



Comparison of machine learning methods for automatic classification of porosities in powder-based additive manufactured metal parts

Nicholas Satterlee¹ · Elisa Torresani¹ · Eugene Olevsky¹ · John S. Kang¹ 

Received: 11 November 2021 / Accepted: 25 March 2022 / Published online: 18 April 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract

An outstanding problem of additive manufacturing is the variability in part quality caused by process-induced defects such as porosity. Image-based porosity detection represents a solution that can be easily implemented into existing systems at a low cost. However, current industry porosity detection software utilizes threshold-based methods which require user calibration and ideal lighting conditions, and thus cannot be fully automated. This paper investigates the application of machine learning methods and compares their ability to classify porosities from cross-section images of 3D printed metal parts. Fifty-one features are manually defined and automatically extracted from the images and the most relevant features among them are selected using feature reduction methods. Six machine learning algorithms that are commonly used for classification problems are trained with those features and used for the porosity classification. The decision tree, one of the six machine learning algorithms, yields 85% accuracy with a processing time of 0.5 s to classify porosities from 691 images. However, manual features may not adequately characterize porosity because they are dependent on user's experience and judgment. Alternatively, deep convolutional neural network (DCNN) that does not require user-defined features is used for the classification problem. The comparison results showed that a DCNN yields the highest accuracy of 95% with a processing time of 1.8 s to classify porosities from the same 691 images.

Keywords Porosity · Powder-based additive manufacturing · Machine learning · Convolutional neural network

1 Introduction

With the advance in the technologies, additive manufacturing (AM) is hitting the market to manufacture complex or customized parts in a short period of time. Compared to conventional manufacturing processes, AM processes have various perceived advantages. It has a potential to approach zero waste manufacturing by maximizing material utilization and to produce parts near final shape which eliminate additional processing. AM has an ability to produce part with greater complexity than traditional processes, without need of additional tooling. Geometries which are difficult or impossible to manufacturing using conventional techniques can be manufactured at the same cost as uncomplicated shapes such as simple cylinder or cube [1]. It increases production flexibility because it requires smaller operational

footprint toward manufacturing large variety of parts with on-demand manufacturing, moving away from projection-based manufacturing with excellent scalability [2]. AM processes are generally categorized depending on the raw material as shown in Table 1 [3]. For example, stereolithography is used for liquid resins, selective laser sintering for powder materials, and fused deposition modeling for solid materials [4].

Powder-based AM (PBAM) is fast, highly precise, available for various materials, and suitable for printing complex shapes [5]. Despite these advantages, the variability in part quality primarily caused by process-induced defects prevents the PBAM technology from scaling and broader acceptance for high-value applications [6–8]. One of the most critical defects is porosity, which can be defined as the presence of small spaces or voids within the part as shown in Fig. 1. The porosity can be induced by the spaces that powder did not occupy while melting or sintering the rest of the material together, which is detrimental to the mechanical properties of the material. Therefore, porosity detection is a crucial step in assessing the quality of the parts printed by PBAM.

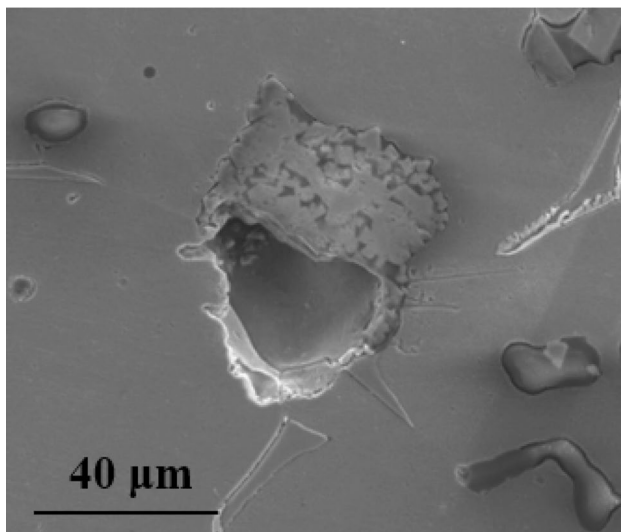
✉ John S. Kang
jkang4@sdsu.edu

¹ Department of Mechanical Engineering, San Diego State University, San Diego, CA 92182, USA

Table 1 Major AM processes based on Hopkinson and Dickens classification [3]

Material	AM process
Liquid based	Stereolithography
	Jetting systems
	Direct light processing
Powder based	Selective laser sintering
	Selective laser melting
	Binder jetting
	Three-dimensional printing
	Fused metal deposition systems
	Electron beam melting
	Selective masking sintering
	Selective inhibition sintering
	Electro photographic layered manufacturing
	High speed sintering
Solid based	Fused deposition modeling
	Sheet stacking technologies

For example, in the sintering-assistive additive manufacturing (SAAM) [9] using the binder jetting [10], one of the PBAM technologies, the primary challenge is to achieve component densities comparable with those produced by conventional processes. The green component printed by the binder jetting is typically characterized by a low powder packing density [11], which results considerable shrinkage during the follow-up sintering process. Because the component density is directly related to the bulk porosity, porosity detection is critical to assess the quality of the additively manufactured parts. In addition, it is important to analyze

**Fig. 1** Typical porosity in SEM images

the sintering behavior of SAAM compared to that of the conventional process.

Porosity detection is generally categorized into in situ and post-processing methods. The post-processing approach involves measurement and analysis of pores and defects using a finished part. Alternatively, the in situ methods provide the advantage of real-time monitoring of the sample during the printing process. The post-processing methods are relatively simpler to perform and are not limited by real-time requirements. In situ generally requires more complex equipment and analysis methods but assesses the reliability of a printed part in real time. A comparison of these technologies is provided in Table 2.

Based on the Table 2 comparison, visual image-based methods are advantageous in both post-processing and in situ methods. Bulk density measurement and ultrasonic wave speed measurement can only measure bulk porosity and cannot provide size, morphology, or pore distribution. Micro X-ray computed tomography has the advantage of 3D imaging, but requires an expensive equipment and X-ray absorption variation in bulk material, and has part size limitations [12]. Coaxial temperature imaging is limited to specific processes such as selective laser sintering and requires extensive modifications of current systems for implementation.

Alternatively, the primary disadvantage of traditional image-based methods is manual classification of the regions in the image. This ultimately makes image-based methods costly and time-consuming. Even industry software for porosity detection was found in [15] to be highly user dependent and without adequate training, resulting in high classification error. The issue with this software is that porosity detection is software-assisted, meaning that the user detects porosity by changing parameters such as threshold limits. These disadvantages could be eliminated by applying machine learning (ML) techniques to automatically process the images reliably and in real time without user assistance.

A literature review of real-time image-based porosity classification was performed. Layer-wise optical imaging to classify porosities in Ti-6AL-4 V during laser powder bed fusion (LPBF) was performed with a support vector machine (SVM), k-nearest neighbor (KNN), and artificial neural network (ANN), achieving accuracies of 89.36%, 78.60%, and 84.40%, respectively [16]. Another study classified porosities in LPBF parts composed of Inconel 625 using a Bayesian classifier resulting in an accuracy of 89.5% [17]. Porosity classification in direct energy deposition (DED) with sponge titanium powder was investigated with convolutional neural networks yielding an accuracy of 91.2% [18]. Porosity classification of parts composed of Al-5083 manufactured with direct metal deposition (LMD) was performed with a random forest classifier yielding an accuracy of 94.41% [15]. SVM was also utilized for porosity

Table 2 Comparison of porosity detection methods

Category	Detection type	Pros	Cons
Post-processing	Bulk density measurement (e.g., Archimedes' measure) [12]	• Simple process • No specialty hardware	• Only available for measuring bulk density (no pore size, location, or morphology data) • Theoretical density highly dependent on exact chemical composition • Small pores that affect material properties overlooked
	Cross-section microscopic image analysis [12]	• Characterization of internal pores at high resolution	• Preparation may alter sample • Prone to human error • Need consistent parameters (sample location, magnification, focus, etc.) • Time-consuming
	Micro X-ray computed tomography [13]	• Non-destructive • Characterization of internal pores at high resolution • 3D spatial distribution of pores	• Requires an expensive equipment • Requires X-ray absorption of the bulk material and defect to be significantly different • Time required for sufficient X-ray transmission may be significant for larger parts • Large parts may need a small sample cut to limit X-ray absorption • Requires two-sided access
In situ	Ultrasonic wave speed measurement [13]	• Non-destructive • Quick response • One-sided access	• Only available for measuring bulk density (No pore size, location, or morphology data) • High porosity increases uncertainty • Uneven surfaces may blur signal and reduce quality • Limited to 16% porosity
	Coaxial temperature imaging [14]	• High-resolution in situ measurement	• Data collection is limited to melt pool • Defects and quality inferred by melt pool data • Limited applications
	Off-axial imaging [14]	• Simple hardware Easy interpretation data • Initial powder layer can be captured before processing	• Large amount of data collection • Need suitable camera depending on capture scenario • Requires complicated processing

classification in stainless steel parts constructed with LPBF yielding accuracy of 85% [19]. A study of porosity classification in Inconel 718 parts constructed with LPBF was performed with deep convolutional neural networks (CNN), yielding accuracies as high as 99.4% [20]. Another study utilized images of Ti-6Al-4 V to detect lack of fusion porosities in LBPF parts and reported an accuracy of 91.5% [21].

While the results of previous studies are promising, these methods rely on specific materials and are not capable of classifying porosity in general. This is problematic since an algorithm trained on a single material has no guarantee of generalization. It may require significant retraining and hyperparameter tuning to include additional materials or may perform poorly even with retraining efforts. In addition, nearly all previous studies have attempted to solve porosity classification by employing a single ML algorithm. However, there are many ML algorithms that are commonly

used for classification, and no studies have explored which algorithms should be employed to best classify porosity for a diverse material set. Therefore, a breath-first approach to porosity classification will compare a variety of common ML algorithms to determine the strengths and weaknesses of each method when applied to a diverse material image.

The overall goal of this study is to use and compare supervised ML algorithms for a real-time image-based porosity classification. A database of cross-sectional images of 3D printed metal parts is developed to train the ML algorithms. Fifty-one features are manually determined and extracted from these images and used to train six ML algorithms that are commonly used for classification problems. However, the manually defined features may not be the optimal for classifying porosity because they are user dependent. Therefore, deep convolutional neural network (DCNN), which does not require user-defined features, is also used for porosity

classification. The results were then compared to determine the pros and cons of each method.

2 Approach

2.1 Dataset generation

To train and test the ML algorithms for porosity classification, we generated a dataset that contains pores and non-pores in the form of cropped images. The dataset was obtained from cross-section images of cubical metal (316L stainless steel, boron carbide, copper, and nickel) specimens that are additively manufactured by SAAM with binder jetting. It is assumed that the approaches explored in this paper can be applied to other AM methods once a dataset is generated for those methods. Due to the time-consuming process of collecting images and creating an image database, the applicability of the presented detection algorithms on other AM methods will be reserved for a future study.

The specimens were first printed using a binder jetting printer (Jet Zprinter[®] 350, Z-Corp, USA) with 5 wt% of binder and 100–250 μm of layer thicknesses. The printed specimens were cured at 80 $^{\circ}\text{C}$ in a vacuum oven (25L 200C Vacuum Oven, MTI Corp., USA) for a half hour to better consolidation and dry. Then the debinding and sintering profile, shown in Fig. 2, was conducted in a tube furnace under vacuum (GSL-1700X-KS-UL-60, MTI Corp., USA). The debinding was performed in two steps: the first is at 210 $^{\circ}\text{C}$ with 20-min hold and the second is at 480 $^{\circ}\text{C}$ for 1-h dwell. These steps were selected to gradually remove the binder from the specimen and were optimized through a thermogravimetric analysis using a TGA/DSC equipment (Q600 SDT, TA instruments, USA). Subsequently, sintering was performed by heating at a rate of 5 $^{\circ}\text{C}/\text{min}$ until 1250 $^{\circ}\text{C}$ was reached and held for 10 h. The average relative density

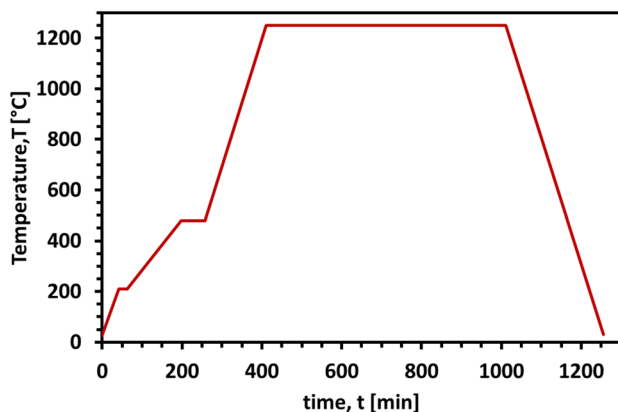


Fig. 2 Debinding and sintering profile

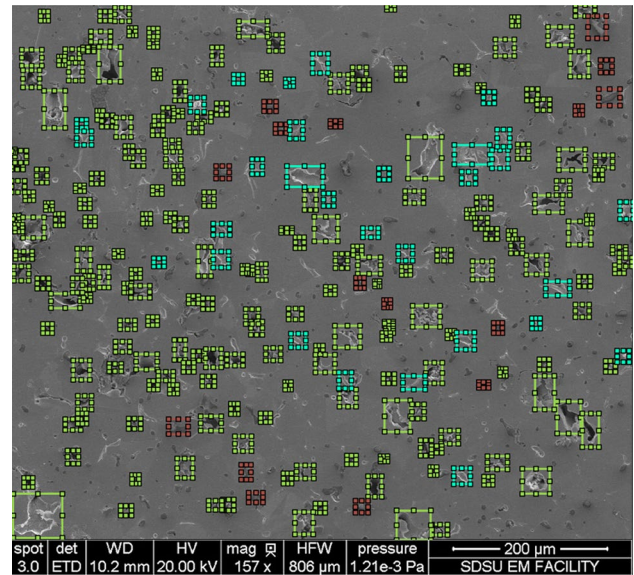


Fig. 3 Labeled image with examples of porous (green), light non-porous (turquoise), and dark non-porous (red) regions

of the as-printed (40%) and sintered specimens (87%) was measured using the Archimedes' method.

To obtain the cross-section images, the specimens were sectioned and the surface polished using sandpaper (180–4000 grit) followed by the finishing using 3 μm and 1 μm diamond suspensions (Struers, Denmark). The microstructure images of the cross-sections were obtained using a SEM (FEI Quanta 450 FEG, Thermo Fisher Scientific, USA) with 197 $\times$ magnification.

Supervised ML requires a prelabeled dataset to train an algorithm. We used an intensity threshold plus gradient filter to select the porous and non-porous regions in cross-section images. However, the algorithm is insufficient to label all true porous and non-porous regions. For example, there are dark non-porous region, which may be inadvertently selected due to the low intensity. Therefore, the selected regions were manually reviewed and labeled as porous or non-porous as shown in Fig. 3. The total number of resulting classified dataset obtained is summarized in Table 3.

There are some limitations with the cross-section image dataset, however. Cross-section images do not fully express porosity size, shape, and distribution due to the loss of a dimension. In practice, many cross-section images per part would be required. Ideally, multiple images at each layer

Table 3 Porosity image dataset

Classification type	Pores	Non-pores (dark)	Non-pores (light)
Number of classifications	2897	1086	631

of the additive manufacturing process would be utilized to represent the part quality.

2.2 ML methods

ML algorithms are highly dependent on the feature space [22]. Generally, features are extracted from images using two distinct methods: (1) construct specific algorithms to extract features using engineering judgment; (2) generalize feature extraction using an automated method. Both methods have pros and cons for implementation. Generally, when manually constructing features, the model is produced via a white box method, that is, the feature space can be easily viewed and interpreted by an individual. This is accomplished by extracting predefined properties from the image, such as average intensity, bounding box size, and canny filters. However, the manual feature set has the disadvantage of specifying features that are not relevant and feature reduction must be performed. It also has the potential to miss features that are critical for classification. Conversely, automated feature definitions are generally black box models, where an individual is unable to easily interpret these features without an accompanying ML algorithm. Subsequent sections evaluate both methods.

2.2.1 Supervised ML methods with user-defined features

Six ML algorithms were selected to be trained with user-defined features. Those ML algorithms are support vector machine (SVM), binary linear classifier (linear), boost trees (BT), decision trees (DT), single-layer neural network, and K-nearest neighbors (KNN). Unless otherwise noted, all ML methods were run with MATLAB R2021a and hyperparameter optimization was employed with built-in MATLAB functions.

Each of these ML algorithms requires input data to perform classification. This data consists of user-generated features extracted from a given porous or non-porous cropped image. Most features were selected based on the OpenCV feature detection documentation [23]. However, some features were selected based on observations of porosities. The light intensity, for instance, was defined as a feature based on the observation that light scatters at the edge of the pore, creating a light intensity area around the porosity. To define this feature, a histogram of the intensity in the image was constructed, then the top 15% of the intensity was used to define the light area. The feature was then used to define other features such as the light area fraction. The total set of features is shown in Table 4.

Table 4 Initial set of features extracted from images using OpenCV libraries [23]

#	Feature	Description
1	Contour area	Pore mask contour area
2	Mean intensity	Mean intensity of mask
3	Light intensity	Mean intensity of light mask
4	Light intensity fraction	Fraction of intensity in light mask
5	Light area fraction	Fraction of area in light mask
6	Convex light area	Determines if light area mask is convex
7	Extent	Ratio of contour area to bounding box area
8	Solidity	Ratio of contour area to convex hull area
9	Equivalent diameter	Diameter of a circle containing the equivalent area of the mask
10	Angle	The angle of the rotated bounding box computed from the aspect ratio
11	Border pixels	Number of pixels around the border of the image
12	Perimeter	Length of the perimeter in pixels
13	Compactness	The ratio of the area of the mask to the area of a circle with the same perimeter
14	Aspect ratio of box	The aspect ratio of the bounding box around the pore
15	Aspect ratio of minimum box	The aspect ratio of the minimum rotated bounding box around the pore
16	Angle of minimum box	The angle of the minimum rotated bounding box computed from the aspect ratio
17	Rectangularity	The ratio of the minimum rotated bounding box area to the mask area
18	Convex	Determines that the mask area is convex
19	Minimum radius	Radius of the minimum enclosing circle
20	Triangle area	Area of minimum enclosing triangle
21–30	Spatial moments	Spatial moments computed with the pixel intensities on a segmentation grid
31–37	Central moments	Spatial moments computer with respect to the center of the segmentation grid
38–44	Central normalized moments	The central moments as described above, normalized and are invariant
45–51	HuMoments	Invariant shape moments

Feature reduction was performed to remove features that do not contribute to porosity classification. Removal of non-contributing features is important for improving prediction accuracy, preventing overfitting, and reducing computation time. Features were reduced using a stochastic coordinate descent algorithm combined with fast ML algorithms such as KNN classification or linear classification, shown in Fig. 4, respectively. For descriptions of ML algorithms utilized, please refer to the following section. The stochastic coordinate descent algorithm trains the ML algorithms on 70% of the data, and the remaining 30% of the data is used for testing. The average accuracy over 20 iterations is selected for each datapoint in the Fig. 4 graph. In every iteration, the training and testing sets were reshuffled. Each result is hashed to reduce runtime if the set is visited again.

Another method of feature reduction was compared using the analysis of variance (ANOVA) general linear model. The feature data extracted from the Table 3 dataset was imported to MATLAB and compared using the ANOVA function. The results indicate that 24 features meet the 5% p value requirements for statistically significant [24]. The results of each feature reduction method are displayed in Table 5.

To compare the feature reduction methods, each of the six ML algorithms was tested with the Table 5 columns along with the entire feature set. The results are summarized in Table 6. The highest accuracy sets were from KNN, DT, and SVM with results within statistical variance. However, notably, the DT set contained the least number of features, but still performed as well as the other feature reduction

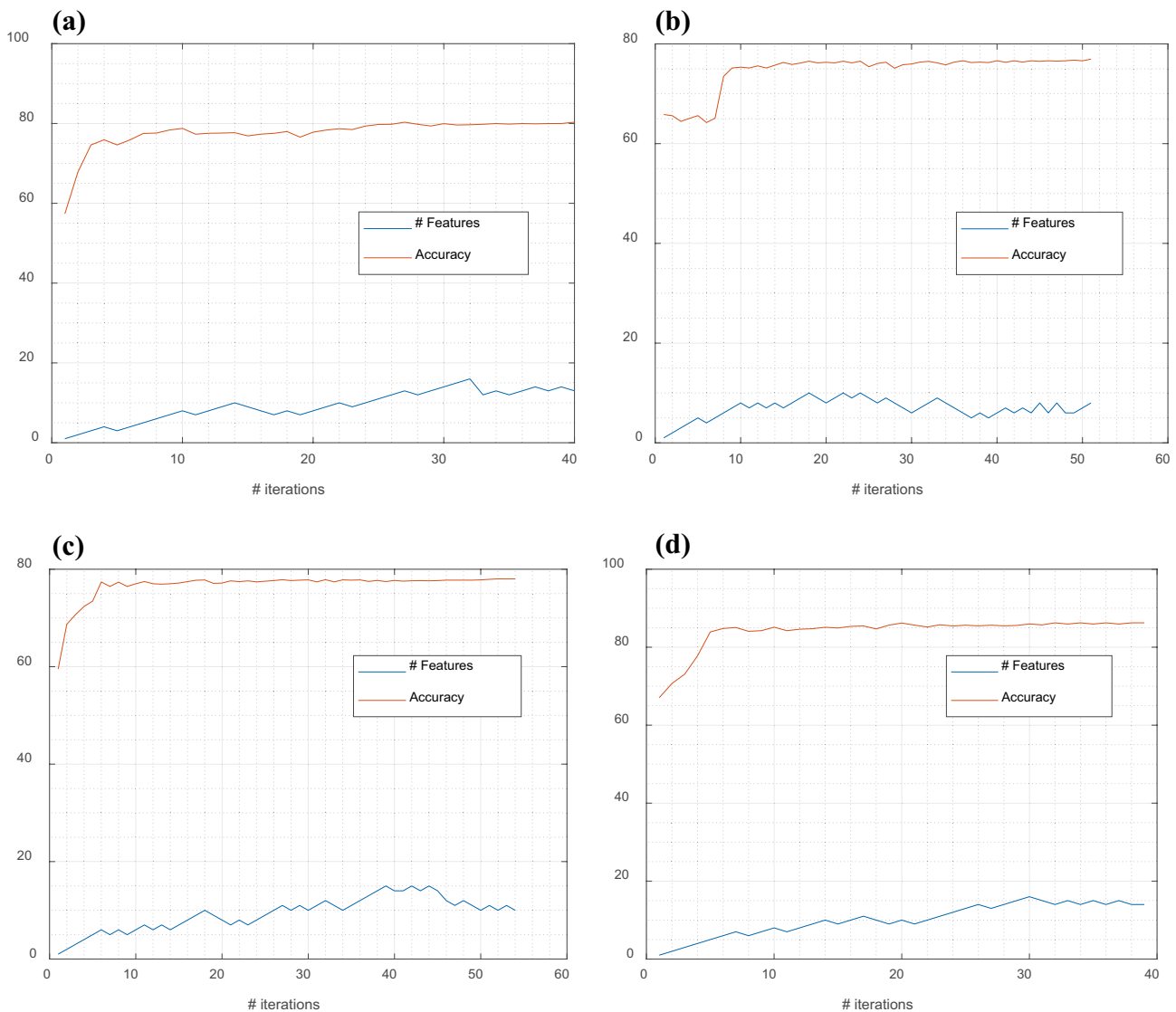


Fig. 4 Feature reduction for KNN (a), linear (b), decision trees (c), and SVM (d)

Table 5 Comparison of feature reduction methods and overlapping features selected

Feature #	KNN	DT	Linear	SVM	ANOVA
1					X
2		X		X	X
3	X				X
4	X	X	X	X	X
5	X	X		X	X
6					
7		X	X	X	X
8			X		X
9					X
10					
11	X				X
12					X
13	X	X		X	X
14					X
15					X
16					X
17	X		X		X
18			X		
19					X
20	X			X	
21				X	X
22					X
23				X	X
24	X			X	
25					X
26					
27					X
28					X
29	X				
30					
31					
32					
33					X
34	X	X			
35					
36					
37	X	X		X	
38					
39					
40					
41	X				
42					
43				X	
44	X	X		X	X
45	X	X		X	
46				X	X
47	X		X	X	
48	X				
49	X	X			

Table 5 (continued)

Feature #	KNN	DT	Linear	SVM	ANOVA
50					
51	X				

methods. It appears that most features are neutral and added or removing them has little impact on the trained model accuracy.

2.2.2 Deep convolutional neural network

DCNNs have transformed image recognition ability of computers over the past decade. These networks have been employed in diverse detection and classification applications such as face detection, marine organism classification, sewer pipe defect detection, solar cell defect classification, wheel hub surface defect detection, and polymeric polarizer defect detection [25–30]. In all cases, accuracies exceeded 90%. This makes DCNNs promising candidates for porosity classification.

Unlike the previous section, where 51 user-defined features were used, this section considers DCNNs that automatically defines and extracts features from the images of porosities and non-porosities. However, these features are embedded in a hidden layer, which is a black box to the users. Raw images are fed into the network with their corresponding labels, and the network relates patterns in the images to the image label via successive layers of convolution. Various networks have been designed and pre-trained for image classification. The major differences between these networks are the number of layers in the network and the design of each layer. A list of some popular pre-trained networks available for image classification is displayed in Table 7.

Each network in Table 7 was pre-trained to detect 1000 different objects from the ImageNet dataset, which contains approximately 1.2 million training images, 50 thousand validation images, and 100 thousand test images [43]. The dataset includes a wide variety of everyday objects as well as similar objects such as various bird species and dog breeds [43]. While these networks are designed to deal with diverse classifications, the present problem only requires binary classification of an image as porous or non-porous. Therefore, the networks must be tested to determine which is best at binary classification of amorphous structures.

However, prior to testing, hyperparameters for training the network must be considered. The relevant parameters are shown in Table 8. The minibatch size was selected to be the largest base two integers before obtaining an out-of-memory error, and the initial learning rate was selected to be 0.001 and reduced by half

Table 6 Accuracy of each training set and method with number of features

ML Algorithm	Feature Reduction Method					
	KNN	DT	Linear	SVM	ANOVA	None
Single Layer ANN	85.8%	84.4%	78.1%	85.6%	84.6%	84.1%
KNN	84.5%	83.6%	77.2%	84.3%	83.2%	78.1%
BT	85.3%	85.6%	77.9%	85.2%	85.2%	85.8%
DT	79.9%	79.8%	70.5%	79.9%	79.4%	79.3%
Linear	68.5%	64.8%	74.2%	68.4%	68.5%	68.4%
SVM	85.0%	85.1%	77.5%	85.2%	81.2%	74.3%
# Features	18	10	6	15	24	51

Highest accuracy per column is shaded and highest accuracy per row is bold

until divergence was no longer observed. The maximum number of epochs was selected by increasing the number of epochs until training and testing accuracy converged. An assessment was performed with the ResNet18 neural network, and the results are shown in Fig. 5. To determine the threshold of convergence, the ResNet18 neural network was trained 25 times on randomly shuffled data and an average of 92.5% with a standard deviation of 1% was observed. It was found that convergence occurred after 9 epochs. Therefore, 10 epochs were used for the MaxEpoch parameter. The validation frequency was set to update every epoch. This ensured that once the weights are updated, the model is saved and assessed. The model with the highest accuracy is then returned. This prevents returning overfit models.

Freezing initial layers without performance impact is a potential benefit using transfer learning on a pre-trained

network. Since the ResNet18 neural network has already been trained to classify 1000 objects, it has also already been trained to detect low-level image features within those initial layers [45]. By skipping over these layers during initial training, training time could be reduced significantly. This would especially be advantageous if further defect classifications were added to the network or additional training images were included in the future.

The number of frozen layers was selected to be zero for all initial assessments. However, since it is advantageous to reduce training time, the ResNet18 neural network was trained multiple times with successive layers frozen. The results, displayed in Fig. 6, indicate that up to 20% of the layers may be frozen without any impact on accuracy and a training time reduction of approximately 12.5%. Therefore, significant time could be saved by freezing approximately 20% of the initial layers.

Table 7 List of popular pre-trained neural networks

Network	Depth	Size (MB)	Parameters (millions)	Image input size	References
SqueezeNet	18	5.2	1.24	227-by-227	[31]
GoogleNet	22	27	7	224-by-224	[32]
Inception v3	48	89	23.9	299-by-299	[33]
MobileNetV2	53	13	3.5	224-by-224	[34]
ResNet18	18	44	11.7	224-by-224	[35]
ResNet50	50	96	25.6	224-by-224	[35]
ResNet101	101	167	44.6	224-by-224	[35]
Xception	71	85	22.9	299-by-299	[36]
Inception-ResNetv2	164	209	55.9	299-by-299	[37]
ShuffleNet	50	5.4	1.4	224-by-224	[38]
Darknet19	19	78	20.8	256-by-256	[39]
EfficientNetB0	82	20	5.3	224-by-224	[40]
VGG16	16	515	138	224-by-224	[41]
NASNetMobile	*	20	5.3	224-by-224	[42]

Table 8 Hyperparameter list for DCNNs [44]

Minibatch size	The population size of the training set used to evaluate the gradient descent function. Increasing this generally decreases training time but requires more computational memory
Initial learning rate	Initial learning rate for stochastic gradient descent model solver. Selecting the correct learning rate will minimize the training time. However, a significantly large learning rate may cause the training to diverge
MaxEpochs	Maximum number of epochs before completing training. Selecting too few epochs will result in an underfit model and too many epochs will result in overfit model
Frozen layers (optional for pre-trained networks)	The number of layers to freeze prior to training. The nodes in the network are not updated while training resulting in a faster training time. Freezing too many layers will result in decreased performance
Validation frequency	Number of iterations before validation set benchmarks the trained model. The validation is used to select the best-performing model at the end of training. Increasing the validation frequency may result in better model selection but will increase training time

3 Results

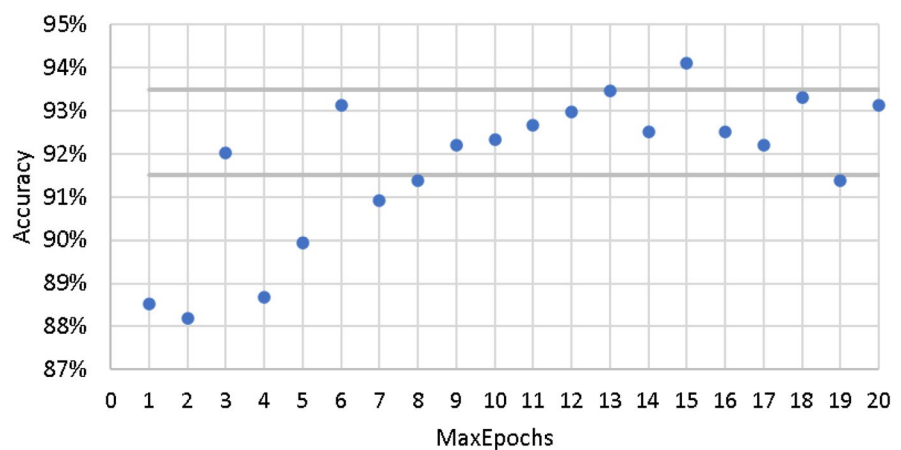
3.1 Computing system

All analysis was performed on Microsoft Windows 10.0.19043 running on an AMD Ryzen 9 3900X 12-Core Processor, NVIDIA GeForce GTX 1080Ti, 32 GB DDR4-3200, and an HP EX900 500 GB SSD.

3.2 Experimental datasets and system

The image dataset containing 4614 cropped images was randomly split into training and testing sets. Seventy percent of the images were selected for the training set, and the remaining 30% images were reserved for the testing set. In the case of DCNNs, the testing set was split into a validation set and a testing set. The validation set is used to select the best model during training and prevent overfitting. The training set is then used to demonstrate consistent results (i.e., the model does not just perform well on the validation set).

Fig. 5 Maximum number of epoch selection with ResNet18



Supervised ML methods with user-defined features were run 20 times on reshuffled data to generate the standard deviation statistics. Due to the long training times required, DCNNs were only run 5 times to generate statistics.

3.3 Metrics

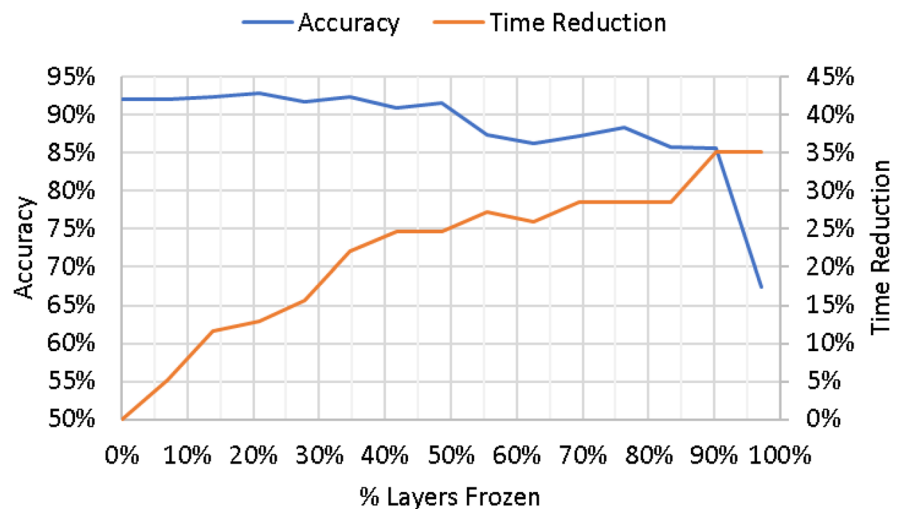
Before comparing ML algorithms, it is important to define some common performance metrics that are used during analysis. The confusion matrix displayed in Fig. 7 defines the true positive (TP), false positive (FP), false negative (FN), and true negative (TN), where TP and TN represent the correct number of positive and negative guesses the classifier made.

The results of the confusion matrix were then used to compute the performance metrics shown below.

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN} = 1 - FNR$$

$$TNR = \frac{TN}{N} = \frac{TN}{TN + FP} = 1 - FPR$$

Fig. 6 Frozen layer impact on accuracy and training time



$$\text{FNR} = \frac{\text{FN}}{\text{P}} = \frac{\text{FN}}{\text{TP} + \text{FN}} = 1 - \text{TPR}$$

$$\text{FPR} = \frac{\text{FP}}{\text{N}} = \frac{\text{FP}}{\text{TN} + \text{FP}} = 1 - \text{TNR}$$

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

where TPR is the true positive rate, TNR is the true negative rate, FNR is the false negative rate, FPR is the false positive rate, and ACC is the accuracy. TPR computes the fraction of positive predictions correctly classified. TNR computes the fraction of negative predictions correctly classified. FNR and FPR similarly measure the incorrect classifications. The ACC metric calculates the fraction of correct classifications.

3.4 Accuracy comparison

A common and straightforward method for comparing ML methods is by graphing the accuracy of each method as shown in Fig. 8. The results presented in Fig. 8 appear

concise and conclusive. However, if the testing datasets are unbalanced, the results can be misleading.

To determine if the testing datasets are unbalanced, the actual positive and negative classifications need only be reviewed from the dataset in Table 3. It was shown that nearly twice the number of pore classifications were included compared to the non-pore classifications. Therefore, it is possible that some ML methods appear more accurate due to the skewed dataset. For instance, an algorithm that classifies everything as a pore would report an accuracy of 68%. To determine how well algorithms are actually performing, the TPR and FNR need to be compared.

The TPR and FNR can be graphed for each algorithm to generate a receiver operating characteristic (ROC) graph [46]. The ROC graph is used as another ML comparison method. The line plotted through the middle of the graph represents the performance of a random classifier. The further to the top left the graph, the better the performance of the algorithm. The ROC graph corresponding to the results from Fig. 8 is presented in Fig. 9 and Table 9 for the six ML algorithms and the best DCNN (ResNet50). Due to the relative similarity of the DCNN results, they were compared separately in Table 10 and Fig. 10.

The results from Fig. 9 make some subtle clarifications to the Fig. 8 accuracy results. What is found is that the six ML algorithms with user-defined features perform similarly

Table 9 Comparison of ML methods with standard deviations

Algorithm	ACC	FNR	TPR	σ_{FNR}	σ_{TPR}
Single-layer ANN	85.8%	0.24	0.91	0.02	0.01
KNN	84.5%	0.29	0.91	0.03	0.01
BT	85.8%	0.33	0.95	0.03	0.01
DT	79.9%	0.32	0.85	0.02	0.01
Linear	74.2%	0.75	0.97	0.03	0.01
SVM	85.2%	0.32	0.94	0.03	0.01
DCNN	95.0%	0.07	0.97	0.02	0.02

		Actual Class	
		P	N
Predicted Class	P	TP	FP
	N	FN	TN

Fig. 7 Confusion matrix

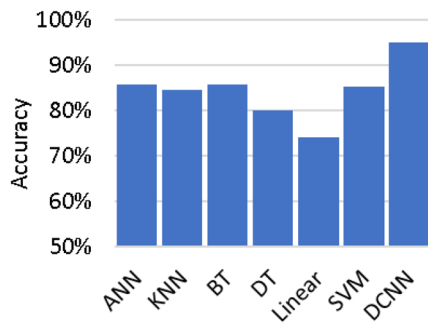


Fig. 8 Chart comparing the best accuracy of each ML method over testing sets

to the DCNN in detecting true porosity (TPR) but perform substantially worse detecting non-pores (FNR). In this way, the DCNN with the automated feature extraction has succeeded in communicating a deficiency in the feature space, namely, that manual feature definition was more concerned with what constitutes a porosity rather than what does not. By explicitly defining features of non-porosities and considering edge cases, it is likely the accuracy of the six ML algorithms with user-defined features can be increased. However, further consideration is required to determine what those features might be.

For the current results, however, the automated feature extraction with DCNNs outperforms the other methods. It was found that the best-performing DCNN algorithms were ResNet50, ResNet18, InceptionResNetv2, and InceptionV3 as shown in Table 10 and Fig. 10. Interestingly, these network designs all share residual layers. Residual layers implement a “shortcut” around multiple convolutional layers [35]. These shortcuts help mitigate the vanishing gradient problem, where nodal weights tend to vanish or blow up during back propagation [35]. Further review of these networks is

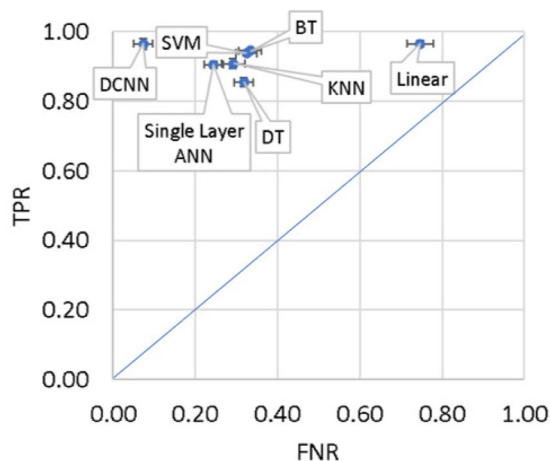


Fig. 9 Comparison of ML methods with TP and FP rates

Table 10 Results of retraining selected neural networks for porosity classification

Network	FNR	TPR	Validation accuracy	Test accuracy
SqueezeNet	0.08	0.92	0.91	0.92
ResNet18	0.07	0.95	0.93	0.95
ShuffleNet	0.11	0.98	0.90	0.95
GoogleNet	0.05	0.91	0.92	0.92
MobileNetv2	0.12	0.96	0.89	0.94
ResNet50	0.07	0.97	0.93	0.95
Darknet19	0.03	0.91	0.91	0.92
VGG16	0.10	0.96	0.93	0.94
Inceptionv3	0.08	0.97	0.91	0.95
NasNetMobile	0.11	0.95	0.91	0.93
EfficientNetB0	0.11	0.94	0.91	0.93
ResNet101	0.13	0.97	0.93	0.94
Xception	0.08	0.93	0.92	0.93
Darknet53	0.11	0.98	0.93	0.95
InceptionResNetv2	0.08	0.96	0.92	0.95

required to determine if they are better suited for transfer learning or small classification sets.

3.5 Accuracy distribution

The distribution of accuracies for each algorithm was evaluated with respect to some potential features of interest such as porosity perimeter, area, aspect ratio, and minimum bounding box angle. All features were used during training and the ResNet18 DCNN was used for the CNN results. The distribution, displayed in Table 11, includes the bin widths and the percent of data contained

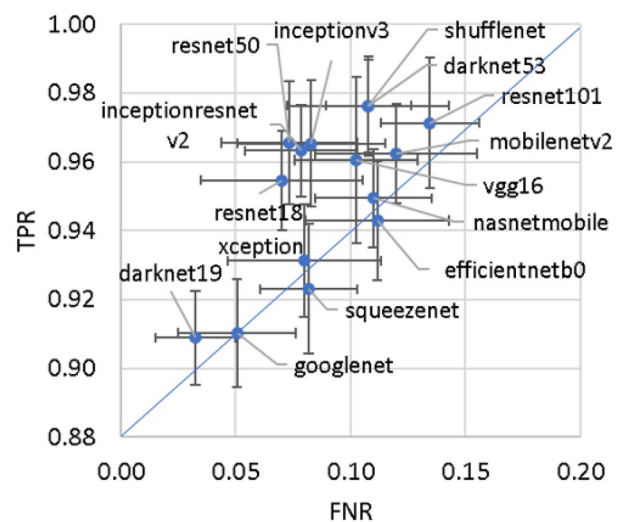


Fig. 10 TPR and FNR comparison of DCNNs

in each bin. Ideally, the distribution would be flat, indicating algorithm invariance and generalizable performance. However, there are some areas where performance notably decreases. Specifically, very small and very large aspect ratios tend to underperform. This is most likely due to the lack of training examples. Small pore areas were also found to underperform in the assessment. This is mostly likely due to the image size being too small to correctly classify the porosity. Bounding boxes with small angles were also found to underperform. Standard deviation of the accuracies for each algorithm was within one another by 1–2%, suggesting that the deficiencies might lie in the dataset. Dataset augmentation could be applied to increase

the number of images in the sparse categories and potentially increase overall accuracy.

3.6 Classification time comparison

Fast classification time is critical for real-time image classification. The time required to extract the 51 features from the 691 cropped images in the testing set and classify them using the six ML methods is displayed in Fig. 11. Each method requires 0.4 s to perform feature extraction from the images which comprises most of the time. Of the fastest performing ML methods, the single-layer ANN had the highest accuracy.

Table 11 Distribution of features of interest with respect to each ML method accuracy. Included are bins with the percent of the dataset they contain

	Bin	% Data	ANN	KNN	BT	DT	Linear	SVM	CNN
Perimeter (um)	25	9%	87%	84%	83%	77%	74%	85%	82%
	45	34%	85%	86%	87%	82%	72%	86%	95%
	70	15%	87%	81%	85%	82%	70%	87%	94%
	90	10%	84%	85%	86%	81%	70%	84%	91%
	110	7%	85%	84%	87%	80%	69%	87%	92%
	130	5%	82%	81%	85%	74%	70%	85%	100%
	150	4%	80%	87%	86%	78%	68%	82%	95%
	170	4%	84%	88%	87%	80%	71%	86%	92%
Area (um ²)	200	12%	86%	87%	87%	80%	74%	88%	91%
	45	19%	83%	84%	83%	78%	74%	84%	84%
	90	18%	85%	86%	87%	79%	70%	86%	95%
	130	14%	87%	84%	85%	81%	71%	85%	95%
	180	7%	86%	85%	90%	84%	77%	89%	95%
	220	6%	85%	83%	85%	83%	70%	89%	100%
	265	5%	86%	85%	85%	82%	67%	89%	91%
	310	3%	85%	83%	88%	77%	67%	86%	100%
	350	3%	81%	85%	84%	77%	79%	87%	100%
	400	25%	85%	87%	87%	80%	74%	87%	94%
Aspect ratio	0.5	2%	68%	70%	75%	64%	72%	82%	50%
	0.7	16%	87%	87%	86%	78%	70%	87%	97%
	0.85	19%	87%	86%	88%	83%	72%	88%	90%
	1	22%	87%	88%	88%	82%	73%	88%	91%
	1.2	17%	82%	83%	84%	81%	73%	85%	91%
	1.4	9%	85%	87%	89%	82%	74%	88%	90%
	1.6	6%	81%	82%	84%	79%	71%	85%	89%
	1.8	3%	89%	88%	84%	83%	71%	88%	79%
Min box angle	2	6%	75%	76%	78%	72%	56%	78%	85%
	5	30%	82%	84%	84%	78%	77%	85%	92%
	15	9%	90%	88%	91%	82%	85%	90%	84%
	25	8%	94%	93%	92%	90%	86%	92%	86%
	35	8%	92%	91%	92%	88%	80%	90%	92%
	45	7%	92%	92%	90%	87%	77%	93%	86%
	55	9%	92%	89%	93%	90%	83%	90%	100%
	65	8%	86%	88%	90%	82%	82%	89%	82%
	75	11%	92%	91%	91%	88%	88%	94%	96%
	85	8%	86%	85%	85%	81%	81%	89%	91%

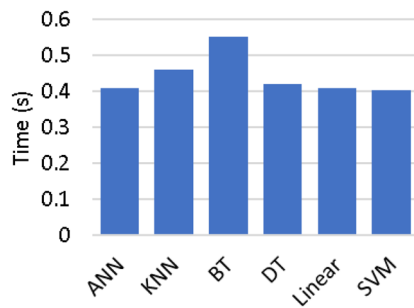


Fig. 11 Classification time comparison

The time required to classify porosities from the 691 images for each DCNN is displayed in Fig. 12. It is found that the best-performing classifiers, ResNet18 and ResNet50, classify images substantially faster at 1.8 s and 3.8 s versus 4.9 s and 9.4 s of the InceptionResNetv2 and InceptionV3, respectively. However, the overall time required for image classification is substantially faster with the manual features coupled with the six ML algorithms.

3.7 Network expansion comparison

Another desirable attribute is the future expansion of the network into classifying other defect types. To perform this assessment, the non-porosity classification is split into two new classifications: light non-porous (LNP) and dark non-porous (DNP). The two top-performing networks, ResNet18 and ResNet50, were then retrained to determine if accuracies

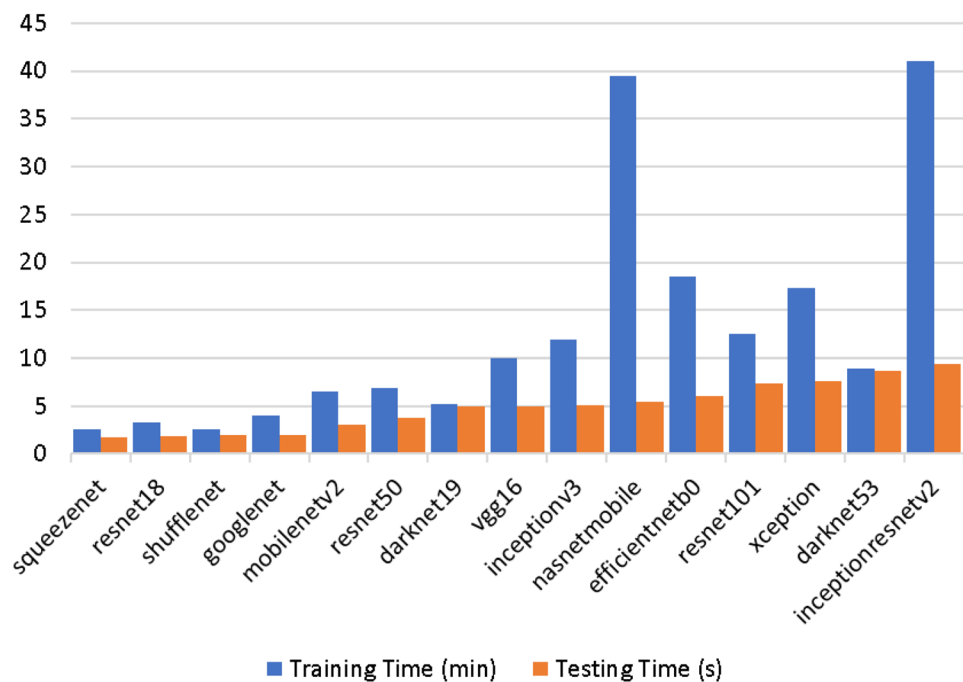
True Class	Predicted Class		
	DNP	LNP	pore
DNP	155	2	2
LNP	5	76	17
pore	3	16	415

Fig. 13 Confusion matrix for ResNet18 network and three classes

remain consistent. The three-classification confusion matrix for the ResNet18 network is displayed in Fig. 13. These results correspond to an accuracy of 93.5%, which is consistent with the previous two class results. The training and classification time also remained constant. The complexity of the porosity database is also noted from this image. Size, shape, depth, and image intensity are all very greatly, yet the DCNN is able to classify correctly in almost all instances.

The three-classification confusion matrix for the ResNet50 network is displayed in Fig. 14. These results correspond to an accuracy of 94.1%, which is consistent with the previous two class results. The training and classification time also remained constant. Some randomly

Fig. 12 CNN training and testing time comparison



True Class	DNP	152	1	2
	LNP	5	79	13
	pore	6	14	419
		DNP	LNP	pore
		Predicted Class		

Fig. 14 Confusion matrix for ResNet50 network and three classes

selected classified images from the testing set are displayed in Fig. 15.

4 Conclusion

In this study, ML algorithms were compared for image-based porosity classification from a diverse and complex porosity image set. The ML methods that require manual feature definition are desirable because they represent a “white box” method where the efficacy of features can be easily analyzed by an observer, and the features can be readily recycled for further material characterization. DCNNs that do not require user-defined features represent a “black box” method where the features are automatically extracted, but unknown to the user.

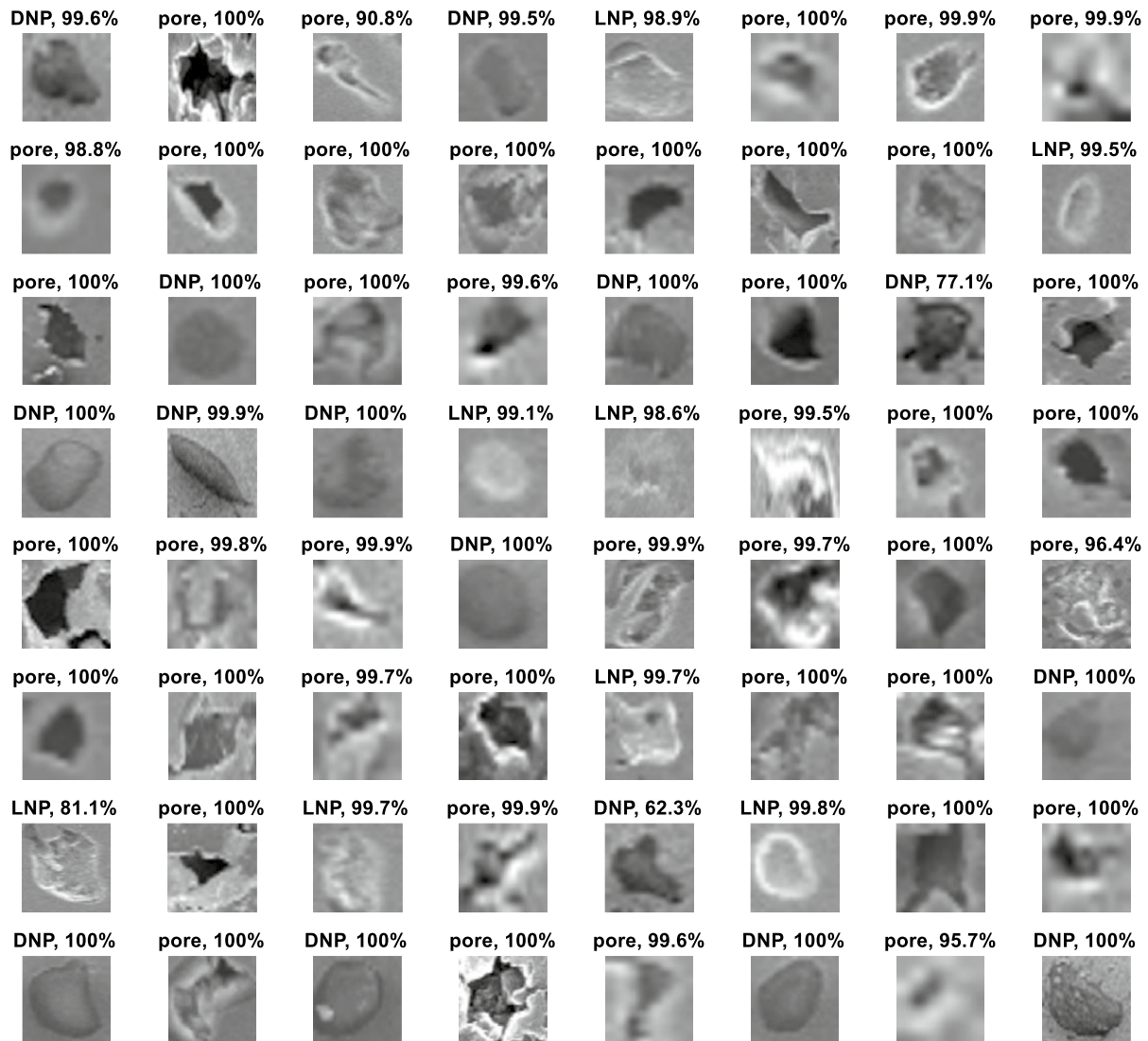


Fig. 15 Three classifier results on 64 randomly selected images

For the six ML methods that are single-layer neural networks, K-nearest neighbors, boost trees, decision trees, binary linear classifier, and support vector machine, 51 features were defined and extracted from the images. From these 51 features, 10 were found to best contribute to porosity classification. Of these algorithms, it was found that K-nearest neighbors, boost trees, single-layer neural networks, and support vector machine performed similarly with accuracies around 85%, and the TPR for each was above 90% and the FNR was below 35%.

For the DCNNs, 15 common networks for image classification were tested. Nearly all the networks performed similarly at around 92–95% accuracy, with TPR around 91–95%, and FNR around 5–11%. However, it was found that the fastest classification and highest accuracy networks were ResNet50 and ResNet18. It was also found that freezing the initial 20% of the layers reduces the training time by 12.5% and does not impact the classification accuracy. Overall, it can be concluded that the smaller ResNet18 network is appropriate for classifying two to three classes; however, as further defects are added, it may be beneficial for the ResNet50 network to be utilized or for other networks to be revisited.

Overall, this study found that there exists merit in the application of ML for porosity detection. The six ML methods with manual feature definition performed at least four times faster; however, the DCNNs performed with the highest accuracy. Another advantage of DCNNs was found in the simplicity of the model. In DCNNs, adding new classes requires only retraining the DCNN with new classified images and the user is not required to define new features. Therefore, a variety of materials and defects can be easily included in the model by simply adding a new training dataset.

Future study includes the application of the DCNN for the porosity detection to analyze and improve AM processes. For example, the analysis of the placement of individual porosities can be used to analyze the quality of the part printed by SAAM with binder jetting compared to that of conventionally processed parts.

Funding This work was supported by the National Science Foundation (award number: DMR-2119832).

Availability of data and materials The data that support the findings of this study will be openly available in MendeleyData at <http://dx.doi.org/10.17632/vhxs23krz4.1>.

Code availability The data that support the findings of this study will be openly available in MendeleyData at <http://dx.doi.org/10.17632/vhxs23krz4.1>.

Declarations

Ethics approval The authors guarantee the study findings have not been previously published.

Consent to participate The authors have consented to participate in the study.

Consent for publication The authors have consented to publish the research findings.

Conflict of interest The authors declare no competing interests.

References

- Goodridge RD, Tuck CJ, Hague RJM (2012) Laser sintering of polyamides and other polymers. *Prog Mater Sci*. <https://doi.org/10.1016/j.pmatsci.2011.04.001>
- Tofail SAM, Koumoulos EP, Bandyopadhyay A, Bose S, O'Donoghue L, Charitidis C (2018) Additive manufacturing: scientific and technological challenges, market uptake and opportunities. *Mater Today*. <https://doi.org/10.1016/j.mattod.2017.07.001>
- Hopkinson N, Hague RJM, Dickens PM (2006) *Rapid manufacturing: an industrial revolution for the digital age*. John Wiley & Sons, Inc
- Bakshi KR (2016) A review on selective laser sintering: a rapid prototyping technology. *IOSR J Mech Civ Eng*. <https://doi.org/10.9790/1684-15008040453-57>
- Chen L, He Y, Yang Y, Niu S, Ren H (2017) The research status and development trend of additive manufacturing technology. *Int J Adv Manuf Technol* 89(9):3651–3660. <https://doi.org/10.1007/s00170-016-9335-4>
- Malekipour E, El-Mounayri H (2018) Common defects and contributing parameters in powder bed fusion AM process and their classification for online monitoring and control: a review. *Int J Adv Manuf Technol*. <https://doi.org/10.1007/s00170-017-1172-6>
- Zhang B, Li Y, Bai Q (2017) Defect formation mechanisms in selective laser melting: a review. *Chinese J Mech Eng* 30(3):515–527. <https://doi.org/10.1007/s10033-017-0121-5>
- Mani M, Donmez M, Feng S, Moylan S, Fesperman R, Lane B (2015) Measurement science needs for real-time control of additive manufacturing powder bed fusion processes. National Institute of Standards and Technology
- Sachs E, Cima M, Cornie J (1990) Three-dimensional printing: rapid tooling and prototypes directly from a CAD model. *CIRP Ann - Manuf Technol*. [https://doi.org/10.1016/S0007-8506\(07\)61035-X](https://doi.org/10.1016/S0007-8506(07)61035-X)
- Srinivas M, Babu BS (2017) A Critical review on recent research methodologies in additive manufacturing. *Mater Today Proc*. <https://doi.org/10.1016/j.matpr.2017.07.258>
- Zhang S, Miyanaji H, Yang L, Zandinejad AA, Dilip JJS, Stucker B (2014) An experimental study of ceramic dental porcelain materials using a 3D print (3DP) process. In: 25th Annual International Solid Freeform Fabrication Symposium & #65533; An Additive Manufacturing Conference, SFF 2014
- Sola A, Nouri A (2019) Microstructural porosity in additive manufacturing: the formation and detection of pores in metal parts fabricated by powder bed fusion. *J Adv Manuf Process*. <https://doi.org/10.1002/amp2.10021>

13. Slotwinski JA, Garboczi EJ, Hebenstreit KM (2014) Porosity measurements and analysis for metal additive manufacturing process control. *J Res Natl Inst Stand Technol*. <https://doi.org/10.6028/jres.119.019>
14. Grasso M, Colosimo BM (2017) Process defects and in-situ monitoring methods in metal powder bed fusion: a review. *Meas Sci Technol* 28(4):44005. <https://doi.org/10.1088/1361-6501/aa5c4f>
15. García-Moreno AI, Alvarado-Orozco JM, Ibarra-Medina J, Martínez-Franco E (2020) Image-based porosity classification in Al-alloys by laser metal deposition using random forests. *Int J Adv Manuf Technol* 110(9–10):2827–2845. <https://doi.org/10.1007/s00170-020-05887-6>
16. Imani F, Gaikwad A, Montazeri M, Rao P, Yang H, Reutzel E (2018) Process mapping and in-process monitoring of porosity in laser powder bed fusion using layerwise optical imaging. *J Manuf Sci Eng Trans ASME* 140(10). <https://doi.org/10.1115/1.4040615>
17. Aminzadeh M, Kurfess TR (2019) Online quality inspection using Bayesian classification in powder-bed additive manufacturing from high-resolution visual camera images. *J Intell Manuf* 30(6):2505–2523. <https://doi.org/10.1007/s10845-018-1412-0>
18. Zhang B, Liu S, Shin YC (2019) In-process monitoring of porosity during laser additive manufacturing process. *Addit Manuf* 28:497–505. <https://doi.org/10.1016/j.addma.2019.05.030>
19. Gobert C, Reutzel EW, Petrich J, Nassar AR, Phoha S (2018) Application of supervised machine learning for defect detection during metallic powder bed fusion additive manufacturing using high resolution imaging. *Addit Manuf* 21:517–528. <https://doi.org/10.1016/j.addma.2018.04.005>
20. Caggiano A, Zhang J, Alfieri V, Caiazzo F, Gao R, Teti R (2019) Machine learning-based image processing for on-line defect recognition in additive manufacturing. *CIRP Ann - Manuf Technol* 68(1):451–454. <https://doi.org/10.1016/j.cirp.2019.03.021>
21. Abdelrahman M, Reutzel EW, Nassar AR, Starr TL (2017) Flaw detection in powder bed fusion using optical imaging. *Addit Manuf* 15:1–11. <https://doi.org/10.1016/j.addma.2017.02.001>
22. Khalid S, Khalil T, Nasreen S (2014) A survey of feature selection and feature extraction techniques in machine learning. In: *Proceedings of 2014 Science and Information Conference, SAI 2014*. <https://doi.org/10.1109/SAI.2014.6918213>
23. O Community (2010) *The OpenCV Reference Manual*. OpenCV
24. Freedman DA (2009) *Statistical models: theory and practice*. Cambridge University Press
25. Cheng JCP, Wang M (2018) Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques. *Autom Constr* 95:155–171. <https://doi.org/10.1016/j.autcon.2018.08.006>
26. Zhang X, Hao Y, Shangguan H, Zhang P, Wang A (2020) Detection of surface defects on solar cells by fusing Multi-channel convolution neural networks. *Infrared Phys Technol* 108:103334. <https://doi.org/10.1016/j.infrared.2020.103334>
27. Sun X, Gu J, Huang R, Zou R, Palomares BG (2019) Surface defects recognition of wheel hub based on improved faster R-CNN. *Electron* 8(5):481. <https://doi.org/10.3390/electronics8050481>
28. Lei HW, Wang B, Wu HH, Wang AH (2018) Defect detection for polymeric polarizer based on faster R-CNN. *J Inf Hiding Multimed Signal Process* 9(6):1414–1420
29. Wu W, Yin Y, Wang X, Xu D (2019) Face detection with different scales based on faster R-CNN. *IEEE Trans Cybern* 49(11):4017–4028. <https://doi.org/10.1109/TCYB.2018.2859482>
30. Huang H, Zhou H, Yang X, Zhang L, Qi L, Zang AY (2019) Faster R-CNN for marine organisms detection and recognition using data augmentation. *Neurocomputing* 337:372–384. <https://doi.org/10.1016/j.neucom.2019.01.084>
31. Ye DH, Zikic D, Glocker B, Criminisi A, Konukoglu E (2013) SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. *ICLR17* 16(Pt 1)
32. Szegedy C et al (2015) Going Deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*
33. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. <https://doi.org/10.1109/CVPR.2016.308>
34. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) MobileNetV2: inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. <https://doi.org/10.1109/CVPR.2018.00474>
35. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. <https://doi.org/10.1109/CVPR.2016.90>
36. Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. <https://doi.org/10.1109/CVPR.2017.195>
37. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, inception-ResNet and the impact of residual connections on learning. In: *31st AAAI Conference on Artificial Intelligence, AAAI*
38. Zhang X, Zhou X, Lin M, Sun J (2018) ShuffleNet: an extremely efficient convolutional neural network for mobile devices. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. <https://doi.org/10.1109/CVPR.2018.00716>
39. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition*. <https://doi.org/10.1109/CVPR.2017.690>
40. Tan M, Le QV (2019) EfficientNet: rethinking model scaling for convolutional neural networks. In: *36th International Conference on Machine Learning, ICML*
41. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*
42. Zoph B, Vasudevan V, Shlens J, Le QV (2018) Learning transferable architectures for scalable image recognition. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. <https://doi.org/10.1109/CVPR.2018.00907>
43. Russakovsky O et al (2015) ImageNet large scale visual recognition challenge. *Int J Comput Vis* 115(3):211–252. <https://doi.org/10.1007/s11263-015-0816-y>
44. MathWorks (2021) Options for training deep learning neural network - MATLAB trainingOptions. <https://www.mathworks.com/help/deeplearning/ref/trainingoptions.html>. Accessed 6 Mar 2021
45. Hussain M, Bird JJ, Faria DR (2019) A Study on CNN transfer learning for image classification. In: Lotfi A, Bouchachia H, Gegov A, Langensiepen C, McGinnity M (eds) *Advances in Computational Intelligence Systems*. UKCI 2018. *Advances in Intelligent Systems and Computing*, vol 840. Springer, Cham
46. Fawcett T (2006) An introduction to ROC analysis. *Pattern Recognit Lett*. <https://doi.org/10.1016/j.patrec.2005.10.010>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.