



Automatic detection and characterization of porosities in cross-section images of metal parts produced by binder jetting using machine learning and image augmentation

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

Received: 8 July 2022 / Accepted: 22 February 2023 / Published online: 1 April 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

In binder jetting followed by sintering, the porosity characterization is critical to understand how the process affects the structure of the printed parts. Image-based porosity detection methods are widely used but the current solutions are limited to specific materials and conditions and require manual tuning that precludes real-time porosity detection. The application of machine learning for automating porosity detection has been also limited to specific materials and conditions and requires a large training dataset for successful implementation. However, large datasets are difficult to acquire experimentally in binder jetting due to prohibited material costs and experiment time. To bridge the knowledge gap, this paper investigates the application of machine learning on automated porosity detection using a small dataset consisting of highly varied cross-section images of metal parts produced by binder jetting followed by sintering. Stylegan3, a type of generative adversarial network, is used to increase the number of training images by image augmentation, and YOLOv5, a convolutional neural network specialized for object detection, is used to detect porosities. The resulting model achieves an F1 score of 88% and detection time of 3–15 ms per image. Generalized porosity detection is also assessed on a set of images containing highly varied materials, resolutions, magnifications, and pore densities. Furthermore, morphological information of the classified porosities such as the distribution of their orientations are automatically extracted using image processing algorithms.

Keywords Porosity · Binder jetting · Sintering · Machine learning · CNN · Image augmentation

Introduction

One of the seven categories of additive manufacturing (AM) technology recognized by ASTM F2792 (ASTM, 2012) is binder jetting (BJ), which was invented in 1993 at the Massachusetts Institute of Technology (Sachs et al., 1990). BJ is commonly used to print green components, which has remarkable advantages in the easy production of components with high-complex geometry (Milewski, 2017; Srinivas & Babu, 2017). BJ uses iterative ink-jet printing of binder on a powder bed to produce metallic (Bai & Williams, 2015; Chen & Zhao, 2016; Do et al., 2017; Doyle et al., 2015; Hong et al., 2016; Mostafaei et al., 2016; Shrestha & Manogharan, 2017; Tang et al., 2016; Wang & Zhao, 2017; Zhou

et al., 2015) or ceramic (Bergmann et al., 2010; Enneti et al., 2018; Gaytan et al., 2015; Gonzalez et al., 2016; Islam & Sacks, 2016; Lanzetta & Sachs, 2001; Lee et al., 2020; Winkel et al., 2012; Yoo et al., 2008) green components. The green components need a post-processing such as sintering to consolidate them and to obtain desired properties. In addition to the advantages mentioned above, BJ can virtually work with any powdered feedstock including functionally graded materials (Moon et al., 2001). Since this technology doesn't need to sinter or melt the powder bed to produce the printed part, the residual thermal stresses can be avoided, and the anisotropic microstructures typical to the powder bed fusion technologies can be controlled (Do et al., 2015, 2017; Miyajagi et al., 2016; Mostafaei et al., 2016; Sachs et al., 1992; Snelling et al., 2015). Moreover, by adding multiple print-heads and nozzles, BJ can be easily scaled and achieved high build rates compared with other AM technologies (Bai & Williams, 2015; Gibson et al., 2015; Winkel et al., 2012). For example, commercially available machines have a large

✉ John S. Kang
jkang4@sdsu.edu

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

build volume up to $2200 \times 1200 \times 600$ mm. These advantages have attracted attention over BJ in recent years.

The primary challenge of BJ followed by sintering is to achieve component densities comparable with those produced by conventional processes. The green components are typically characterized by a low powder packing density (Zhang et al., 2014), which results considerable shrinkage during the follow-up sintering process. The control of dimensional changes during sintering is crucial to achieve the desired dimensional tolerances. Because the bulk porosity in the printed component is directly related to the density of the component, the porosity detection is critical to determine and control the quality of the component.

The porosity and its evolution play a fundamental role to understand and model the sintering behavior (Olevsky, 1998). For example, in the models based on the continuum mechanics (Olevsky, 1998), material properties and constitutive parameters (i.e., normalized shear and bulk modulus and sintering stress) are defined as a function of porosity and its evolution (Manière & Olevsky, 2017). Since these models were developed for the sintering in the traditional powder processing, they might need to be modified to consider the effects of the BJ process on the porosity and its evolution in sintering. Therefore, the capability to detect and quantify the porosity in the parts produced by BJ followed by sintering is fundamental to understand how the process, from the printing to the sintering, influences the structure of the final component.

Cross section image analysis using industry software such as Avizio, ImageJ, and Pace are commonly used for the porosity analysis (García-Moreno et al., 2020). However, the industry software has limitations to detect all true pores because it uses intensity thresholding method that requires non-gradient background and consistent pore intensity (García-Moreno et al., 2020). It also requires user interaction on determining the threshold, which is time-consuming and hinders automation. In addition, it is error-prone for complex pores and lighting conditions. These disadvantages of the traditional image-based porosity detection methods could be eliminated by applying machine learning (ML) techniques to process the cross-section images of printed components (García-Moreno et al., 2020).

A few studies used ML techniques for porosity detection, however they are optimized for non-complex pore shapes, single materials, or do not provide high porosity detection accuracy. For example, image-based porosity detection for AM has been previously performed with support vector machines; however, the performance was low yielding an F1 score of 62% from a precision of 64% and a recall of 60% (Gobert et al., 2018). Porosity detection in Ti–6Al–4V was performed during laser powder bed fusion (LPBF) achieving accuracies of 89.36% for support vector machine (SVM),

78.60% for k-nearest neighbor (KNN), and 84.40% for Artificial Neural Network (ANN) (Imani et al., 2018). Another LPBF study focused on classifying porosities in Inconel 625 using a Bayesian classifier resulting in an accuracy of 89.5% (Aminzadeh & Kurfess, 2019). Another study used optical images of Ti–6Al–4V parts during LPBF to identify porosities resulting from lack of fusion reported a 91.5% accuracy (Abdelrahman et al., 2017). SVM was also employed to classify porosities in stainless steel parts created with LPBF and yielded accuracy of 85% (Gobert et al., 2018). CNNs were used for pore classification in parts created from sponge titanium powder with direct energy deposition and yielded an accuracy of 91.2% (Zhang et al., 2019). Another study focused on pore classification in Al-5083 composed parts produced with direct metal deposition using a random forest classifier resulting in an accuracy of 94.41% (García-Moreno et al., 2020). However, the results were performed on a single material with sparse porosity. Therefore, a generalized image-based porosity detection algorithm that can be applied to any materials, pore density, and varied image resolutions is required.

The objective of this paper is to develop image based automated porosity detection methods on a limited dataset. This is accomplished by first augmenting the dataset with a generative adversarial neural network (GAN), then training two popular CNNs: the Faster R-CNN and YOLOv5. Transfer learning was applied to the GAN network to train cross-section images within a reasonable time, and hyperparameter tuning was performed on the CNNs to optimize performance. The final selected model was assessed on various porous images obtained from literature containing various materials, magnifications, and resolutions. The results indicate that the model can perform well on cross-section images beyond the training set.

Approach

CNNs are widely used for classifying images with defects (Scime & Beuth, 2018). A common CNN classifier was found to be the Faster Region Based CNN (Faster R-CNN) (Ren et al., 2017). Faster R-CNN has previously been used for face detection resulting in precision and recall rates at over 95% (Wu et al., 2019). Faster R-CNN has also been used in automating the detection of maize tassels with unmanned arial vehicles yielding accuracies from 88 to 95% (Liu et al., 2020). High accuracies were also obtained with Faster R-CNN for detecting marine organisms with data augmentation (Huang et al., 2019). It has also been utilized extensively in defect detection including sewer pipe, solar cell, wheel hub surface, and polymeric polarizer defect detections (Cheng & Wang, 2018; Lei et al., 2018; Sun et al., 2019; Zhang et al., 2020). However, the training times of Faster R-CNN is

around several days (Ren et al., 2017). To address the potential training time deficiency of Faster R-CNN, YOLOv5, another common CNN-based classifier was investigated for image-based defect detection (Bochkovskiy et al., 2020; Jocher et al., 2020). Accuracies with YOLOv5 are reported to be similar to Faster R-CNN but can perform image detection in real time, with image processing on the order of milliseconds (Bochkovskiy et al., 2020; Jocher et al., 2020; Zhao et al., 2021). YOLOv5 has been used in a variety of applications including wheat spike detection with an accuracy of 94.1%, kiwi fruit defect detection with accuracies of 94.7%, and 94.4% accuracy for crack detection in asphalt pavement (Li et al., 2021; Yao et al., 2021; Zhao et al., 2021).

Equally important to the model is the dataset the model is trained on. Because of the prohibited material costs and experiment time, it is difficult to experimentally obtain large datasets in BJ. Therefore, the study approach includes image augmentation to increase the training data set. As an additional complication, the images are composed of varied materials, yielding a highly varied image set. Due to the small image set and high diversity of images, unsupervised ML for detection was not considered. However, unsupervised learning for image augmentation was incorporated using neural networks, specifically stylegan3, a type of GAN network, to increase the existing dataset size (Heusel et al., 2017; Karras et al., 2021).

This section describes how we designed and tuned the CNN, Faster R-CNN, YOLOv5, and GAN models for the application of porosity detection, followed by the dataset description. The classification metrics used to evaluate and compare those ML models are introduced first followed by a description of each algorithm.

Classification metrics

The performance of the proposed ML models must consider how many of the pores in the image are classified correctly (precision) and how many true pores in the image are classified (recall). The F1 Score is used to combine both precision and recall into a single reporting metric (Sasaki, 2007). The equations to calculate the precision, recall, and F1 score are provided Eqs. 1, 2, and 3.

$$Precision = \frac{\# \text{ True Objects Detected}}{\text{Total} \# \text{ Objects Detected}} \quad (1)$$

$$Recall = \frac{\# \text{ True Objects Detected}}{\text{Total} \# \text{ True Objects}} \quad (2)$$

$$F1_{score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

The ML models are retrained over a specified number of epochs and the mean squared error over a validation curve

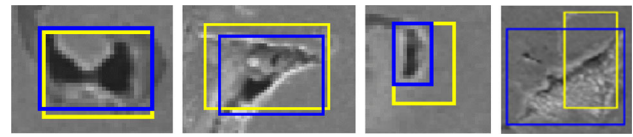


Fig. 1 IoU of 95% (left), 67% (left-middle), 49% (right-middle), 34% (right). Blue represents the ground truth label and yellow is the trained detector label (Color figure online)

is calculated to determine the performance of the models at each epoch using the Eqs. 4, 5, and 6.

$$MSE_P = \frac{\sum_N (1 - Precision)^2}{N} \quad (4)$$

$$MSE_R = \frac{\sum_N (1 - Recall)^2}{N} \quad (5)$$

$$MSE_{net} = \frac{MSE_R + MSE_P}{2} \quad (6)$$

where N is the number of threshold values evaluated for the precision and recall obtained at each threshold. An ideal classifier would have an MSE of 0 for each equation, indicating a perfect model.

Another metric of interest is the intersection of union (IoU). The true porosities were manually classified by specifying a bounding box around the pore. Nonetheless, the predicted bounding box will not be in the exact same location as the training set. Therefore, the bounding boxes must be compared, and some limiting overlap threshold must be specified to determine if the pore has been accurately detected. The metric used to determine this minimum intersection is the IoU. This is defined as the area of the intersection divided by the area of union. The equation for IoU is presented in Eq. 7.

$$IoU = \frac{A_I}{A_U} = \frac{A_I}{A_{GT} + A_D - A_I} \quad (7)$$

where A_I is the area of intersection, A_U is the area of union, A_{GT} is the ground truth area (manually labeled area), and A_D is the area marked by the trained detector. The union area is simply the area of the ground truth plus the area of the detector minus the intersection area. The IoU is defined to determine if an area is correctly marked. A demonstration of IoUs is displayed in Fig. 1.

A final metric to consider is the mean average precision (mAP). The precision and the recall are both functions of the confidence threshold of the detector model. When the model performs a detection, it is accompanied by a confidence value associated with the probability of the object belonging to a specific classification. Generally, setting this value to low results in high recall, but low precision, and setting the value high results in high precision but low recall. For the model

to be independent of the confidence threshold, it is common to integrate over the threshold for the precision metric.

$$AP = \int_0^1 Precision(T) dT \quad (8)$$

where T is the confidence threshold and AP is the average precision. The closer AP is to 1, the better the classifier. For a model that performs multiple classifications, the AP is computed for each class then the mean is computed to obtain the mAP .

$$mAP = \frac{\sum AP_i}{N} \quad (9)$$

where N is the number of classes. When a model performs single class detection, (8) and (9) are equal.

CNN for image-based porosity detection

Developing a deep neural network from scratch is an iterative and time-consuming process. ResNet deep CNN is commonly used for image localization and classification. Its high accuracy pretrained networks are readily available and are used to classify 1000 everyday objects. We retrained ResNet50 deep CNN (He et al., 2016) using transfer learning (Hussain et al., 2019) to use it for porosity detection (Satterlee et al., 2022). This is accomplished by removing the final two layers of the ResNet50 network: the fully connected layer and the classification output. Since only pores and non-pores are of concern, these final layers must be replaced with layers that perform only two classifications. As more defects are added to the image dataset, the layers can be further modified to account for these defects. The initial layers recognize basic shapes and contours of the image (Hussain et al., 2019). Since these layers perform fundamental image processing that can be applied to any images, these layers need not be updated. Therefore, the weights of the initial 10 layers are not updated when the network is retrained with the pore dataset. This allows the model to be fully trained to detect porosity in just 13 epochs (just under 15 min). Further in-depth information on the setup and training of this model are discussed in (Satterlee et al., 2022).

The CNN is trained on manually segmented images containing pores and non-pores from the training dataset. The testing dataset is automatically segmented using the methods discussed below to produce candidate regions of interest. The trained CNN classifies the proposed regions to obtain the results. Potential porous regions to be classified were selected using a threshold plus gradient method further discussed in (Satterlee et al., 2022). This initial method required manual selection of the intensity threshold and the gradient to select regions. It was performed as a scoping exercise to determine if automating the threshold and gradient selection

would provide a viable method for region proposals. During initial testing on the original full dataset, the recall and the precision for the CNN classifying the proposed regions were 56% and 70%, respectively, for an IoU of 50%. However, the recall and precision are not considered sufficient for industrial applications, and the method was not further considered. Therefore, another region proposal method was considered.

Another regional proposal method with higher recall was employed to test the CNN. This method utilized edge detection to find contours around regions of interest (Dollár & Zitnick, 2015). These regions however are large and require segmentation, so a dark mask was applied to the original image to create seeds for the OpenCV watershed algorithm (Community, 2010). These dark regions are potential centers of pores. With this method hundreds of regions are proposed for classification by the CNN. Using this method, the recall was increased to 72%, however, the precision remained nearly constant at 71%.

From this assessment, it was found that the region proposal algorithms selected for the CNN did not perform adequately. As a result, additional research was performed to determine a ML algorithm that both proposes regions and classifies them, which is introduced in the next section.

Faster R-CNN for image-based porosity detection

A potential solution comes from an extension of the CNN to the R-CNN. This method combines region proposals with a CNN to classify the region of interest (RoI). There are three types of R-CNNs that have been developed: R-CNN, Fast R-CNN and Faster R-CNN (Ren et al., 2017). The primary difference between Faster R-CNN and the other two networks is that Faster R-CNN is an integrated part of the deep learning network. This means that the image is only passed through the network once. R-CNN, on the other hand, requires a proposal network to first extract regions of interest, then each region is extracted, scaled, and independently classified. This ultimately results in longer image classification time and lower accuracy.

The training time of a Faster R-CNN network is significantly longer than a normal CNN. Therefore, a small network was trained to select hyperparameters, then a larger network was finally trained to increase result accuracy. Typically, the trend in accuracy and training time is proportional to the size of the neural network. For context, the time for training ResNet-18, ResNet-50, or ResNet-101 were about two hours, twelve hours, or sixty hours, respectively.

In Faster R-CNN, RoIs are determined by predefined anchor boxes using a Region Proposal Network (RPN) displayed in Fig. 2. Just as a CNN has classification outputs, the Faster R-CNN also classifies RoIs. As the detector is looking for pores, it uses a sliding window that moves across

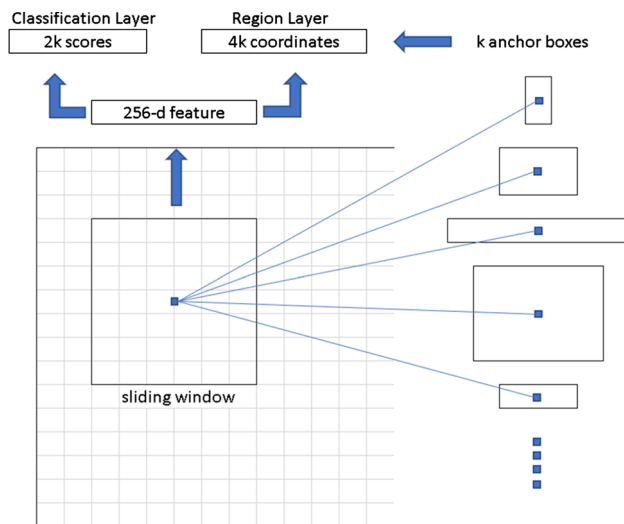


Fig. 2 Faster R-CNN region proposal network (Ren et al., 2017)

Table 1 Number of anchor boxes vs IoU over 15 trials

Num boxes	Mean IoU	Max IoU
100	0.85	0.88
50	0.83	0.87
35	0.81	0.85
25	0.78	0.84
15	0.77	0.82
10	0.70	0.78

the image. Within the window it uses the previously specified anchor boxes built into the RPN to propose regions and classify them. Therefore, it is critical to select anchor boxes that can accurately bound porosities. The number of anchor boxes, however, is less important than the dimensional distribution of the boxes selected. Table 1 was generated creating a cumulative distribution function (CDF) of the training data bounding boxes then selecting anchor boxes based on this data using a Monte Carlo method. The process was repeated 15 times for each number of boxes to obtain some statistics. The mean IoU for the generated anchor boxes were computed and the average and max IoUs were extracted over the trials. It is clear from Table 1 that there are diminishing returns for selecting a greater number of anchor boxes.

Ultimately, anchor boxes with IoUs greater than 70% for a ground-truth box will be marked as positive boxes and boxes that overlap less than 30% for all ground-truth boxes will be marked as negative boxes during the training (Ren et al., 2017). Any boxes between these values are unused during training. Effectively, moving from 50 to 100 boxes will require more time for the boxes to be classified since it is known that the additional 50 boxes added only contribute

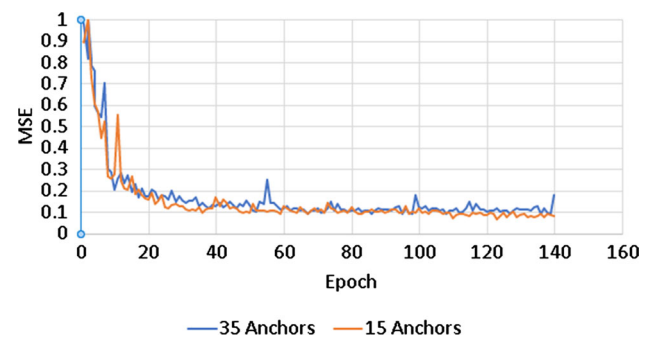


Fig. 3 ResNet50 MSE for 35 and 15 anchor boxes

to a 1% increase in IoU. In addition, many of these boxes are not likely meet the 70% requirement over the larger dataset.

Investigation was performed on larger sets of anchor boxes; however, it was found at testing that increasing the number of bounding boxes resulted in lower precision and the detector would not perform at all at probability thresholds greater than 10%. Note that the ResNet18 deep neural network was used to reduce training time. The observation reveals that increasing the number of anchor boxes from 15 to 35 does not appear to have a major impact, however, increasing to 100 results in a notable change to the regions of interest. Based on these results, an anchor box of 15 was selected for training. This assumption was later tested by running the ResNet50 network with 35 anchor boxes to 140 epochs. The resulting MSE per epoch for the test set is displayed in Fig. 3. It was found that decreasing the number of anchor boxes resulted in a slightly better outcome. However, caution should be applied in considering this outcome in generality since the boxes were randomly selected based on the training set and more iterations may be required to achieve minimum error.

Another hyperparameter to consider is the learning rate. A learning rate of 2×10^{-3} was used in the ResNet18 bounding box assessment after finding that a rate of 5×10^{-3} diverges. After testing the learning rate of 2×10^{-3} on the ResNet50 network, the loss was found to diverge. Therefore, the rate was reduced to 1×10^{-3} and convergence was observed.

YOLOv5 for image-based porosity detection

The YOLOv5 model consists of three stages, the backbone, the neck and head. The backbone and neck extract features from the image and aggregates the features, respectively. The aggregated features are then passed to the head for bounding box prediction and classification. Since the features are used for both bounding box prediction and classification, the algorithm only makes one pass over the image (Zhu et al., 2021). The model produces multiple outputs to facilitate a larger range of bounding box sizes, which is a distinct difference from the Faster R-CNN moving window method.

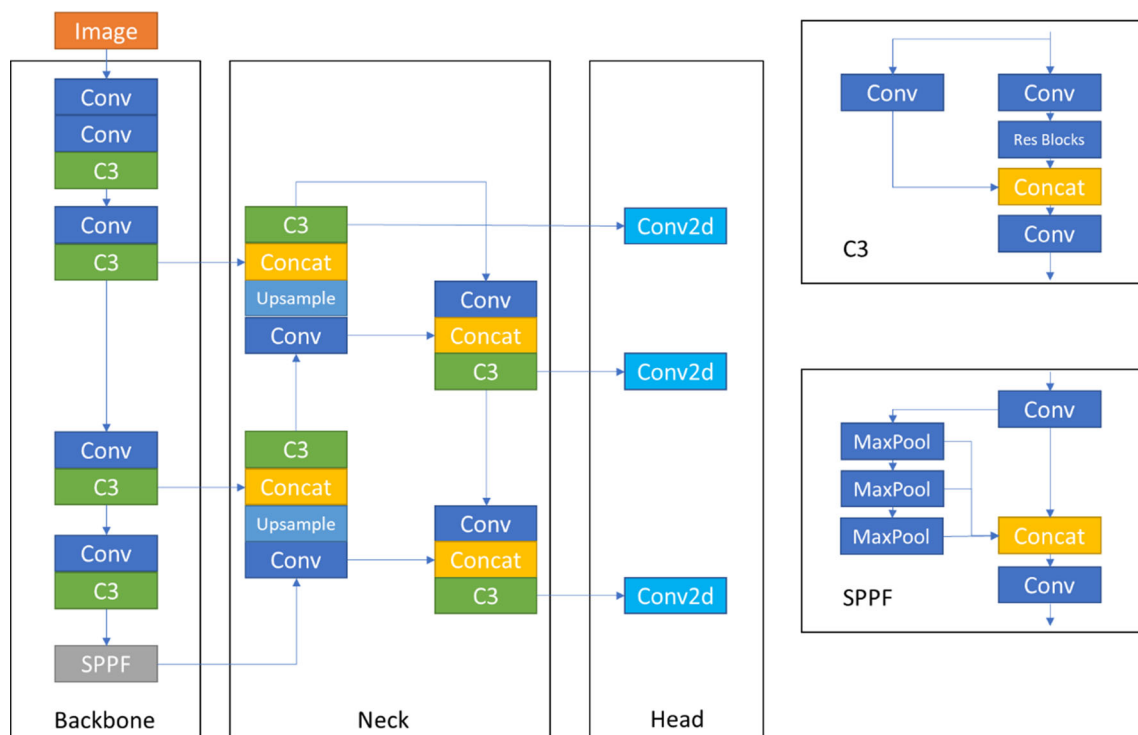


Fig. 4 YOLOv5 high level architecture

Five default models are provided for YOLOv5, YOLOv5n6, YOLOv5s6, YOLOv5m6, YOLOv5l6, and YOLOv5x6. Each model provides a tradeoff between detection time and accuracy. YOLOv5n6 is the smallest model (3.2 M parameters) and thus provides the fastest detection time with the lowest accuracy. YOLOv5x6 is the largest model (140.7 M parameters) and provides the longest detection time with the highest accuracy (Jocher et al., 2021). The “6” at the end of the model name indicates the updated version over the initial models offered in the release version. These models contain an additional output layer, and they consistently outperform the base models in pretrained benchmarks (Jocher et al., 2021).

The basic YOLOv5 architecture consists of three stages, the backbone, neck, and head as shown in Fig. 4. The backbone primarily performs feature extraction, neck constructs a feature pyramid, and the head provides the detected region, classes, and confidence used for computing the loss and labeling the image. Specific architecture for each model differs in the number of residual blocks used in the backbone. During training, labeled images from the training set including the Stylegan3 network generated images are fed into the backbone and the network head outputs the image labels which are used to compute the loss and perform backpropagation. The trained network performs similarly with the network taking

an input image and producing the labels; however, the confidence output is used to limit the number of labels generated based on a threshold.

An advantage of the YOLOv5 model is the ease of implementation. The model is set up with default hyperparameters being applicable to most training scenarios. Manual deviations from the default hyperparameters did not result in notable performance increases. The base learning rate of 0.01 was decreased to 0.005 but the model failed to converge. The learning rate was increased to 0.05 but resulted in reduced performance. The weight decay parameter was also increased to $1e-3$ and $5e-3$ from the initial $5e-4$ value but no performance gains were observed. However, in addition to the learning parameters, there are various image augmentation parameters which can also be tuned. Therefore, additional hyperparameters tuning was performed with the built-in genetic algorithm (Jocher et al., 2021) for 10 epochs over 100 iterations for each model. In general, the hyperparameters tuning resulted in around 1–2% performance gain during tuning. The performance gain per iteration of the best model is presented in Fig. 5.

The final training hyperparameters selected from the 100 iterations of tuning are provided in Table 2. Class BCE weight (cls_pw) was not modified because there is only a single class. However, the class weight (cls_pw) is used as the weight for the implemented IOMA loss function described

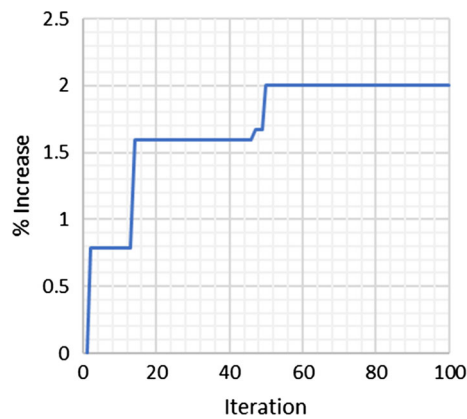


Fig. 5 Percent increase from base parameters of best predicted model per iteration during genetic hyperparameter tuning

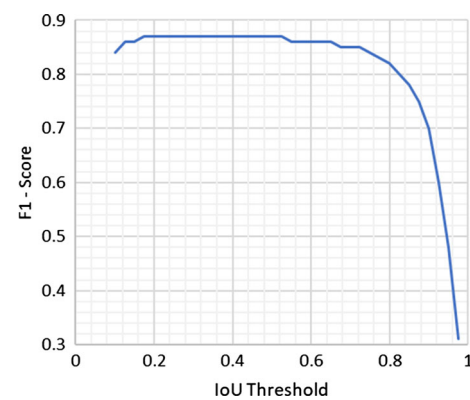


Fig. 6 F1 score vs IOU threshold using non-maximum suppression on final full dataset

Table 2 Initial and final hyperparameters following tuning

Hyperparameter	Final	Initial
lr0	0.00908	0.01
lrf	0.01068	0.01
momentum	0.93651	0.937
weight_decay	0.00082	0.0005
warmup_epochs	3.5766	3
warmup_momentum	0.47282	0.8
warmup_bias_lr	0.09904	0.1
box	0.0292	0.05
cls	0.47673	0.5
cls_pw	1.0	1.0
obj	0.82419	0.7
obj_pw	0.73048	1.0
iou_t	0.25474	0.2
anchor_t	5.5188	4
hsv_h	0.00665	0.015
hsv_s	0.63794	0.7
hsv_v	0.22542	0.4
translate	0.06897	0.1
scale	0.3162	0.5
shear	0.2	0.2
flipud	0.5	0.5
fliplr	0.5	0.5
mosaic	1	1

in this section, so it was not fixed. Also, an interesting outcome was that the scale was decreased significantly from 0.5 to 0.32. This is likely due to small pores in the image that cannot be resolved if the image is scaled down excessively. However, the overall results of the fine tuning did not substantially improve the results, indicating that the default values are close to optimal.

Preliminary testing was performed on the five different models via the full image set with results displayed in Table 3. There was large increase in performance from the YOLOv5n6 model to the YOLOv5s6 model, then smaller performance increase with each subsequent set. The time increase, however, is substantially higher for the larger models. The YOLOv5x 6 took 8 times longer to train than the YOLOv5s6 model and 2.5 times longer for image detection but only provided a 5% performance increase. Since there is no clear advantage to any model, they were all tested on the final full dataset.

A fundamental challenge in porosity detection is ambiguity. It is often possible for bounding boxes to localize a pore correctly in more than one way. This can lead to a pore being labeled multiple times erroneously. PyTorch includes a built-in function that perform non-maximum suppression. This function takes the proposed bounding boxes along with the box score to remove low-scoring, redundant boxes greater than a specified threshold. Unlike other hyperparameters, this one is not included in the genetic tuning algorithm, so manual scoping was performed with results displayed in Fig. 6. The default IOU value of 0.6 specified in YOLOv5 was found to produce a slightly lower F1 score compared to a lower IOU value such as 0.4 on the final full dataset.

Another method of limiting bounding boxes was attempted by enforcing the number of fully overlapping bounding boxes via the loss function (Fig. 7). This was performed by computing the number of times each box overlaps fully with other boxes (i.e., one box is contained within another box). To determine if one box is contained within another box, the intersection over minimum area (IOMA) is computed and compared to a threshold value of 0.9 (e.g., 90% of one box area is contained within another). True bounding boxes (that are accurately predicted), however, should not be penalized for overlapping with other true boxes as shown in Fig. 8. To determine if a box is false, the IOU of the predicted and ground truth box was computed. An IOU less than 0.5

Table 3 Preliminary YOLOv5 training results on full image set

Model	TP	F1	mAP	Train time (min)	Detect time (ms)
YOLOv5n6	0.76	0.71	0.69	14	3.5
YOLOv5s6	0.77	0.73	0.74	20	4.4
YOLOv5m6	0.80	0.75	0.76	24	6.1
YOLOv5l6	0.81	0.76	0.77	55	6.9
YOLOv5x6	0.82	0.77	0.76	173	11.5

Fig. 7 Python implementation of IOMA loss function in Eq. 10

```

def compute_ioma(bboxes, iou_boxes, hyp):
    ioma_thres = 0.9
    iou_thres = 0.5
    bad = iou_boxes.clone() <= iou_thres
    good = iou_boxes
    bboxes_g = bboxes
    bboxes_b = bboxes[bad].clone()
    g = bboxes_g.shape
    s = bboxes_b.shape
    # compare one box with all other boxes
    if (g[0]>0) and (s[0]>0):
        # concat all boxes s[0] times
        ext_box_g = torch.concat([bboxes_g]*s[0],1)
        # reshape tensor so repeat occurs every s[0] times
        ext_box_g = torch.reshape(ext_box_g, (g[0]*s[0],4))
        # concat bad boxes g[0] times
        ext_box_b = torch.concat([bboxes_b]*g[0],0)
        # compute areas and minimum areas for each pair
        a1 = ext_box_g[:,2]*ext_box_g[:,3]
        a2 = ext_box_b[:,2]*ext_box_b[:,3]
        a_min = torch.minimum(a1, a2)
        # compute intersection of box pairs
        inter = torch.squeeze(torch.iou(bbox_i(ext_box_b, ext_box_g), 1))
        # compute ioma and increment pairs greater than threshold
        iou_min = torch.gt(torch.div(inter, a_min), ioma_thres)
        # rearrange so number times overlap occurred can be summed
        iou_min = iou_min.view(g[0], s[0])
        # sum the number of times overlapping occurred
        loss_update = torch.sum(iou_min, 1)
        # remove case where self overlap occurs
        loss_update = loss_update - 1
        # divide by number of cases
        niou = loss_update / (s[0]*g[0])
        # compute the mse
        mse_iou = torch.mean(torch.pow(niou, 2)).unsqueeze(0)
    else:
        mse_iou = torch.zeros(1)
    return mse_iou.to("cuda:0")

```

was used to designate a false box. The IOMA loss was then calculated by summing the number of overlaps false boxes make with other boxes greater than the 0.9 threshold. For a set of n boxes containing m false boxes, the loss is computed as follows then multiplied by a gain and added to the base loss function.

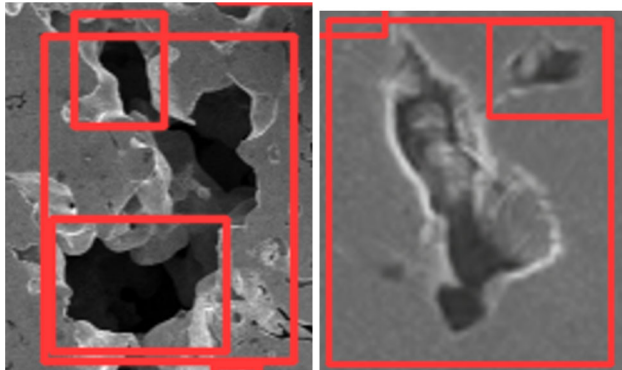
$$loss_{IOMA} = \left(\frac{\sum_{i=1}^m \sum_{j=1}^n IOMA_{ji} > 0.9}{m * n} \right)^2 \quad (10)$$

The addition of the loss function to enforce reduced full bounding box overlap resulted in a minor increase to the

YOLOv5n6 model but the performance of the YOLOv5s6 model remained mostly unchanged as shown in Table 4. This indicates the YOLOv5n6 model was underperforming due to the excessive and redundant labeling, while the YOLOv5s6 model likely did not contain any redundancies. A redundant area found in the YOLOv5n6 results was compared with the YOLOv5s6 and the updated loss function results for both as shown in Fig. 9. In the YOLOv5n6 example, the redundant areas are successfully reduced, and the pore is instead segmented into multiple boxes with box intersection occurring along the necking regions. In the YOLOv5s6 example, there

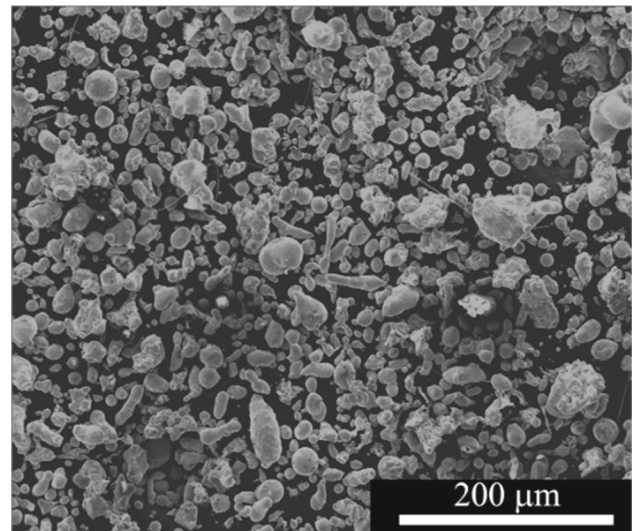
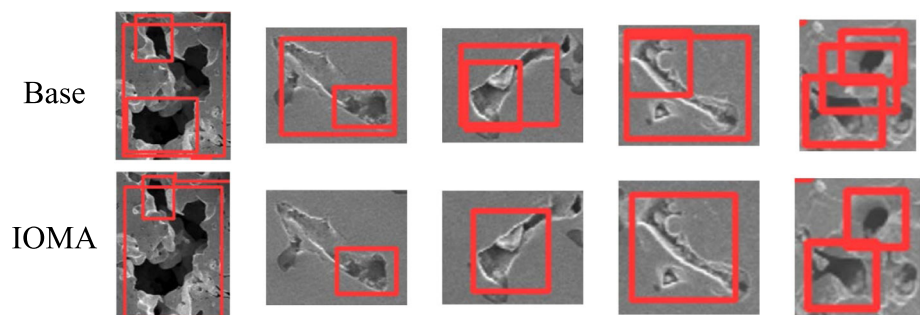
Table 4 Comparison of base loss function and modified loss function with IOMA for false cases overlapping with true cases

Model	TP	F1	mAP
YOLOv5n6	0.76	0.71	0.69
YOLOv5s6	0.77	0.73	0.72
YOLOv5l6	0.81	0.76	0.77
YOLOv5n6_ioma	0.81	0.73	0.73
YOLOv5s6_ioma	0.79	0.74	0.73
YOLOv5l6_ioma	0.84	0.77	0.79

**Fig. 8** Example of incorrect bounding box overlap (left) and correct bounding box overlap (right)

are no bounding boxes meeting the redundant criteria, however with the updated loss function, the pore is segmented into multiple parts like the YOLOv5n6 model. The total number of redundant examples were search for in the original full dataset. It was found that only a one examples could be found for the YOLOv5s6 model, while 17 were found for the YOLOv5n6. This indicates that the loss function update can increase detection performance if redundant detection occurs often. However, the IOMA loss function is computationally expensive and roughly doubles the training time.

Comparisons of overlapping regions observed in the YOLOv5n6 base results are presented with their updated loss function counterparts in Fig. 9. It is observed that the updated loss functions solve the redundant labeling in each

Fig. 9 Overlapping areas compared with base loss function (left) and IOMA loss function (right)**Fig. 10** Stainless-steel powder used in the present work

case, however, the results are still ambiguous since it might be contended that of the overlapping boxes, a better box may have been selected.

Loss function testing was performed on the original full image dataset, so additional testing on the final full dataset will need to be performed to determine the applicability of loss function modifications on the final model. As seen in the analysis, the applicability depends on how prevalent the bounding box redundancy is for a given model. Furthermore, enhanced results may be obtained by tuning the IOMA overlap, the IOU overlap thresholds, and the gain, however due to the long training time this tuning activity was not pursued.

Dataset preparation

Experimental dataset

Cross-section images of metal specimens produced by BJ followed by sintering were obtained and used to test and validate the proposed ML approaches. The specimens are cubes composed of four different types of materials that are stainless steel 316L, boron carbide, copper, and nickel with

Table 5 Particle size distribution for the stainless steel 316L used in the present work

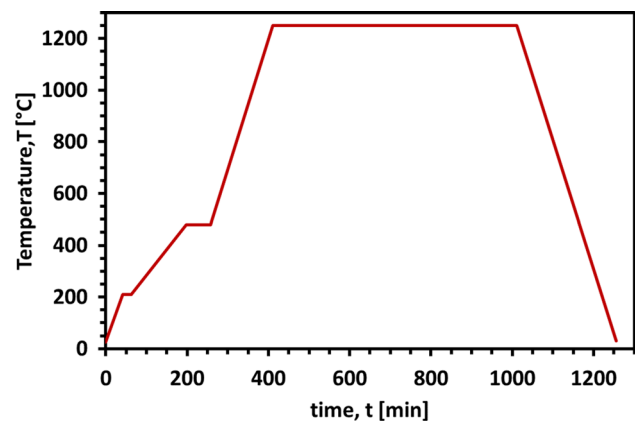
Distribution	Size (μm)
D ₁₀	8.4
D ₅₀	29.5
D ₉₀	55.9

10 mm edge dimension which were printed using a BJ printer (Jet Zprinter[®] 350, Z-Corp, USA) with 5%w of binder, varying the layer thickness (100–250 μm). For example, Fig. 10 shows the particle shapes of the water atomized stainless-steel powder (OzoMetal, USA) used in this study.

The average particle size is 32 μm with and the size distribution has been measured using a particle size analyzer (PSA 1090DL, Anton Paar, Austria). In Table 5 the value obtained for the particle size distribution for powder involved in the present work is shown.

The printed cubes were subjected to curing at 80 °C in a vacuum oven (25L 200C Vacuum Oven, MTI Corp., USA) for a half hour: this helped to better consolidate and dry the printed specimens. The profile of the debinding and sintering conducted in a tube furnace (GSL-1700X-KS-UL-60, MTI Corp., USA) with vacuum are shown in Fig. 11.

The debinding is composed by two steps; the first is at 210 °C with 20 min hold and the second is at 480 °C for one-hour 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, with a heating rate of 5 °C/min the sintering temperature 1250 °C was reached and held for 10 h. The average relative densities

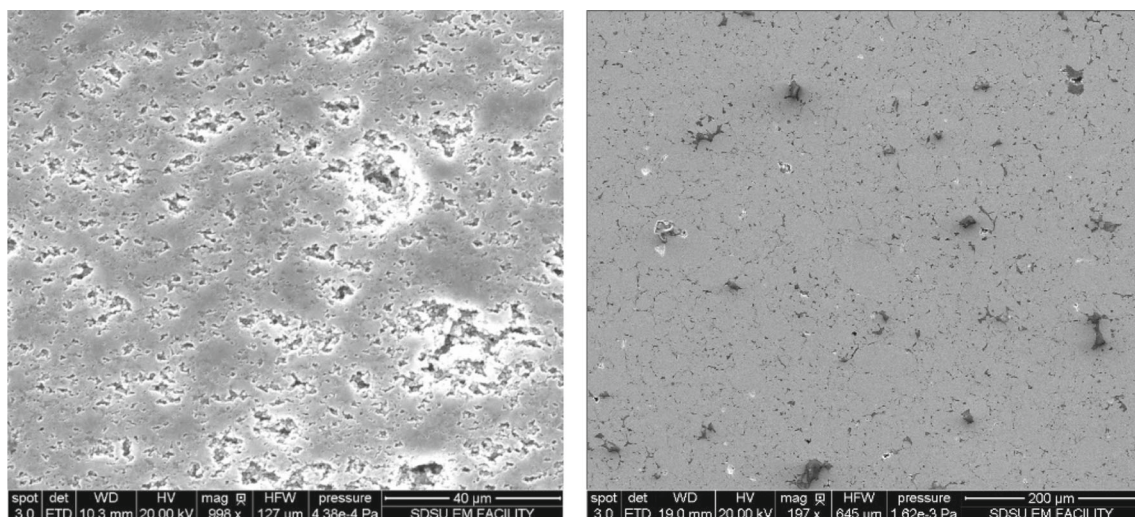
**Fig. 11** Debinding and sintering profile

of the as printed and sintered specimens were measured using the Archimedes method.

To obtain the cross-section images, metallographic specimens were produced by sectioning the cubes and polishing the surface using sandpaper (180 grit -4000 grit) followed by the finishing using the 3 μm and 1 μm diamond suspensions (Struers, Denmark). 67 images of the microstructure were obtained using a scanning electron microscope (SEM) (FEI Quanta 450 FEG, Thermo Fisher Scientific, USA). The SEM cross-section images were divided into two groups: 58 images with sparse porosities and 9 images with dense porosities. Examples of the SEM cross-section images with sparse and dense porosities are shown in Fig. 12.

Image augmentation

The initial original full dataset consisted of 67 images containing 4525 pores assessed via manual labeling by an

**Fig. 12** Example of dense (left) and sparse (right) porous images

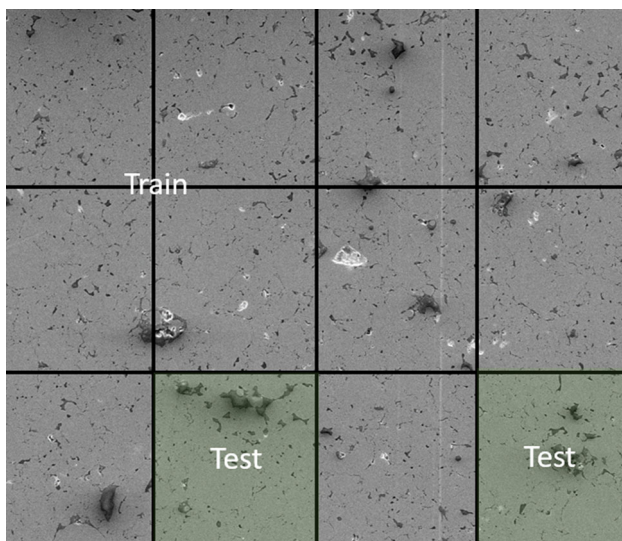


Fig. 13 Image segmenting scheme to increase dataset size

experienced expert. In order to increase the dataset size to improve detection accuracy, initial image augmentation was performed by segmenting the image and training a Generative Adversarial Network (GAN) using Stylegan3 (Karras et al., 2021). Stylegan3 is an aliasing-free GAN network which solves the issue of texture sticking in which textures appear to remain static when traversing the latent space. Conceptually, coarse features are simply filled by fine texture features without consideration for the precise location resulting in them occupying the same pixel coordinates in each image (Karras et al., 2021). In the task of image detection, we train an algorithm to extract features from an image and use these features to find patterns in order to locate and classify an object of interest. Therefore, it is advantageous to remove any synthetic patterns from the image such as previously mentioned texture sticking.

To perform image augmentation, the original full 67 images in the dataset were segmented into a 4×3 grid as shown in Fig. 13, which resulted in the 804 image segmented dataset. Segmenting the images is advantageous for three reasons: (1) it ensures that similar images are presented during training, validation, and testing; (2) it increases the dataset size for image augmentation; (3) it reduces the image size for augmentation. The final point is important when limited hardware resources are available as training time will be drastically reduced (Karras et al., 2021).

The stylegan3 model was trained using transfer learning by freezing the first 10 layers of the model. The pretrained model selected was a 256×256 pixel landscape generation network (Pinkney, 2022). This model was selected since generated images contain little symmetry and contain features similar to the cross-section images of metal parts. In contrast, transfer learning from a network trained to generate



Fig. 14 Transfer learning using stylegan3 network pretrained on human faces transfers symmetry to the cross-section images within the same latent space

human faces results in poor quality because the symmetry is transferred over to the cross-section images. Figure 14 demonstrates symmetry in images generated within same latent space in the original human face model and retrained cross-section model. Alternatively, transfer learning from the landscape images exhibits more natural results as shown in Fig. 15.

The stylegan3 model was trained using the hyperparameter settings recommended by the authors (Karras et al., 2021). A single Nvidia RTX 3080 GPU was used during training with the batch size set to 32, gamma to 2, and batch-gpu to 8. The batch-gpu parameter is used when a larger batch size is desired, but there is insufficient memory to support the batch size. Larger batch sizes are usually preferred when batch normalization is implemented in the neural network. During batch normalization, the results are normalized over the entire batch and used to update the network weights. Therefore, the batch size becomes a hyperparameter that can be tuned. In the present case, a synthetic batch size of 32 is obtained by normalizing four iterations of 8 batches before the network weights are updated. While this may increase training time, it allows for consistent results among various hardware configurations.

The stylegan3 model was trained until Fréchet Inception Distance (FID) converged to a minimum (Heusel et al., 2017). The FID is a metric used to score the similarity of real images to the generated ones. A low score indicates a higher similarity. The FID reported at the end of training was 14.2 at the end of 800 epochs over a period of 6 days. FID scores reported

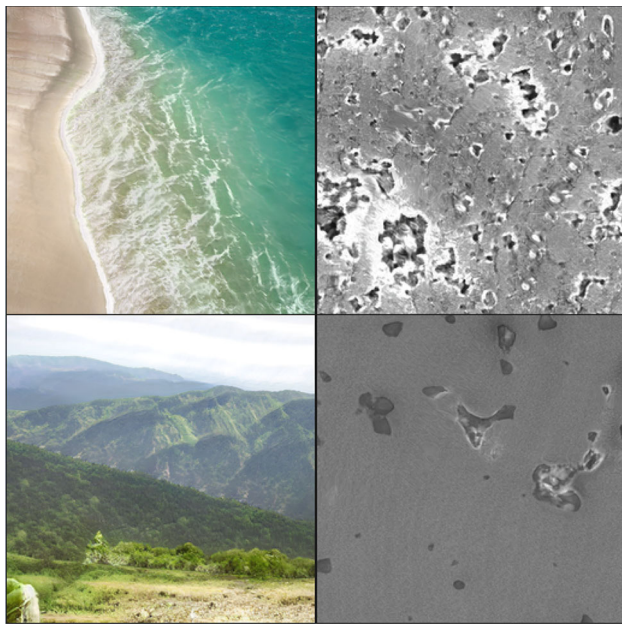


Fig. 15 Transfer learning using stylegan3 network pretrained on landscape images retain no obvious features within the same latent space

for stylegan3 range from 3–19 depending on the number of training images with lower FID scores corresponding to very large image sets and high FID score corresponding to small image sets. For the small image set implemented in this study, an FID result of 14 is reasonable (Karras et al., 2021). Generated images with their real image counterparts are displayed in Fig. 16.

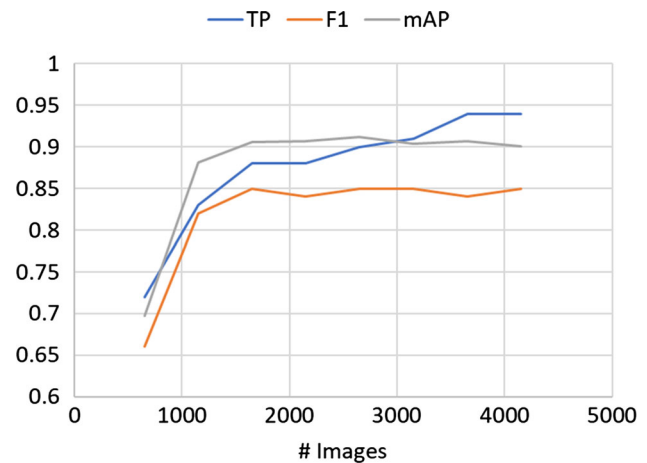


Fig. 17 True positive (TP), F1, and mean average precision (mAP) validation metrics vs the number of images in the training set

The number of augmented images was determined using the YOLO5n6 model due to its short training time compared to the other models. The segmented dataset was incrementally increased by adding 500 augmented images each iteration. The results, shown in Fig. 17, indicate that no substantial increases are observed after 3653 images are used for training the YOLO5n6 model. Therefore, this dataset size was utilized.

It is noted that the testing image set is isolated throughout the whole process as shown in Fig. 18. Among the 804 segmented images, 641 is used for training the models and 163 is used for testing the model performances. The StyleGan3 is applied to the 641 training images and generated 3102

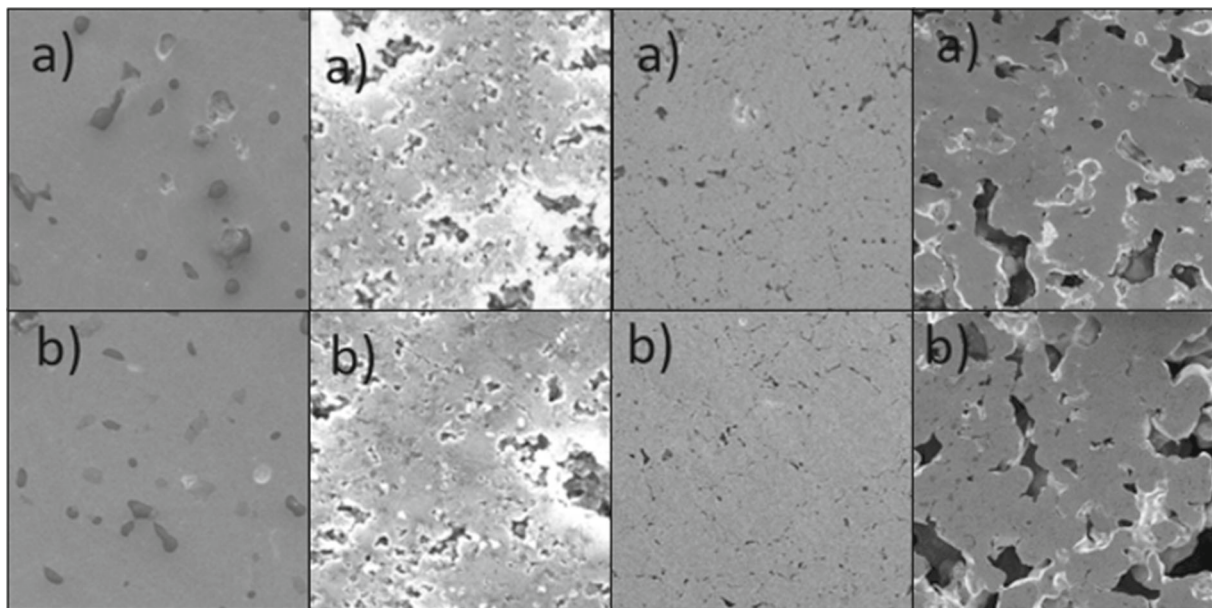
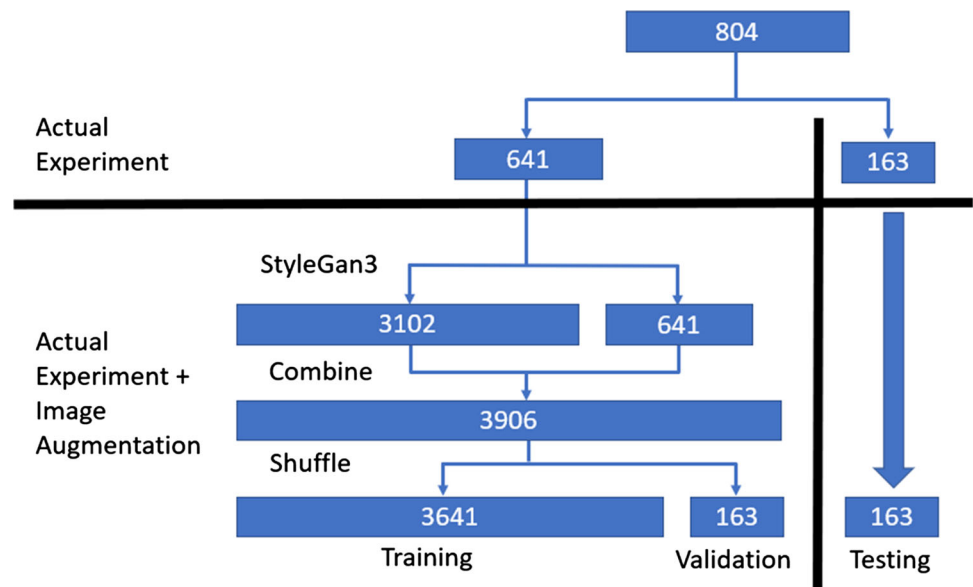


Fig. 16 Image comparison between **a** Stylegan3 generated images and **b** real images

Fig. 18 Dataset generation flowchart**Table 6** Datasets created for model training

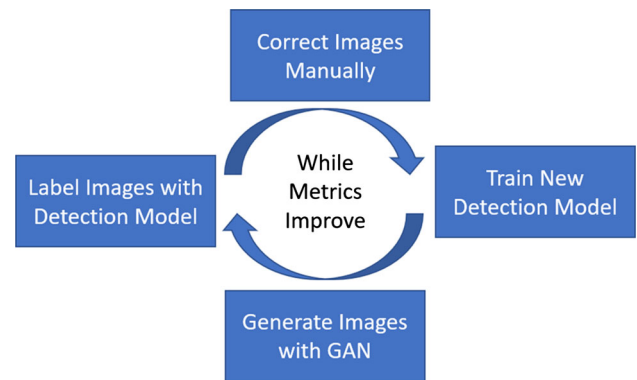
Dataset	Total images	Ave pores/Img
Original sparse	58	34
Original full (sparse + dense)	67	68
Segmented	804	6
Augmented	3102	7
Final full (segmented + augmented)	3906 ^a	7

^a3641 is used for training, 163 for validation, and 163 for testing.

augmented images. Those 3102 augmented images are combined with the 641 real training images, and the combined images are divided into two groups after random shuffle: (1) the final training dataset containing 3641 images and (2) the validation dataset containing 163 images. Table 6 shows the total number of images and their average porosities for each of the datasets defined in this study.

Image Labeling

An iterative technique was implemented for labeling the many images generated from the stylegan3 model. The original 67 images containing 4525 pores can be manually labeled over a couple of days, but the final full dataset with 27,294 pores would take weeks to label by hand. To reduce the manual labor, the YOLOv5 model trained on the segmented dataset was used to produce the initial bounding boxes for 500 images. These bounding boxes were then manually updated for accuracy, reducing the labeling time from days to hours. With each training iteration, the model improves

**Fig. 19** Semi-supervised training with images generated from GAN

incrementally, and less manual correction is required. Since the detection model provides the initial labels to the new dataset entries, the technique is semi-supervised. The basic flowchart for this process is displayed in Fig. 19.

Labeled pore properties

The final size distribution of the labeled pores can influence the performance of each algorithm and can be used to troubleshoot poor performance in testing. The histograms for the pore width, height, and aspect ratio are provided in Figs. 20, 21, and 22, respectively. The maximum pore bounding box width was found to be 229 pixels, the maximum height was 175 pixels, and the minimum width and height is 5 and 4 pixels, respectively. However, with image augmentation, the image is randomly scaled between $\pm 30\%$, so the peak in the training images is effectively broadened by 30%.

Fig. 20 Histogram of pore bounding box width

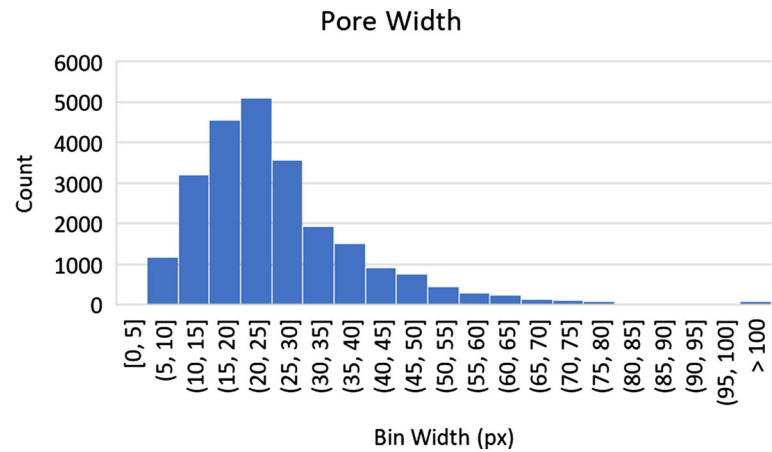


Fig. 21 Histogram of pore bounding box height

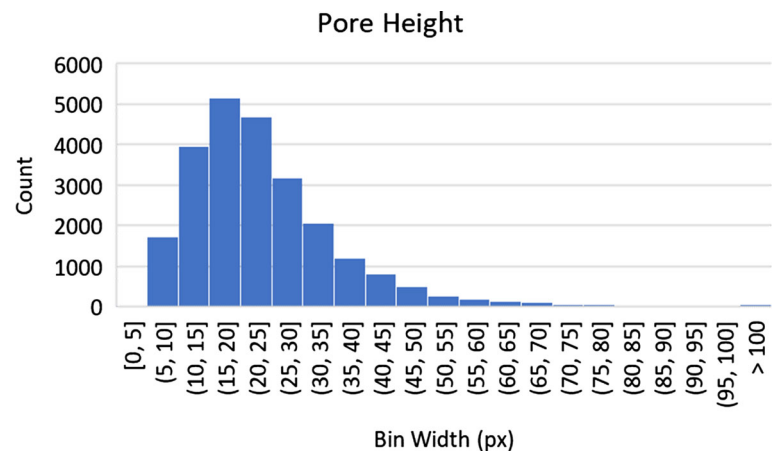
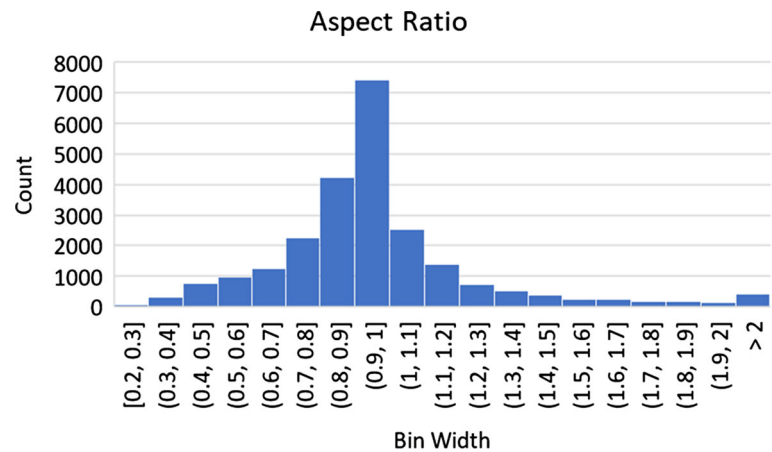


Fig. 22 Histogram of pore bounding box aspect ratio



Porosity characterization

Once bounding boxes are placed over the porous locations, a mask was placed over those pores to obtain morphological information. To obtain the mask, the localized pore is cropped out of the full image and passed to an edge detection algorithm (Dollár & Zitnick, 2015), the background mask is then obtained by applying the OpenCV floodfill function (Community, 2010). Once the pore mask is obtained, the contour

of the mask was used to compute the morphological information that are the contour area, the diameter of fitted circle, the area, angle, and aspect ratio of fitted ellipse, and curvature. All morphological properties except curvature were obtained using the built-in OpenCV functions (Community, 2010). The curvature, k , was computed from the raw contour points (around the edge of the pore) using the following

Eq. 11.

$$k = \frac{|x'y'' - y'x''|}{(x'^2 + y'^2)^{\frac{3}{2}}}, \quad (11)$$

where x and y are the interpolated parametric functions from the perimeter pixels. The curvature at each pixel is estimated and used to compute the average curvature. Smaller curvature values indicate a straighter line and may denote elongated pores or cracks. It is anticipated that curvature will be a useful feature in differentiating pores and cracks in the future along with the other features computed.

Results

Model comparison results

Each of the models were initially trained with the original sparse dataset. The resulting training time, detection time, and F1 score was used to determine which algorithms would be selected for training on the final full datasets. 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.

The CNN method with custom region proposal resulted in an F1 score of 71% on the sparse dataset. However, the region proposal method implemented was found to be time consuming, requiring about 10 s to process an image, making it the lowest performer and longest classifier. For these reasons, the CNN model was eliminated as a candidate.

The Faster R-CNN model was selected by considering the training time required and the resulting accuracy for the sparse dataset. The testing performed with ResNet18 resulted in an F1 score of 74%. The total training time for these results was about 2 h. The ResNet50 network was then trained resulting in an F1 score of 78%. Training time for the ResNet50 was approximately 12 h. For the ResNet101 model the recall MSE converged to F1 score of 84% with a training time of approximately 30 h. Image classification time for the Faster R-CNN ranged from 0.5 s for the ResNet18 to 3 s for the ResNet101. Shorter classification times could be obtained by eliminating bounding box fusion, but at the cost of accuracy.

On the original full dataset, the Faster R-CNN model performed substantially worse, producing an F1 score of 68% with the ResNet101 model. On the segmented dataset, the ResNet101 required 6 days of training and produced an F1 score of 64%. The model displayed very high precision, but low recall. The decrease in performance is likely due to the 224×224 window used to localize and classify objects

Table 7 TP, F1 score, and mAP for each of the YOLOv5

Model	TP	F1	mAP
YOLOv5n6	0.94	0.84	0.91
YOLOv5n6_ioma_TF	0.94	0.87	0.92
YOLOv5s6	0.95	0.88	0.93
YOLOv5s6_ioma_TF	0.95	0.87	0.92
YOLOv5m6	0.94	0.87	0.93
YOLOv5l6	0.91	0.87	0.94
YOLOv5x6	0.94	0.85	0.92

within the image. For the original sparse dataset, there are no very large pores present unlike the other datasets. Since a large pore may exceed the size of the sliding window it will not be classified. A potential solution would be to resize the images and classify them multiple times but would further increase the detection time. Another solution would be to increase the 224×224 layer size, but this also increases detection and training time and increases the GPU memory requirements. On the final full dataset (segmented + augmented), a 17% performance gain to a F1 score of 75% is observed, indicating the stylegan3 augmentation can substantially increase performance.

In comparison to the Faster R-CNN model, the YOLOv5 model performed consistently across the original sparse, original full, and segmented datasets. The final model was selected by running each on the final full dataset, as presented in Table 7. The updated loss function was found to provide similar performance gains for the YOLOv5n6 model in the final full dataset compared to the scoping exercise. However, YOLOv5s6 with updated loss did not display any performance enhancements. It is anticipated that further turning of the hyperparameter thresholds could increase the performance, however, due to the increased training time, additional tuning was not performed. It was found that models larger than YOLOv5n6 performed similarly. Therefore, the YOLOv5s6 model is selected since it benefits from faster detection compared to the other models. It is noted however, that the larger models may be preferable if they are trained on multiple classes (e.g. pores, cracks, precipitates).

The stylegan3 model was used to increase the image set and resulted in a substantial increase in the YOLOv5 detector performance. The percent increase of each metric from the original full dataset is displayed in Fig. 23. Smaller models tend to benefit more from the stylegan3 augmentation compared to the larger models. This may indicate that the augmented set contains similar images, and the larger models are overfitting due to similar data. Since smaller models are less prone to overfitting, this may partially explain why they benefit more from the large image set.

Fig. 23 TP, F1 score, and mAP performance increase of each YOLOv5 model due to dataset expansion with stylegan3

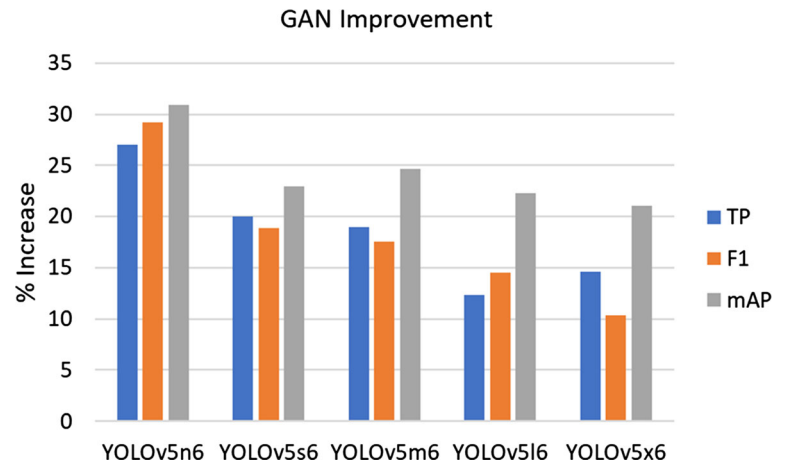


Table 8 F1 Score and classification time for each algorithm and dataset

Dataset	# Images	# Pores	Faster R-CNN	YOLOv5
Original Sparse	58	1960	84%	77%
Original Full	67	4525	68%	77%
Segmented	804	4525	64%	79%
Final Full	3966	27,294	75%	88%
Training Time	–	–	5–10 days	3–5 h
Detection time	–	–	0.5–3 s	3–15 ms

Table 8 compares the experimental results of the best Faster R-CNN and YOLOv5 models. Overall, the YOLOv5 model tends to outperform the other models except for in the sparse test, where the Faster R-CNN takes a distinct lead. This indicates that for the case of a small, similar image set, the Faster R-CNN may be the preferred model. The final YOLOv5 model was selected for testing and the results were consistent with the validation score, resulting in a F1 score of 88%.

Table 9 Average porosity morphological results

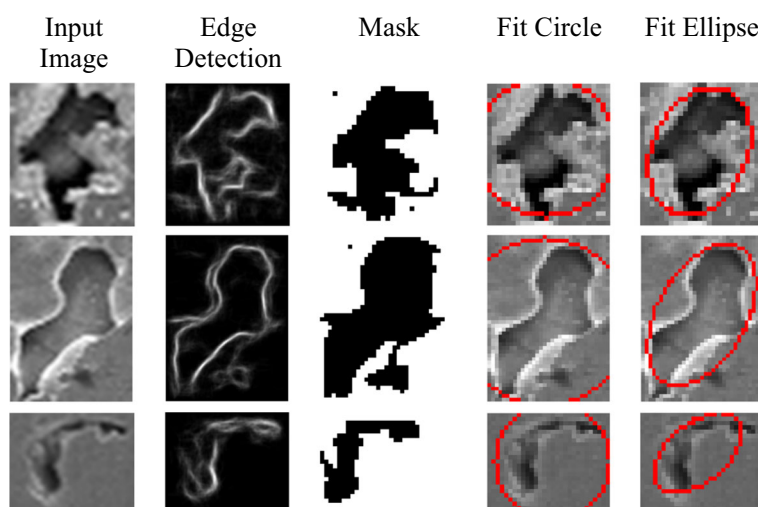
Parameter	Result
Average contour area	275 μm^2
Average diameter of equivalent circles	25 μm
Average area of bounding ellipses	537 μm^2
Average angle of bounding ellipses	89.8°
Average aspect ratio (AR) of bounding ellipses	1.2
Average curvature of contours	0.34

Porosity characterization results

After running the final model, we extracted the morphological information of the porosities by finding their edges, masks, best-fitted circles and ellipses. Examples of the edges, masks, fit circles, and fit ellipses are displayed in Fig. 24. The resulting morphological information obtained from the classified porosities is displayed in Table 9 (average) and Table 10 (distribution). This information is useful for characterizing SEM images and can be used for mapping process parameters to defects using ML. The characterization could also be used for future work on GANs inversion for SEM images (Xia et al., 2021). Specifically, an input image could be mapped to the GAN latent space and then modified based on user specified parameters such as those specified in Table 10.

Real world testing

To test our model as a general porosity detection tool, 15 SEM images containing porosities were obtained from various publications (An et al., 2016; Andreeva et al., 2016; Antunes et al., 2014; Baek et al., 2012; Chen et al., 2015; Chia, 2012; Cui et al., 2014; Da'as et al., 2013; Jamion et al., 2017; Y. Li et al., 2014; Ozols et al., 2004; Scheiba, 2021; Yao-jian et al., 2009). The images were selected to represent a large diversity of pore sizes, porosity, surface textures, intensities, resolutions, materials and magnifications. The pore information for each image is listed in Table 11, image sizes and scales are listed in Table 12, and image materials are shown in Fig. 25. The results, displayed in Fig. 25, indicates

Fig. 24 Steps for obtaining morphological information from four pores**Table 10** Distributions of porosity morphological results

Distribution of contour areas						
Area (μm ²)	0–250	250–500	500–750	750–1000	> 1000	
#	2339	165	99	65	224	
Distribution of diameters of equivalent circles						
Diameter (μm)	0–30	30–60	60–90	90–120	> 120	
#	2231	359	178	76	48	
Distribution of areas of bounding ellipses						
Area (μm ²)	0–620	620–1240	1240–1860	1860–2480	> 2480	
#	2309	251	131	62	139	
Distribution of angles of bounding ellipses						
Angle (°)	0–30	30–60	60–90	90–120	120–150	150–180
#	433	396	591	654	401	417
Distribution of ARs of bounding ellipses						
AR	0–0.75	0.75–1.5	1.5–2.25	2.25–3	> 3.75	
#	638	1585	492	111	43	
Distribution of curvature						
Curvature	0.15	0.3	0.45	0.6	> 0.75	
#	3	1031	1548	258	27	

that the detection model performs well on images of various sizes, various materials, and at various magnifications. Please note that the images were captured from publications are likely not the original resolution. It is also likely that compression has been applied.

Compared to manual bounding box labeling, the final model produced an F1 score of 85%, TP of 86%, and mAP

of 90%. The morphological results are found to be comparable to the F1 score with exception to the average pore area and average ellipse area. The reason for this is likely two-fold. Average sized pores that are missed may not significantly reduce the other average metrics, however, missing a large pore or mislabeling a large area incorrectly as a pore may significantly increase the area prediction. There is also the issue of overlapping boxes. Large areas may be labeled

Table 11 Average pore properties of each image extracted from ground source bounding boxes

Image	Porosity	Area	Eq. diameter	Ellipse area	Angle	AR	Perimeter	Curvature
A	0.47	975	32	954	74	1.09	114	0.09
B	0.36	2054	36	1427	93	1.15	152	0.09
C	0.05	499	23	531	94	1.14	88	0.12
D	0.20	465	22	458	92	0.90	80	0.11
E	0.20	3622	60	3827	95	1.28	241	0.05
F	0.17	308	16	267	85	1.17	59	0.17
G	0.40	250	17	299	83	1.16	64	0.19
H	0.30	7772	85	6464	97	1.00	313	0.04
I	0.20	1036	33	948	104	0.94	110	0.07
J	0.24	241	16	233	94	1.11	53	0.17
K	0.25	318	17	301	94	1.04	65	0.14
L	0.31	901	28	731	90	1.37	105	0.13
M	0.13	5733	58	2611	91	1.05	206	0.15
N	0.18	234	15	278	99	1.02	62	0.20
O	0.12	517	24	521	82	1.04	83	0.11
Max	0.47	7772	85	6464	104	1.37	313	0.20
Min	0.05	234	15	233	74	0.90	53	0.04
Mean	0.24	1662	32	1323	91	1.10	120	0.12

Note that units are in pixels (px) for more direct comparison

Table 12 Properties of each image found in literature

Image	Width (px)	Height (px)	Width (μm)	Intensity average	Intensity SD
A	564	383	3303	116	83
B	561	406	198	123	76
C	561	406	60	117	70
E	947	710	NA	108	36
F	532	320	55	135	43
G	499	390	310	134	44
L	500	500	8	124	63
H	761	672	298	85	42
J	409	308	251	117	41
I	865	621	1219	166	49
K	282	282	2	76	35
M	454	454	98	116	37
N	320	290	52	180	51
D	504	449	94	80	38
O	580	369	23	80	38

with overlapping boxes multiple times. For the average pore computations, the overlapping represents a source of error. However, for porosity assessment, a blank image the same resolution as the source image is produced and the mask for each pore is projected upon this image. Therefore, overlapping pores do not influence porosity assessment (Table 13).

Of the publication images, the model struggled most with images B and C. Image B contains multiple large oddly

shaped pores and is unable to concisely place a bounding box on the upper pore or fully bound the lower pore. Image C contains a highly textured surface and the model exhibits difficulty distinguishing areas of the rough surface from pores. The pores in image M were fully bounded, however, splitting up the large pore on the right resulted in significant deviations from the ground truth average pore properties.

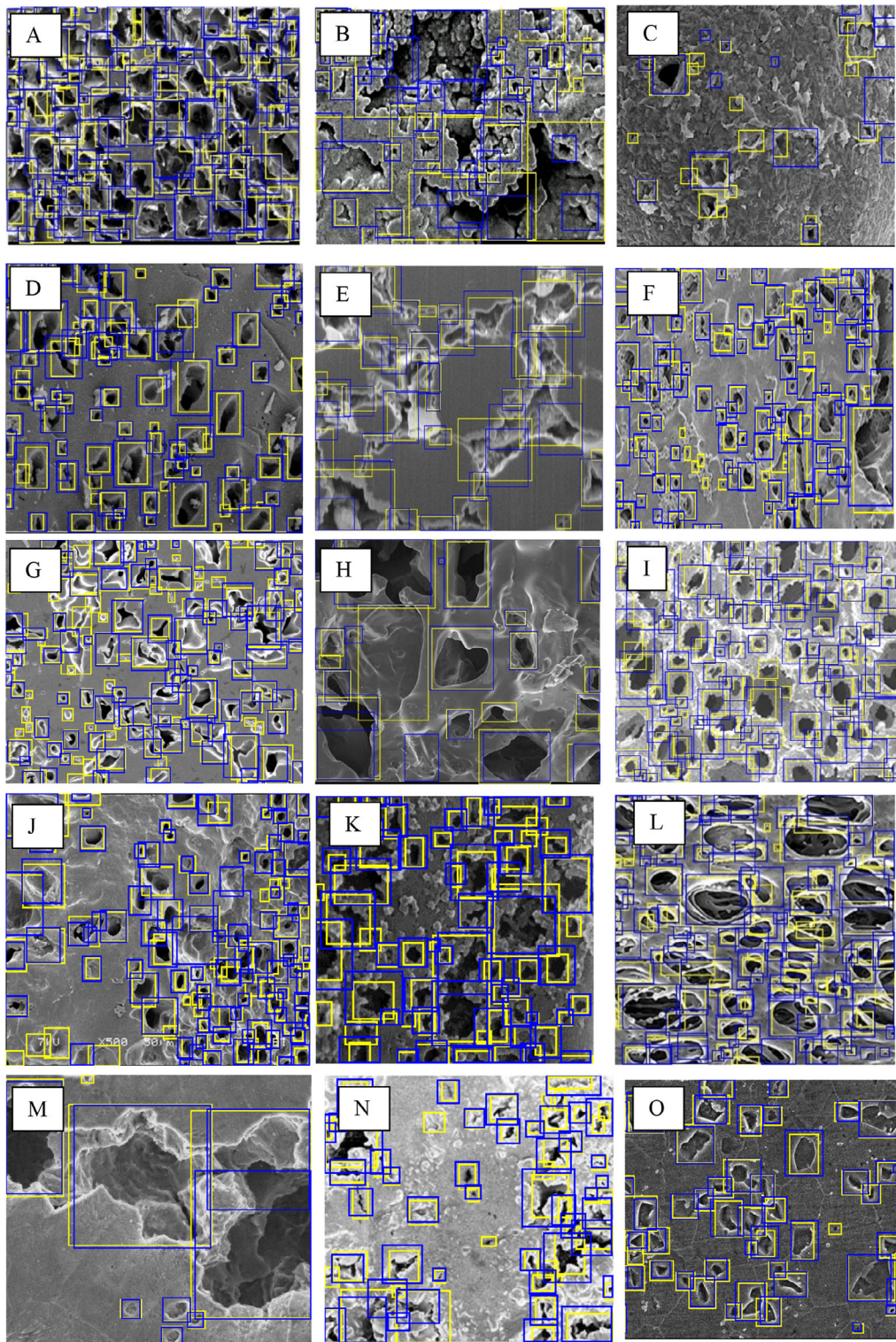


Fig. 25 Pore detection on porous images scraped from publications. Yellow denotes ground truth bounding boxes and blue are model predictions. From top left to bottom right the images are **A** porous composite scaffolds consisting of poly (3-hydroxybutyrate-co-3-hydroxyvalerate) and hydroxyapatite (Baek et al., 2012), **B** flattened magnetite (Antunes et al., 2014), **C** titania nanosheets (Cui et al., 2014), **D** activated carbon (Jamion et al., 2017), **E** aged graphite anode of Li-ion battery (Scheiba, 2021), **F** impact polypropylene copolymer (Chen et al., 2015), **G** 316 steel by gel cast (Ozols et al., 2004), **H** Polyethylene Glycol

Scaffold (Chia, 2012), **I** porous titanium fabricated by slurry foaming (Li et al., 2014), **J** Rice grain surface after pressure cooking, **K** Al_2O_3 film (Andreeva et al., 2016), **L** Polyethersulfone membrane filter (Fisher Scientific, 2022), **M** carbon anode (Dojcinovic et al., 2012), **N** 4-dinitrofurazanofuroxan and glycidyl azide polymer (An et al., 2016), **O** IN100 superalloy (Wusatowska-Sarnek et al., 2003). Note that images displayed here have been formatted and are not the original size or aspect ratio (Color figure online)

Table 13 Predicted percent error of average pore properties of each image

Image	Porosity (%)	Area (%)	Eq. diameter (%)	Ellipse area (%)	Angle (%)	AR (%)	Perimeter (%)	Curvature (%)
A	3	8	6	13	4	1	5	7
B	31	86	11	22	7	4	11	9
C	53	48	24	40	2	4	23	22
D	9	27	16	34	10	7	16	20
E	8	3	7	15	8	17	4	11
F	11	11	8	15	7	10	5	10
G	7	15	10	20	4	7	7	13
H	27	34	12	24	2	4	10	12
I	9	21	10	22	8	12	10	12
J	10	25	10	18	1	2	6	12
K	10	23	14	29	8	9	15	16
L	1	9	6	13	1	14	1	12
M	6	23	27	47	7	16	21	30
N	8	27	15	31	2	5	16	17
O	0	1	0	1	0	4	1	1
Mean	13	24	12	23	5	8	10	14

Conclusions

In this paper, the efficacy of CNNs for image-based detection of porosities in the metal parts produced by BJ followed by sintering is investigated. It was found that the typical CNN used for image classification performed poorly due to the region proposal algorithms. Based on this deficiency, the Faster R-CNN and YOLOv5 ML models were investigated for their ability to combine region proposal and classification. To support this investigation, a stylegan3 model was trained to expand the dataset from 67 images containing 4545 pores to 3966 images containing 27,294 pores. The YOLOv5 model produced an 88% F1 score on the expanded dataset (original + augmented images), a 15% increase over the 79% F1 score from the original dataset. A real-world test was performed with the final model on a set of cross-section SEM images containing various materials, porosities densities, and resolutions yielding a F1 score of 85%. The Faster R-CNN model was found to only perform best on the original sparse dataset with an F1 score of 84%, however its performance on the final full dataset was notably lower than the YOLOv5 model. Training time for the YOLOv5 model was also substantially less than the Faster R-CNN model, taking hours opposed to weeks on the larger dataset. Classification time was also orders of magnitude less for the YOLOv5 model requiring 3–15 ms opposed to 0.5–3 s on the Faster R-CNN model.

Based on the results, we demonstrated a novel use of the YOLOv5 network for the generalized image-based porosity detection in the BJ metal parts using a small yet varied training datasets, which can be applied to any materials, pore density, and varied image resolutions. To achieve these results on a small training set, classic image augmentation was applied to increase the initial dataset size which was then used to train a GAN network to further increase the dataset size. The proposed approach can be widely used for more accurate and rapid image-based porosity characterization and quality control in the BJ and sintering process, which in turn will contribute to the AM industry, especially for the scaling of the BJ and sintering process.

Despite the positive results, there remain many areas of improvement. It is anticipated that performance can be substantially increased by using a masking or polygon outlining algorithms opposed to bounding boxes to eliminate labeling ambiguity. However, this would likely increase detection time. It is also noted that the trained model originated from a small set of 67 images, and only so much information can be extracted from those images. Therefore, it is likely that performance could be increased with a larger initial image set. Other tools could also be used to increase the detection performance such employing an upscaling GAN to synthesize many image magnifications from a single lower magnification image.

Acknowledgements The support of the National Science Foundation, under Award DMR-2119832 is gratefully acknowledged.

Funding Funding was provided by National Science Foundation (Grant Number 2119832).

Data availability The data that support the findings of this study are openly available in MendeleyData at <https://data.mendeley.com/datasets/vhxs23krz4/draft?a=37f6a434-bd7e-4895-8995-e32547f42421> [<https://doi.org/10.17632/vhxs23krz4.1>].

References

- Abdelrahman, M., Reutzel, E. W., Nassar, A. R., & Starr, T. L. (2017). Flaw detection in powder bed fusion using optical imaging. *Additive Manufacturing*, 15, 1–11. <https://doi.org/10.1016/j.addma.2017.02.001>
- Aminzadeh, M., & Kurfess, T. R. (2019). Online quality inspection using Bayesian classification in powder-bed additive manufacturing from high-resolution visual camera images. *Journal of Intelligent Manufacturing*, 30(6), 2505–2523. <https://doi.org/10.1007/s10845-018-1412-0>
- An, C., Wen, X., Wang, J., & Wu, B. (2016). GAP/DNTF based PBX explosives: A novel formula used in small sized explosive circuits. *Central European Journal of Energetic Materials*. <https://doi.org/10.22211/cejem/64992>
- Andreeva, R., Stoyanova, E., Tsanev, A., & Stoychev, D. (2016). Corrosion behavior of anodically formed oxide films on aluminum, sealed in cerium-ions containing solutions. *Bulgarian Chemical Communications*, 48, 96–102.
- Antunes, R. A., Ichikawa, R. U., Martinez, L. G., & Costa, I. (2014). Characterization of corrosion products on carbon steel exposed to natural weathering and to accelerated corrosion tests. *International Journal of Corrosion*. <https://doi.org/10.1155/2014/419570>
- ASTM. (2012). *F2792-12a, Standard Terminology for Additive Manufacturing Technologies*, (Withdrawn 2015). ASTM International.
- Baek, J. Y., Xing, Z. C., Kwak, G., Yoon, K. B., Park, S. Y., Park, L. S., & Kang, I. K. (2012). Fabrication and characterization of collagen-immobilized porous PHBV/HA nanocomposite scaffolds for bone tissue engineering. *Journal of Nanomaterials*. <https://doi.org/10.1155/2012/171804>
- Bai, Y., & Williams, C. B. (2015). An exploration of binder jetting of copper. *Rapid Prototyping Journal*. <https://doi.org/10.1108/RPJ-12-2014-0180>
- Bergmann, C., Lindner, M., Zhang, W., Koczur, K., Kirsten, A., Telle, R., & Fischer, H. (2010). 3D printing of bone substitute implants using calcium phosphate and bioactive glasses. *Journal of the European Ceramic Society*. <https://doi.org/10.1016/j.jeurceramsoc.2010.04.037>
- Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: optimal speed and accuracy of object detection. arXiv Preprint. <https://doi.org/10.48550/ARXIV.2004.10934>
- Chen, F., Qiu, B., Ye, Y., Lv, Y., Shangguan, Y., Song, Y., & Zheng, Q. (2015). Reconstruction of core-shell dispersed particles in impact polypropylene copolymer during extrusion. *Chinese Journal of Polymer Science (English Edition)*. <https://doi.org/10.1007/s10118-015-1616-0>
- Chen, H., & Zhao, Y. F. (2016). Process parameters optimization for improving surface quality and manufacturing accuracy of binder jetting additive manufacturing process. *Rapid Prototyping Journal*, 22(3), 527–538. <https://doi.org/10.1108/RPJ-11-2014-0149>
- Cheng, J. C. P., & Wang, M. (2018). Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques. *Automation in Construction*. <https://doi.org/10.1016/j.autcon.2018.08.006>
- Chia, H. (2012). Southern California Society for microscopy and microanalysis. Retrieved June 16, 2022, from <http://www.scsmm.org/gallery.html>
- Community, O. (2010). *The OpenCV reference manual*. October.
- Cui, X., Shi, J., Ye, Z., Zhang, Z., Xu, B., & Chen, S. (2014). Layer-by-layer assembly and photocatalytic activity of titania nanosheets on coal fly ash microspheres. *International Journal of Photoenergy*. <https://doi.org/10.1155/2014/823078>
- Da'as, E. H., Irvine, J. T. S., Traversa, E., & Boulfrad, S. (2013). Controllable impregnation via Inkjet printing for the fabrication of solid oxide cell air electrodes. *ECS Transactions*. <https://doi.org/10.1149/05701.1851ecst>
- Do, T., Kwon, P., & Shin, C. S. (2017). Process development toward full-density stainless steel parts with binder jetting printing. *International Journal of Machine Tools and Manufacture*, 121, 50–60. <https://doi.org/10.1016/j.ijmachtools.2017.04.006>
- Do, T., Shin, C. S., Stetsko, D., Vanconant, G., Vartanian, A., Pei, S., & Kwon, P. (2015). Improving structural integrity with boron-based additives for 3D printed 420 stainless steel. *Procedia Manufacturing*. <https://doi.org/10.1016/j.promfg.2015.09.019>
- Dojcinovic, M., Eric, O., Rajnovic, D., Sidani, L., & Balos, S. (2012). The morphology of ductile cast iron surface damaged by cavitation. *Metallurgical & Materials Engineering*, 18(3), 165–176.
- Dollár, P., & Zitnick, C. L. (2015). Fast edge detection using structured forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. <https://doi.org/10.1109/TPAMI.2014.2377715>
- Doyle, M., Agarwal, K., Sealy, W., & Schull, K. (2015). Effect of layer thickness and orientation on mechanical behavior of binder jet stainless steel 420 + bronze parts. *Procedia Manufacturing*. <https://doi.org/10.1016/j.promfg.2015.09.016>
- Enneti, R. K., Prough, K. C., Wolfe, T. A., Klein, A., Studley, N., & Trasorras, J. L. (2018). Sintering of WC-12%Co processed by binder jet 3D printing (BJ3DP) technology. *International Journal of Refractory Metals and Hard Materials*. <https://doi.org/10.1016/j.jrmhm.2017.10.023>
- Fisher Scientific. (2022). MilliporeSigma™ Express™ PLUS membrane filters. Fisher Scientific Product Catalog. Retrieved November 20, 2022, from <https://www.fishersci.com/shop/products/emd-millipore-express-plus-membrane-filters-10/p-3303398>
- García-Moreno, A. I., Alvarado-Orozco, J. M., Ibarra-Medina, J., & Martínez-Franco, E. (2020). Image-based porosity classification in Al-alloys by laser metal deposition using random forests. *International Journal of Advanced Manufacturing Technology*. <https://doi.org/10.1007/s00170-020-05887-6>
- Gaytan, S. M., Cadena, M. A., Karim, H., Delfin, D., Lin, Y., Espalin, D., MacDonald, E., & Wicker, R. B. (2015). Fabrication of barium titanate by binder jetting additive manufacturing technology. *Ceramics International*. <https://doi.org/10.1016/j.ceramint.2015.01.108>
- Gibson, I., Rosen, D., & Stucker, B. (2015). *Additive manufacturing technologies: 3D printing and direct digital manufacturing*. Springer.
- Gobert, C., Reutzel, E. W., Petrich, J., Nassar, A. R., & Phoha, S. (2018). Application of supervised machine learning for defect detection during metallic powder bed fusion additive manufacturing using high resolution imaging. *Additive Manufacturing*, 21, 517–528. <https://doi.org/10.1016/j.addma.2018.04.005>
- Gonzalez, J. A., Mireles, J., Lin, Y., & Wicker, R. B. (2016). Characterization of ceramic components fabricated using binder jetting additive manufacturing technology. *Ceramics International*. <https://doi.org/10.1016/j.ceramint.2016.03.079>
- 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* (Vol. 2016–December). <https://doi.org/10.1109/CVPR.2016.90>

- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in neural information processing systems* (Vol. 2017–December).
- Hong, D., Chou, D. T., Velikokhatnyi, O. I., Roy, A., Lee, B., Swink, I., Issaev, I., Kuhn, H., & Kumta, P. N. (2016). Binder-jetting 3D printing and alloy development of new biodegradable Fe–Mn–Ca/Mg alloys. *Acta Biomaterialia*. <https://doi.org/10.1016/j.actbio.2016.08.032>
- Huang, H., Zhou, H., Yang, X., Zhang, L., Qi, L., & Zang, A. Y. (2019). Faster R-CNN for marine organisms detection and recognition using data augmentation. *Neurocomputing*. <https://doi.org/10.1016/j.neucom.2019.01.084>
- Hussain, M., Bird, J. J., & Faria, D. R. (2019). A study on CNN transfer learning for image classification. *Advances in Intelligent Systems and Computing*. https://doi.org/10.1007/978-3-319-97982-3_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. *Journal of Manufacturing Science and Engineering, Transactions of the ASME*. <https://doi.org/10.1115/1.4040615>
- Islam, M. N., & Sacks, S. (2016). An experimental investigation into the dimensional error of powder-binder three-dimensional printing. *International Journal of Advanced Manufacturing Technology*. <https://doi.org/10.1007/s00170-015-7482-7>
- Jamion, N. A. B., Haffif, N. H. B. A., Halim, N. H. A., Izzaddin, S. A., Ghazali, S. M., & Sapari, J. M. (2017). Preparation of date seed activation for surfactant recovery. *Malaysian Journal of Analytical Science*. <https://doi.org/10.17576/mjas-2017-2105-06>
- Jocher, G., Stoken, A., Borovec, J., Christopher, S., Changyu, L., Hogan, A., Laughing, A., Diaconu, L., Poznanski, J., Yu, L., Rai, R., Ferri, R., Sullivan, T., Claramunt, E. R. (2020). ultralytics/yolov5: v2.0. *GitHub*.
- Jocher, G., Stoken, A., Borovec, J., Chaurasia, A., Changyu, L., Laughing, A., Hogan, A., Hajek, J., Diaconu, L., Marc, Y. (2021). ultralytics/yolov5: v5.0—YOLOv5-P6 1280 models, AWS, Super-verse.ly and YouTube integrations. *Zenodo*.
- Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., & Aila, T. (2021). Alias-free generative adversarial networks. *arXiv Preprint*. <https://doi.org/10.48550/ARXIV.2106.12423>
- Lanzetta, M., & Sachs, E. M. (2001). The line formation with alumina powders in drop on demand three dimensional printing. In *Proceedings of the 1st international seminar on: Progress in innovative manufacturing engineering (PRIME 2001)*.
- Lee, G., Carrillo, M., McKittrick, J., Martin, D. G., & Olevsky, E. A. (2020). Fabrication of ceramic bone scaffolds by solvent jetting 3D printing and sintering: Towards load-bearing applications. *Additive Manufacturing*. <https://doi.org/10.1016/j.addma.2020.101107>
- Lei, H. W., Wang, B., Wu, H. H., & Wang, A. H. (2018). Defect detection for polymeric polarizer based on faster R-CNN. *Journal of Information Hiding and Multimedia Signal Processing*, 9(6), 1414–1420.
- Li, S., Gu, X., Xu, X., Xu, D., Zhang, T., Liu, Z., & Dong, Q. (2021). Detection of concealed cracks from ground penetrating radar images based on deep learning algorithm. *Construction and Building Materials*. <https://doi.org/10.1016/j.conbuildmat.2020.121949>
- Li, Y., Yang, C., Zhao, H., Qu, S., Li, X., & Li, Y. (2014). New developments of ti-based alloys for biomedical applications. *Materials*. <https://doi.org/10.3390/ma7031709>
- Liu, Y., Cen, C., Che, Y., Ke, R., Ma, Y., & Ma, Y. (2020). Detection of maize tassels from UAV RGB imagery with faster R-CNN. *Remote Sensing*. <https://doi.org/10.3390/rs12020338>
- Manière, C., & Olevsky, E. A. (2017). Porosity dependence of powder compaction constitutive parameters: Determination based on spark plasma sintering tests. *Scripta Materialia*. <https://doi.org/10.1016/j.scriptamat.2017.07.026>
- Milewski, J. O. (2017). Additive manufacturing metal, the art of the possible. In *Springer series in materials science*. Springer. https://doi.org/10.1007/978-3-319-58205-4_2
- Miyajima, H., Zhang, S., Lassell, A., Zandinejad, A., & Yang, L. (2016). Process development of porcelain ceramic material with binder jetting process for dental applications. *JOM Journal of the Minerals Metals and Materials Society*. <https://doi.org/10.1007/s11837-015-1771-3>
- Moon, J., Caballero, A. C., Hozer, L., Chiang, Y. M., & Cima, M. J. (2001). Fabrication of functionally graded reaction infiltrated SiC–Si composite by three-dimensional printing (3DPTM) process. *Materials Science and Engineering A*. [https://doi.org/10.1016/S0921-5093\(00\)01282-x](https://doi.org/10.1016/S0921-5093(00)01282-x)
- Mostafaei, A., Stevens, E. L., Hughes, E. T., Biery, S. D., Hilla, C., & Chmielus, M. (2016). Powder bed binder jet printed alloy 625: Densification, microstructure and mechanical properties. *Materials and Design*. <https://doi.org/10.1016/j.matdes.2016.06.067>
- Olevsky, E. A. (1998). Theory of sintering: From discrete to continuum. *Materials Science and Engineering R: Reports*, 23(2), 41–100. [https://doi.org/10.1016/S0927-796X\(98\)00009-6](https://doi.org/10.1016/S0927-796X(98)00009-6)
- Ozols, A., Thern, G., Rozenberg, S., Barreiro, M., & Marajofsky, A. (2004). Moldeo de Acero AISI 316 por Colada Gel. *CONGRESO CONAMET/SAM*.
- Pinkney, J. (2022, June 12). Awesome pretrained StyleGAN3. *GitHub*. Retrieved June 14, 2022, from <https://github.com/justinpinkney/awesome-pretrained-stylegan3>
- Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. <https://doi.org/10.1109/TPAMI.2016.2577031>
- Sachs, E., Cima, M., & Cornie, J. (1990). Three-dimensional printing: Rapid tooling and prototypes directly from a CAD model. *CIRP Annals—Manufacturing Technology*. [https://doi.org/10.1016/S0007-8506\(07\)61035-X](https://doi.org/10.1016/S0007-8506(07)61035-X)
- Sachs, E., Cima, M., Williams, P., Brancazio, D., & Cornie, J. (1992). Three dimensional printing: Rapid tooling and prototypes directly from a CAD model. *Journal of Manufacturing Science and Engineering, Transactions of the ASME*. <https://doi.org/10.1115/1.2900701>
- Sasaki, Y. (2007). *The truth of the F-measure*. Teach Tutor mater.
- Satterlee, N., Torresani, E., Olevsky, E., & Kang, J. S. (2022). Comparison of machine learning methods for automatic classification of porosities in powder-based additive manufactured metal parts. *The International Journal of Advanced Manufacturing Technology*, 120(9), 6761–6776. <https://doi.org/10.1007/s00170-022-09141-z>
- Scheiba, F. (2021). *Carbon based materials for electrochemical energy storage*. Institute for Applied Materials Energy Storage Systems. Retrieved June 16, 2022, from <https://www.iam.kit.edu/english/667.php>
- Scime, L., & Beuth, J. (2018). A multi-scale convolutional neural network for autonomous anomaly detection and classification in a laser powder bed fusion additive manufacturing process. *Additive Manufacturing*, 24, 273–286. <https://doi.org/10.1016/j.addma.2018.09.034>
- Shrestha, S., & Manogharan, G. (2017). Optimization of binder jetting using Taguchi method. *JOM Journal of the Minerals Metals and Materials Society*, 69(3), 491–497. <https://doi.org/10.1007/s11837-016-2231-4>
- Snelling, D., Li, Q., Meisel, N., Williams, C. B., Batra, R. C., & Druschitz, A. P. (2015). Lightweight metal cellular structures fabricated via 3D printing of sand cast molds. *Advanced Engineering Materials*. <https://doi.org/10.1002/adem.201400524>

- Srinivas, M., & Babu, B. S. (2017). A critical review on recent research methodologies in additive manufacturing. *Materials Today: Proceedings*. <https://doi.org/10.1016/j.matpr.2017.07.258>
- Sun, X., Gu, J., Huang, R., Zou, R., & Palomares, B. G. (2019). Surface defects recognition of wheel hub based on improved faster R-CNN. *Electronics (Switzerland)*. <https://doi.org/10.3390/electronics8050481>
- Tang, Y., Zhou, Y., Hoff, T., Garon, M., & Zhao, Y. F. (2016). Elastic modulus of 316 stainless steel lattice structure fabricated via binder jetting process. *Materials Science and Technology (United Kingdom)*. <https://doi.org/10.1179/1743284715Y.0000000084>
- Wang, Y., & Zhao, Y. F. (2017). Investigation of sintering shrinkage in binder jetting additive manufacturing process. *Procedia Manufacturing*. <https://doi.org/10.1016/j.promfg.2017.07.077>
- Winkel, A., Meszaros, R., Reinsch, S., Müller, R., Travitzky, N., Fey, T., Greil, P., & Wondraczek, L. (2012). Sintering of 3D-printed glass/HAp composites. *Journal of the American Ceramic Society*. <https://doi.org/10.1111/j.1551-2916.2012.05368.x>
- Wu, W., Yin, Y., Wang, X., & Xu, D. (2019). Face detection with different scales based on faster R-CNN. *IEEE Transactions on Cybernetics*. <https://doi.org/10.1109/TCYB.2018.2859482>
- Wusatowska-Sarnek, A. M., Ghosh, G., Olson, G. B., Blackburn, M. J., & Aindow, M. (2003). Characterization of the microstructure and phase equilibria calculations for the powder metallurgy superalloy IN100. *Journal of Materials Research*. <https://doi.org/10.1557/JMR.2003.0371>
- Xia, W., Zhang, Y., Yang, Y., Xue, J.-H., Zhou, B., & Yang, M.-H. (2021). GAN inversion: A survey. arXiv Preprint. <https://doi.org/10.48550/ARXIV.2101.05278>
- Yao, J., Qi, J., Zhang, J., Shao, H., Yang, J., & Li, X. (2021). A real-time detection algorithm for kiwifruit defects based on yolov5. *Electronics (Switzerland)*. <https://doi.org/10.3390/electronics10141711>
- Yao-jian, R., Zhi, S., Jie, T., & Wei, L. (2009). Study on electrolytic aluminium carbon anode preparation with calcined anthracite. In *Procedia Earth and planetary science* (Vol. 1). <https://doi.org/10.1016/j.proeps.2009.09.109>
- Yoo, J., Cima, M., Sachs, E., & Suresh, S. (2008). Fabrication and microstructural control of advanced ceramic components by three dimensional printing. In *Proceedings of the 19th annual conference on composites, advanced ceramics, materials, and structures—B: Ceramic engineering and science proceedings*. <https://doi.org/10.1002/9780470314784.ch17>
- Zhang, B., Liu, S., & Shin, Y. C. (2019). In-Process monitoring of porosity during laser additive manufacturing process. *Additive Manufacturing*. <https://doi.org/10.1016/j.addma.2019.05.030>
- Zhang, S., Miyajima, H., Yang, L., Zandinejad, A. A., Dilip, J. J. S., & 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; An additive manufacturing conference, SFF 2014*.
- 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 Physics and Technology*. <https://doi.org/10.1016/j.infrared.2020.103334>
- Zhao, J., Zhang, X., Yan, J., Qiu, X., Yao, X., Tian, Y., Zhu, Y., & Cao, W. (2021). A wheat spike detection method in UAV images based on improved YOLOv5. *Remote Sensing*, 13(16), 3095. <https://doi.org/10.3390/rs13163095>
- Zhou, Y., Tang, Y., Hoff, T., Garon, M., & Zhao, F. Y. (2015). The verification of the mechanical properties of binder jetting manufactured parts by instrumented indentation testing. *Procedia Manufacturing*. <https://doi.org/10.1016/j.promfg.2015.09.038>
- Zhu, X., Lyu, S., Wang, X., & Zhao, Q. (2021). TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. In *Proceedings of the IEEE international conference on computer vision* (Vol. 2021–October). <https://doi.org/10.1109/ICCVW54120.2021.00312>

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.