Classification of Microstructural Defects in Selective Laser Melted Inconel 713C alloy using Convolutional Neural Networks

Ethan L. Edmunds and Meurig Thomas\*

Interdisciplinary Programmes in Engineering, University of Sheffield, Mappin Street, Sheffield, S1 3JD, United Kingdom

\*Corresponding Author ([meurig.thomas@sheffield.ac.uk](mailto:meurig.thomas@sheffield.ac.uk))

**Abstract**

Microstructural defects are commonly found in additively manufactured materials and can have significant effects on the material’s bulk properties. This warrants defect detection and classification during microstructural evaluation, which is often laborious, costly, and can yield sub-optimal results if done manually. Previous attempts to facilitate automated classification in additively manufactured nickel-alloys have used supervised machine learning methods, such as kth-nearest neighbour classification and decision trees. This study proposes and evaluates the use of convolutional neural networks for microstructural defect classification in selective laser melted Inconel 713C samples. It outlines the process used to create, augment and expand the dataset, as well as hyperparameter selection for the neural network architecture to yield optimal classification performance.

**Keywords:** Additive Manufacturing; Nickel Alloys; Microstructural Defects; Machine Learning; Deep Learning

**1. Introduction**

Additive manufacturing (AM) is a class of manufacturing processes, that enables the creation of components, layer-by-layer using a high-energy heat source, from three-dimensional models. The characteristics of AM lead to various advantages when compared to traditional manufacturing processes. For instance, AM enables the production of intricately complex parts with enhanced design flexibility, optimises structural topology to improve mechanical properties, consolidates multiple components into single units to reduce weight and cost, and supports on-demand production, collectively enhancing efficiency, cost-effectiveness, and environmental sustainability (1–3). While these advantages are significant, further exploration is needed to deepen our understanding of the 'process-microstructure-property' relationship in AM, which would enhance the design and optimisation of materials and manufacturing processes (4). In particular, components produced using AM methods are typically prone to the presence of microstructural defects, such as lack of fusion defects, gas porosities, solidification cracking and keyhole collapses, which pose risks in structural applications, especially when under cyclic loading conditions (5,6). For these reasons, studying the “process-microstructure-property” relationship is vital to advancing the performance of engineering components. A part of the investigation of these relationships is correctly detecting and classifying microstructural defects from a variety of microstructural images. However, when performed manually, the detection and classification of microstructural defects can often be laborious, costly and yield suboptimal results, suggesting the need for an alternative strategy to detection and classification (7).

Machine Learning (ML) is a branch of Artificial Intelligence that offers a solution to data analysis in the field of materials science. It focuses on pattern identification and cognitive acquisition concepts, which optimise performance criteria based on example data or past experiences (8,9). In the field of AM, supervised and unsupervised machine learning methods have been used for a variety of microstructural detection and classification applications. For instance, unsupervised learning has been used to classify gas pores, keyhole pores resulting from excessive energy input, and lack of fusion pores caused by insufficient energy (10). Unsupervised machine learning has been employed to identify and classify anomalies in a laser powder bed additive manufacturing process using a moderately-sized training database of image patches (11). K-means clustering, a supervised ML model, and support vector machines, a unsupervised ML model, have been used to detect porosities in thin-walled aluminium alloy structures with an average classification accuracy of 99% (12).

Deep Learning, another subset of ML algorithms, has been pivotal in advancing the state-of-the-art across various domains, including speech and visual object detection and classification (13,14). Deep learning algorithms are more capable at learning complicated patterns from more complex datasets, leading to their increased use in materials science applications. For instance, it has been used to distinguish between metallic morphologies with dendrites and those without (15). It has also been used for lath-bainite segmentation in complex-phase steel, achieving an accuracy of 90% which rivals the accuracy of expert segmentations (16). Deep learning has also found use in the design of new materials and to link microstructural characteristics to the material’s bulk properties (17,18). For an overview on the previous use of deep learning approaches in microstructural characterisation, the reader is directed to these resources for more information (16,19,20).

In particular for AM…

This paper builds on the work of Aziz et al. (21) who used k-th Nearest Neighbours (kNN) and decision tree algorithms to classify defects found in a variety of nickel-alloys with 89.8% and 90-92% accuracy respectively (21). In this study, convolutional neural networks (CNNs) are used to automatically classify the different types of defects that are present in Inconel 713C alloy. It will outline the process to sourcing binary images of individual defects from micrographs of the alloy samples, the preprocessing that the dataset underwent, as well as the methodologies used for CNN optimisation including hyperparameter selection and data augmentation. It will then evaluate the efficacy of neural networks as a method to classifying defects, comparing the network’s performance to that of the supervised machine learning methods used by Aziz. et al. (21).

**2. Methodology**

*2.1 Dataset Creation and Preparation*

A range of micrographs of the IN713C alloy samples were produced using selective laser melting (SLM) during a research program at the University of Sheffield, with varying settings for power, beam velocity and hatch spacing (22). The micrographs, an example of which is shown in Fig. 1, were then imported into MATLAB and converted into a binary image, clearly showing the microstructural defects. The scale, shown on the bottom, along with defects that intersected the border of the micrograph were removed, to ensure the quality of the dataset. In addition, defects that were too small (below 10 pixels in area), were excluded as low-resolution images made manual classification challenging. After filtering unwanted data, individual binary images of the defects in the micrographs were created, as illustrated in Fig 1b. The images were categorised into specific classes: ‘crack’, ‘pore’, ‘lack of fusion’, and ‘pore with crack’ defects. This process was applied to 18 micrographs, yielding a dataset containing 4800 binary images of microstructural defects. In this dataset, there were 1807 ‘crack’ defects, 900 ‘lack of fusion’ defects, 1830 ‘pore’ defects, and 263 ‘pore with crack’ defects.

A close-up of a grey and white speckled surface

Description automatically generated

Figure 1: Example micrograph used to create the dataset, with example binary images of each defect type that were found in the micrographs. This includes b) ‘crack’, c) ‘pore’, d) ‘lack of fusion, and e) ‘pore with crack’ defects.

*2.2 Data Augmentation*

The defect dataset images were standardised to a size of 200x200 pixels. It should be noted that resizing images can lead to some loss or distortion of the original information, as shrinking an image may lead to **loss of detail**, and enlarging an image may lead to **pixelation** (23). As a result, this may **affect the accuracy of subsequent image analysis or machine learning models that rely on high-quality data**. Data augmentation was also applied to the dataset. Deep learning models typically demand extensive datasets to gather sufficient information and patterns for effective training and performance. Data augmentation can support with avoiding overfitting to training data, which would result in the model’s poor generalisation performance on unseen samples (24,25). In this work, augmentation involved a random reflection on the X and Y axis of the image, a random rotation of the image from 20 to -20 degrees around its centre, and a random scaling of 0.9 to 1.1. Following augmentation, the dataset was compiled into a training dataset (70%) and a validation dataset (30%).

*2.4 Baseline Model*

To begin, a baseline neural network was created as a foundation model for optimisation. It used a filter size of 5x5 which was kept consistent for all subsequent model training. The baseline model consisted of 4 blocks and 10 filters per block. Each block is of a standard architecture, containing a 2d convolution layer, batch normalisation layer, rectified linear unit (ReLU) layer and a max pooling layer. By varying the number of batches and stacking them on top of each other, we achieve a network with different architectures, modifying its ability to learn the patterns from the training data and improving overall performance.

*2.5 Hyperparameter tuning and Optimisation*

An important feature of designing and optimising neural networks is hyperparameter tuning. In machine learning models, there are two types of parameters to consider: model parameters, which are initialised internally in the model and vary as the machine learning algorithm learns (such as weights of neurons in a neural network), and hyperparameters, which are freely-selectable external parameters that do not vary while the model is learning (26,27). In the context of neural networks, hyperparameters are settings that can be changed to influence the learning process (such as training time, learn rate, or batch size) and the architecture of a neural network (such as number of neurons, number and type of hidden layers) (28). Hyperparameter tuning and optimisation by automated means is useful because it reduces the human effort in identifying the best combination of hyperparameters for best performance, improves the performance of the algorithm by customising it to the given dataset/application, and it improves reproducibility and facilitates comparison between models (29). In this work, two algorithms were used to tune hyperparameters to their optimal configuration: i) Grid-search and ii) Bayesian optimisation.

Grid-search is a basic hyperparameter tuning method where a model is built using each combination of hyperparameters, trained, and then evaluated based on its performance (30). In this grid search, the number of batches and filters in the CNN were explored, to evaluate which combination of these two CNN characteristics yields the optimal performance. A disadvantage to grid search is that as the range of values being explored increases, the computational efficiency of the algorithm drastically decreases. To combat this, each model trained during grid search used 25% of the baseline training dataset, improving computational efficiency of the algorithm. The CNN architecture that yielded the optimal results in grid search was then trained with the full baseline training dataset, to evaluate its performance on the full dataset. During grid search, CNN architectures with 2-7 blocks and 5-25 filters (with increments of 5) were explored.

Bayesian optimisation is a more advanced approach to hyperparameter tuning. The algorithm builds a response surface model using the mean and uncertainty predictions to guide the selection of subsequent data collection (31). Due to its ability to learn from previous iterations, Bayesian optimisation offers a more efficient approach to hyperparameter tuning and Neural Architecture Search when compared to grid search (32). Due to this improvement compared to grid search, the scope of hyperparameters explored during optimisation was widened. This included varying the number of batches, number of filters, initial learn rate, mini batch size, learn rate drop factor, and learn rate drop period.

**3. Results and Discussion**

*3.1 Original Dataset*

Confusion matrices describing the performance of the baseline model and models produced using optimisation methods are shown in Figure 3. Confusion matrices display the counts of actual versus predicted classifications provided by the model, with each cell in the matrix representing the frequency of predictions for each class, including both correct and incorrect predictions. Confusion matrices allow for the evaluation of overall model accuracy and assists with identifying the model’s weaknesses when classifying certain types of defects. Table 1 presents the performance of each model in accurately classifying various defects within the validation dataset along with its total accuracy. The network produced by gridsearch optimisation contained 7 batches and 10 filters, whereas Bayesian optimisation created a model with 8 batches and 26 filters. There was a steady increase in model classification performance as different optimisation methods were used to vary the hyperparameters of the model, with the Bayesian optimised model yielding the highest overall classification performance. This suggests that CNNs with more complex architectures yield better results, as they are better able to capture distinctive features between different types of microstructural defects, which has been found in prior research (33).

Similar to supervised ML models produced by Aziz et al. (21), the CNNs faced consistent challenges in accurately classifying ‘pore with crack’ defects. These defects combine the characteristics of ‘pore’ and ‘crack’ defects, which already exist as separate classes in the dataset. The CNNs struggle due to overlapping features between these classes, making it difficult to distinguish and classify them effectively. 'Pore with crack' defects may overlap with 'pore' or 'crack' defects due to shared visual features such as round void shapes from pores and linear, sharp features from cracks. Moreover, this disparity could also stem from the imbalance in the dataset, where there were only 263 instances of ‘pore with crack’ defects out of a total of 4773 defects. This imbalance may suggest that the model is overfitting to the more prevalent defect types (34,35).

Despite it’s poor performance in correctly classifying ‘pore with crack’ defects, the models showed strong capability at distinguishing between ‘lack of fusion’, ‘pore’ and ‘crack’ defects. The final model produced by Bayesian optimisation showed a overall performance of 91.6%, which is similar to that of models produced by Aziz et al. (21). It should be noted when compared to kNN classification and decision trees, the training and optimisation of CNNs for classification is significantly more time-consuming and computationally intensive. Moreover, the performance of the optimal models do vary each time the model was trained as shown in Figure 2. Consistent with findings from other research, the variability observed in our model's performance across different training sessions underscores the influence of factors such as initial weight settings (36), data shuffling (37), and hyperparameter choices (38). This highlights the importance of training models multiple times and reporting average performance metrics to ensure the reliability and generalisability of the model conclusions. However, given the time-consuming and computationally expensive nature of deep learning training, this process can be particularly laborious and costly. Baseline models converged to roughly 85-88% in classification accuracy, grid search optimised models to 88-91% and Bayesian optimised models to 90-92%. The overall accuracy of models produced by Bayesian optimisation is similar to that of deep learning models used for a variety of other applications in materials science, such as models that classify dendritic and non-dendritic microstructures and models that segment lath-bainite in complex-phase steel, which are both roughly 90% accurate (15,16).

**A diagram of a model

Description automatically generated with medium confidence**

Figure 2: Confusion matrices showing the performance of the baseline, grid-search optimised, and Bayesian optimised models for classifying ‘crack’, ‘pore’, ‘lack of fusion’ and ‘pore with crack’ defects in the original dataset. The validation dataset consists of 1439 instances.

Table 1: Summary of accuracy for baseline, grid-search optimised, and Bayesian optimised models in correctly classifying defects in the original dataset. The validation dataset consists of 1439 instances.

|  |  |  |  |
| --- | --- | --- | --- |
| **Defect Type** | **Baseline Model** | **Grid-search Optimised Model** | **Bayesian Optimised Model** |
| Crack | 93.3% | 96.3% | 96.5% |
| Lack of Fusion | 84.8% | 92.8% | 85.1% |
| Pore | 97.4% | 95.2% | 97.8% |
| Pore with Crack | 5.3% | 9.2% | 35.5% |
| **Total** | **88.6%** | **90.6%** | **91.6%** |

A graph showing a number of data

Description automatically generated with medium confidence

Figure 3: Variation in training accuracy and validation accuracy with number of training iterations for the Baseline model trained on the original dataset.

*3.2 Synthetic Dataset*

To address the poor performance in classifying 'pore with crack' defects, the original dataset was expanded using data augmentation techniques to create a dataset containing ‘synthetic’ data. Data augmentation is a valuable tool in deep learning, enabling the expansion of datasets for improved training and validation (39,40). In this study, the augmentation process included randomly reflecting the images along the X and Y axes, rotating the images between 20 and -20 degrees around their centre, and scaling the images randomly between 0.9 and 1.1. During augmentation, the 'pore with crack' defect class was expanded to 1,052 - 4 times larger than the original size. After augmentation, the data was divided into a training set (70%) and a validation set (30%). The ‘crack’, ‘pore’ and ‘lack of fusion’ classes were not expanded as they were well classified in previous models. The baseline model was trained with 4 batches and 10 filters. Grid-search optimisation identified the optimal network configuration with 8 batches and 20 filters, while Bayesian optimisation yielded a CNN with 9 batches and 28 filters. The confusion matrices for the models' performance are provided in Figure 4, and their accuracies are shown in Table 3.

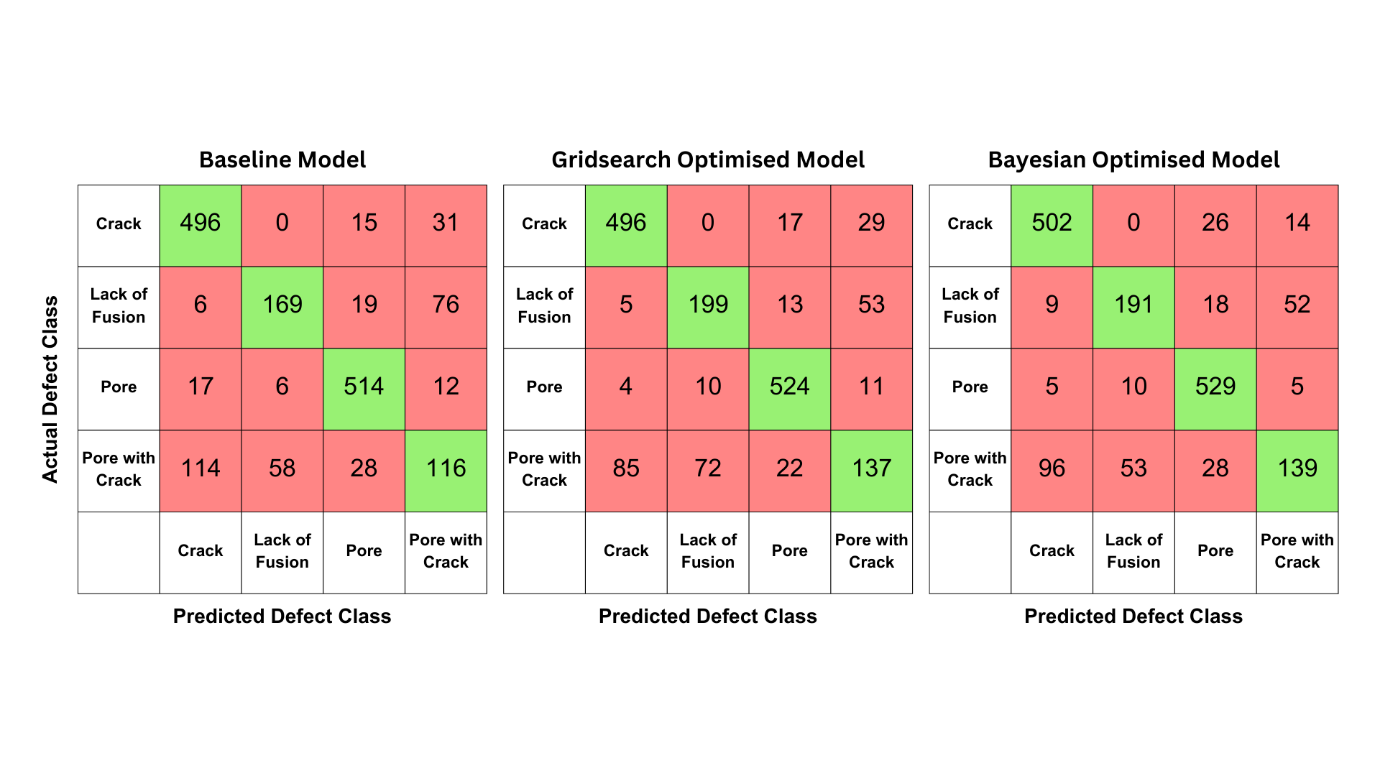


Figure 4: Confusion matrices showing the performance of the baseline, grid-search optimised, and Bayesian optimised models for classifying ‘crack’, ‘pore’, ‘lack of fusion’ and ‘pore with crack’ defects in the synthetic dataset. The validation dataset consists of 1677 instances.

Table 3: Summary of accuracy for baseline, grid-search optimised, and Bayesian optimised models in correctly classifying defects in the synthetic dataset. The validation dataset consists of 1677 instances.

|  |  |  |  |
| --- | --- | --- | --- |
| **Defect Type** | **Baseline Model** | **Grid-search Optimised Model** | **Bayesian Optimised Model** |
| Crack | 91.5% | 91.5% | 92.6% |
| Lack of Fusion | 62.6% | 73.7% | 70.7% |
| Pore | 93.6% | 95.4% | 96.4% |
| Pore with Crack | 36.7% | 43.4% | 44.0% |
| **Total** | **72.2%** | **80.9%** | **81.2%** |

Models trained on the synthetic dataset showed a 10% improvement in ‘pore with crack’ classification for Bayesian optimised models. However, the overall accuracy decreased by roughly 10%, and ‘lack of fusion’ classification accuracy dropped by approximately 15%. This occurs because training on an imbalanced dataset often leads to model bias towards majority classes, resulting in poor generalisation on minority classes (34,35). By balancing the dataset through augmentation, this bias is reduced, leading to more evenly distributed classification performance across all classes.

*3.3 Model Evaluation and Comparison*

It should be noted that other forms of performance metrics, such as sensitivity, precision and recall are also available for model performance evaluation (21). A key consideration is that the synthetic dataset exhibits a different class distribution to the original dataset, with more ‘pore with crack’ defects. For this reason, performance measurement tools must consider both the size of the dataset and the distribution of class labels. By doing so, the models trained on the original dataset and synthetic dataset can by compared, while reflecting these differences. This ensures that the comparisons accurately reflect the models' abilities to generalise and perform well across varied data distributions and sizes. The Matthews Correlation Coefficient (MCC) is an example of such a performance metric, that was initially used to compare chemical structures (41), and was then proposed as a standard performance metric for binary and multi-class machine learning applications (42,43). The MCC has been found to be a reliable metric for evaluating model performance, particularly when comparing different datasets, because it accounts for all four categories of the confusion matrix: true positives, true negatives, false positives, and false negatives (44–46). Unlike simple accuracy, which only measures the proportion of correctly classified instances, MCC provides a more nuanced assessment by considering the full range of classification outcomes. This makes it especially useful in both binary and multi-class classification problems and when dealing with imbalanced datasets (47). The MCC ranges from -1 to +1. An MCC of +1 indicates perfect prediction, 0 suggests no correlation and predictions are no better than random chance, and -1 reflects complete disagreement with entirely incorrect predictions. Table 4 shows the MCC scores for the various models trained.

Table 3: MCC scores for 'crack,' 'lack of fusion,' 'pore,' and 'pore with crack' defects using baseline, grid-search optimised, and Bayesian optimised models on original and synthetic datasets.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Original Dataset** | | | **Synthetic Dataset** | | |
| **Defect Type** | **Baseline Model** | **Grid-search Optimised Model** | **Bayesian Optimised Model** | **Baseline Model** | **Grid-search Optimised Model** | **Bayesian Optimised Model** |
| **Crack** | 0.862 | 0.898 | 0.914 | 0.767 | 0.815 | 0.806 |
| **Lack of Fusion** | 0.828 | 0.853 | 0.847 | 0.618 | 0.668 | 0.679 |
| **Pore** | 0.885 | 0.915 | 0.930 | 0.871 | 0.898 | 0.880 |
| **Pore with Crack** | 0.139 | 0.229 | 0.434 | 0.315 | 0.415 | 0.458 |

Table 4 presents the MCC scores for each class across the different models developed in this work. The scores indicate that the models proficiently classify ‘crack’, ‘pore’, and ‘lack of fusion’ defects, whereas ‘pore with crack’ defects are generally less accurately classified. Notably, models optimised using Bayesian optimisation achieved the highest MCC scores overall. While there is a slight improvement in the MCC score for ‘pore with crack’ defects between the original and synthetic datasets, there is a significant decrease in MCC for the ‘crack’, ‘pore’, and ‘lack of fusion’ defects. This behaviour reflects the trade-off involved in balancing a dataset, as more balanced datasets lead to models that distribute their ability to correctly classify different defects more evenly.

**Conclusions**

This paper has proposed and explored the use of convolutional neural networks (CNN) in the classification of microstructural defects found in additively manufactured nickel-based superalloys. The various CNNs produced have shown good capabilities in classifying the variety of microstructural defects found in additively manufactured nickel-alloys. Classification performance increases as the CNN’s architecture becomes more complex, with increased number of layers and filters in the model. CNNs trained on the original dataset and produced by Bayesian optimisation have shown the best performance (approximately 92% accuracy) which rivals the accuracy of supervised ML models produced by Aziz et al. (21), and deep learning models produced for other tasks in materials science (15,16). Despite its success, CNNs have consistently shown poor performance in correctly classifying ‘pore with crack’ defects, due to the lack of data for this defect class and the overlapping features it shares with the ‘pore’ and ‘crack’ classes. To improve the model’s classification ability with ‘pore with crack’ defects, data augmentation techniques were used to increase the size of the dataset, creating more ‘pore with crack’ defects for models to train on. While these models, which used synthetic data, showed better performance with ‘pore with crack’ classification, it’s classification ability for the ‘crack’, ‘pore’ and ‘lack of fusion’ defects reduced. Supervised ML models produced by Aziz et al. (21) also show difficulty with correctly classifying these ‘pore with crack’ defects, suggesting alternative strategies are needed to accurately predict these defects using automated means.

**Suggested future works…**

Increase the fucking dataset size

Transfer learning, in which you use previously trained models, can be very useful on sparse sets of data (20).

Alternative machine learning powered algorithms could be applied to account for the variability in size of the input images such as fully connected networks or more advanced algorithms such as panoptic segmentation. REFERENCE

**Acknowledgements**

**Declaration of Interests**

None

1. Mumith A, Thomas M, Shah Z, Coathup M, Blunn G. Additive manufacturing: current concepts, future trends. The Bone & Joint Journal. 2018;100-B(4):455–60.

2. Brischetto S, Maggiore P, Ferro CG. Special Issue on “Additive Manufacturing Technologies and Applications”. Technologies [Internet]. 2017;5(3). Available from: https://www.mdpi.com/2227-7080/5/3/58

3. ZHU J, ZHOU H, WANG C, ZHOU L, YUAN S, ZHANG W. A review of topology optimization for additive manufacturing: Status and challenges. Chinese Journal of Aeronautics. 2021;34(1):91–110.

4. Mojtaba Khanzadeh MM Sudipta Chowdhury, Mark A Tschopp, Haley R Doude, Bian L. In-situ monitoring of melt pool images for porosity prediction in directed energy deposition processes. IISE Transactions. 2019;51(5):437–55.

5. Gorelik M. Additive manufacturing in the context of structural integrity. International Journal of Fatigue. 2017;94:168–77.

6. Brennan MC, Keist JS, Palmer TA. Defects in Metal Additive Manufacturing Processes. Journal of Materials Engineering and Performance. 2021 Jul 1;30(7):4808–18.

7. Lorena G, Robinson G, Stefania P, Pasquale C, Fabiano B, Franco M. Automatic Microstructural Classification with Convolutional Neural Network. In: Botto-Tobar M, Barba-Maggi L, González-Huerta J, Villacrés-Cevallos P, S. Gómez O, Uvidia-Fassler MI, editors. Information and Communication Technologies of Ecuador (TICEC). Cham: Springer International Publishing; 2019. p. 170–81.

8. Lantz B. Machine Learning with R [Internet]. 1st Edition. Birmingham, UNITED KINGDOM: Packt Publishing, Limited; 2013. Available from: http://ebookcentral.proquest.com/lib/sheffield/detail.action?docID=1343653

9. Alpaydin E. Introduction to machine learning. Third edition. Cambridge, Massachusetts: MIT Press; 2014. (Adaptive computation and machine learning).

10. Snell R, Tammas-Williams S, Chechik L, Lyle A, Hernández-Nava E, Boig C, et al. Methods for Rapid Pore Classification in Metal Additive Manufacturing. JOM. 2020 Jan 1;72(1):101–9.

11. Scime L, Beuth J. Anomaly detection and classification in a laser powder bed additive manufacturing process using a trained computer vision algorithm. Additive Manufacturing. 2018;19:114–26.

12. Nalajam PK, V R. Microstructural porosity segmentation using machine learning techniques in wire-based direct energy deposition of AA6061. Micron. 2021;151:103161.

13. Ketkar N. Deep learning with Python : a hands-on introduction. 1st ed. 2017. Place of publication not identified]: Apress; 2017.

14. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015 May 1;521(7553):436–44.

15. Chowdhury A, Kautz E, Yener B, Lewis D. Image driven machine learning methods for microstructure recognition. Computational Materials Science. 2016;123:176–87.

16. Durmaz AR, Müller M, Lei B, Thomas A, Britz D, Holm EA, et al. A deep learning approach for complex microstructure inference. Nature Communications. 2021 Nov 1;12(1):6272.

17. Tan RK, Zhang NL, Ye W. A deep learning–based method for the design of microstructural materials. Structural and Multidisciplinary Optimization. 2020 Apr 1;61(4):1417–38.

18. Beniwal A, Dadhich R, Alankar A. Deep learning based predictive modeling for structure-property linkages. Materialia. 2019;8:100435.

19. Holm EA, Cohn R, Gao N, Kitahara AR, Matson TP, Lei B, et al. Overview: Computer Vision and Machine Learning for Microstructural Characterization and Analysis. Metallurgical and Materials Transactions A. 2020 Dec 1;51(12):5985–99.

20. Wang C, Tan XP, Tor SB, Lim CS. Machine learning in additive manufacturing: State-of-the-art and perspectives. Additive Manufacturing. 2020;36:101538.

21. Aziz U, Bradshaw A, Lim J, Thomas M. Classification of defects in additively manufactured nickel alloys using supervised machine learning. Materials Science and Technology. 2023;39(16):2464–8.

22. Liu C. Selective laser melting of nickel superalloys for aerospace applications: defect analysis and material property optimisation [Internet] [PhD Thesis]. [Sheffield, United Kingdom]: The University of Sheffield; 2021. Available from: https://etheses.whiterose.ac.uk/30067/

23. Wang Q, Yuan Y. High quality image resizing. Neurocomputing. 2014;131:348–56.

24. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research. 2014;15(1):1929–58.

25. Sietsma J, Dow RJF. Creating artificial neural networks that generalize. Neural Networks. 1991;4(1):67–79.

26. Yang L, Shami A. On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing. 2020;415:295–316.

27. Feurer M, Hutter F. Hyperparameter Optimization. In: Hutter F, Kotthoff L, Vanschoren J, editors. Automated Machine Learning: Methods, Systems, Challenges [Internet]. Cham: Springer International Publishing; 2019. p. 3–33. Available from: https://doi.org/10.1007/978-3-030-05318-5\_1

28. Diaz GI, Fokoue-Nkoutche A, Nannicini G, Samulowitz H. An effective algorithm for hyperparameter optimization of neural networks. IBM Journal of Research and Development. 2017;61(4/5):9:1-9:11.

29. Raiaan MAK, Sakib S, Fahad NM, Mamun AA, Rahman MA, Shatabda S, et al. A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks. Decision Analytics Journal. 2024;11:100470.

30. Antal-Vaida C. Basic Hyperparameters Tuning Methods for Classification Algorithms. Informatica Economica. 2021;25(2).

31. Ungredda J, Branke J. Bayesian Optimisation for Constrained Problems. ACM Trans Model Comput Simul [Internet]. 2024 Apr;34(2). Available from: https://doi.org/10.1145/3641544

32. Kandasamy K, Neiswanger W, Schneider J, Póczos B, Xing EP. Neural architecture search with Bayesian optimisation and optimal transport. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates Inc.; 2018. p. 2020–9. (NIPS’18).

33. Wen H, Huang C, Guo SM. The Application of Convolutional Neural Networks (CNNs) to Recognize Defects in 3D-Printed Parts. Materials. 2021 May;14:2575.

34. Buda M, Maki A, Mazurowski MA. A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks. 2018;106:249–59.

35. He H, Garcia EA. Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering. 2009;21(9):1263–84.

36. Skorski M, Temperoni A, Theobald M. Revisiting Weight Initialization of Deep Neural Networks. In: Balasubramanian VN, Tsang I, editors. Proceedings of The 13th Asian Conference on Machine Learning [Internet]. PMLR; 2021. p. 1192–207. (Proceedings of Machine Learning Research; vol. 157). Available from: https://proceedings.mlr.press/v157/skorski21a.html

37. Nguyen TT, Trahay F, Domke J, Drozd A, Vatai E, Liao J, et al. Why Globally Re-shuffle? Revisiting Data Shuffling in Large Scale Deep Learning. In: 2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS). 2022. p. 1085–96.

38. Taylor R, Ojha V, Martino I, Nicosia G. Sensitivity Analysis for Deep Learning: Ranking Hyper-parameter Influence. In: 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI). 2021. p. 512–6.

39. Taylor L, Nitschke G. Improving Deep Learning with Generic Data Augmentation. In: 2018 IEEE Symposium Series on Computational Intelligence (SSCI). 2018. p. 1542–7.

40. Shorten C, Khoshgoftaar TM. A survey on Image Data Augmentation for Deep Learning. Journal of Big Data. 2019 Jul 6;6(1):60.

41. Matthews BW. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochimica et Biophysica Acta (BBA) - Protein Structure. 1975;405(2):442–51.

42. Baldi P, Brunak S, Chauvin Y, Andersen CAF, Nielsen H. Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics. 2000 May;16(5):412–24.

43. Gorodkin J. Comparing two K-category assignments by a K-category correlation coefficient. Computational Biology and Chemistry. 2004;28(5):367–74.

44. Chicco D, Jurman G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics. 2020 Jan 2;21(1):6.

45. Chicco D, Jurman G. The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification. BioData Mining. 2023 Feb 17;16(1):4.

46. Chicco D, Warrens MJ, Jurman G. The Matthews Correlation Coefficient (MCC) is More Informative Than Cohen’s Kappa and Brier Score in Binary Classification Assessment. IEEE Access. 2021;9:78368–81.

47. Jurman G, Riccadonna S, Furlanello C. A comparison of MCC and CEN error measures in multi-class prediction. 2012;