

Article

A Tool for Rapid Analysis Using Image Processing and Artificial Intelligence: Automated Interoperable Characterization Data of Metal Powder for Additive Manufacturing with SEM Case

Georgios Bakas ¹, Spyridon Dimitriadis ¹, Stavros Deligiannis ², Leonidas Gargalis ², Ioannis Skaltsas ¹, Kyriaki Bei ¹, Evangelia Karaxi ² and Elias P. Koumoulos ^{1,*}

¹ IRES—Innovation in Research & Engineering Solutions, Rue Koningin Astritlaan 59B, 1780 Wemmel, Belgium

² Conify, Lavrion Technological and Cultural Park (LTCP), Lavrion Ave. 1, 19500 Lavrion, Greece

* Correspondence: epk@innovation-res.eu

Abstract: A methodology for the automated analysis of metal powder scanning electron microscope (SEM) images towards material characterization is developed and presented. This software-based tool takes advantage of a combination of recent artificial intelligence advances (mask R-CNN), conventional image processing techniques, and SEM characterization domain knowledge to assess metal powder quality for additive manufacturing applications. SEM is being used for characterizing metal powder alloys, specifically by quantifying the diameter and number of spherical particles, which are key characteristics for assessing the quality of the analyzed powder. Usually, SEM images are manually analyzed using third-party analysis software, which can be time-consuming and often introduces user bias into the measurements. In addition, only a few non-statistically significant samples are taken into consideration for the material characterization. Thus, a method that can overcome the above challenges utilizing state-of-the-art instance segmentation models is introduced. The final proposed model achieved a total mask average precision (mAP50) 67.2 at an intersection over union of 0.5 and with prediction confidence threshold of 0.4. Finally, the predicted instance masks are further used to provide a statistical analysis that includes important metrics such as the particle size distinction.

Keywords: metal powder; scanning electron microscope; computer vision; instance segmentation; additive manufacturing; deep learning; convolutional neural network



Citation: Bakas, G.; Dimitriadis, S.; Deligiannis, S.; Gargalis, L.; Skaltsas, I.; Bei, K.; Karaxi, E.; Koumoulos, E.P. A Tool for Rapid Analysis Using Image Processing and Artificial Intelligence: Automated Interoperable Characterization Data of Metal Powder for Additive Manufacturing with SEM Case. *Metals* **2022**, *12*, 1816. <https://doi.org/10.3390/met12111816>

Academic Editor: Amir Mostafaei

Received: 18 September 2022

Accepted: 19 October 2022

Published: 26 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Metal powder physical characteristics, such as particle size distribution (PSD), shape/morphology, porosity, and their flowability/spreadability capabilities, are considered critical parameters for the additive manufacturing (AM) industry since the structural integrity of a built part relies almost entirely on these characteristics [1]. On that basis, emphasis has been given to metal powder characterization and the measurements and protocols which can provide vital information about printability. Especially for the morphological characterization of metal AM powder feedstock, ISO/ASTM 52907-19 “Additive manufacturing—Feedstock materials—Methods to characterize metal powders”, proposes employing a scanning electron microscope (SEM), as well as energy dispersive spectroscopy (EDS) for (semi)quantitative chemical.

1.1. Applications on Additive Manufacturing

Currently, most metal additive manufacturing (AM) processes rely upon the use of metal powders. Manufacturing metal parts via AM depends on consistent powder characteristics to ensure repeatable build results [1]. Characterization of powder feedstock is needed, and therefore, several analytical techniques are employed, including microscopic observation, spectroscopy and destructive/non-destructive evaluation, in order to

determine chemical composition, particle size, size distribution, morphology/shape, and internal porosity often caused during the gas atomization process.

1.2. Relevant Work—SotA

Previous studies have utilized different deep learning (DL) techniques for the analysis of surface roughness. Bhandari B. conducted a comparative study of the performance of state-of-the-art DL models used for real-time classification of surface roughness using machining sound and force data [2]. Furthermore, Bhandari Binayak and Park Gijun used the light and shade composition of the surface [3].

DeCost et al., have developed a method to cluster, compare, and analyze powder micrographs using SIFT-VLAD descriptors from computer vision [4]. The developed system classifies powder images into the correct material systems with high accuracy. Although this methodology achieves a high accuracy level, it uses static filters that lack generalization aspects.

Xin Zhou et al. [5], have introduced a machine learning method into 3D metal powder particles classification. Their method provides an efficient tool for accurate and automated powder characterization utilizing the features provided by the PointNet++ model [6] while also using twelve user-defined features from the X-ray computed tomography input data. Although their results show that the accuracy of the model reaches 93.8%, the specific strategy used for classification not only requires X-ray computed tomography to obtain three-dimensional volume data of metal powder particles. Ryan Cohn et al., proposed the state-of-the-art instance segmentation Mask R-CNN model to detect the masks of particles found within metal powder SEM images. Two separate models were trained to segment the individual powder particles and satellites in each image, respectively. Although their approach reaches high levels of precision and recall, the model is only trained to detect the mask of each particle without further classifying them into particle categories [7].

1.3. AI and Computer Vision

Computer vision (CV) is a combination of fields, such as information technology, mathematics, and image processing, which aims to develop methods for computers to process and detect visual information in videos or images. CV emerged in the late 1960s at universities that were pioneering artificial intelligence (AI), and the initial idea was to mimic the human visual system in order to endow robots with intelligent behavior. A variety of applications in the area of CV are closely related to everyday life examples, such as medical diagnostics, driverless vehicles, camera monitoring, or smart filters, to name a few. In recent years, deep learning technology has greatly enhanced CV systems' performance. It can be said that the most advanced CV applications are nearly inseparable from deep learning [8].

In this manner, convolutional neural networks (CNNs) are introduced. CNNs are a powerful family of neural networks designed to take into account the spatial characteristics of an image. While traditional neural networks discarded each image's spatial structure by flattening them into one-dimensional vectors, a CNN leverages our prior knowledge that nearby pixels are typically related to each other, building efficient models for learning from image data. CNN-based architectures are now ubiquitous in the field of CV and have become so dominant that hardly anyone today would develop a commercial application related to image recognition, object detection or semantic segmentation, without building off of this approach.

Modern CNNs owe their design to inspirations from biology, group theory, and experimental tinkering. CNNs tend to be computationally efficient while managing to achieve accurate models both because they require fewer parameters than fully connected neural network architectures and because convolutions are easy to parallelize across graphics processing unit (GPU) cores. As a result, practitioners often apply CNNs whenever possible, and increasingly, they have emerged as credible competitors even on tasks with a one-dimensional sequence structure.

1.4. Mask R-CNN and Previous Models

Region-based CNNs (R-CNNs) are a pioneering approach that applies deep models to object detection [9]. R-CNN models first select several proposed regions from an image (for example, anchor boxes are one type of selection method) and label their categories and bounding boxes. A CNN backbone is used to perform forward computation and extract features from each proposed area. Afterwards, the features of each proposed region (region of interest, RoI) are used to predict their categories and bounding boxes. An improved architecture of R-CNN model is the faster R-CNN model [10] which is illustrated in Figure 1.

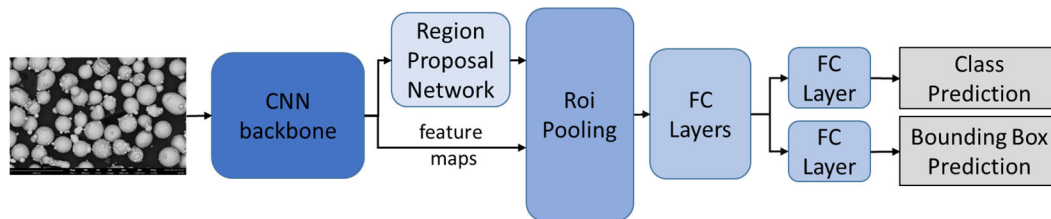


Figure 1. The faster R-CNN architecture; the information flows from left to right.

If training data are labelled with the pixel-level positions of each object in an image, a mask R-CNN [11] model can effectively use these detailed labels to further improve the precision of object detection. Mask R-CNN is a modification to the faster R-CNN model. It replaces the RoI pooling layer with an RoI alignment layer. This allows the use of bilinear interpolation to retain spatial information of feature maps, making it better suited for pixel-level predictions. The RoI alignment layer outputs feature maps of the same shape for all RoIs. This allows not only predictions of categories and bounding boxes of the proposed RoIs but also the use of the additional fully convolutional network to predict the pixel-level positions of objects. The full architecture of Mask R-CNN model is shown in Figure 2.

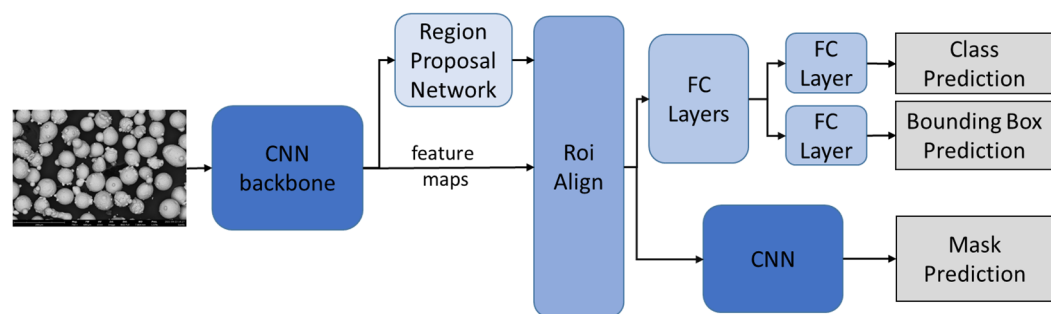


Figure 2. The mask R-CNN architecture; the information flows from left to right.

1.5. Semantic Segmentation

In this subsection, semantic segmentation is introduced, which attempts to segment images into regions with different semantic categories. These semantic regions label and predict objects at the pixel level.

In the computer vision field [12], there are two methods of significant importance related to semantic segmentation:

Image segmentation: Divides an image into several constituent regions. This method uses the correlations between pixels in an image. During training, labels are not needed for image pixels. However, this method cannot ensure that the segmented regions have the semantics an end user needs.

Instance segmentation: Also known as simultaneous detection and segmentation. This method tries to identify the pixel-level regions of each object instance in an image. In contrast to semantic segmentation, instance segmentation not only distinguishes semantics but also different object instances. For example, if an image contains two objects of a

desired class, instance segmentation will distinguish which pixels belong to each object of the aforementioned class.

1.6. Problem Description

In the scope of material science and specifically SEM images, it is crucial that the substructure of metal powders be further analyzed. Thus, the developed software aims to identify structures within SEM images as well as properties that can be derived from them. These structures are grouped into classes, such as particles with satellites or nodular, elongated, or spherical (circular) particles.

SEM provides images that reveal the structure of a material in micro-scale to users without any options for processing or analysis. Thus, software is necessary for further analysis of the aforementioned images. For further analysis a user is obligated to use a third-party software in order to calculate specific aspects taken from the image.

The common approach is to manually select structures from the image and further analyze them. Then, using third-party software, the user is able to extract information about the selected structures. This approach involves several disadvantages. The first major disadvantage is that it requires a large amount of time. The user must select a specific structure by looking every image separately, making this procedure long and tedious. Furthermore, this approach is characterized by bias as the user only selects structures that they believe are related to the respective analysis. As a result, several cases can be misjudged, and the output of an analysis can become inaccurate. In this manner, manual processing does not provide information about the whole image because the user selects only a handful of structures to analyze.

2. Materials and Methods

Depending on the AM process, the particle size distribution (PSD) of powder feedstock may differ, most commonly from 15 to 53 μm for PBF machines and between 45 to 90 μm for DED systems, respectively. In the present study, three AISI H13 tool steel powder samples were procured from Oerlikon, GKN Additive and Atomizing. Table 1 shows key characteristics for each powder according to providers' technical data sheets (TDSs). All powders had PSDs between 45 μm and 150 μm to be used in a directed energy deposition (DED) 3D printer. In order to be characterized, all powders were firstly subjected to sampling using a scoop to collect a representative sample from each batch container, according to ASTM B215-20: "Standard practices for sampling metal powders". After the sampling process, a thin layer of powder was deposited on an SEM sample holder for morphological, microstructural, and chemical analysis [13]. The electron microscope employed for the characterization process was the Phenom ProX Desktop SEM model by Thermofisher Scientific with integrated EDS. Figure 3a–c show micrographs for each powder sample. In each micrograph, nodular, satellited, non-spherical, or elongated particles can be observed that must be morphologically classified and statistically quantified in order to evaluate their printability performance.

Table 1. AM metal powder characteristics provided by producers' TDSs.

Brand Name	Chemical Composition (wt. %)							Production Process	PSD (Nominal Range)	Apparent Density	Morphology
	Fe	Cr	Mo	Si	V	Mn	C				
Oerlikon Metco H13	Balance	5.2	1.3	1.0	1.0	-	0.4	Gas Atomization	45–90 μm	>4.0 g/cm ³	Spheroidal
GKN Additive H13	Balance	5.2	1.3	0.8	1.2	0.3	0.4	Gas Atomization	50–150 μm	-	Spheroidal
Atomizing H13	Balance	5.2	1.3	0.8	1.2	0.3	0.4	Gas Atomization	45–90 μm	-	Spheroidal

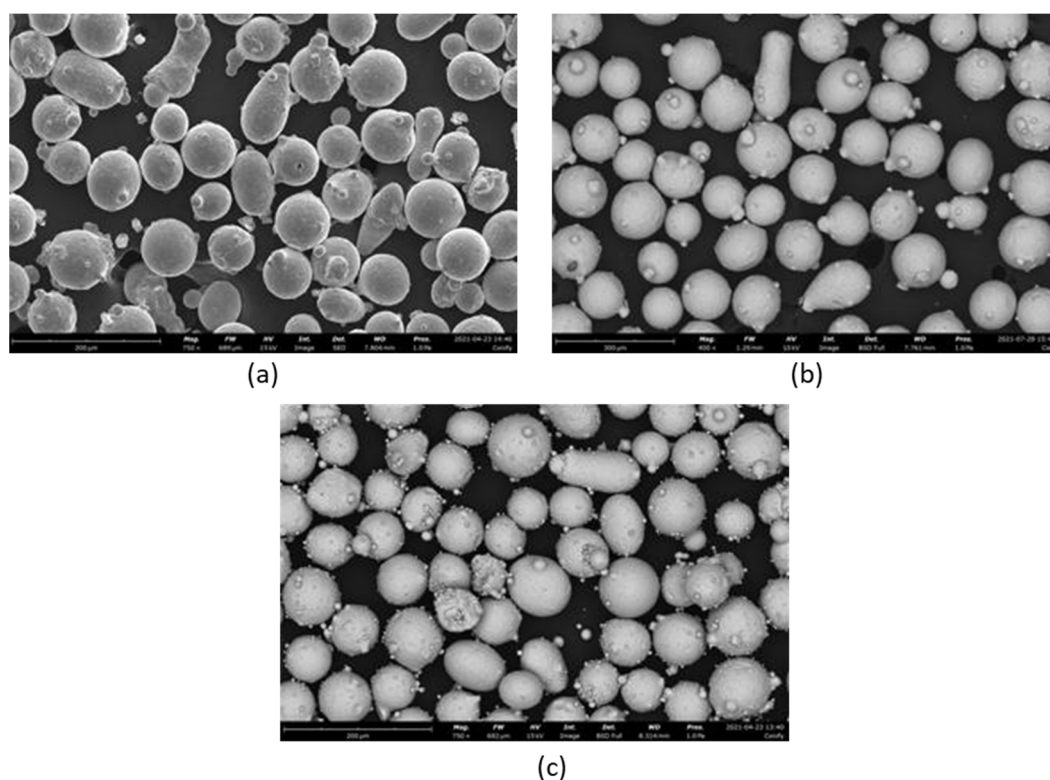


Figure 3. Backscattered electron micrographs of (a) Oerlikon (scale at 200 μm , magnitude 750 \times), (b) GKN Additive (scale at 300 μm , magnitude 400 \times), and (c) Atomizing AM (scale at 200 μm , magnitude 750 \times) metal powders.

2.1. Training and Methodology

The goal of this software is to provide users with an automated solution when analyzing metal powder for additive manufacturing. The intuition behind the software comes with some steps, and the following sections provide information that is related to the developed software methodology. This software is implemented in Python programming language [14].

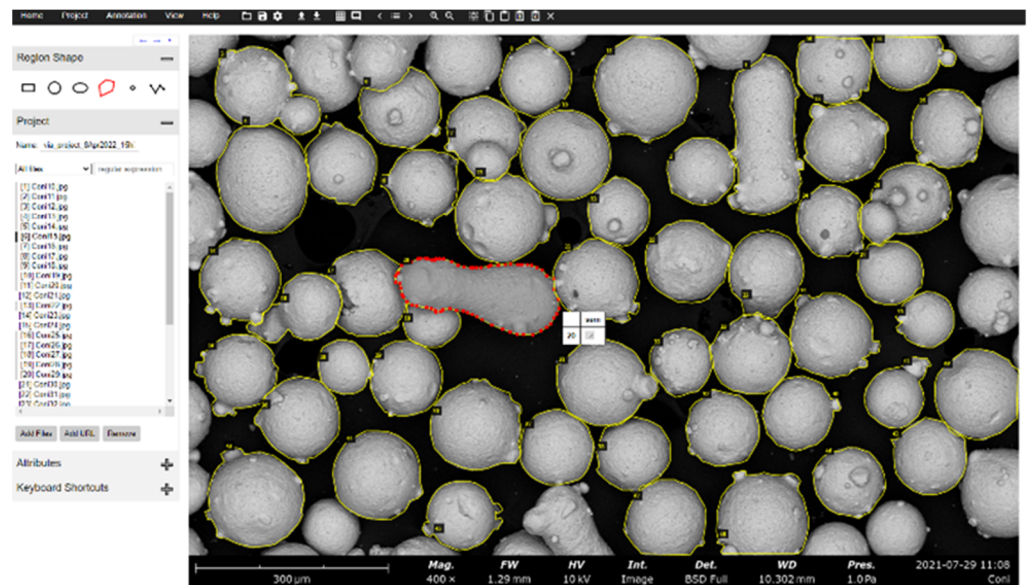
Annotation Labels

The acquisition of a large number of high-quality annotated images requires considerable manpower. Thus, the first step was the segmentation of the particles' area within the SEM images by utilizing an active learning [15] approach. A small number of images was manually annotated. Then, the PointRend [16] model was trained on these images to produce the masks for each particle instance of the remaining images. Then, the predicted masks were reviewed manually and corrected. This approach delivered the particle masks for all images in an efficient manner.

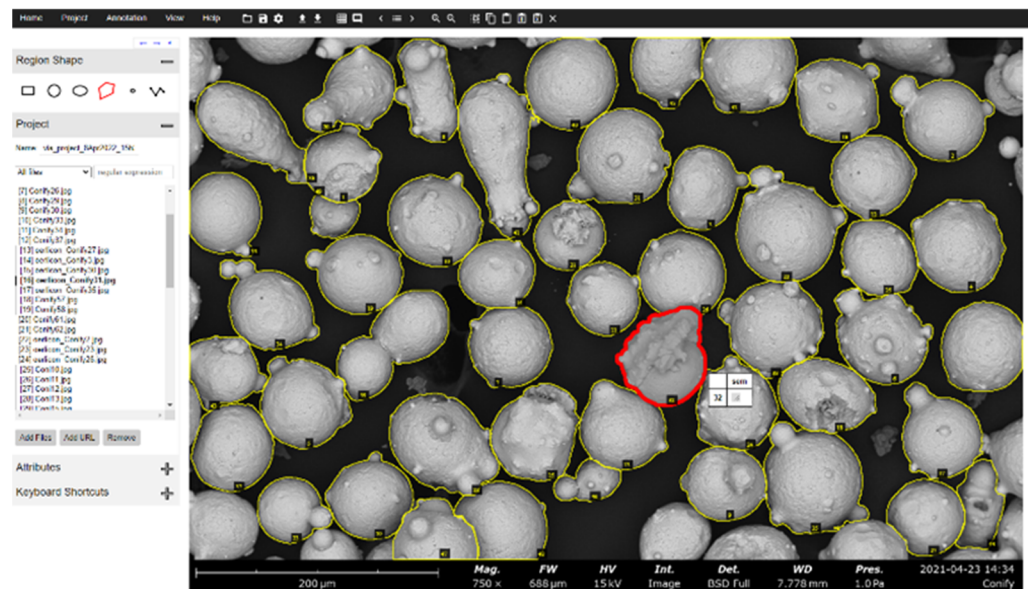
PointRend is an instance segmentation model that can efficiently render high-quality label maps by using a subdivision approach to select non-uniform sets of points at which to compute labels. It can be incorporated into SotA architectures, such as mask R-CNN, which is used in this implementation. PointRend implementation computes high-resolution segmentation maps using an order of magnitude fewer floating-point operations than direct, dense computation. In this work, the PointRend was implemented by Detectron2 [16], which is built on top of the PyTorch library [17].

For the manual annotation of the particles' area, the VGG Image Annotator (VIA) [18] was used. VIA is an open-source manual annotation software for images, audio, and video running on a web browser without any installation or setup, which makes it very user-friendly. Each particle was segmented with at least 15 vertices to annotate its perimeter, resulting in a json file. The constructed dataset included only one category, the existence

of a particle. A manually annotated image is shown in Figure 4a, and an output of the PointRender model is shown in Figure 4b. At this point, it is of great importance to stress the fact that PointRender was only used for the annotation of the particles' area within training data images. Training and serving of the PointRender model require a great amount of computing resources (GPU memory utilization) and time. Thus, for the rest of the developed methodology, the Detectron2 implementation of the mask R-CNN model was used.



(a)



(b)

Figure 4. Manual segmented particles' (a) and PointRender result particles' segmentation (b).

The next step was the categorization the metal powder elements into several types of structures. Four types of structures were defined for this analysis, as shown in Table 2.

Table 2. The table contains information about the categorization of the structures found within the images.

Type of Structure	Category
Particles with Satellites	Satellites
Nodular Particles (Agglomerated and Splat Cap)	Nodular
Elongated Particles	Elongated
Circular Particles	Circular

The output segmentation results from the PointRend model were converted into a VIA-compatible object, which was further categorized based on the structures defined in Table 2, utilizing the VIA tool. A collaboration of data scientists along with SEM specialists annotated the images that were used to create a training dataset that served as input for training of the mask R-CNN model. The annotation procedure resulted in a dataset of the json file format that included the location of the vertices for each segmented object in the x-y image plane as well as the category of the segmented object. There are 73 annotated SEM images, and Table 3 shows the number of instances for each category.

Table 3. The number of each particle category found within the images.

Category	Annotated Label Instances
Satellites	1614
Nodular	1109
Elongated	377
Circular	422

2.2. Training

In this section, the technical characteristics of the model are described. First, the dataset was split into train, validation, and test sets containing 54, 15, and 4 images, respectively.

The mask R-CNN model is used to predict and categorize the structures found in Table 2. The code has been modified in order to address the current scope of the described problem. The model generates bounding boxes and segmentation masks for each instance of a detected object in the image. It is a model that is based on the feature pyramid network (FPN) [19] and uses the ResNet50 [20] as a backbone. Using ResNet101 as a backbone did not significantly improve the performance for the described problem. Thus, the lighter ResNet50 was used. The Mask-RCNN used the pretrained weights on COCO [21] dataset as initial weights. These weights were retrieved from the open source model zoo repository by Detectron2.

The model has been fine-tuned in order to serve the purposes of the described problem in the most effective manner. It was trained by fine-tuning all the layers of the selected mask R-CNN architecture. The AdamW optimizer [22] and cyclical learn rate have been used [23] in order to accelerate the convergence of the model. The selection of the hyperparameters is performed by utilizing Bayesian optimization [24], minimizing the validation loss. As tool for Bayesian hyperparameter optimization, the sweeps from the Weights and Biases [25] library were used. The chosen distributions of the hyperparameters to search for were: for learning rate, $\log_{\text{uniform}}(10^6, 10^4)$; for weight decay, $\log_{\text{uniform}}(10^6, 10^4)$. The search distribution of the threshold on non-maximum suppression of region proposal network was uniform (0.5, 0.9); the distribution of the threshold on non-maximum suppression of region of interest heads was uniform (0.5, 0.9); the distribution of the score threshold of the region of interest on evaluation was uniform (0.4, 0.7); and the choices of the batch size per image of the region of interest heads was a choice from {128, 256, 512}. Before the Bayesian optimization, a manual search was performed to find the suitable search area of the distributions for each hyperparameter. The total number of trials utilizing sweeps was 50. After manual search for the optimum number of epochs, 53 was chosen. The early stopping method was not used because the one-cycle learning rate scheduler was utilized. In order for the model to visit also the low-learning-rate values at the end of the scheduler,

it should not be stopped earlier. The configuration of the best derived hyperparameters for the trained model consists of the following attributes described in Table 4.

Table 4. The table contains the model configuration attributes.

Category	Value	Description
GPU_COUNT	1	Number of GPUs utilized for training
BATCH_SIZE	2	Number of images on one train step
NUM_CLASSES	4	Number of classes
EPOCHS	53	Number of training passes utilizing the whole dataset
LEARNING_RATE	10^{-4}	Maximum value of learning rate
WEIGHT_DECAY	10^{-5}	Weight decay regularization
TRAIN_ROIS_PER_IMAGE	256	Number of ROIs per image to feed to classifier/mask heads
RPN_NMS_THRESHOLD	0.7	Non-maximum suppression threshold of region proposal network
ROI_NMS_THRESHOLD	0.65	Non-maximum suppression threshold of region of interest heads
DETECTION_MIN_CONFIDENCE	0.45	Minimum probability value to accept a detected instance
IMAGE_MIN_DIM	1000	Resized image minimum size
IMAGE_MAX_DIM	1333	Resized image maximum size

Along with all previously described configurations, augmentation was applied on each training batch during training. One or more of the following image augmentation methods was applied:

- Horizontal flip
- Vertical flip
- Rotation of [0, 90, 180, 270] angle
- Random brightness with intensity in (0.9, 1.1)
- Random contrast with intensity in (0.9, 1.1)
- Random saturation with intensity in (0.9, 1.1)
- Random lighting with standard deviation of principal component weighting equal to 0.9

The purpose of the models' training was to minimize the loss function, which is an average estimation of three losses [11]. The total mask R-CNN loss is defined in the following equation by incorporating losses from prediction class labels, bound box predictions, and binary predicted segmentation masks per instance.

$$L_{Total} = L_{Class} + L_{Box} + L_{Mask}$$

where

$$L_{Class} = -\log p_u$$

$$L_{Box} = \sum_{i \in \{x, y, w, h\}} smooth_{L1}(t_i^u - v_i)$$

in which $smooth_{L1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$

$$L_{Mask} = -\frac{1}{N} \sum_{i=1}^N [y_i \log s_i + (1 - y_i) \log(1 - s_i)]$$

where in L_{Class} , u is a vector with the true classes; $p = (p_1, \dots, p_K)$, over K categories, is the predicted probabilities which are calculated using a softmax function over the K outputs of the fully connected layer. In L_{Box} , $v = (v_x, v_y, v_w, v_h)$ is a vector with the ground true coordinates (x, y) of the center, the width, and the height of the box. The vector $t^u = (t_x, t_y, t_w, t_h)$ is the predicted box of class u . In L_{Mask} , the y corresponds to the ground

truth pixel label (0 or 1), and s corresponds to the predicted probabilities for pixel i . The training was performed on one NVIDIA GPU Tesla P100-16GB for both the PointRender model and the Mask R-CNN.

2.3. Framework—Software

The overall goal of this analysis was to create a user-friendly environment in which the user can upload SEM images and receive material characterization result. The software was developed in 3 steps.

The Flask [26] framework was used to create a web-based application. Both front- and back-end were developed using this framework. The front-end was designed in order to serve the needs of SEM experts. Thus, on the initial page, the user is prompted to create a user account to view and edit previously performed analyses. As a next step, the user provides information for the SEM image (specimen specifications) that are required for the analysis (e.g., magnification of the image). The user uploads an image or a batch of images for processing. The UI is shown in Figure 5.

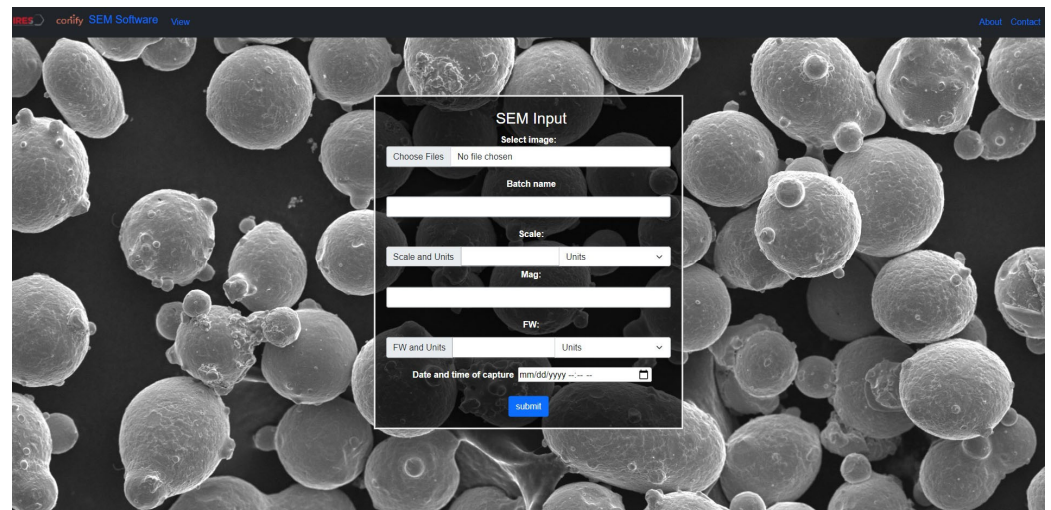


Figure 5. Step 1 of the user interface.

The next step of the software is the connection of the user interface with the mask R-CNN model. As discussed in the previous sections, the most accurate developed model was used for inference. The configuration file of the model that included the hyperparameters was given as a static input to the software, as were the output weights of the trained mask R-CNN model. For inference, the software makes use of the CPU while processing the images in batches. The results (both output images and statistical evaluation) are saved into an in-house-developed relational database.

The result consists of the segmented image that depicts the detected particle masks along with their respective class. The software uses the masks to assess shape properties and characteristics. Statistical analysis of the shape properties of the detected particles is performed per batch. The metrics that are calculated are given in the following Table 5 [27].

The calculated properties from Table 5 serve as total key morphological powder characteristics vital for the additive manufacturing process. In this scope, every calculated property for each detected particle is saved into a database and can be downloaded as a .csv file. The distribution of each property is provided to the user after evaluation.

Table 5. Statistical evaluation of the results.

Property	Symbol	Equation	Applied Particle Type	Units
Area	A	$\sum Pixels_i$	All	(nm ² or μm ²)
Perimeter	P	$\sum Perimeter\ Pixels_i$	All	(nm or μm)
Elongation	E	$1 - \frac{S_{axis}}{I_{axis}}$	Elongated	[0, 1]
Boxivity	B	$A - x_2 - x_1 \times y_2 - y_1 $	Nodular	[0, 1]
Circular Equivalent Diameter	χ_A	$\sqrt{\frac{4A}{\pi}}$	Circular, Satellites	(nm or μm)
Spherical Equivalent Volume	V	$\frac{1}{6}\pi(\chi_A)^3$	Circular, Satellites	(nm ³ or μm ³)

3. Results and Discussion

3.1. Trained Mask R-CNN Model Metrics

The most efficient model was fine-tuned for approximately 45 min with 53 epochs using the configuration settings presented in Section 2.2. The losses are presented in Table 6 and the AP metrics in Table 7.

Table 6. Losses of the defined model.

Loss Type	Train Value	Validation Value
Total Loss	0.325	0.789
Class Loss	0.169	0.410
Bounding box Loss	0.069	0.168
Mask Loss	0.047	0.114

Table 7. Validation AP50 overall and on the different classes.

Dataset	AP50
Overall	67.2
Elongated	69.5
Satellites	74.5
Circular	57.2
Nodular	60.7

For the evaluation metrics the following definitions are followed [19,28,29]:

- True positive (TP) is when a prediction–target mask (and label) pair has an IoU score which exceeds a predefined threshold.
- False positive (FP) indicates a predicted object mask that has no associated ground truth object mask.
- False negative (FN) indicates a ground truth object mask that has no associated predicted object mask.
- True negative (TN) is the background region correctly not being detected by the model, these regions are not explicitly annotated in an instance segmentation problem. Thus, we chose not to calculate it.
- Accuracy = $\frac{TP}{TP+FP+FN}$
- Precision = $\frac{TP}{TP+FP}$
- AP50 = $\frac{1}{n} \sum_{i=1}^n AP_i$, for n classes

Figure 6 shows an indicative result provided to the end user at the end of a batch detection analysis. The user is provided with AM particle analytics, as indicated in Table 5. The results include the percentages of the detected particle categories (circular, elongated, satellites, nodular) in the given batch. Additionally, the user is presented with the cumulative particle size distribution (CDF), from which the d10, d50, and d90 percentile values are extracted. These are statistical parameters that are read directly from the cumulative

particle size distribution. They indicate the size below which 10, 50, or 90% of all particles are found, respectively. Finally, distributions of all the metrics provided in Table 5 are provided to the end user using a dropdown list that includes the corresponding metric. The plots provided in this section are interactive, i.e., the user can zoom in or out, save the plot distribution locally, or even export the values of the given distribution.

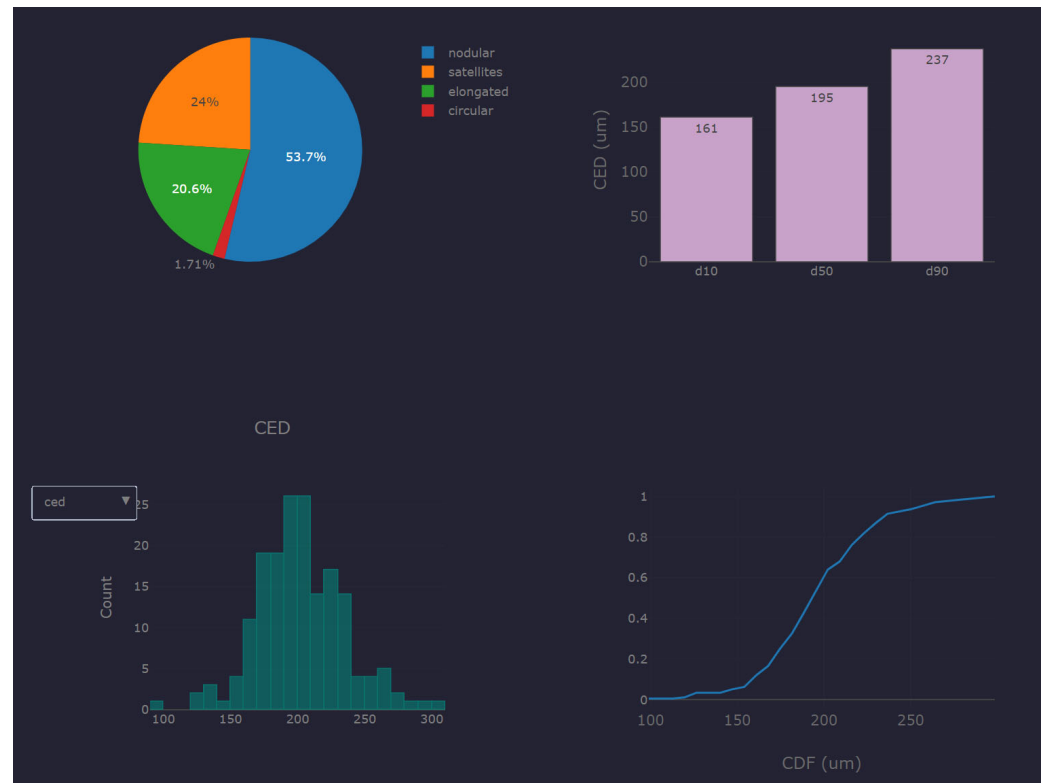


Figure 6. Indicative detection results provided to user for AM particle analysis for a given batch of images.

While the user is able to directly download the batch results, the user is also provided with the ability to dynamically search the previous results. All of the data are stored in an in-house database. The schema of the database is based on the output metric csvs. For each provided .csv, there is a corresponding table in the database. The user can search dynamically based on the batch name, timestamp, or even the name of the input files. Additionally, the user can download a .zip file containing previously detected images as well as the distributions with the corresponding statistical analysis metrics.

In Figure 7, an example of the statistics based on Table 5 is displayed. The histograms are calculated using a batch prediction outputs. For plotting the histograms of the area and perimeter, the predictions from all classes were used. For the circular equivalent diameter and the spherical equivalent volume histograms, both the circular and satellites classes were used. Finally, to plot the boxivity and elongation histograms, the classes nodular and elongated were used, respectively. Figure 7 show the histograms of the statistics, which gives an overview of the particle morphology.

In addition to the statistics, the tool displays the SEM images with their particles segmented and classified into the four chosen categories. The analyzed images are stored in a private cloud repository. The user is able to access the output images online or locally by downloading it. Table 8 shows the color convention that is used to visualize the segmentations of the particles in the results (an example is depicted in Figure 8).

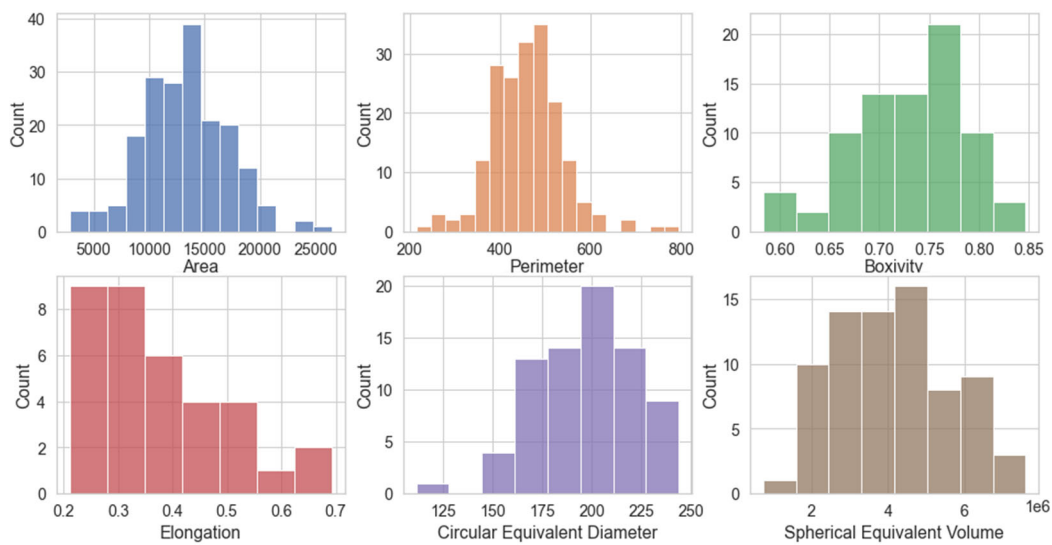


Figure 7. The calculated statistics based on a batch prediction from Mask-RCNN.

Table 8. Color Mapping for the predicted batch visualization of the defined classes.

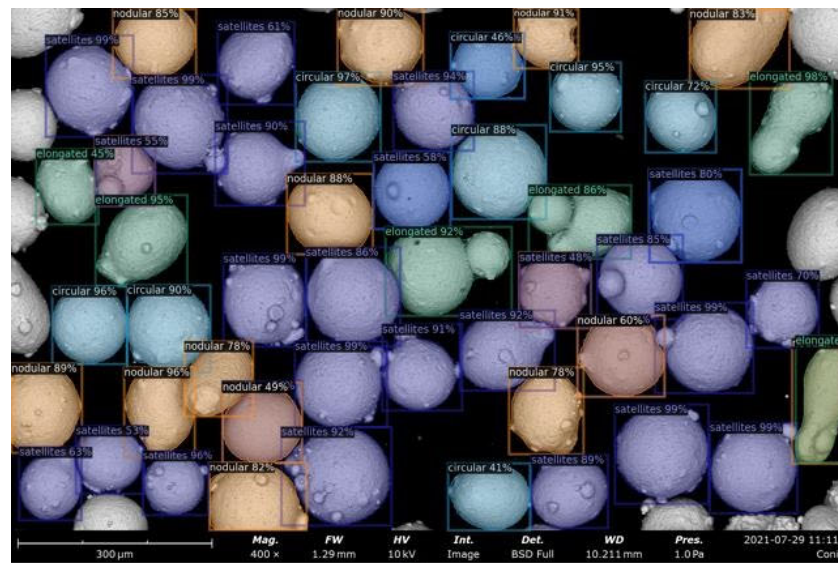
Particle Type	Color Coding
Elongated	Green
Satellites	Purple
Circular	Cyan
Nodular	Orange

3.2. Discussion

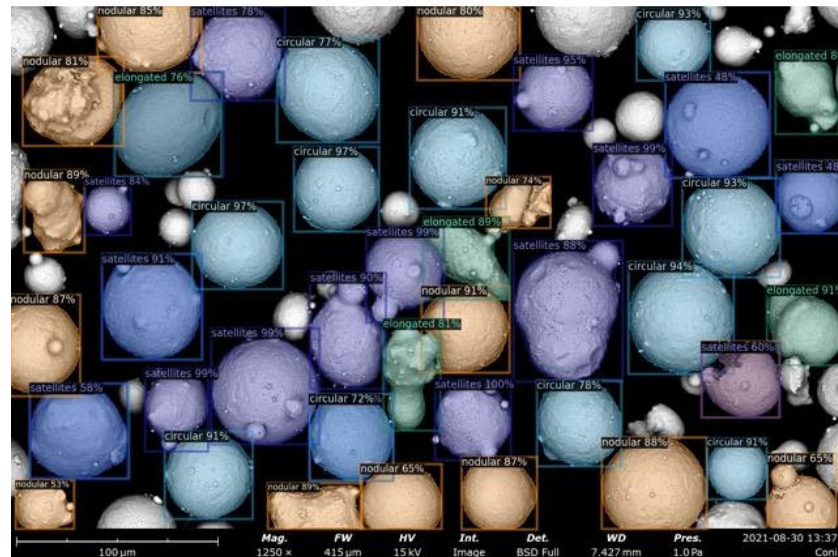
The software provides statistical information based on the mask R-CNN model results. The results are analyzed dynamically by providing distributions of key morphological powder characteristics for metal AM. The analyzed data are stored in an in-house relational database. The tool provides interaction with a user-friendly environment which incorporates the post-processed picture of the provided analysis. Table 8 displays the color mapping for the various defined particle classes.

The results from Section 3 show that mask R-CNN was successfully applied to classify and segment particles from SEM images. Additionally, transfer learning allowed high model performance while utilizing a small number of training images, only 54. Table 7 shows that the satellites category achieved the highest AP50. This was expected, as it is the category with the most annotated label instances. Additionally, the elongated category reached 69.5 AP50, as this category is the most distinguishable category based on the elliptical shape of its instances, which is depicted in the SEM images. The categories in which the model did not perform in such rate were the nodular and circular categories, with AP50 60.7 and 57.2, respectively. The main issue regarding the low performance of the model on these categories is explained due to the high similarities found between them.

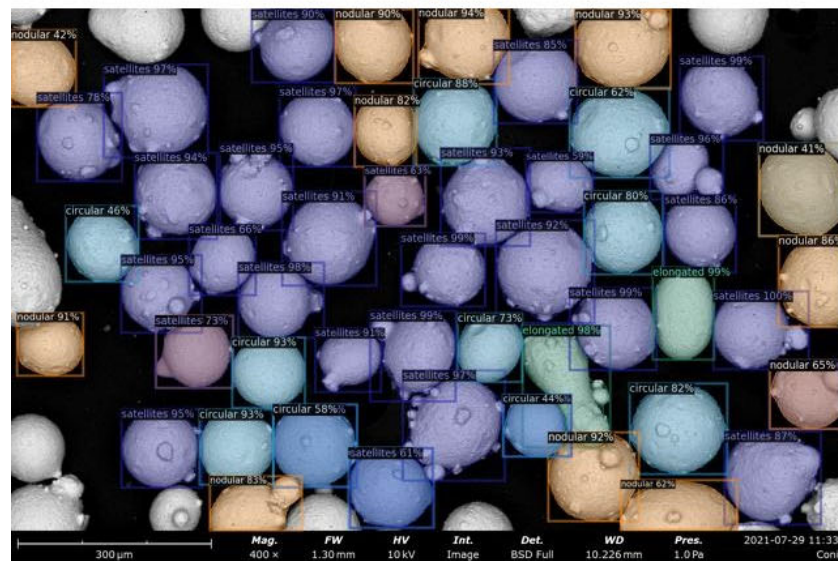
From a metallurgical point of view, the aim is to characterize metal AM powder particles based on specific features both quantitatively and qualitatively. Given a fully trained powder classifier, the next challenge is to demonstrate that it can recognize images taken from physical samples that were not included in its training dataset.



(a)



(b)



(c)

Figure 8. (a–c) The model predictions on a batch of backscattered electron images.

For each powder, a batch of SEM micrographs was used to validate the extracted results from the software (AI). This tool provides a first evaluation (as shown in Figure 8) of the particles' classification, which is unbiased due to the fact that no subjective criteria are used in the measurement method and the measurement of features. The results provide a clear indication of the AI tool's capabilities of autonomous classification of powder particles via backscattered SEM images, which can be extended to other powder-based materials. The automated approach equips the SEM expert with a non-biased, fast solution for a complete batch analysis on powders used for metal AM. The tool is able to capture most the particles classes and characteristics within a given image in an efficient way while also combining the results into a statistical analysis. It must be pointed out that particles located at the edges of the SEM micrographs are mostly identified as nodular due to insufficient image information, and they were excluded from the validation procedure.

Generalization of the proposed solution. The solution can scale up to a potential product such as 3rd party applications (e.g., ImageJ [30]). A vast amount of new annotated pictures will lead to a fast-paced solution able to generalize the categorization, segmentation, and statistical evaluation of powders used not only in metal additive manufacturing but also in pharmaceutical powders. Additionally, a potential scale-up of this approach is able to lead to less bias within the powder characterization industry.

4. Conclusions & Future Work

An automated tool for the analysis of metal powder SEM images using deep learning techniques has been developed and presented; more specifically, the state-of-the-art mask RCNN model for the both the classification and segmentation of metal AM powder particles are introduced. The application can assist the characterization of key material characterization characteristics, such as high sensitivity circularity, circular equivalent diameter, spherical equivalent volume, etc. The developed application eliminates bias and aims to save time as well as increase the accuracy and precision of the measurements compared to the already-existing methodologies and third-party software. Future work includes further development of the application using a larger number of training images in order to eliminate particles detected at the edge of any given SEM micrographs.

Author Contributions: Conceptualization, E.P.K., E.K.; methodology, G.B. and S.D. (Spyridon Dimitriadis), E.P.K. and E.K.; formal analysis, G.B., S.D. (Spyridon Dimitriadis), E.P.K. and S.D. (Stavros Deligiannis); investigation, G.B., S.D. (Spyridon Dimitriadis), E.K., L.G. and E.P.K.; resources, E.P.K.; writing—original draft preparation, G.B., S.D. (Spyridon Dimitriadis), S.D. (Stavros Deligiannis), L.G., I.S., K.B., E.K., E.P.K.; writing—review and editing, S.D. (Spyridon Dimitriadis), G.B. and E.P.K.; supervision, E.P.K.; project administration, E.P.K.; funding acquisition, E.P.K. and E.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by European Commission under Grant Agreement numbers 101016262 and 952869.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Data available on request due to restriction, on case specific.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Slotwinski, J.A.; Garboczi, E.J.; Stutzman, P.; Ferraris, C.; Watson, S.; Peltz, M. Characterization of Metal Powders Used for Additive Manufacturing. *J. Res. Natl. Inst. Stand. Technol.* **2014**, *119*, 460–493. [[CrossRef](#)] [[PubMed](#)]
2. Binayak, B. Comparative Study of Popular Deep Learning Models for Machining Roughness Classification Using Sound and Force Signals. *Micromachines* **2021**, *12*, 1484.
3. Binayak, B.; Gijun, P. Non-contact surface roughness evaluation of milling surface using CNN-deep learning models. *Int. J. Comput. Integr. Manuf.* **2022**, 1–15. [[CrossRef](#)]
4. DeCost, B.; Jain, H.; Rollett, A.; Holm, E. Computer Vision and Machine Learning for Autonomous Characterization of AM Powder Feedstocks. *JOM* **2016**, *69*, 456. [[CrossRef](#)]

5. Zhou, X.; Dai, N.; Cheng, X.; Thompson, A.; Leach, R. Intelligent classification for three-dimensional metal powder particles. *Powder Technol.* **2022**, *397*, 117018. [CrossRef]
6. Kirillov, A.; Wu, Y.; He, K.; Girshick, R. PointRend: Image Segmentation as Rendering. *arXiv* **2019**, arXiv:1912.08193.
7. Cohn, R.; Anderson, I.; Prost, T.; Tiarks, J.; White, E.; Holm, E. Instance Segmentation for Direct Measurements of Satellites in Metal Powders and Automated Microstructural Characterization from Image Data. *JOM* **2021**, *73*, 2159–2172. [CrossRef]
8. Chai, J.Z. Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Mach. Learn. Appl.* **2021**, *6*, 100134. [CrossRef]
9. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 142–158. [CrossRef]
10. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2015**, arXiv:2016.2577031.
11. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2007; pp. 2961–2969. [CrossRef]
12. Zhang, A.; Lipton, Z.C.; Li, M.; Smola, A.J. Dive into Deep Learning. *arXiv* **2021**, arXiv:2106.11342.
13. Thomas, M.; Drawin, S. Role of Metal Powder Characteristics in Additive Manufacturing. In Proceedings of the PM2016 World Congress, Hamburg, Germany, 9–13 October 2016.
14. Van Rossum, G.; Drake, F.L., Jr. *Python Reference Manual*; Centrum Voor Wiskunde en Informatica: Amsterdam, The Netherlands, 1995.
15. Ren, P.; Xiao, Y.; Chang, X.; Huang, P.-Y.; Li, Z.; Gupta, B.B.; Wang, X. A Survey of Deep Active Learning. *arXiv* **2020**, arXiv:2009.00236.
16. Wu, Y.; Kirillov, A.; Massa, F.; Lo, W.-Y.; Girshick, R. Detectron2 Open Source Repository. 2019. Available online: <https://github.com/facebookresearch/detectron2> (accessed on 1 October 2022).
17. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Chintala, S.; PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., Alché-Buc, F., Fox, E., Garnett, R., Eds.; 2019. Available online: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf> (accessed on 1 October 2022).
18. Dutta, A.; Zisserman, A. The VIA Annotation Software for Images, Audio and Video. In Proceedings of the 27th ACM International Conference on Multimedia 2019, Nice, France, 21–25 October 2019; pp. 2276–2279. [CrossRef]
19. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. *arXiv* **2016**, arXiv:1612.03144.
20. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.
21. Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Dollár, P. Microsoft COCO: Common Objects in Context. *arXiv* **2014**, arXiv:1405.0312.
22. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. *arXiv* **2017**, arXiv:1711.05101.
23. Smith, L.N. Cyclical Learning Rates for Training Neural Networks. *arXiv* **2015**, arXiv:1506.01186.
24. Snoek, J.; Larochelle, H.; Adams, R.P. Practical Bayesian Optimization of Machine Learning Algorithms. *arXiv* **2012**, arXiv:1206.2944.
25. Biewald, L. Experiment Tracking with Weights and Biases. 2020. Available online: <https://www.wandb.com/> (accessed on 1 October 2022).
26. Grinberg, M. *Flask Web Development: Developing Web Applications with Python*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2018.
27. Shanbhag, G.; Vlasea, M. Powder Reuse Cycles in Electron Beam Powder Bed Fusion—Variation of Powder Characteristics. *Materials* **2021**, *14*, 4602. [CrossRef]
28. Arase, K.M. Rethinking Task and Metrics of Instance Segmentation on 3D Point Clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop, Seoul, Korea, 27–28 October 2019.
29. Yang, F.Y. Accurate Instance Segmentation for Remote Sensing Images via Adaptive and Dynamic Feature Learning. *Remote Sens.* **2021**, *13*, 4774. [CrossRef]
30. Schneider, C.A.; Rasband, W.S.; Eliceiri, K.W. NIH Image to ImageJ: 25 years of image analysis. *Nat. Methods* **2012**, *9*, 671–675. [CrossRef] [PubMed]