Computer Vision Approaches for Segmentation of Nanoscale Precipitates in Nickel-Based Superalloy IN718

Nishan M. Senanayake¹ · Jennifer L. W. Carter¹

Received: 4 September 2020 / Accepted: 18 November 2020 / Published online: 16 December 2020
© The Author(s) 2020

Abstract

Extracting accurate volume fraction and size measurements of γ'' and γ' precipitates in iron-based superalloys from micrographs is challenging and conventionally involves manual image processing due to their smaller size, and similar crystal structures and chemistries. The co-precipitation of composite particles further complicates automated segmentation. In this work, different types of traditional machine learning approaches and a convolutional neural network (CNN) were compared to a non-machine learning approach, for the segmentation of the composite particles of γ'' and γ' precipitates. The objective was to optimize metrics of segmentation accuracy and the required computational resources. The data set contains 47 experimentally generated scanning electron micrographs of IN718 alloy samples, computationally increased to 188 images (900×900 px). All algorithms are containerized using singularity, publicly available, and can be modified without dependencies. The CNN and the random forest models achieve 95% and 94% accuracy, respectively, on the test images with better computational efficiency than the non-machine learning algorithm. The CNN tested accurately over a range of imaging conditions.

Keywords Disc superalloy · Image processing · Machine learning · Phase segmentation

Introduction

IN718, a nickel–iron-based superalloy, is an important candidate for additive manufacturing for both its weldability and high-temperature properties [1, 2]. The γ'' ($\text{Ni}_3\text{Nb}-\text{DO}_{22}$ crystal structure), and γ' ($\text{Ni}_3\text{Al}-\text{L1}_2$ crystal structure) precipitate phases mainly contribute overall favorable high-temperature properties [1]. The measurement of these individual microstructural features is important for implementation of process-structure-property models and calibration of precipitation simulation models that inform the optimization of processing routes [3–6]. Due to the nanoscale size, morphology, and chemical similarities of phases, the quantitative characterization of these precipitates is challenging and leads many studies to not include quantified values of the γ'' and γ' strengthening precipitates [7–10]. Even CALPHAD-based simulations for microstructure modeling require experimental calibration with precise measurement of the

precipitate dimensions, volume fractions, and number densities for IN718 [11, 12]. The most cost-time effective experimental methods of extracting the mentioned microstructure metrics has been manual segmentation of scanning electron microscopy (SEM) images [10, 13]. Other approaches include transmission electron microscopy (TEM) [8] and atom probe tomography, but due to the extensive sample preparation, these approaches have limited applicability for industrial environments [14].

The kinetics of the precipitates first result in formation of γ'' , then γ' precipitates in both the nickel matrix and at the matrix- γ'' interphase boundary to create both individual and composite particles, respectively [15]. Segmenting the composite particles to individual γ' and γ'' is a challenging task when using standard image processing methods such as watershed, standard thresholding, etc., because the contours of visual objects intersecting with each other in an SEM micrograph contain insufficient visible geometrical evidence. Gradient-based algorithms are incapable of producing accurate segmentation because a strong gradient is not present; the large area fraction of composite particles produces a large variation in orientation, size, and shape of the objects, which leads to a more complex

✉ Nishan M. Senanayake
nms105@case.edu

¹ Department of Materials Science and Engineering, Case Western Reserve University, Cleveland, OH 44106, USA

segmentation problem. Therefore, this segmentation process previously involved manual segmentation based on human interpretation.

We recently implemented an image processing method using mathematical algorithms (non-machine learning) which achieves up to 94% accuracy compared to ground truth as determined by manual segmentation [16]. The approach was based on finding the edge points, assigning symmetrical points to the edge points in a given radius, and the classical ellipse fitting to estimate the missing part of the contours. The algorithm assumes edge contours form elliptical shapes and approximate shapes of partially overlapping objects are as ellipses. Therefore, the algorithm is not able to accurately capture cuboidal particle shapes when they grow and evolve. The development and implementation of non-machine learning algorithms is a challenging task for those in the non-computer science disciplines because they require extensive knowledge of mathematics and computer science [17]. In addition, most existing mathematical algorithms are not compatible with graphics processing units (GPU). Therefore, even though a central processing unit (CPU) in high-performance computing (HPC) cluster can be used to run the algorithm, accelerating the running time with the advantage of GPU infrastructure is a difficult task. In comparison, convolution neural network (CNN) learning approaches can be trained using HPC infrastructures with GPU capabilities; once trained the algorithms can be run on CPU architecture. This makes open-source, trained CNN algorithms valuable for industrial applications.

In computer vision, due to rapid development of the ecosystem for machine learning and the complexity of mathematical-based algorithms, machine-learning approaches have rapidly become a potential solution for classification problems in scientific images [18, 19]. Random forest (RF) and support vector machine (SVM) are two traditional supervised machine learning algorithms that have been used for both classification and the regression problems [20]. Both of these approaches are suited for small, sparse data sets that emerge in microstructural analysis problems unsuitable for deep neural network approaches.

RF generates a collection of decision trees (a forest), a graphically represented model, that can be used to classify data sets effectively [21]. The primary disadvantage of individual decision trees is overfitting, which is prevented by using multiple collections of random decision trees. These multiple decision trees are generated using bootstrapped datasets and a randomly picked subset of features. The features for the root node of each tree are selected using Gini impurity criterion [22]. RF classifies the input values based on the majority vote given by all the decision trees in the collection. The SVM is based on the minimization of the loss function and the principle from unseen patterns [23]. SVM finds the hyperplane that partitions and

draws a margin between the classes based on the distance values. The input data can be classified by selecting the relevant hyperplanes found by the SVM approach. Two of the disadvantages of these traditional machine learning models are manual feature extraction and that the images are not directly taken as input. Additionally, overfitting can occur in the traditional machine learning models [24].

CNN is a deep neural network approach that is specifically designed for image and video classification [25]. The U-Net is a CNN designed for pixel-wise image segmentation that consists of a down-sampling (encoding) path and an up-sampling (decoding) path to reconstruct the original image [26]. U-Net architecture allows for training of extremely deep networks and has shown reasonable results nearly all the time for image classification problems. In CNN, transfer learning which is the capability of a model to recognize and employ the knowledge learned from a previous domain to a new domain can be used, and is considered to be a robust technique for building and training architectures faster than developing the network from the scratch [27]. In transfer learning, the final layers of the pre-trained model's architecture are modified accordingly along with the fine-tuning for the certain problem [27]. Hence, the same CNN-based approach can be applied to many different applications with slight modifications [27]. Another advantage of CNN over RF and SVN is that CNN does not require manual feature extraction; it has the ability to deduce the features and is optimally tuned for the desired outcome.

The disadvantage of using either traditional machine learning algorithms (RF and SVM) or CNN instead of the mathematical, non-machine, learning image processing algorithm is the need for large quantities of “ground truth” manually labeled training data. The CNN approach requires more data than the traditional machine learning approaches to obtain accuracy and prevent model overfitting [28]. Data augmentation is a simple, but powerful tool that can be used to avoid overfitting the data. There are a variety of accepted methods of computationally augmenting existing data to increase the amount of training data [29].

In this work, we have compared the previously validated mathematical algorithm [16] with traditional (RF and SVM) and novel (CNN with U-Net architecture) machine learning approaches to segment the composite particles in SEM images of IN718. The different methods of computing-containerizing and possible usage of GPUs have been applied to each of the approaches in order to compare the resource allocation and the time effectiveness. Comparison of the optimization and satisfaction metrics gives the option to choose the best approach that can be applied to each specific problem accordingly, based on the hypothesis that the presented approaches can be generalized, and thus applied to a variety of segmentation in micrographs.

Experimental

IN718Net Dataset

The SEM micrographs in *IN718Net* were acquired at the NASA Glenn Research Center and the representative sample preparation and imaging conditions are presented in T.M. Smith et al. [16]. The dataset consists of 47 unique 900×900 px micrographs acquired from additively manufactured IN718 samples. The pixels in the original image (Fig. 1a A low-kV, secondary electron detector, SEM micrograph of etched surface of superalloy IN718, and a (b) labeled image with three classes (γ'' : brown, γ' : light blue, and background blue).) were manually labeled into three classes (γ'' : brown, γ' : light blue and matrix: blue) (Fig. 1a A low-kV, secondary electron detector, SEM micrograph of etched surface of superalloy IN718, and a (b) labeled image with three classes (γ'' : brown, γ' : light blue, and background blue). for the training of the machine learning models. All the training and labeled images were preprocessed to the same size and shape to speed up the training process. Default center cropping has been used for resizing, which grabs the center area of the image and resizes it to a given value. During preprocessing, the input pixel intensities were normalized by converting to a tensor with a mean of zero and a standard deviation of one, speeding up training of the deep learning model. A standard data augmentation method of symmetric flipping was used to increase the number of micrographs. Each image was flipped horizontally, vertically, and a combination of the two. Therefore, the 47 of experimentally generated 900×900 px SEM images were increased up to 188 images by using data augmentation.

Mathematical Image Processing (Non- machine Learning Approach)

The algorithm, based on the approach of Zafari et al. [30], automates the identification and segmentation of circular γ' and elliptical γ'' phases from grayscale images, including segmentation of partially overlapping γ' and γ'' co-precipitates. The algorithm is written in Python (v3.6) using standard NumPy, Matplotlib, Skimage, OpenCV, Pylab, Math, and Scipy libraries. Since this is not a machine learning approach, it does not require a labeled “ground truth” training dataset. The five major algorithm steps are described briefly below and the detailed explanation and the results are presented in Smith et al. [16].

Step 1: Preprocessing The SEM image was converted to a grayscale and then to a binary image using Otsu thresholding. Objects smaller than 9 pixels are removed using a denoising filter. Then, an erosion filter was applied to shrink the bright (precipitate) regions and enlarge the dark (matrix) region, which helps to increase the separability of co-precipitates.

Step 2: Seed and Edge Points Extraction Edge points were detected by using the Canny edge detector. The average geometric central seed points of the objects were detected using fast radial symmetry transform (FRS) [31].

Step 3: Assign Edge Points to Seed Points Each group of adjacent edge-points (contour) was assigned to a seed-point based on highest relevance. The edge-to-seed point correlation combines the distance and the divergence index to assign edge pixel points to the seed pixels.

Fig. 1 **a** A low-kV, secondary electron detector, SEM micrograph of etched surface of superalloy IN718, and **a** labeled image with three classes (γ'' : brown, γ' :light blue, and background blue)

Step 4: Ellipse Fitting Classical ellipse fitting approach was used to estimate the missing contours of co-precipitates from the partially observed ellipse-shape objects. If the angle between the long axes of the overlapping ellipses is $\leq 15^\circ$, the algorithm considers the overlapping ellipses to be one particle.

Step 5: Separate γ' and γ'' The γ' precipitates were then differentiated from the γ'' precipitates based on the 2.25 cut-off aspect ratio determined by using STEM/EDS [16]. The algorithm chooses the γ'' over the γ' whenever their contours overlap based on assumptions of the precipitate kinetics.

Traditional Machine Learning Approach

Random Forest (RF) and Support Vector Machine (SVM) traditional supervised machine learning algorithms were implemented using the OpenCV and scikit-image libraries in Python [32, 33]. After removing the noise from the images, 52 different (statistical, textural, gradient, context, projections) extractors that are commonly used in scientific image segmentation were ingested to the RF and SVM classifiers to assess important features [34–36]. In particular, Harris corner detector [37], Oriented FAST and Rotated BRIEF (ORB) [38], Gabor, Canny, Roberts, Sobel, Scharr, Prewitt, Gaussian, Median, Variance filters, etc., were used [39]. Some of the extracted features are shown in Fig. 2. Original pixel values and some of the features extracted by Canny, Sobel, Gaussian, Gabor, Variance, and Median filter.

The extraction of important features and the hyperparameter optimization were done for the RF and SVM classifiers by using a subset of *IN718Net* (10 images). Only the features that are informative for classifiers were selected by generating a map of importance using the feature_importances library [40]. Thirteen informative features of the fifty two were identified, and their importance to the classifier are mapped in Fig. 3. Area map of the rank-order importance of different feature extractors used for classifiers. Only the thirteen extractor methods shown, of the fifty two attempted methods result in features of importance for IN718 micrographs. It can be seen in the features map that the Gabor filter with different parameter values (gamma, theta, lambda) [35] which generates various frequency and orientation

Fig. 3 Area map of the rank-order importance of different feature extractors used for classifiers. Only the thirteen extractor methods shown, of the fifty two attempted methods result in features of importance for IN718 micrographs

representations, provides the highest degree of classifier information. Linear smoothing filters such as Gaussian and Median also contributed information to the classifiers. The different types of edge detectors such as Sobel, Canny, Scharr, Roberts, Prewitt [41] that identify and highlight gradient edges were also informative. The Harris filter extracts corners and infer features of an image has also provided a significant amount of information for the classifiers. The classifiers also have used information extracted from Oriented FAST and Rotated BRIEF (ORB) [38].

The RF and SVM model parameters were randomly sampled from the distribution of possible parameter values, using a randomized optimization search over hyperparameters implemented by the RandomizedSearchCV library [42]. The hyperparameters selected are listed in Table 1. Listed are the optimized hyperparameters chosen by randomized optimization search for RF (left) and SVM (right) that give the best accuracies. It is not the intention of this work to explain these parameters, and details can be found in the literature [21, 23].

After the selection of important features and the optimization of the hyperparameters, the RF and SVM models were trained from a larger set of *IN718Net*. This two-step training approach makes the process faster than trial and error method or feature selection and hyperparameter

Fig. 2 Original pixel values and some of the features extracted by Canny, Sobel, Gaussian, Gabor, Variance, and Median filter

Table 1 Listed are the optimized hyperparameters chosen by randomized optimization search for RF (left) and SVM (right) that give the best accuracies

Random forest		Support vector machine	
Number of trees	250	Regularization parameter	1.0
Max features	Auto	Kernel	rbf
Max depth	None	Gamma	Scale
Minimum sample split	2	Max iterations	−1
Minimum sample leaf	1	Decision function shape	ovr
Bootstrap	True		

optimization with the entire dataset. The steps of models training are shown in Fig. 4. The pipeline of model training for the entire dataset followed by hyperparameter optimization and important feature selection. The arrow represents the process of using the chosen parameters and features for the final model training with the entire dataset. A hundred and sixty, randomly selected, SEM images and their “ground truth” labels were used for final model training and twenty eight images from *IN718Net* were used for the testing and validation.

Convolution Neural Network

The CNN used the U-net structure with a modified, pre-trained, residual, 50 layer deep, network (ResNet-50) [43]. The ResNet architecture includes residual blocks (shown in Fig. 5a Schematic representation of the residual block that may contain different combinations of layers with the identity connection that is shown in the red arrow. (b) Schematic representation of the neural network that 0.2 percent of activations were removed (represented in red color.) that diminish the gradient vanishing problem and the issue of degradation of deep networks by introducing the identity connection. The modified ResNet-50 model includes groups of convolution, batch-norm, maxpooling,

Fig. 5 **a** Schematic representation of the residual block that may contain different combinations of layers with the identity connection that is shown in the red arrow. **b** Schematic representation of the neural network that 0.2 percent of activations were removed (represented in red color)

identity block and the newly added final layers [43]. The entire architecture was developed using the Fastai [44] and standard Python libraries. Initially, a smaller pretrained ResNet-34 was utilized to train the newly added final layers before proceeding to training the larger ResNet-50 architecture. This stepwise training is an efficient method to increase the model accuracy and use less memory in the training phase.

Regularization

Weight decay (L2 regularization) [45] and dropout [46] regularization techniques were iteratively used to solve the problem of high variance that causes model overfitting. The weight decay technique modifies the cost function using the constant times the sum of the square of the weight parameters as shown in Eq. 1.

Fig. 4 The pipeline of model training for the entire dataset followed by hyperparameter optimization and important feature selection. The arrow represents the process of using the chosen parameters and features for the final model training with the entire dataset

$$L(w, b) = \text{Normalized lost unction} + \frac{\lambda}{m} w^2, w^2 = w^T w \quad (1)$$

where L is the modified lost function which is a function of weights (w) and biases (b), m is the number of samples, and λ is the weight decay constant. Previous studies have shown that λ of 0.1 is one of the optimal values [47], which was used for this work. In dropout, a percentage of the activations in the hidden layers (not the weights or parameters) are removed, a dropout of 0.2 was used for this work and is shown schematically in Fig. 5a Schematic representation of the residual block that may contain different combinations of layers with the identity connection that is shown in the red arrow. (b) Schematic representation of the neural network that 0.2 percent of activations were removed (represented in red color)..

Cross Entropy Loss

The normalized cross entropy loss function was employed (Eq. 2) for model training [48]. This is optimized in a way that correct predictions have a smaller probability of loss and the wrong predictions should have a higher probability of loss. The softmax activation [49] function was included in the final layer of the model architecture for compatibility with the cross entropy loss function.

$$H(y, \hat{y}) = - \sum_i y_i \log_e (\hat{y}_i) \quad (2)$$

where $\hat{y}$ is the predicted probability vector from the softmax function and y is the ground-truth vector.

Adaptive Moment Estimation

Adaptive moment estimation (Adam) is the optimizer that iteratively updates the weights and biases of the CNN based on training data [50] using the momentum and root mean square propagation (RMSProp) [51]. Adam calculates the exponentially weighted average and weighted average of squares of past gradients of weights with the bias correction and includes the weight update formulas accompanied with β_1 and β_2 constants. In the CNN, standard values of 0.9 and 0.999 were used for β_1 and β_2 constants, respectively.

CNN Model Training

The schematic representative of the implemented model is shown in Fig. 6. Schematic representation of the encoder that shows the resizing layers, the layers of the pretrained model, and newly added layers.. The first layer resizes the image to be compatible with the pretrained model, then are the layers of the pretrained models (Resnet-34 or Resnet-50),

Fig. 6 Schematic representation of the encoder that shows the resizing layers, the layers of the pretrained model, and newly added layers

and finally, the newly added layers for this particular image segmentation application.

The model training follows through recursive steps to achieve the optimal hyperparameter values. The model training steps in CNN approach can be summarized into four major steps.

Step 1 The layers of the pretrained model (RES-net 34) were frozen and only the newly added layers (weights with a Rectified Linear Unit (ReLU) activation) [52] were trained until the best accuracy was reached. To find the best learning rate parameter for new layers, the learning rate finder was performed and generated the values of loss function against the learning rate graph for mini batches (4 images) as shown in Fig. 7a Loss against the learning rate. The model was trained with the range of learning rate highlighted in the red rectangle with the steepest slope region and (b) comparing the accuracy to select the optimized learning rate. The range of learning rate values at the steepest slope, as indicated by box in Fig. 7a Loss against the learning rate. The model was trained with the range of learning rate highlighted in the red rectangle with the steepest slope region and (b) comparing the accuracy to select the optimized learning rate. The learning rate of $5e^{-4}$ that yielded the highest accuracy was chosen to train the added layers. Initially, the model was trained with a smaller image size of 300×300 px (by using default center cropping) for the first cycle of training, whereas the full size images (900×900 px) were used for the final training. Starting with smaller images and increasing the size for the final

Fig. 7 **a** Loss against the learning rate. The model was trained with the range of learning rate highlighted in the red rectangle with the steepest slope region and **b** comparing the accuracy to select the optimized learning rate

Fig. 8 The number of epochs, divisions in time, against the accuracy; **a** for the step 1 training of the newly added layers and **b** the step 2 training of all the layers including the newly added layers

cycles of training (progressive resizing) is a technique for robust model training [53]. The model achieved 90.2% of accuracy as shown in Fig. 8a.

Step 2 The entire model was unfrozen and trained with discriminative learning rates again with smaller image size of 300×300 px (by using default center cropping), which set a gradient of learning rates along all the layers in the network [54]. The layers of the pre-trained model that were well optimized for detecting the primary features were trained with a smaller learning rate ($5e^{-5}$). The newly added layers, as indicated in Fig. 8. The number of epochs, divisions in time, against the accuracy (a) for the step 1 training of the newly added layers and (b) the step 2 training of all the layers including the newly added layers, require more optimization than the layers in the pretrained model and were thus trained with a greater learning rate ($5e^{-4}$), as determined in step 1. Applying discriminative learning rates accelerates the model training and also helps to tune hyperparameters [55]. The accuracy of the model improved to 92.1% as shown in Fig. 8. The number of epochs, divisions in time, against the

Table 2 Total number of parameters of the Resnet-34 (left) and Resnet-50 (right) models

	Resnet-34	Resnet-50
Total parameters	41, 221, 268	339, 070, 292
Total trainable parameters	19, 953, 620	315, 615, 380
Total non-trainable parameters	21, 267, 648	23, 454, 912

accuracy (a) for the step 1 training of the newly added layers and (b) the step 2 training of all the layers including the newly added layers.

Step 3 The trained model was loaded to the GPU and trained further with the larger images (900×900 px). The accuracy has been raised to 93.4%.

Step 4 The model was trained with the larger RES-net 50 architecture, number of parameters of Resnet-34 and Resnet-50 shown in Table 2. Total number of parameters

of the Resnet-34 (left) and Resnet-50 (right) models. The accuracy of the final model was 95%.

Accuracy Metrics of the Test Images

The accuracy scores were calculated using the set of test images from *IN718Net*; these images were not used to train the models. The optimizing metric of accuracy of the four approaches (Mathematical, RF, SVN, CNN) were evaluated by comparing the percentage of correctly predicted pixel values and the ground truth pixel values using the following equation (Eq. 3).

$$\text{Accuracy} = \frac{\text{Correctly predicted pixels}}{\text{Total number of pixels}} \quad (3)$$

Computer Resources Optimization

The High Performance Cluster (HPC) Environment at CWRU was used for this work. A load-balancing router which distributes sessions across login nodes connects the workstations to the HPC and three login nodes work as gateways to the cluster. The HPC system contains over 200 compute nodes and 4000 compute cores. All the nodes are equipped with Intel Xeon $\times 86$ 64 @ 2.10 GHz processors. Both the mathematical algorithm (non-machine learning) and traditional machine learning (RF and SVM) classifiers were run on 2 processors of Xeon phi 5110 and 12 cores with the 32 GB of memory in a high performance computing cluster due to their incompatibility with the GPUs. CNN model was trained on the 16 Gb Memory, 4 CPU and GPU Nvidia v100.

The Singularity, an open-source cross-platform container solution that resolves the issue of system dependencies, was employed to facilitate the reproducibility of the scientific computing and high-performance computing [56]. The runtime environment and system dependencies can be defined for the implemented singularity container image and the containers can be executed on any singularity installed hosts without reinstalling the required software packages and system dependencies. The representation of computer hardware and software infrastructure that the models were built on is shown in Fig. 9. Hardware and software infrastructure that was used to implement the containerized solutions for computer vision models.

Results and Discussion

Mathematical Image Processing Algorithm

The running time of the algorithm was 67 s, and achieved 94% accurate predictions. Figure 10 represents the visual results from the non-machine learning algorithm compared with the ground truth (labeled by hand). The main reason for the discrepancy between the predicted and ground truth is caused by the algorithm assumption of predefined shapes (ellipse and circle). Thus, it is not able to accurately predict the rare, complex-shaped, γ' and γ'' precipitates as shown in numbered particles in Fig. 10.

Traditional-Machine Learning Models

The running time and corresponding accuracy scores of RF and SVM models are 4 s and 94%, and 128 s and 91%,

Fig. 9 Hardware and software infrastructure that was used to implement the containerized solutions for computer vision models

Fig. 10 Visual representation of ground truth vs. the predicted image. The numbers indicate the over labeled particles due to the predefined shapes in predicted image compared to the ground truth image. The segmented layer is translucent and stacked on the original images

respectively. The examples of predicted images from RF and SVM models are shown in Fig. 11. The comparison of the RF and SVM predicted output images with the ground truth image indicates oddly shaped anomalies that are inconsistent with precipitate kinetics. The segmented layer is translucent and stacked on the original images. In both cases, the models mislabel pixels that result in the formation of precipitates that are inconsistent with precipitation kinetics (i.e. γ' particles inside the γ'' particles). The discrepancy between the ground truth and the predictions is caused by the nature of the extracted features and classifiers. The SVM classifier requires longer running time, as it is trained using the dual objective. Therefore, with N examples, the matrix contains N^2 elements [23].

Convolution Neural Network

The comparison between the ground truth and predicted images is shown in Fig. 12. This compares the ground truth (left) and the CNN prediction (right). The numbers compare

the interpretations, which are consistent with physics and are equally likely solutions without additional chemical information. The segmented layer is translucent and stacked on the original images. The running time for the prediction was one second, and the accuracy was 95%. The numbered cases compared between the ground truth and the predicted in Fig. 12. This compares the ground truth (left) and the CNN prediction (right). The numbers compare the interpretations, which are consistent with physics and are equally likely solutions without additional chemical information. The segmented layer is translucent and stacked on the original images, indicating that the accuracy of the CNN might arise from human bias in the ground truth. In the ground truth in Fig. 12, the area of the γ'' particle at location 1 was over labeled. At location 2 in the ground truth, the particle is separated into two pairs of γ' and γ'' particles, whereas the CNN labeled them as a single γ'' particle. The CNN prediction seems more probable because of the human bias in the ground truth image. At number 3, over-labeling results in the ground truth identifying two γ'' particles. The CNN

Fig. 11 The comparison of the RF and SVM predicted output images with the ground truth image indicates oddly shaped anomalies that are inconsistent with precipitate kinetics. The segmented layer is translucent and stacked on the original images

Fig. 12 This compares the ground truth (left) and the CNN prediction (right). The numbers compare the interpretations, which are consistent with physics and are equally likely solutions without additional chemical information. The segmented layer is translucent and stacked on the original images

predicted them as a one single γ' and γ'' instead of two γ'' particles. At number 4, the two γ' particles are connected in the ground truth image, but they are separated in the prediction image. At numbers 5 and 6 the prediction image mislabeled γ'' inside a γ' particle. The variations in predicted particles is within the error that might be expected if comparing ground truth interpretations from two different researchers.

SEM images can be acquired with varying brightness, contrast, and resolution conditions. Therefore, to assess the efficacy of the CNN model predictions when subjected to variations in these conditions, the CNN model was subsequently tested with images from IN718Net that were computationally modified for different, independent, imaging conditions by using pillow 3.0 python package [57]. Three of the applied brightness, contrast and resolutions scales are shown in Fig. 13. Three relative scales of contrast, brightness (0.1, 0.5, 0.9) and resolutions (9, 18, 72 pixels/inch) are presented. The bolded values are the inherent IN718Net image conditions.

As shown in Fig. 14a, the CNN model maintains 95% accuracy in predicting pixel phase identification with in the relative contrast range of 0.4–0.8 and brightness range of 0.3–0.7. The accuracy decreases to 92% for the images with the relative maximum brightness and the minimum contrast due to vanishing image features. The model retains 95% accuracy for images with resolution as low as 36 pixels/inch, as shown in Fig. 14b, or close to a 50% reduction in resolution. At this magnification, a resolution of 36 pixels/inch is 1.41 pixels/nn. The accuracy of the model decreased to 93.6% for the images with a very low resolution of 9 pixels/inch. The average γ' size was 35 nm in diameter and the γ'' long axis was 100 nm size were previously determined [16]. Therefore, we recommend an image resolution of at least 24 pixels/ γ' diameter for accurate CNN model predictions.

In general, the benefits and disadvantages of the different techniques are worth exploring so that one may decide the

Fig. 13 Three relative scales of contrast, brightness (0.1, 0.5, 0.9) and resolutions (9, 18, 72 pixels/inch) are presented. The bolded values are the inherent IN718Net image conditions

best model choice for future specific tasks. Table 3. Performance summary of the four approaches listed in increasing accuracy rates shows the performance metrics of the different approaches with regard to processing time, but it neglects to account for the human set-up time of each approach. For example, it would be inconceivable to apply a “from-scratch” CNN solution when a researcher needs to quantify two or even ten micrographs; in this situation, manual interpretation, which can require hours of time, would be the most-time efficient option (Though now that these approaches are incorporated into a singularity, the next researcher could use this CNN). When the number of micrographs increases to 20 or 30, the standard mathematical image processing techniques become a viable solution if issues like lighting variability

Fig. 14 The graphs illustrate the accuracy of the CNN model at predicting pixel phases against the **a** relative scale of brightness and contrast and **b** resolution

Table 3 Performance summary of the four approaches listed in increasing accuracy rates

Approaches	Compute	Running time (seconds)	Accuracy (%)
SVM	CPU	122	91
Non-machine learning algorithm	CPU	67	94
RF	CPU	4	94
CNN	GPU	1	95

and contrast can be controlled because the time needed to make an algorithm might balance the time needed to manually interpret the images. At 50+ micrographs, the computational time needed for the standard mathematical image processing technique become prohibitive for quality control applications (i.e., 50 image, 122 s/image, 1.7 h of computational time on an HPC). At this point, the CNN models can be trained with transfer learning approaches for different training sets and can be applied to similar problem domains (i.e., phase identification in grayscale SEM images). In the implementation phase, machine learning approaches are less challenging than the non-machine learning algorithm due to the fast developing ecosystem. The main advantage of the non-machine learning algorithms is that it does not require a large data set likely saving time and reducing the cost of collecting and manual labeling data.

Conclusion

In this work, we have presented four computer vision approaches, one based on conventional, mathematical, non-machine learning algorithms and three based on machine

learning approaches. The resulting algorithms result in up to 95% accurately at segmenting phases in IN718 SEM images by using the *IN718Net* dataset. The CNN model was shown to have robust accuracy under a range of realistic variations in imaging parameters that might be observed. The usage of singularity containers presented here facilitates deployment of these models in different environments by avoiding the system dependencies. The comparison of optimizing and satisfying metric (Table 3) shows that the CNN and RF have similar accuracy with less running time than the non-machine learning algorithm, but the interpretation of the micrographs indicates that RF predictions are inconsistent with the kinetics of precipitation. It can be concluded that the SVM is not the best classifier for this task because of the higher time consumption and lower accuracy.

Most of the python libraries that are used to develop non-machine learning and traditional machine models are not compatible with graphical processing units (GPU) because this limits the speed of training and running of the algorithms. The libraries that were used to implement CNN are compatible to run in GPUs. When we compare the accuracy, speed, hardware infrastructure and the developing ecosystem, the CNN approach is the most promising solution for future scientific image analysis in IN718.

Acknowledgements This work was supported by the National Science Foundation under Grant No 1152716. SEM graphs of IN718 material were provided by NASA Glenn Research Center. High Performance Computing (HPC) clusters at CWRU were accessed for model training, testing, and deployment. Technical support for implementation of the singularity containerize environment was provided by SDLE center at CWRU. The code required to reproduce these findings is available to download from <https://github.com/CWRU-MSL/GammaDoublePrime> and/or from the corresponding author upon request.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing,

adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Strondl A, Palm M, Gnauk J, Frommeyer G (2011) Microstructure and mechanical properties of nickel based superalloy IN718 produced by rapid prototyping with electron beam melting (EBM). *Mater Sci Technol* 27(5):876–883
- Amato KN, Gaytan SM, Murr LE, Martinez E, Shindo PW, Hernandez J, Medina F (2012) Microstructures and mechanical behavior of Inconel 718 fabricated by selective laser melting. *Acta Mater* 60(5):2229–2239
- Cozar R, Pineau A (1973) Morphology of γ' and γ'' precipitates and thermal stability of inconel 718 type alloys. *Metall Transac* 4(1):47–59
- Brooks JW, Bridges PJ (1988) Metallurgical stability of Inconel alloy 718. *Superalloys* 88:33–42
- Oblak JM, Paulonis DF, Duvall DS (1974) Coherency strengthening in Ni base alloys hardened by DO 22 γ' precipitates. *Metall Transac* 5:1–143
- Senanayake NM, Mukhopadhyay S, Carter JLW (2020) High-throughput approaches to establish quantitative process–structure–property correlations in Ni-base superalloys. In: *Superalloys 2020*. Springer, Cham. https://doi.org/10.1007/978-3-030-51834-9_66
- Kulawik K, Buffat PA, Kruk A, Wusatowska-Sarnek AM, Czyrska-Filemonowicz A (2015) Imaging and characterization of γ' and γ'' nanoparticles in Inconel 718 by EDX elemental mapping and FIB–SEM tomography. *Mater Charact* 100:74–80
- Dubiel B, Kruk A, Stepniowska E, Cempura G, Geiger D, Formanek P, Czyrska-Filemonowicz A (2009) TEM, HRTEM, electron holography and electron tomography studies of γ' and γ'' nanoparticles in Inconel 718 superalloy. *J Microsc* 236(2):149–157
- Xie X, Dong J, Wang G, You W, Du J, Zhao C, Loria E (2005) The effect of Nb, Ti, Al on precipitation and strengthening behavior of 718 type superalloys. *Superalloys* 718:625–706
- Sarosi PM, Viswanathan GB, Whitis D, Mills MJ (2005) Imaging and characterization of fine γ' precipitates in a commercial nickel-base superalloy. *Ultramicroscopy* 103(1):83–93
- Semiatin SL, Kim SL, Zhang F, Tiley JS (2015) An investigation of high-temperature precipitation in powder-metallurgy, gamma/gamma-prime nickel-base superalloys. *Metall Mater Transac A* 46(4):1715–1730
- Gabb TP et al. (2016) Comparison of γ – γ' phase coarsening responses of three powder metal disk superalloys. p. 44
- Smith TM, Bonacuse P, Sosa J, Kulis M, Evans L (2018) A quantifiable and automated volume fraction characterization technique for secondary and tertiary γ' precipitates in Ni-based superalloys. *Mater Charact* 140:86–94
- Blavette D, Cadel E, Deconihout B (2000) The role of the atom probe in the study of nickel-based superalloys. *Mater Charact* 44(1–2):133–157. [https://doi.org/10.1016/S1044-5803\(99\)00050-9](https://doi.org/10.1016/S1044-5803(99)00050-9)
- Phillips PJ, McAllister D, Gao Y, Lv D, Williams REA, Peterson B, Mills MJ (2012) Nano γ'/γ'' composite precipitates in Alloy 718. *Appl Phys Lett* 100(21):211913
- Smith TM, Senanayake NM, Sudbrack CK, Bonacuse P, Rogers RB, Chao P, Carter J (2019) Characterization of nanoscale precipitates in superalloy 718 using high resolution SEM imaging. *Mater Charact* 148:178–187
- Carter JL, Verma AK, Senanayake NM (2020) Harnessing legacy data to educate data-enabled structural materials engineers. *MRS Adv* 5(7):319–327
- Dong H, Yang G, Liu F, Mo Y, Guo Y (2017) Automatic brain tumor detection and segmentation using U-Net based fully convolutional networks. *Annual conference on medical image understanding and analysis*. Springer, Cham, pp 506–517
- Wen S, Kurc TM, Hou L, Saltz JH, Gupta RR, Batiste R, Zhu W (2018) Comparison of different classifiers with active learning to support quality control in nucleus segmentation in pathology images. *AMIA Summits Transl Sci Proceed* 2018:227
- Rodriguez-Galiano V, Sanchez-Castillo M, Chica-Olmo M, Chica-Rivas MJOG (2015) Machine learning predictive models for mineral prospectivity: an evaluation of neural networks, random forest, regression trees and support vector machines. *Ore Geol Rev* 71:804–818
- Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32
- Archer KJ, Kimes RV (2008) Empirical characterization of random forest variable importance measures. *Comput Stat Data Anal* 52(4):2249–2260
- Suykens JA, Vandewalle J (1999) Least squares support vector machine classifiers. *Neural Process Lett* 9(3):293–300
- Ma J, Sheridan RP, Liaw A, Dahl GE, Svetnik V (2015) Deep neural nets as a method for quantitative structure–activity relationships. *J Chem Inf Model* 55(2):263–274
- Simard PY, Steinkraus D, Platt JC (2003) Best practices for convolutional neural networks applied to visual document analysis. *Icdar* 3:2003
- Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. *International conference on medical image computing and computer-assisted intervention*. Springer, Cham, pp 234–241
- Tan C, Sun F, Kong T, Zhang W, Yang C, Liu C (2018) A survey on deep transfer learning. *International conference on artificial neural networks*. Springer, Cham, pp 270–279
- Tang A, Tam R, Cadrin-Chênevert A, Guest W, Chong J, Barfett J, Poudrette MG (2018) Canadian association of radiologists white paper on artificial intelligence in radiology. *Can Assoc Radiol J* 69(2):120–135
- Li W, Chen C, Zhang M, Li H, Du Q (2018) Data augmentation for hyperspectral image classification with deep CNN IEEE. *Geosci Remote Sens Lett* 16(4):593–597
- Zafari S, Eerola T, Sampo J, Kälviäinen H, Haario H (2015) Segmentation of overlapping elliptical objects in silhouette images IEEE. *Trans Image Process* 24(12):5942–5952
- Loy G, Zelinsky A (2003) Fast radial symmetry for detecting points of interest IEEE. *Trans Pattern Anal Mach Intell* 25(8):959–973
- Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Vanderplas J (2011) Scikit-learn: machine learning in Python. *J Mach Learn Res* 12:2825–2830
- Culjak I, Abram D, Pribanic T, Dzapo H, Cifrek M (2012) A brief introduction to OpenCV. In: *2012 proceedings of the 35th international convention MIPRO* (pp. 1725–1730). IEEE
- Souza A, Oliveira LB, Hollatz S, Feldman M, Olukotun K, Holton JM, Nardi L (2019) Deepfreak: learning crystallography diffraction patterns with automated machine learning. *arXiv preprint arXiv 1904.11834*

35. Kamarainen JK (2012) Gabor features in image analysis. In: 2012 3rd international conference on image processing theory, tools and applications (IPTA) (pp. 13–14). IEEE
36. Choras RS (2007) Image feature extraction techniques and their applications for CBIR and biometrics systems. *Int J Biol Biomed Eng* 1(1):6–16
37. Alkaabi S, Deravi F (2004) Candidate pruning for fast corner detection. *Electron Lett* 40(1):18–19
38. Rublee E, Rabaud V, Konolige K, Bradski G (2011) ORB: An efficient alternative to SIFT or SURF. In: 2011 International conference on computer vision (pp. 2564–2571). IEEE
39. Krig S (2016) Computer vision metrics. Springer, Berlin, Germany, pp 187–246
40. Paper D, Paper D (2020) Scikit-learn classifier tuning from simple training sets. *Hands-on Scikit-learn for machine learning applications: data science fundamentals with Python*. Apress, Berkeley, pp 137–163
41. Acharjya PP, Das R, Ghoshal D (2012) Study and comparison of different edge detectors for image segmentation. *Glob J Comput Sci Technol*
42. Kramer O (2016) Scikit-learn. *Machine learning for evolution strategies*. Springer, Cham, pp 45–53
43. Targ S, Almeida D, Lyman K (2016) Resnet in resnet: generalizing residual architectures. *arXiv preprint arXiv 1603:08029*
44. Howard J, Gugger S (2020) Fastai: a layered API for deep learning. *Information* 11:2–108
45. Cortes C, Mohri M, Rostamizadeh A (2012) L2 regularization for learning kernels. *arXiv Preprint arXiv 1205:2653*
46. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15(1):1929–1958
47. Rognvaldsson TS (1998) A simple trick for estimating the weight decay parameter. *Neural networks: tricks of the trade* Springer, Berlin, Heidelberg, pp 71–92
48. Zhang Z, Sabuncu M (2018) Generalized cross entropy loss for training deep neural networks with noisy labels. *Adv Neural Inf Process Syst* 31:8778–8788. <https://papers.nips.cc/paper/2018/file/f2925f97bc13ad2852a7a551802feea0-Paper.pdf>
49. Goodfellow I, Bengio Y, Courville A (2016) Deep feedforward networks. *Deep learning*. 168–227
50. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. *arXiv Preprint arXiv 1412:6980*
51. Tieleman T, Hinton G (2012) Lecture magnitude. *Neural Netw Mach Learn* 4(2):26–31
52. Agarap AF (2018) Deep learning using rectified linear units (relu). *arXiv Preprint arXiv 1803:08375*
53. Binguitcha-Fare AA, Sharma P (2019) Crops and weeds classification using convolutional neural networks via optimization of transfer learning parameters. *Int J Eng Adv Technol (IJEAT)* 8(5):2249–8958
54. Bickel S, Brückner M, Scheffer T (2007) Discriminative learning for differing training and test distributions. In: *proceedings of the 24th international conference on Machine learning*, pp 81–88
55. Smith LN (2018) A disciplined approach to neural network hyper-parameters: part 1-learning rate, batch size, momentum, and weight decay. *arXiv Preprint arXiv 1803:09820*
56. Kurtzer GM, Sochat V, Bauer MW (2017) Singularity: scientific containers for mobility of computers. *PLoS One* 12(5):e0177459
57. Clark A (2015) Pillow (PIL fork) documentation