.0336</td>
<td>0.0391</td>
<td>0.0445</td>
<td>0.0500</td>
<td></td>
</tr>
<tr>
<td></td>
<td>93.10%</td>
<td>94.58%</td>
<td>95.16%</td>
<td>95.40%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.0032)</td>
<td>(0.0025)</td>
<td>(0.0035)</td>
<td>(0.0061)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>95.38%</td>
<td>95.24%</td>
<td>95.07%</td>
<td>94.87%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.0035)</td>
<td>(0.0061)</td>
<td>(0.0061)</td>
<td>(0.0060)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>94.67%</td>
<td>94.43%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.0064)</td>
<td>(0.0069)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>62.64%</td>
<td>75.24%</td>
<td>79.91%</td>
<td>81.99%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.0287)</td>
<td>(0.0234)</td>
<td>(0.0100)</td>
<td>(0.0118)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>82.63%</td>
<td>82.70%</td>
<td>82.57%</td>
<td>82.30%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.0216)</td>
<td>(0.0190)</td>
<td>(0.0179)</td>
<td>(0.0190)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>81.95%</td>
<td>81.44%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.0190)</td>
<td>(0.0212)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Fig. 10. Anomaly detection by statistical process monitoring on top of the convolutional auto-encoders.](image)

random proportions of the data from the normal cluster as training data for each of the control chart configuration, 100 trials were conducted. Table 3 provides the results for performance measurements criteria. The standard deviation of results is given in parenthesis below each entry. The highest scores for each measurement have been emboldened in the table. All the measurements are either monotonically increasing or decreasing with respect to $\alpha^*$, except for accuracy and F1-score that reach their maximum at $\alpha^* = 0.0228$. On rare occasions where the true classification of normal samples is preferable to the practitioner, the high score of Specificity can be used to select the $\alpha^*$. Fig. 10 illustrates a slice of phase-II $T^2$ control charting scheme constructed based on $\alpha^* = 0.0228$. It includes the entire samples from anomalies and covers a portion of training and test data. The data are sorted according to their labels to provide a better visual intuition into the control charting method’s performance regarding the true classification of samples.

4.5. Comparison with hand-crafted features

As it can be observed by the right-hand side images in Fig. 9, the deep representation of melt pool images allows for the detection of eccentric types of anomalies that are not detectable by the handcrafted feature engineer methods. For example, two melt pool images might encompass the same amount of bright pixels or share the same melt pool area, but the splashing effect could still distinguish them from each other. The NBEM model [10] described briefly in Section 2 is an example study that uses the area of fitted ellipses on the thresholded images of melt pools as the feature of interest. It should be noted that this work and NBEM study both share the same experiment setting and data set of melt pool images. In order to extract the features according to [10], the threshold value of 150 is chosen to segment the melt pools from surrounding areas.

Similar to the procedure described in Subsection 4.4, normal melt pool images are employed to construct the phase-I control charting and estimate the parameters. However, in this case, a univariate monitoring scheme composed of a $I$ and a $MR$ chart is deployed. The $I$ chart is provided to monitor the individual observations, and their moving range is monitored by the $MR$ chart. The individual observations represented by $x_i$ are simply the measured melt pool areas, and their moving range can be calculated as $MR_i = |x_i - x_{i-1}|$. The $UCL$, $CL$, and $LCL$ for the $I$ chart to monitor $x_i$ is obtained as follows:

$$UCL = \tau + \frac{MR}{d_2}$$

$$CL = \tau$$

$$LCL = \tau - \frac{MR}{d_2}$$  \hspace{1cm} (15)

where $\tau = \frac{\sum_{i=1}^{N} x_i}{N}$, $MR = \frac{\sum_{i=1}^{N} MR_i}{N}$, and $d_2$ is a predetermined constant equal to 1.128. Moreover, the $UCL$, $CL$, and $LCL$ for the $MR$ chart can be obtained as follows:

$$UCL = D_4 MR$$

$$CL = MR$$

$$LCL = D_3 MR$$  \hspace{1cm} (16)

where $D_3$ and $D_4$ are predetermined constants equal to 0 and 3.267, respectively. Fig. 11 illustrates the phase-II $I - MR$ control charting scheme constructed based on the above control limits. It includes the entire samples from anomalies and covers the same portion of training and test data as Fig. 10. Similarly, the data are sorted according to their
The table provides the results based on the results are provided in the first row of Table 4. The second row in the method based on melt pool areas in terms of anomaly detection. The set proposed deep representation learning approach outperforms the label to provide better visualization. As it is obvious by the comparison between the two figures, the control charting scheme based on the proposed deep representation learning approach outperforms the method based on melt pool areas in terms of anomaly detection. The set of performance measurement criteria was calculated for 100 trials, and the results are provided in the first row of Table 4. The second row in the table provides the results based on $\alpha = 0.0228$ for the proposed deep representation learning method. The low values for performance measurement criteria, especially sensitivity, precision, and $F_1$-score compared to the proposed method, prove that simple characterization of melt pools by their area results in losing valuable information vital to maintaining a quality process monitoring and control.

5. Conclusions

Sensor data, including melt pool images, carry the most important signatures of the Laser Powder Bed Fusion (LPBF) process and other additive manufacturing technologies. Data-driven in-situ process monitoring and control methods mine this information to shed light on the correlation between process parameter settings and quality characteristics of interest without accounting for the actual physics-based modeling. While the state-of-the-art methods often rely on handcrafted feature engineering methods to extract a set of features from sensor data, this paper developed a methodology to gain a set of smart features that provide a more in-depth description of the data. Specifically, a configuration of Convolutional Auto-Encoder (CAE) neural networks processes the collected melt pool images to learn a low-dimensional but deep representation from data. The extracted features are plugged into an agglomerative clustering algorithm to tag the anomalies and automatically annotate the data due to the burdensome and expensive manual annotation process. Subsequently, a Hotelling’s $T^2$ and $S^2$ control charting scheme is developed to monitor the process’s stability by keeping track of the learned representations and residuals obtained from the reconstruction of original images. Finally, the numerical results from a real-world case study are provided to prove the proposed method’s effectiveness, applicability, and accuracy. The frequency of data acquisition is usually high, resulting in a large volume of data. Therefore, in addition to the spatial correlation within each sample, it is also important to account for the temporal interrelation between samples. Future studies can be directed toward investigating deep learning techniques for time series analysis by considering instances as sequences of samples. Furthermore, this study examined the detection of anomalies in real-time. It is also worth studying the incorporation of control methods to the model and optimize the process parameter settings to ensure certain targets for the quality characteristics are met online.

Table 4

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Specificity</th>
<th>Sensitivity</th>
<th>Precision</th>
<th>$F_1$-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>NBEM</td>
<td>89.20%</td>
<td>98.38%</td>
<td>17.98%</td>
<td>59.77%</td>
<td>27.32%</td>
</tr>
<tr>
<td>Deep representation learning</td>
<td>95.38%</td>
<td>97.47%</td>
<td>82.10%</td>
<td>83.34%</td>
<td>82.63%</td>
</tr>
</tbody>
</table>

Fig. 11. Anomaly detection by statistical process monitoring on top of the melt pool areas.

References


