

Full length article

Anomaly detection and classification in a laser powder bed additive manufacturing process using a trained computer vision algorithm



Luke Scime*, Jack Beuth

NextManufacturing Center, Department of Mechanical Engineering, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, United States

ARTICLE INFO

Article history:

Received 3 May 2017

Received in revised form

25 September 2017

Accepted 16 November 2017

Available online 16 November 2017

Keywords:

Additive manufacturing

Computer vision

Machine learning

In-situ monitoring

Powder spreading anomalies

ABSTRACT

Despite the rapid adoption of laser powder bed fusion (LPBF) Additive Manufacturing by industry, current processes remain largely open-loop, with limited real-time monitoring capabilities. While some machines offer powder bed visualization during builds, they lack automated analysis capability. This work presents an approach for in-situ monitoring and analysis of powder bed images with the potential to become a component of a real-time control system in an LPBF machine. Specifically, a computer vision algorithm is used to automatically detect and classify anomalies that occur during the powder spreading stage of the process. Anomaly detection and classification are implemented using an unsupervised machine learning algorithm, operating on a moderately-sized training database of image patches. The performance of the final algorithm is evaluated, and its usefulness as a standalone software package is demonstrated with several case studies.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

In recent years, Additive Manufacturing, colloquially known as 3D-printing, has experienced immense growth as an industry; this is particularly true for machines and processes producing net-shape metal parts [1]. Additive Manufacturing promises to be well-suited for aerospace and medical applications [2] as well as for producing mission-critical parts, on-site, at remote locations [3]. However, these applications require a degree of part quality assurance and process reliability that are difficult to achieve with the systems currently on the market [1]. It is commonly recognized that implementation of in-situ process monitoring and closed-loop control is necessary to meet the stringent requirements of these applications [1].

Laser Powder Bed Fusion (LPBF) machines operate by spreading a thin layer (typically 20 μm–60 μm thick) of metal powder over a build plate using a recoater blade. After powder spreading, a laser beam is used to selectively melt the powder in locations corresponding to a 2D slice of a 3D part. After the lasing is complete, the build plate is lowered, another layer of powder is spread (now over an existing powder bed, Fig. 1), and the process repeats until the part is finished. The entire process of creating a part is often referred to as a build. There has been extensive work performed on

monitoring builds in-situ [4–6], with a particular focus on tracking both the size of the melt pool produced by the laser beam [7–9] as well as the powder bed temperature [10]. Many of the flaws in a final part, as well as the overall reliability of the build process, are directly related to interactions between the recoater blade and the powder bed. As a result, several groups have begun paying special attention to this stage of both the LPBF and Electron Beam PBF processes [7,11–17]. The focus of the presented work is to monitor the powder bed for indications of flaws in final parts, as well as anomalies that may impact the stability of the process as a whole.

For this work, six types of anomalies (not including the anomaly-free case), summarized in Table 1, were identified. These anomalies range in severity from recoater hopping which may only indicate the onset of a more severe problem, to super-elevation which can be quite serious. Some anomalies (such as part failure) may indicate flaws in the final part, while others, such as recoater streaking, suggest damage to the machine itself; further description of the anomalies is provided in Section 2.2. Detection of recoater streaking has been explored by Craeghs et al. [7] and various methods for detecting super-elevation (albeit at a different size scale) have been proposed by Jacobsmühlen et al. [12]. Recent work by Abdellrahman et al. [13] demonstrates layer-wise detection of general flaws via comparison of post-fusion optical images with the CAD model. Little work has been done to comprehensively address all of these anomaly types simultaneously, particularly over the entire build volume and using only hardware directly available from an AM machine manufacturer. Furthermore, much of the exist-

* Corresponding author.

E-mail address: lsleine@andrew.cmu.edu (L. Scime).

Table 1

Brief description of anomaly classifications and their respective color codes used throughout this document. Additional descriptions can be found in Section 2.2.

Anomaly	Description	Color Code
Anomaly-Free/Okay	No significant anomalies in the powder bed.	Green (3D figures) Clear (overlays)
Recoater Hopping	Caused by the recoater blade striking a part, characterized by repeated vertical (perpendicular to the recoater spreading direction, Fig. 1) lines.	Teal/Light Blue
Recoater Streaking	Caused either by the recoater blade dragging a contaminant across the powder bed or by damage to the blade. Characterized by horizontal (parallel to the recoater spreading direction, Fig. 1) lines.	Dark Blue
Debris	Debris or other small to mid-sized discrepancies located in the powder bed but not directly over any parts.	Black (plots) White (overlays)
Super-Elevation	Occurs when a part warps or curls upwards out of the powder layer. Typically the result of a buildup of residual thermal stresses.	Red
Part Failure	General classification for any significant damage to a part. Characterized by a variety of signatures.	Magenta/Purple
Incomplete Spreading	Occurs when an insufficient amount of powder is repeatedly fetched from the powder dispenser (Fig. 1). Results in a lack of powder, the severity of which is highest nearest the powder collector (Fig. 1).	Yellow

ing work relies on human-created detectors for specific anomalies, e.g. line profiles [7] and segmentation [13], while the presented methodology makes use of contemporary machine learning techniques to construct the anomaly detectors. It is worth noting that LPBF machine manufactures (including EOS GmbH [18]) are now releasing process monitoring solutions that include analyses of the powder bed. Unfortunately, many of the details about the methodologies used by these systems are currently unavailable.

To accomplish the goal of comprehensive powder bed monitoring, this work presents an algorithm that implements contemporary machine learning and computer vision techniques to detect and classify the enumerated anomalies using only hardware provided by the LPBF machine manufacturer. In the computer vision community, machine learning has become immensely popular, though many of the methods are typically applied to entire images [19]. This presents a challenge as each powder bed image may contain hundreds of uniquely-identifiable anomalies; to compensate, this work modifies a standard approach (Section 3) to allow for classification of multiple objects within a single image, an approach also pursued by Winn et al. [20]. Even as stand-alone software (e.g. not integrated with the LPBF machine control system), this algorithm has proven valuable in analyzing build failures and in analyzing final part quality.

2. Experimental procedure and methods

All of the work presented herein is performed on an EOS M290 LPBF machine (EOS GmbH, Germany). No modifications are made to the EOS hardware, e.g. only the stock camera and lighting configurations are used. Images of the build plate and powder bed are taken through a viewport located (almost) directly above the build chamber. Grayscale images with a resolution of 1280 pixels × 1024 pixels are automatically captured immediately after a new powder layer is spread. All software was developed in the MATLAB R2015a and R2016a programming environments.

2.1. Image pre-processing

The raw images (Fig. 1) captured by the EOS M290 present several difficulties that prevent their direct usage in a machine learning-based algorithm. Fortunately, the camera mounting and lighting conditions remain consistent throughout a build as well as between different builds, so many of the required image enhancements can be greatly simplified.

Out of the necessity of avoiding the laser optic train, the camera is mounted such that its axis is not parallel to the normal vector of the build plate. This distortion is corrected using a fully-constrained

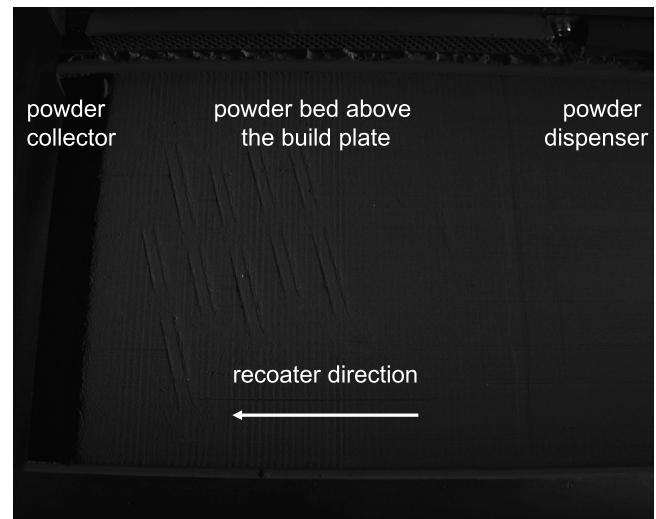


Fig. 1. Raw powder bed image collected by the EOS M290.

Homography matrix [21] which warps and scales the raw image such that a square build plate in the image will appear square. Because the camera positioning and orientation are fixed, manual measurements of a powder-free build plate (within the camera's field of view) were taken and no fiducial (e.g. corner) detection was implemented to inform the Homography matrix. The image is then cropped to include only the region of the powder bed directly above the build plate. The spatial resolution (not synonymous with resolving power [22]) of the camera setup is between 290 μm/pixel and 340 μm/pixel (the existence of a range of resolutions is the result of the misalignment between the camera axis and the normal vector of the build plate). After the described warping and cropping, each pixel represents a 290 μm × 290 μm field of view; note that no anomalies with a dimension less than 2.9 mm are reported by the algorithm (Section 3.3).

During printing, the powder bed is lit by a single bank of white LEDs on the right side of the build chamber. This side lighting increases (compared to top lighting) the contrast of any 3D features (e.g. hills and valleys), but it also results in uneven lighting conditions. The uneven lighting causes a haloing effect in the images that is detrimental to the training process. To remedy this, an anomaly-free powder bed image was used to generate a baseline intensity mask. Stochastic noise present in the mask was reduced using a Gaussian filter. This mask is applied to each future powder bed image to levelize the lighting across the powder bed.

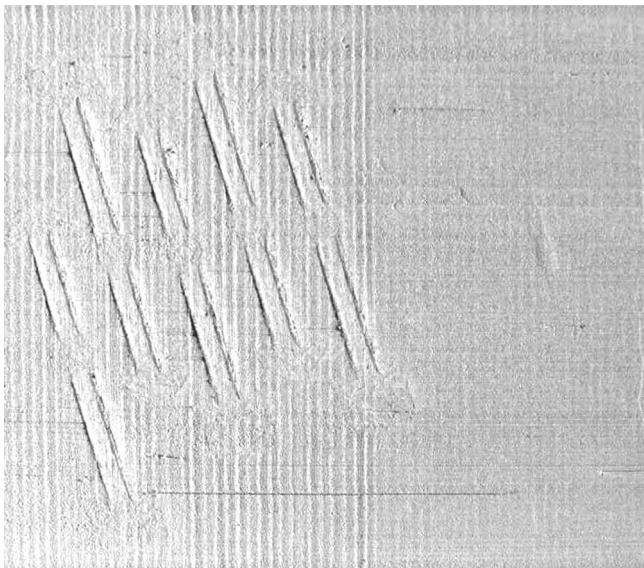


Fig. 2. Fig. 1 after pre-processing.

Fig. 2 shows a fully pre-processed image. Preliminary results (not presented in this manuscript) indicate that the algorithm is fully capable of operating on datasets generated by other (i.e. non-CMU) EOS M290s, provided that the pre-processing steps, described above, are followed. Additionally, the algorithm has been successfully used to analyze several powder bed images taken from a camera mounted on an SLM 280 machine (SLM Solutions GmbH). Note that the recoater blade moves from right to left in all powder bed images presented in this document (as shown in **Fig. 1**).

2.2. Selection of the powder spreading anomaly classes

The powder bed anomaly classifications chosen to illustrate this work are briefly enumerated in **Table 1** and further described in this section. Importantly, the operation of the algorithm presented in Section 3 is not dependent upon the specific anomaly groupings chosen by the authors; the machine learning approach only requires that the chosen anomaly types are self-consistent and sufficiently distinct from each other. Of course, a sufficiently large number of training examples must be available for each anomaly

type, as discussed in Section 2.2. For the purposes of the consolidation heuristics described in Section 3.4, the anomaly types in order of increasing severity are: recoater hopping, recoater streaking, debris, super-elevation, part failure, and incomplete spreading.

Recoater hopping typically occurs when the recoater blade (relatively) lightly strikes a part just below the powder layer. Such a strike results in a periodic “chatter” of the recoater blade which is visible as repeated vertical lines in the powder bed as shown in **Fig. 3a**. Recoater streaking occurs either when the recoater blade is damaged (i.e. “nicked”) or when the recoater blade drags a piece of debris or a clump of powder across the powder bed. Recoater streaking is visible as individual horizontal lines in the powder bed (**Fig. 3b**); because of its relatively small size, it is one of the most challenging anomalies to detect with the presented method (see Section 4.1). The debris classification encompasses most (see incomplete spreading, below) disturbances to the powder bed that are not directly located over a part. **Fig. 3c** shows several examples of debris.

Super-elevation occurs when a part visibly warps up above the powder layer; the warping is often the result of a buildup in residual thermal stresses within the part. Super-elevation anomalies have a wide range of appearances due to the wide range of extant part geometries, but all generally contain long edges at varying orientations. **Fig. 3d** shows several representative examples of super-elevation. Part failure encompasses significant disturbances to the powder bed directly over a part and generally indicates damage to a part caused by a major strike by the recoater blade. Unlike super-elevation, long edges are typically not present as the part has sustained damage; **Fig. 3e** shows several examples of part failure. Finally, incomplete spreading occurs when insufficient powder is fetched from the powder dispenser (**Fig. 1**) and spread across the build plate. This mechanism results in large disturbances to the powder bed which initially occur near the powder collector (left-hand side of the powder bed, **Fig. 1**). Over the duration of a build, disturbances due to incomplete spreading may encroach further and further into the powder bed; **Fig. 3f** shows an example of incomplete spreading.

2.3. Selection of training images

Each powder bed image may contain hundreds of distinct examples of different anomalies and anomaly-free regions. For this reason, training of the machine learning algorithm is performed using image patches: sub-regions of the full powder bed images

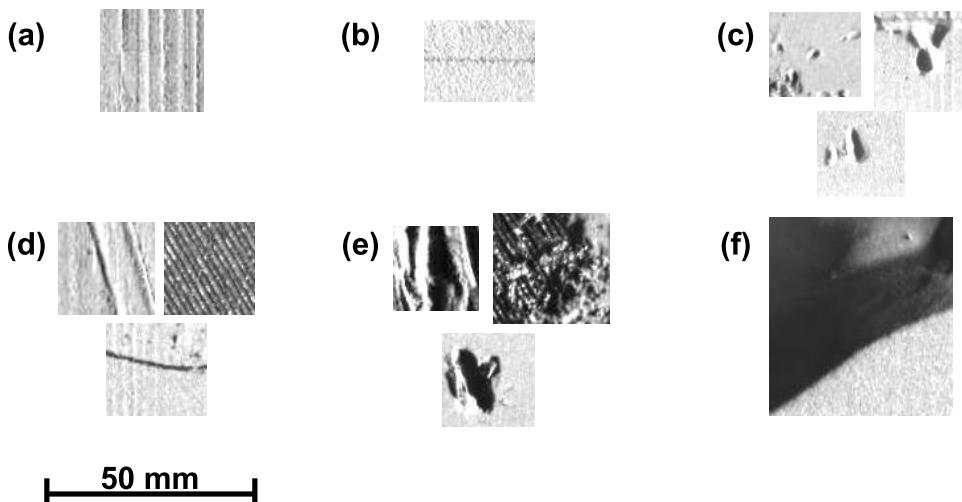


Fig. 3. Representative examples of the six different powder bed anomaly classes chosen by the authors. Note that the relative sizes between the anomalies have been preserved. Specifically, the anomalies are: (a) Recoater hopping, (b) Recoater streaking, (c) Debris, (d) Super-elevation, (e) Part failure, and (f) Incomplete spreading.

such as those shown in Fig. 3. To develop the training database, a human manually selects rectangular image patches from multiple powder bed images collected from multiple builds. Note that while the exact size of the training image patches is not constrained, they were selected such that they are similar in size to the patches introduced in Section 3.3. Before patch selection, the powder bed images are pre-processed as described in Section 2.1. Each image patch is given a ground-truth label as one of the six anomaly types listed in Table 1 or as anomaly-free/okay. The patches and their attached labels are stored in a database for access by the machine learning algorithm. The current training database includes a total of 2402 image patches, composed of 1040 anomaly-free patches, 264 recoater hopping patches, 228 recoater streaking patches, 187 debris patches, 314 super-elevation patches, 264 part failure patches, and 105 incomplete spreading patches. Algorithm performance continues to improve as more image patches, from additional builds, are added to the training database. Because of this behavior, no formal convergence study has yet been performed on the size of the database.

2.4. CAD information extraction

The accuracy and the usefulness of the powder bed monitoring algorithm are improved by the inclusion of information regarding the shape, position, and orientation of the parts being built. This information is contained within the EOSPRINT slice (.SLI) files. Because the slice files are encrypted [23], this information is extracted using an automated Windows 10 script that captures

screenshots of each layer of the build within the EOSPRINT v1.3 software environment [24]. The screenshots are processed and converted into a 3D point cloud of the build. This CAD information is overlaid onto the powder bed images during anomaly classification. Explicit registration (alignment) between the powder bed camera images and the CAD models is not performed. Instead, the entire powder bed image is warped and cropped (Section 2.1) such that the CAD information can be directly overlaid on top of the image. Further discussion of the algorithm's usage of the CAD information is provided in Section 3.4.

3. Theory

This work is an application of a widely-used machine learning technique, known as bag-of-keypoints (or words) [25], often applied to computer vision problems. A machine learning approach was pursued by the authors, as opposed to the manual creation of anomaly detectors, due to its inherent flexibility and potential to be a “one-size-fits-all” method for detection and classification of multiple anomaly types. While the bag-of-keypoints technique can be applied to multiple feature types such as SIFT [26,27], the authors chose filter responses (Section 3.1) for their ability to preserve scale information (i.e. the size of a potential anomaly influences its filter response). This section is intended to provide an overview of this technique, and describe its application to Additive Manufacturing. Fig. 4 is a flowchart of this portion of the powder bed monitoring algorithm and is referred to extensively.

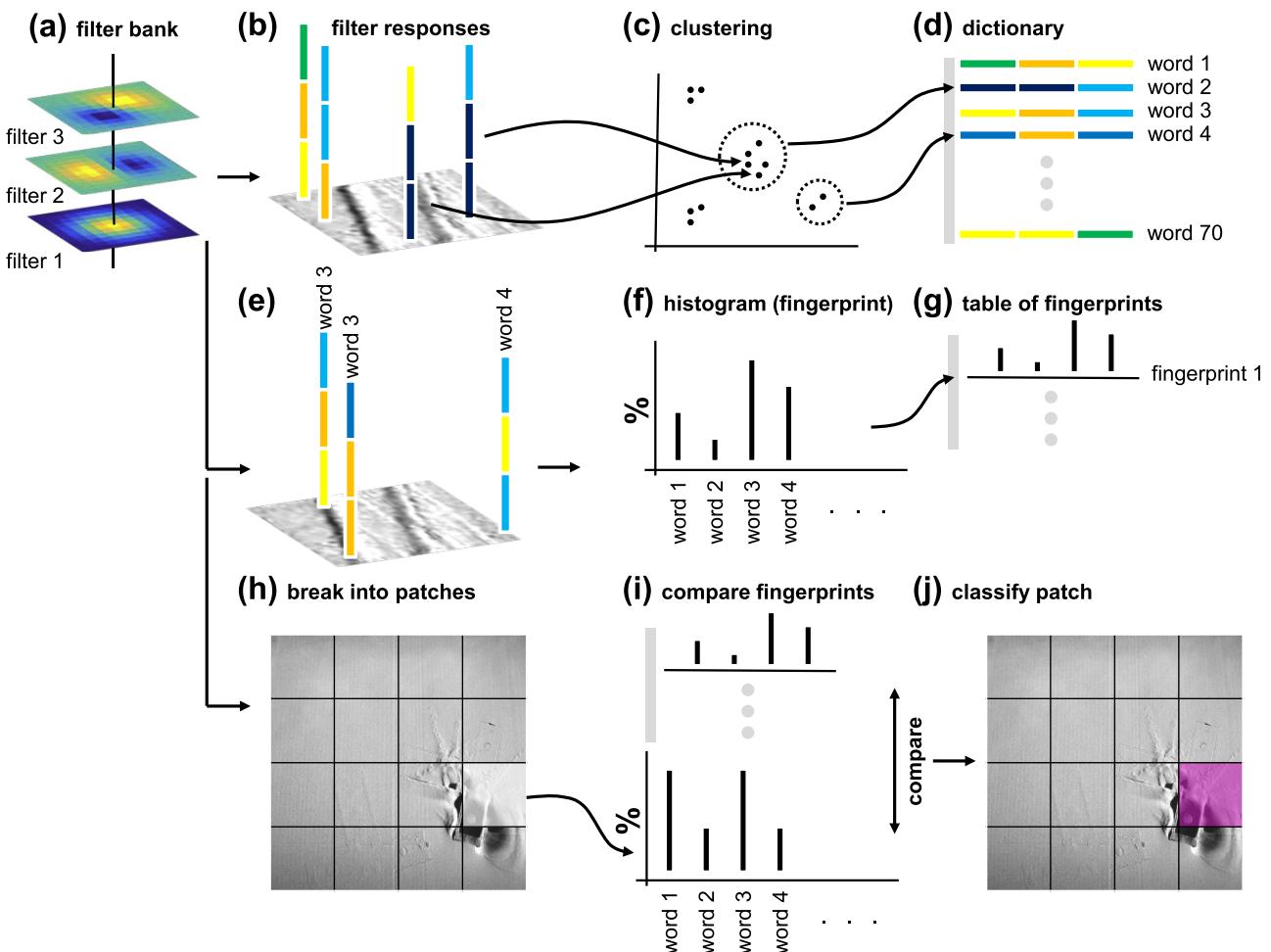


Fig. 4. Flowchart of the machine learning process implemented in this work.

Table 2

A brief description of the composition of the filter bank.

Filter Type	Description	Num. of Size Scales/Variants
Gaussian	Standard filters, designed to detect blobs.	3
Uniform Averaging Disk	Standard filters, designed to detect blobs. The response is generally stronger than that of a Gaussian filter for blobs of particularly uniform intensity (i.e. darkness).	1
Difference of Gaussian (DoG)	Standard filter, designed to detect edges at all orientations.	3
Oriented Edge Detectors	Standard, asymmetric first derivative filters designed to detect edges at specific orientations (e.g. 0° and 90°).	6
Oriented Line Detectors	Non-standard combinations of oriented edge detectors designed to detect lines at specific orientations (e.g. 0° and 90°). Particularly effective at detecting super-elevation anomalies.	18
Streak Detectors	Non-standard combinations of oriented edge detectors designed to detect recoater streaking and recoater hopping anomalies.	4
Gabor [30]	Standard filters designed to produce strong responses when convolved with an image containing spatially-encoded frequency information. Note that only the numerically real component is considered.	2

3.1. Development of the filter bank

To first order, all machine learning algorithms operate by identifying similar and dissimilar *features* within a set of training data. The prevalence of, location of, and correlation between these detected *features* within the training data are compared to the *features* identified within the data set that is being analyzed. For the purposes of this work, the training data set is the database of image patches and their corresponding labels (Section 2.3), the data being analyzed are the powder bed images of new builds, and the *features* are the pixel-level responses generated by the application of a *filter bank* to a 2D image.

In this context, a *filter* refers to a discretized, 2D function or pattern; three examples of *filters* can be seen within the representative *filter bank* shown in Fig. 4a. These *filters* are “passed over,” or convolved [28] with, an image (either a training image patch or a powder bed image). The output of the convolution operation is a 2D image with the same dimensions as the original image. The value of each pixel of the output image is the response of the original image to the convolution with the applied *filter*; its value depends upon the values of the corresponding pixel and its surrounding pixels in the original image.

Different *filters* produce either stronger or weaker *responses* depending on the distribution of the information encoded in the original image. For example, a Gaussian *filter* (e.g. filter 1 in Fig. 4a) produces a strong *response* when convolved with pixels in the vicinity of a dark blob, while an asymmetric first derivative *filter* (e.g. filter 2 in Fig. 4a) produces a strong *response* centered on any dark, vertical edges. The *response* of each pixel to each *filter* in the *filter bank* is stored in a vector, represented by the vertical bars in Fig. 4b; the three colored subsections of each bar represent the responses to the three *filters* in the representative *filter bank*. These *response* vectors are the *features* on which the machine learning algorithm operates.

The *filter bank* used in this work contains a total of thirty-seven filters as enumerated in Table 2. Multiple trials were performed to determine an effective *filter bank* size and composition. To mitigate the impact of edge effects on the training process, the borders of the image patches are padded with the pixels surrounding that patch and the borders of the powder bed images are padded with replicated pixels [29].

3.2. Training

The *filter bank* described in Section 3.1 is passed over every training image patch, such that there is a response vector for every pixel in every image patch. No subsampling of the training data is performed, i.e. all of the response vectors are included in the training process. Vectors with similar values in each element (i.e. vectors

that share a similar set of responses) are grouped together using a standard k-means unsupervised clustering algorithm [31], represented by Fig. 4c. For this work, cluster initialization was performed using random seeding, with preference given to a uniform spacing between clusters. The requested number of clusters was systematically increased until the final powder bed monitoring algorithm results ceased to noticeably improve; the final clustering produces 100 groups. Cluster seeding is repeated 100 times to reduce the chance of the algorithm converging to a poor solution; e.g. a shallow local minimum instead of a global, or at least a deeper local, minimum.

Each group is represented by a mean response vector. The 100 mean response vectors are commonly referred to as *visual words*, and are stored in a *dictionary*, represented by Fig. 4d. The *visual words* are the *features* that will be searched for in future data sets (e.g. new powder bed images). After the *dictionary* is constructed, the *filter bank* is again convolved with each training image patch. But this time the filter response vectors at each pixel are matched to the closest (pair-wise distance) *visual word* in the *dictionary* (Fig. 4e). For each training image patch, the percentage of pixels matched to each *visual word* is calculated. This information can be represented by a histogram (Fig. 4f). As it uniquely identifies each image patch, it is often referred to as a *fingerprint*. The *fingerprint* of each training image patch is stored in a table (Fig. 4g). Ideally, training images containing similar anomalies will have similar *fingerprints*, while training images containing different anomalies will have dissimilar *fingerprints*. The final output of the training process is a table containing 2402 histograms/fingerprints (one for each training image patch) that are each 100 elements long; the corresponding ground-truth anomaly labels are stored in the training database (Section 2.2).

3.3. Classification of the powder bed images

The steps described in Section 3.2 are only performed during training. During implementation of the powder bed monitoring algorithm, the *filter bank* is convolved with the entire powder bed image and each pixel is assigned its closest-matching *visual word*. The layer image is then broken up into *patches* (Fig. 4h). These *patches* are rectangular but vary in size and aspect ratio to better detect specific anomalies. A total of three different *patch* types are used in the current work: a 20 pixel × 20 pixel square that is expected to detect most anomaly types, a 10 pixel × 40 pixel rectangle designed to detect recoater streaking, and a 100 pixel × 100 pixel square designed to detect incomplete spreading. A *fingerprint* for each *patch* is generated and compared (using a binary singleton expansion function [32]) to the table of *fingerprints* (Fig. 4i); the top three matches are considered in the anomaly classification deci-

sion process. It is at this stage that the labels associated with each *fingerprint* in the table are extracted from the training database.

These top three matches are then weighted according to their respective degree of agreement and the *patch* is classified as the anomaly with the highest total weighting among the top three matches. For example, if the top match is for recoater hopping, but the next two matches are for recoater streaking, the algorithm would classify the patch as recoater streaking if the second and third matches have a stronger combined agreement than the first match by itself. This approach was experimentally shown to produce more accurate results, possibly by mitigating the impact of over-fitting [33]. Additionally, if any of the top three matches are an anomaly-free case, the patch is immediately classified as anomaly-free. This restriction reduces the number of false positives, which were deemed more problematic than false negatives (see Section 5). The final anomaly classification for that *patch* is then applied to every pixel within that *patch*.

3.4. Context driven heuristics

The overlapping results from the three *patch* type analyses are combined with relevant CAD information (Section 2.4) in a series of context driven heuristics to determine the appropriate anomaly classification for each pixel in the powder bed image (Fig. 4j). The inclusion of the heuristics layer allows information about the location of the anomaly detections with respect to the build plate, the parts, and the surrounding anomaly detections to factor into the final anomaly classification decisions made by the algorithm. Each of the heuristic rules is discussed, briefly, in the following two paragraphs.

If the initial classification of a pixel not lying on top of a part (as defined by the extracted CAD data) is either part failure or super-elevation, the classification is converted to the debris category as it can be visually similar to the aforementioned anomalies, which, by definition (Section 2.2), can only occur on top of a part. Similarly, any pixels lying on top of a part and initially classified as debris, are converted to the part failure category. Because incomplete spreading anomalies are detected with the largest scale *patches*, not all of the pixels labeled as incomplete spreading may be truly anomalous. To increase the chances that as-built part quality is accurately reflected by the algorithm output, any pixels lying on top of a part and initially labeled as incomplete spreading are retroactively unlabeled. In other words the results from the largest *patches* (Section 3.3) are not included in the final, multi-*patch* consolidation process for the pixels lying above parts. An additional constraint is imposed on the incomplete spreading category to reduce the occurrence of false positives: Extremely large disturbances to the powder bed, such as that visible in Fig. 9, may appear visually similar to incomplete spreading despite being the result of a different mechanism. Therefore, incomplete spreading detections are ignored if there are no incomplete spreading detections near the left-hand edge (Fig. 1) of the build plate (see Section 2.2 for reasoning).

At this point in the implementation of the heuristics, the results from the three different *patch* types are combined. Because the *patch* results overlap and may be in disagreement, consolidation is effected by classifying each pixel as the highest possible severity anomaly, according the severity ranking described in Section 2.2. After consolidation, the false positive rates for recoater hopping are reduced by confirming that multiple detections are present in a vertical line (i.e. parallel to the y-axis, Fig. 1). A similar approach is pursued for recoater streaking, except multiple detections in a horizontal line are required. To reduce the computational burden (Section 4.1) of the heuristic layer, many of the context rules are implemented by convolution of a mask of the initial anomaly detections with various templates and filters (note that these are conceptually similar to the *filters* described in Section 3.1). While

most of the heuristics discussed above (e.g. those relating to debris vs. part failure and incomplete spreading vs. part failure) are fully objective (i.e. they are derived from the anomaly definitions presented in Section 2.2), the heuristics for recoater streaking and recoater hopping have a degree of subjectivity in that the required "line length" can be adjusted. The "line length" parameters ultimately chosen for the recoater streaking and recoater hopping heuristics are based upon the analysis of over two dozen separate builds.

4. Results and discussion

The overall algorithm performance is first discussed in Section 4.1. Because this work did not receive external funding, no test artifacts were created specifically for this research. Therefore, the results are presented by reviewing the analyses of three case studies, using data from builds performed on CMU's EOS M290. The algorithm described in Section 3 is used to analyze powder bed images from each layer of a build; results for the entire build can be visualized in multiple formats. Recall that color codes for each detected anomaly are listed in Table 1.

4.1. Algorithm performance

Analysis of each layer requires approximately 4 s on a single 4.00 GHz i7-4770 processor; the analysis computer has 32 GB of installed RAM. At the time of submission, 29 builds had been analyzed, the largest of which is 5452 layers tall (at 20 $\mu\text{m}/\text{layer}$). Due to the substantial burden posed by 3D data, a compression algorithm was written such that the largest build analysis is only 60 MB on the hard disk and 2 GB loaded into RAM. Decompression for viewing detailed analysis results requires 20 milliseconds per layer on the computer specified above. Extraction of the CAD information from the EOSPRINT software (Section 2.4) requires approximately 3 s per layer. Builds of multiple material systems have been analyzed, including: Ti-6Al-4V (four powder types), AlSi10Mg, Inconel 718 (two powder types), stainless steel 316L, stainless steel 17-4, and a bronze alloy. The training database includes samples from most of these material systems; the algorithm has proved robust and appears to perform consistently (without the need for retraining) across all of these material systems.

It is common to evaluate machine learning algorithms using a metric known as a *confusion matrix* [34]. Ground truths were determined by manually labeling (with anomaly classifications) hundreds of thousands of pixels across twenty representative powder bed images. None of the representative powder bed images were included in the training process; such a separate dataset may be considered similar to a validation dataset. These ground truth labels at each pixel were then compared to the labels supplied by the powder bed monitoring algorithm. The results are presented in Figs. 5 and 6. As more builds with unique and anomaly-rich layers are analyzed, the *confusion matrices* can be updated.

The values along the diagonal of the left *confusion matrix* (Fig. 5) represent the percentage of pixels for which the correct anomaly classifications were guessed by the algorithm, e.g. 83.4% of pixels identified by a human as part failure were correctly classified by the algorithm. Most of the incorrectly classified part failure pixels were misidentified as, or "confused by," anomaly-free/okay regions (7.0%) or super-elevation regions (7.8%). The values along the diagonal of the right *confusion matrix* (Fig. 6) represent the percentage of pixels for which a label guess was correct, e.g. 70.7% of pixels classified by the algorithm as part failure were truly part failure anomalies (as labeled by a human). Most of the pixels incorrectly classified by the algorithm as part failure were truly super-elevation pixels (28.9%). An algorithm that is more aggressive



Fig. 5. (left): A confusion matrix showing the percentage of each ground-truth anomaly that was guessed correctly; the absolute numbers of pixels labeled as each anomaly type are shown in parentheses on the vertical axis.

in classifying a particular anomaly will produce a higher diagonal-element percentage in the left confusion matrix for that anomaly, but often a lower diagonal-element percentage in the right confusion matrix for that same anomaly. Essentially, the left confusion matrix indicates how well the algorithm detects each anomaly while the right confusion matrix indicates how well the algorithm avoids producing false-detections of each anomaly.

A common metric for algorithm performance is the total guessing accuracy (98%); because the overwhelming majority of tested pixels are anomaly-free/okay (see the parenthetical numbers on the vertical axis of Fig. 5), the authors do not consider this metric to be fully informative in this situation. Therefore, three additional performance metrics are also considered. The total guessing accuracy among the anomalies (exempting the anomaly-free rows and columns) is 95%, e.g. if the algorithm correctly identifies that an anomaly exists, in 95% of cases it will classify that anomaly correctly. The algorithm is able to correctly detect that no anomaly is present (i.e. that the pixel is anomaly-free) in 100% of cases (to three reported significant figures). Finally, the algorithm is able to correctly detect that some anomaly is present (i.e. that the pixel



Fig. 6. (right): A confusion matrix showing the percentage of anomaly guesses made by the algorithm that were correct. This can be considered a measure of how well the algorithm avoids false positives.

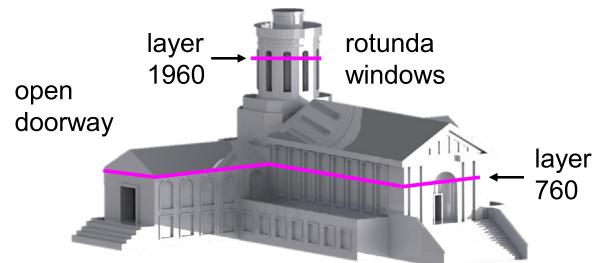


Fig. 7. A CAD model of Hamerschlag Hall, the home of CMU's Additive Manufacturing Laboratory. Two sub-scale models of these were built out of Ti-6Al-4V on a single build plate.

is not anomaly-free) in 89% of cases. The algorithm has the greatest difficulty detecting recoater streaking (50.6% accuracy). This is likely the result of several factors including the low number of available training images for this anomaly type, the small area (in pixels) that this anomaly typically occupies on a powder layer (see Fig. 3), and its frequent co-location with the debris anomaly type. More

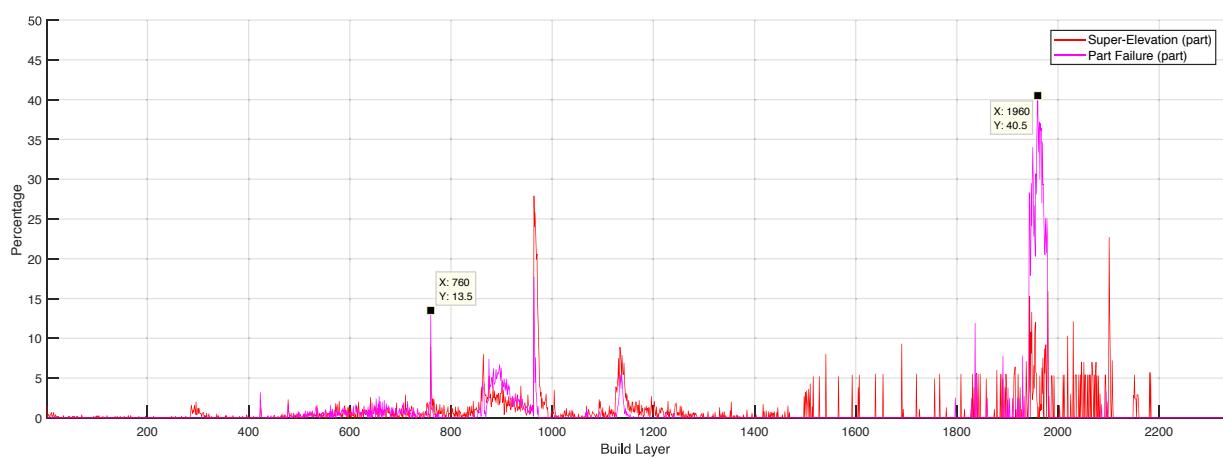


Fig. 8. A global build report showing the number of pixels (as a percentage of part area, based on the CAD model) classified as super-elevation and part failure anomalies at each layer of the build.

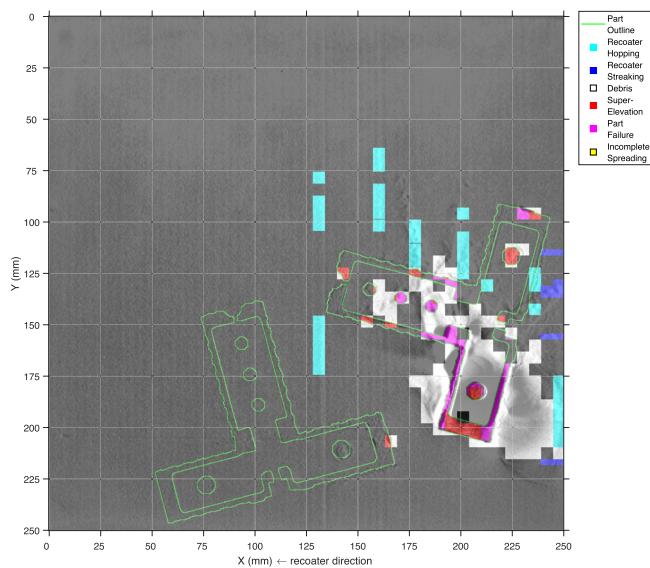


Fig. 9. Layer 760 (22.80 mm above the build plate). The green pixels show the CAD outline of the parts at that layer. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

information about interpreting *confusion matrices* can be found in [34].

4.2. Hamerschlag Hall model - delamination and overhangs

After each layer has been analyzed, the percentage of each anomaly classification in each layer can be displayed as a function of build height in a *build report*. Fig. 8 shows a simplified *global build report* for the construction of two model Hamerschlag Halls (Fig. 7). Anomaly detection spikes are visible at several layers; two of these layers, 760 and 1960 are shown in Figs. 9 and 10, respectively.

The anomaly spike at layer 760 (Fig. 9) is a detection of a delamination that occurred during an accidental cool-down of the EOS build chamber. At this layer, the residual thermal stresses within the right Hamerschlag Hall were sufficient to tear it from some of the supports anchoring it to the build plate. This sudden delamination “tossed” powder off of the part, and this lack

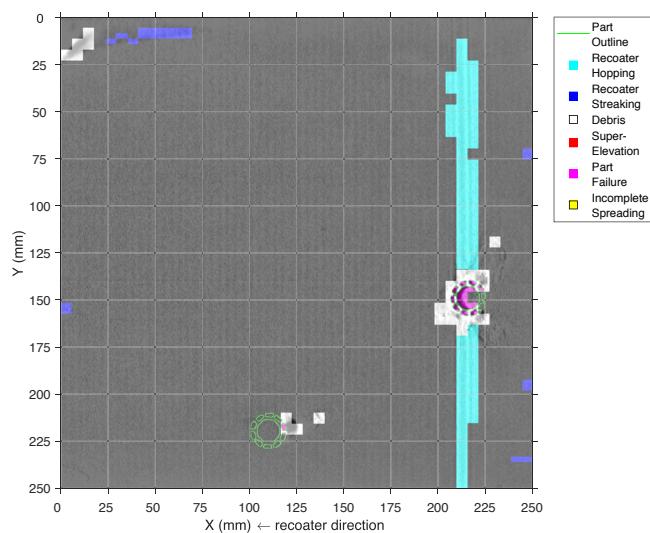


Fig. 10. Layer 1960 (58.80 mm above the build plate). The green pixels show the CAD outline of the parts at that layer. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

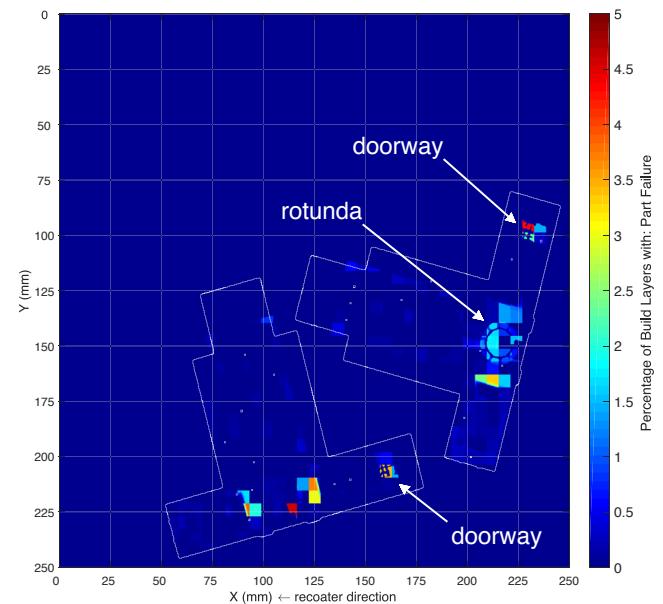


Fig. 11. A heat map showing the percentage of layers (throughout the build height) in which a part failure was detected at each pixel. The positions of the parts on the build plate are shown as white outlines.

of powder was correctly classified as super-elevation (red) and part failure (magenta). The disturbances to the surrounding powder layer were correctly classified as debris (white). Note that the left Hamerschlag Hall does not experience the same issue at this layer. At layer 1960 (Fig. 10) the algorithm detects multiple part failures (magenta) which correspond to observed collapses in the rotunda windows (overhang regions) in the final part. The recoater impacted the region undergoing part failure leading to detectable levels of recoater hopping (vertical teal line).

Note that part failures were primarily detected on the right Hamerschlag model and this is consistent with the post analysis of the final parts. This difference is particularly observable in Fig. 11; note that the rotunda on the right model has a high percentage of

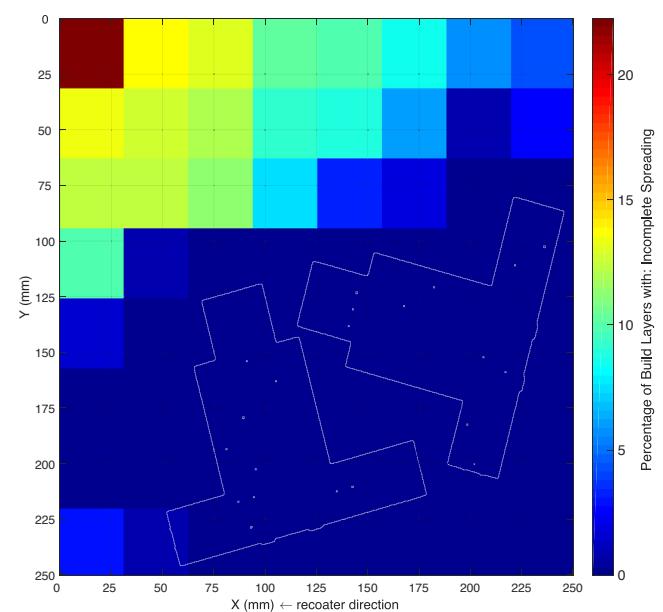


Fig. 12. A heat map showing the percentage of layers (throughout the build height) in which an incomplete spreading anomaly was detected at each pixel. The positions of the parts on the build plate are shown as white outlines.

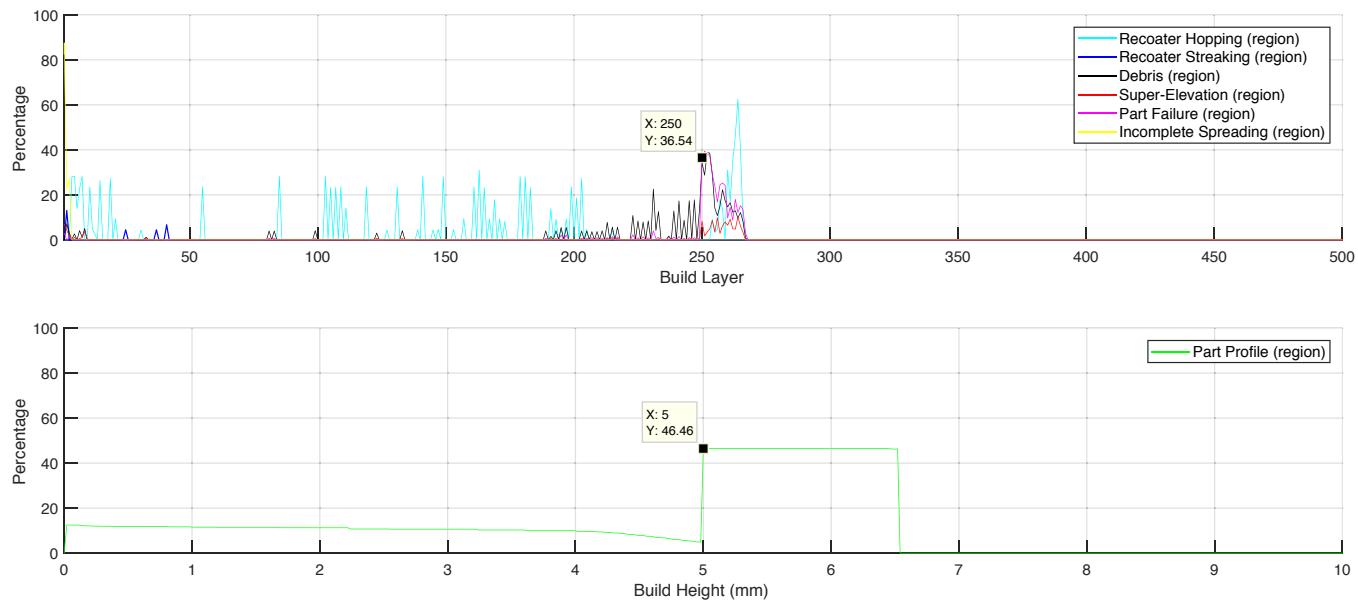


Fig. 13. Local build report and vertical part profile for a horizontally-oriented tensile bar.

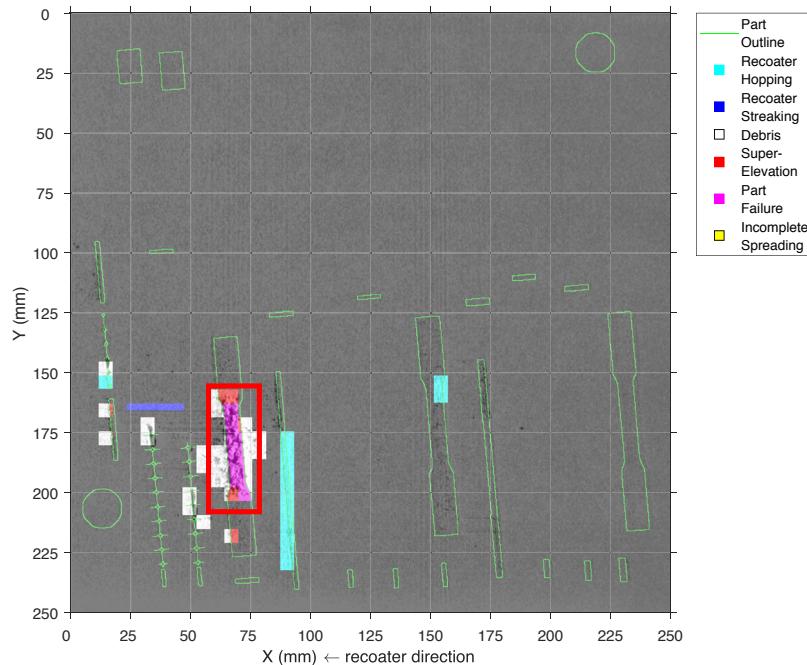


Fig. 14. (left): Note the part failures (magenta) detected at layer 250 within the red bounding box. The green pixels show the CAD outline of the parts at that layer. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

detected part failures. The other regions where a large number of part failures were detected correspond to the open doorways of both models, these doorways represent significant overhangs and the quality of the final part was noticeably poor in those locations. In Fig. 12, a significant number of incomplete spreading anomalies are observed for this build. This is an accurate observation, due to insufficient available powder, the powder dosing factor (amount of powder fetched from the dispenser (Fig. 1) each layer) was set quite low for the duration of this build. None of the incomplete spreading anomalies extended into regions of the build plate containing the Hamerschlag Hall models. Assuming sufficient powder supplies, such an issue could be corrected by a feed-back control system which would increase the powder dosing factor, ideally before any parts were negatively affected.

4.3. Tensile bars – building high-Aspect ratio structures

Multiple tensile bars were built out of stainless steel 316L, in varying orientations and with different support schemes, for the Bettis Naval Nuclear Laboratory. Fig. 8 showed a *global build report*, containing information about an entire powder bed. In a build with many different parts, this becomes difficult to interpret. Fig. 13 and 16 show *local build reports* that include only anomaly detections relevant to specific tensile bars – one specimen built horizontally, the other built vertically. Note that while all of the tensile bars (23) and cylindrical witness coupons (2) are visible as green outlines in Figs. 14 and 17, the specific tensile bars being discussed are bounded by red boxes.

The horizontally-oriented tensile bar used minimal support structures underneath the gage section in an attempt to reduce their influence on the properties of the final test specimen. From Fig. 13 it is clear that significant part failures (magenta, Fig. 14) are detected as soon as the part transitions from the support material to the tensile bar itself. This occurs at 5 mm of build height, or layer 250. The lower subplot (green) of Fig. 13 shows the percentage of pixels within the red bounding box (Fig. 14) that lie on top of the tensile bar itself (based on CAD data, Section 2.4) as a function of build height. In other words, the lower subplot shows the percentage of the bounded area that is being fused at that layer. Recognize that an increase in this percentage represents an overhang region, and a sharp increase implies a largely unsupported overhang. It is inferred that the support structures are insufficient for the substantial overhangs presented by the horizontally-oriented tensile bar. As a result, part failure occurs once the recoater blade passes back over that region of the powder bed, striking the first layer of the tensile bar (above the supports). Fig. 15 shows the as-built horizontally-oriented tensile bar.

From Fig. 16 it is apparent that the vertically-oriented tensile bar built well throughout most of its height, but part failures (magenta) are suddenly detected around layer 3524 (70.48 mm of build height). This corresponds to the tensile bar transitioning from the gage section to the upper grip section, a period during which a delicate overhang is produced. Based on Figs. 17–19, it is surmised that the un-supported overhang warped upwards (due to residual thermal stress) enough to be impacted by the recoater blade. This impact is highlighted by the extensive recoater hopping detections (vertical teal line). The impact caused the tensile bar to bend and then “spring back,” tossing powder away and leaving a powder cavity to the right of the tensile bar (highlighted by debris detections in white). This cavity prevented proper powder coverage of this region in subsequent layers, worsening the situation.

4.4. Impeller blades – build geometry orientation with respect to the recoater blade

An impeller was built out of stainless steel 316L to demonstrate the capabilities of the EOS M290 and as part of CMU's NextManu-



Fig. 15. (right): The as-built horizontal tensile bar.

facturing industry training initiative. The thin impeller blades and 20 μm thick layers make this part a challenge to build successfully.

Fig. 20 shows the as-built impeller blade. Note that one half of the part built correctly while the blades on the other half collapsed. The algorithm was employed to determine the possible reason(s) for the partial build failure and identify any potential strategies for improving build quality in the future.

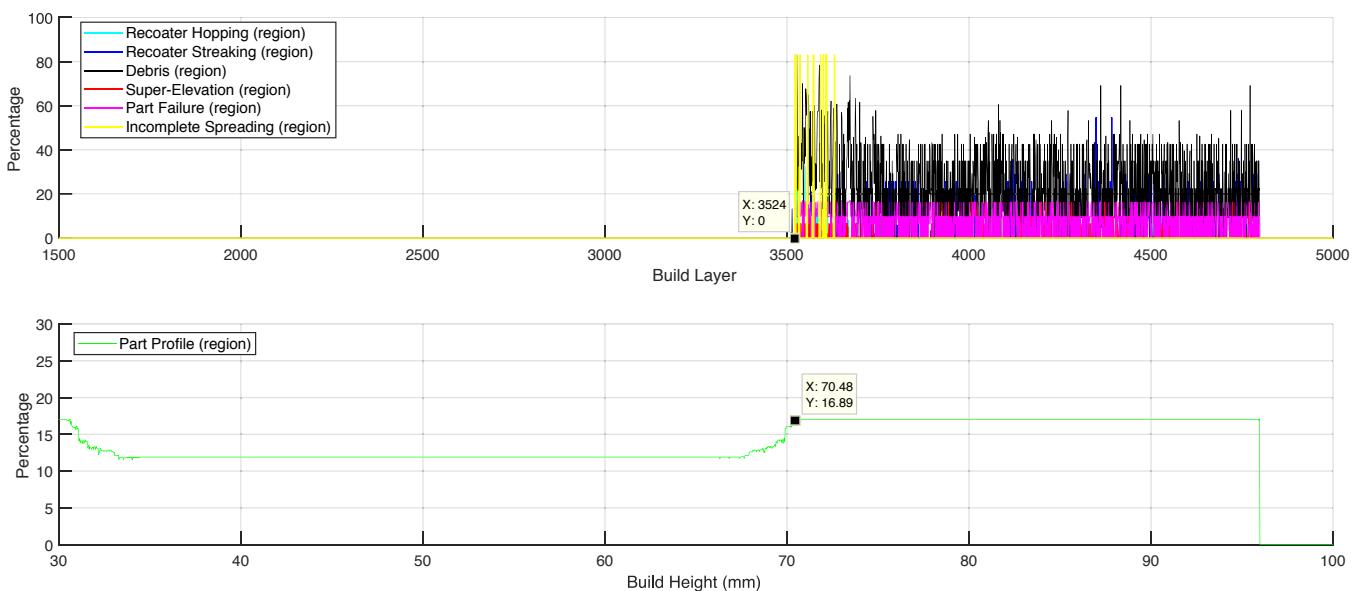


Fig. 16. Local build report and vertical part profile for a vertically-oriented tensile bar.

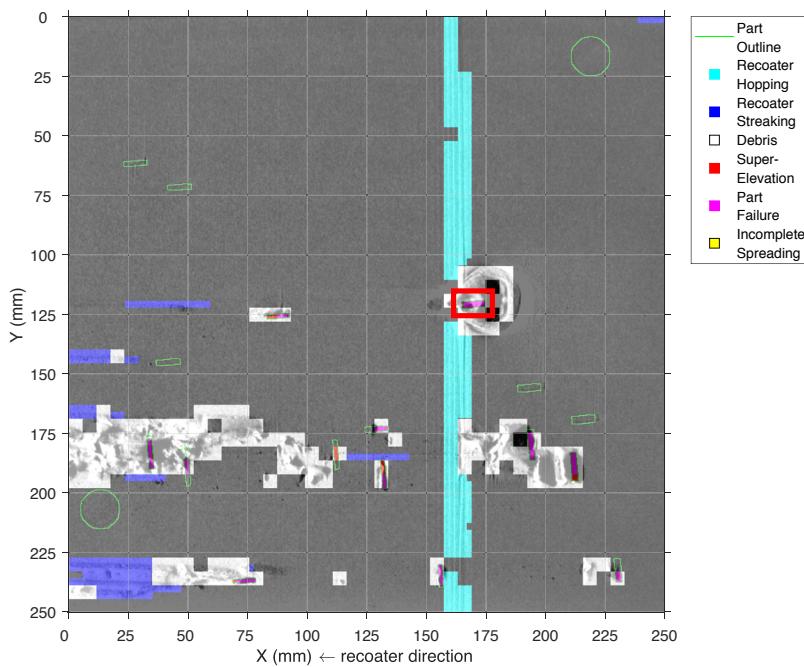


Fig. 17. (left): Note the part failures (magenta) detected at layer 3524 within the red bounding box. The green pixels show the CAD outline of the parts at that layer. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

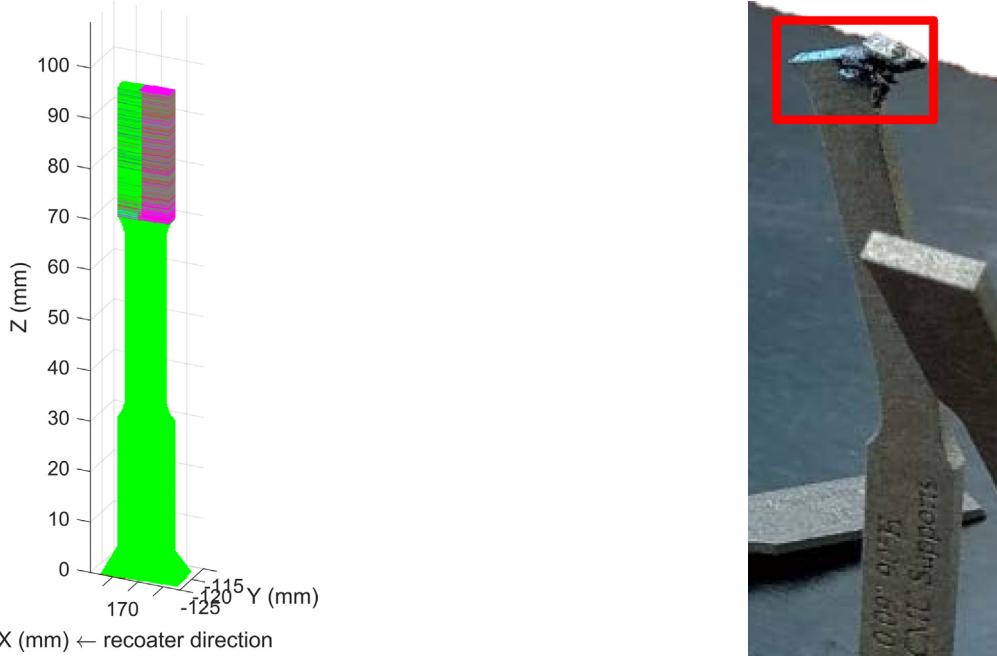


Fig. 18. (center): A 3D rendering of the vertical tensile bar with part failures highlighted in magenta.

While it was immediately clear that the thin impeller blades failed due to repeated impacts with the recoater blade, it was initially unclear what caused only the impeller blades on one half to fail.

Fig. 21 makes it clear that super-elevation occurred on both halves of the impeller – suggesting that the recoater blade likely impacted most of the impeller blades, including many that did not fail. Fig. 22 highlights the impeller blades that collapsed during the build, note that only impeller blades with their leading-edge pointed away from the direction of the incoming recoater blade (Orientation 2) failed to build correctly.

Fig. 19. (right): The as-built vertical tensile bar with the visible defects bounded by a red box. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Based on Figs. 21 and 22, the authors hypothesize that some plastic deformation occurred when the impeller blades were struck by the recoater. For the impeller blades that built correctly (Orien-



Fig. 20. The as-built impeller after being cut with a wire EDM to preserve only the half that built correctly.

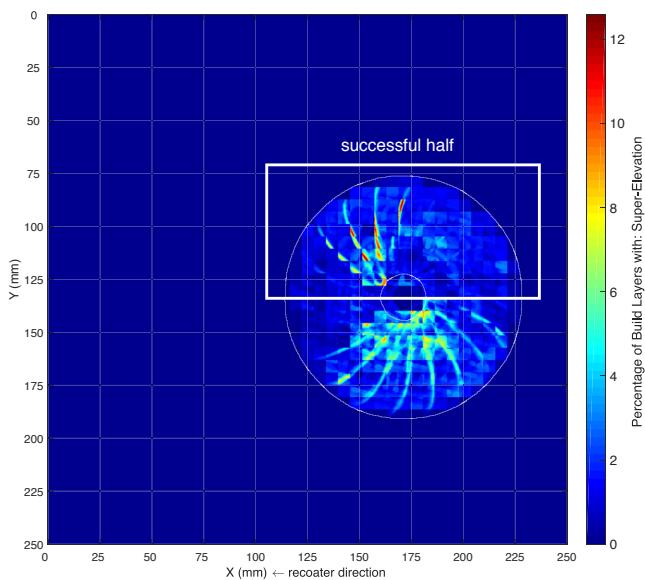


Fig. 21. A heat map showing the percentage of layers (throughout the build height) in which super-elevation was detected at each pixel. The white box encloses the half of the impeller that built correctly. The positions of the parts on the build plate are shown as white outlines.

tation 1), this deformation would be in approximately the same direction as the shift in blade geometry in the subsequent layer, while the deformation and geometry shift would be in opposite directions for blades in Orientation 2. The “geometry shifts” refer to the discrete, layer-wise, in-plane shifts that are used to construct the smooth 3D curvature of the impeller. As a result, plastic deformation inflicted on the Orientation 1 impeller blades would be less likely to cause cascading failures in subsequent layers than the same deformation inflicted on the Orientation 2 impeller blades. Because of this analysis of the algorithm results, the impellers built for the subsequent industry training event were reduced to only the half oriented in the more robust direction; complete collapses of the impeller blades due to recoater impacts were not observed for this subsequent build. Successfully building an entire impeller of this design would likely require increasing the thickness of the impeller blades and/or increasing the layer thickness to reduce the severity of the recoater impacts caused by super-elevation.

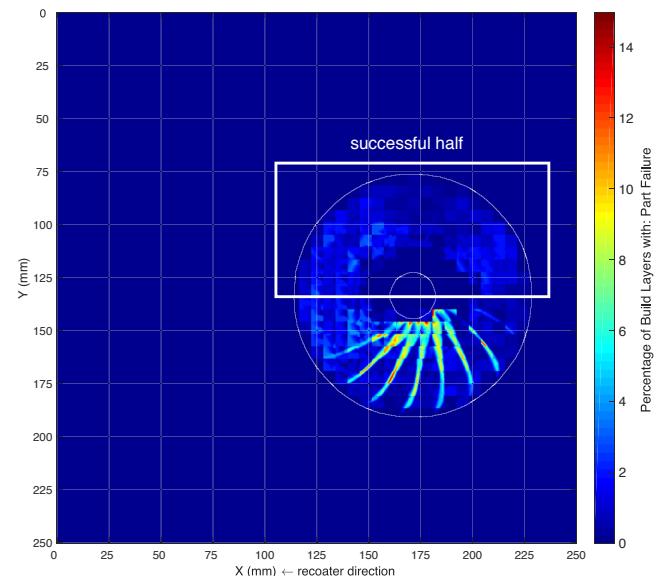


Fig. 22. A heat map showing the percentage of layers (throughout the build height) in which a part failure was detected at each pixel. The white box encloses the half of the impeller that built correctly. The positions of the parts on the build plate are shown as white outlines.

5. Conclusions

An algorithm was developed that is capable of autonomously detecting and classifying consequential anomalies related to the powder spreading process in an EOS LPBF machine. This algorithm is a novel application of contemporary machine learning and computer vision techniques to Additive Manufacturing. Only hardware standard to the EOS M290 was used to capture and illuminate the images analyzed by the algorithm. In its current configuration the algorithm functions well as a post-build analysis tool, allowing a user to identify failure modes and locate regions within a final part that may contain macroscopic (millimeter-scale) flaws. Given the relatively low computational burden and execution time of the software, it is likely that a similar algorithm could be implemented for real-time process monitoring as part of a feedback control loop for the building process.

The presented algorithm has proved to be a robust and invaluable tool; it is currently used to analyze every EOS M290 build performed at Carnegie Mellon University. It has correctly predicted locations of poor build quality in a final part, and demonstrated the ability to detect anomalies that could be corrected with a feedback control system (Section 4.2). It was used extensively to analyze the effectiveness of different support structure schemes as part of a feasibility study for an external customer (Section 4.3). The designs of subsequent builds were informed based on the algorithm’s findings (Sections 4.3 and 4.4). Perhaps most promisingly, the algorithm has demonstrated the ability to detect possible flaws and anomaly correlations within a final part that would not otherwise have been noticed, including cyclical anomaly behavior that appears to be related to EOS’ rotating laser scan pattern [35].

Before the algorithm is ready to be deployed in a real-time in-situ monitoring environment, its classification accuracy should be improved. Ideally its ability to detect each anomaly should increase, without increasing the probability of a false-detection; operators will likely be reticent to use monitoring software that is overly-aggressive in suggesting or applying countermeasures. A higher resolution camera and an additional lighting source, mutually perpendicular to the current light source and the camera axis, may be beneficial [7,13,16]. Given the effectiveness of the data compression algorithm, the authors do not anticipate any immedi-

ate restrictions on increasing the camera resolution. The authors strongly suspect that accuracy can also be improved by implementing deep learning [36] to develop a more effective *filter bank* (Fig. 4a). The deployment of the algorithm as a real-time powder bed monitoring tool will be pursued by the authors in the near future.

Acknowledgements

The authors would like to acknowledge Bettis Naval Nuclear Laboratory for sponsoring one of the builds presented in this paper as well as Prof. Deva Ramanan (CMU) for teaching a course (16-720) on computer vision that was used as a foundation for some of this work. The authors would like to recognize Michelle Ma, Eric Myers, and Ryan Pearce who created the original CAD model of Hamerschlag Hall as their final project for the class “Additive Manufacturing for Engineers” (24-210) at CMU. The authors would like to thank Colt Montgomery (CMU) for his role in running many of the builds analyzed as part of this project as well as his assistance in user-testing the powder bed monitoring software. Finally, the authors would like to thank Dr. Sneha Prabha Narra and Brian Fisher of CMU for their continuing advice on this topic. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The Additive Manufacturing terminology used herein complies as closely as possible with ISO/ASTM 52900:2015 [37].

References

- [1] M. Bourell, D. Leu, Roadmap for additive manufacturing: identifying the future of freeform processing, in: Solid Freeform Fabrication Proceedings, Austin, TX, 2009, <http://wohlersassociates.com/roadmap2009A.pdf> (Accessed 23, May 2017).
- [2] K.M. Taminger, R.A. Hafley, Electron beam freeform fabrication for cost effective near-net shape manufacturing, NATO/RTOAVT-139 Specialist's Meeting on Cost Effective Manufacturing via Net Shape Processing (2006).
- [3] U.S. Navy, Shipboard additive manufacturing (AM)/3D printing SBIR 2016.1 – N161-038, 2016. <http://www.navysbir.com/n16.1/N161-038.htm> (Accessed 01, January 2016).
- [4] T.G. Spears, S.A. Gold, In-process sensing in selective laser melting (SLM) additive manufacturing, Integr. Mater. Manuf. Innov. 5 (2016) 1–25, <http://dx.doi.org/10.1186/s40192-016-0045-4>.
- [5] G. Tapia, A. Elwany, A review on process monitoring and control in metal-based additive manufacturing, J. Manuf. Sci. Eng. 136 (2014), <http://dx.doi.org/10.1115/1.4028540>.
- [6] S.K. Evertson, M. Hirsch, P. Stravroulakis, R.K. Leach, A.T. Clare, Review of in-situ process monitoring and in-situ metrology for metal additive manufacturing, JMADE 95 (2016) 431–445, <http://dx.doi.org/10.1016/j.matedes.2016.01.099>.
- [7] T. Craeghs, S. Clijsters, E. Yasa, J.-P. Kruth, Online quality control of selective laser melting, Solid Freeform Fabrication Proceedings (2011) 212–226 <http://utwired.engl.utexas.edu/lff/symposium/proceedingsarchive/pubs/Manuscripts/2011/2011-17-Craeghs.pdf>.
- [8] P. Lott, H. Schleifenzbaum, W. Meiners, K. Wissenbach, C. Hinke, J. Bültmann, Design of an optical system for the in situ process monitoring of Selective Laser Melting (SLM), Phys. Procedia 12 (2011) 683–690, <http://dx.doi.org/10.1016/j.phpro.2011.03.085>.
- [9] A.R. Nassar, E.W. Reutzel, J.P. Brown, J.P. Morgan, D.J. Natale, R.L. Tutwiler, D.P. Feck, J.C. Banks, Sensing for directed energy deposition and powder bed fusion additive manufacturing at Penn State University, Laser 3D Manufacturing III of SPIE 9738 (2016), <http://dx.doi.org/10.1117/12.2217855>.
- [10] F. Rodriguez, D. Medina, C. Espalin, D. Terrazas, C. Muse, R.B. Henry, Integration of a thermal imaging feedback control system in electron beam melting, in: Solid Freeform Fabrication Proceedings, Austin, TX, 2012, pp. 945–961.
- [11] S. zur Jacobsmuhlen, D. Kleszczynski, G. Schneider, High resolution imaging for inspection of laser beam melting systems, IEEE International Instrumentation and Measurement Technology Conference (I2MTC) (2013) 707–712, <http://dx.doi.org/10.1109/I2MTC.2013.6555507>.
- [12] S. Zur Jacobsmuhlen, G. Kleszczynski, D. Witt, Detection of elevated regions in surface images from laser beam melting processes, Conference of the IEEE Industrial Electronics Society (IECON) (2016) 1270–1275, <http://dx.doi.org/10.1109/IECON.2015.7392275>.
- [13] M. Abdelrahman, E.W. Reutzel, A.R. Nassar, T.L. Starr, Flaw detection in powder bed fusion using optical imaging, Addit. Manuf. 15 (2017) 1–11, <http://dx.doi.org/10.1016/j.addma.2017.02.001>.
- [14] J. Schwerdtfeger, R.F. Singer, C. Korner, In Situ flaw detection by IR imaging during electron beam melting, Rapid Prototyp. J. 18 (2012) 259–263, <http://dx.doi.org/10.1108/13552541211231572>.
- [15] E.W. Foster, A.R. Reutzel, B.T. Nassar, S.W. Hall, C.J. Brown, Optical, layerwise monitoring of powder bed fusion, in: Solid Freeform Fabrication, Austin, TX, 2015, pp. 295–307, <http://dx.doi.org/10.1017/CBO9781107415324.004>.
- [16] M. Grasso, B.M. Colosimo, Process defects and in situ monitoring methods in metal powder bed fusion: a review, Meas. Sci. Technol. 28 (2017) 44005, <http://dx.doi.org/10.1088/1361-6501/aa5c4f>.
- [17] M. Aminzadeh, A Machine Vision System for In-Situ Quality Inspection in Metal Powder-Bed Additive Manufacturing, 2016.
- [18] C. Scott, EOS Introduces EOSTATE Exposure OT, First Commercial Optical Tomography System for Additive Manufacturing, 3DPrint.com, 2017. <https://3dprint.com/178624/eos-eostate-exposure-ot>.
- [19] J. Xiao, K.A. Hays, A. Ehinger, SUN database: large-scale scene recognition from abbey to zoo, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2010) 3485–3492.
- [20] J. Winn, A. Criminisi, T. Minka, Object categorization by learned universal dictionary, International Conference on Computer Vision (ICCV) (2005).
- [21] E. Dubrofsky, Homography Estimation, The University of Vancouver, 2007, <http://dx.doi.org/10.1117/1.3364071>.
- [22] ISO International, ISO-12233:2014 Photography – Electronic still picture imaging – Resolution and spatial frequency responses, 2014. <https://www.iso.org/standard/59419.html>.
- [23] K. Zeng, N. Patil, H. Gu, H. Gong, Layer by layer validation of geometrical accuracy in additive manufacturing processes, in: Solid Freeform Fabrication, Austin, TX, 2013, pp. 76–87 <http://utwired.engl.utexas.edu/lff/symposium/proceedingsarchive/pubs/Manuscripts/2013/2013-06-Zeng.pdf>.
- [24] EOS GmbH, EOS Software for Additive Manufacturing, n.d. https://www.eos.info/systems_solutions/software (Accessed 31, May 2017).
- [25] G. Csurka, C. Dance, L. Fan, J. Willamowski, B. Cedric, Visual categorization with bags of keypoints, Workshop on Statistical Learning in Computer Vision (ECCV) (2004) 1–22 <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.72.604>.
- [26] D.G. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the International Conference on Computer Vision, Washington, D.C., 1999, p. 1150.
- [27] B.L. DeCost, H. Jain, A.D. Rollett, E.A. Holm, Computer vision and machine learning for autonomous characterization of AM powder feedstocks, J. Mater. 69 (2017) 456–465, <http://download.springer.com/static/pdf/233/art%253A10.1007%252F252Fs11837-016-2226-1.pdf?originUrl=http%3A%2F%2Flink.springer.com%2Farticle%2F10.1007%2Fs11837-016-2226-1&token2=exp=14966891043~acl=%2Fstatic%2Fpdf%2F233%2Fart%25253A10.1007%25252F252Fs11837-016-222> (Accessed 07, June 2017).
- [28] S.W. Smith, The Scientist and Engineer's Guide to Digital Signal Processing, 1st ed., California Technical Publishing, 1997 <http://www.dspsguide.com/>.
- [29] MathWorks, imfilter, 2017. https://www.mathworks.com/help/images/ref/imfilter.html?s_tid=gn_loc_drop (Accessed 04, January 2017).
- [30] MathWorks, gabor, 2017. <https://www.mathworks.com/help/images/ref/gabor.html> (Accessed 04, July 2017).
- [31] MathWorks, k-Means Clustering, 2016. <https://www.mathworks.com/help/stats/k-means-clustering.html> (Accessed 30, December 2016).
- [32] MathWorks, bsxfun, 2017. <https://www.mathworks.com/help/matlab/ref/bsxfun.html> (Accessed 04, July 2017).
- [33] M.K. Smith, Overfitting, The University of Texas at Austin. n.d. <http://www.ma.utexas.edu/users/mks/statmistakes/overfitting.html> (Accessed 30, December 2016).
- [34] T. Fawcett, An introduction to ROC analysis, Pattern Recognit. Lett. 27 (2005) 861–874, <http://dx.doi.org/10.1016/j.patrec.2005.10.010>.
- [35] A. Anam, J.J.S. Dilip, D. Pal, B. Stucker, Effect of scan pattern on the microstructural evolution of Inconel 625 during Selective Laser Melting, Solid Freeform Fabrication (2014) 363–376, <http://dx.doi.org/10.13140/2.1.1256.6089>.
- [36] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444, <http://dx.doi.org/10.1038/nature14539>.
- [37] ASTM, Standard Terminology for Additive Manufacturing, 2015. <https://compass.astm.org/download/ISOASTM52900.23551.pdf> (Accessed 03, May 2017).