



Overview: Computer Vision and Machine Learning for Microstructural Characterization and Analysis

ELIZABETH A. HOLM, RYAN COHN, NAN GAO, ANDREW R. KITAHARA, THOMAS P. MATSON, BO LEI, and SRUJANA RAO YARASI

Microstructural characterization and analysis is the foundation of microstructural science, connecting materials structure to composition, process history, and properties. Microstructural quantification traditionally involves a human deciding what to measure and then devising a method for doing so. However, recent advances in computer vision (CV) and machine learning (ML) offer new approaches for extracting information from microstructural images. This overview surveys CV methods for numerically encoding the visual information contained in a microstructural image using either feature-based representations or convolutional neural network (CNN) layers, which then provides input to supervised or unsupervised ML algorithms that find associations and trends in the high-dimensional image representation. CV/ML systems for microstructural characterization and analysis span the taxonomy of image analysis tasks, including image classification, semantic segmentation, object detection, and instance segmentation. These tools enable new approaches to microstructural analysis, including the development of new, rich visual metrics and the discovery of processing-microstructure-property relationships.

<https://doi.org/10.1007/s11661-020-06008-4>

© The Minerals, Metals & Materials Society and ASM International 2020

I. INTRODUCTION: THE QUANTIFICATION OF MICROSTRUCTURE

IN 1863, the geologist Henry Clifton Sorby examined acid-etched and polished steel under a microscope and observed a complex collection of substructures that we now call microstructure.^[1] Over the next two decades, Sorby related these visual entities to the chemistry, history, and behavior of various steel alloys, making the first connections between materials structure, composition, processing, and properties.^[2] Sorby's observations were necessarily qualitative; for example, he wrote, "There is also often great variation in the size of the crystals [grains] of iron...and one cannot but suspect that such great irregularities might be the cause of the fracture..."^[2] However, by the early 1900s, methods for

measuring microstructural features had been developed, and in 1916 the first ASTM metallographic standard E 2 to 17T included planimetric grain size measurement.^[3]

Throughout the 20th century, metallurgists and materials scientists continued to create and refine an ever-growing catalog of microstructural metrics.^[6] The field of quantitative stereology linked two-dimensional (2D) cross-sectional structures to their true three-dimensional (3D) counterparts,^[7] techniques such as electron backscatter diffraction (EBSD) microscopy were especially amenable to digital analysis, and 3D microstructural representations became feasible, if not routine, with the advent of 3D X-ray diffraction microscopy (3DXRD).^[8,9] Despite the sustained progress in microstructural science, however, an overarching microstructural metric remained elusive. Instead, each individual metric was developed and applied on a case-by-case basis. The reason for this is simple: The visual phase space of microstructural images is immense, as illustrated in Figure 1. This leads to the two fundamental problems of quantitative microstructural science:

1. What to measure: For any given microstructure, it is not always self-evident which metric is most strongly related to the process or property of interest. Consider the Hall-Petch relationship $\sigma_y = kd^{1/2}$, which directly links average grain

ELIZABETH A. HOLM, RYAN COHN, NAN GAO, ANDREW R. KITAHARA, BO LEI, and SRUJANA RAO YARASI are with the Department of Materials Science and Engineering, Carnegie Mellon University, Pittsburgh, PA 15213. Contact e-mail: eaholm@andrew.cmu.edu THOMAS P. MATSON is with the Department of Materials Science and Engineering, Carnegie Mellon University and also with the Department of Materials Science and Engineering, MIT, Cambridge, MA 02139.

Manuscript submitted May 3, 2020.

Article published online September 29, 2020

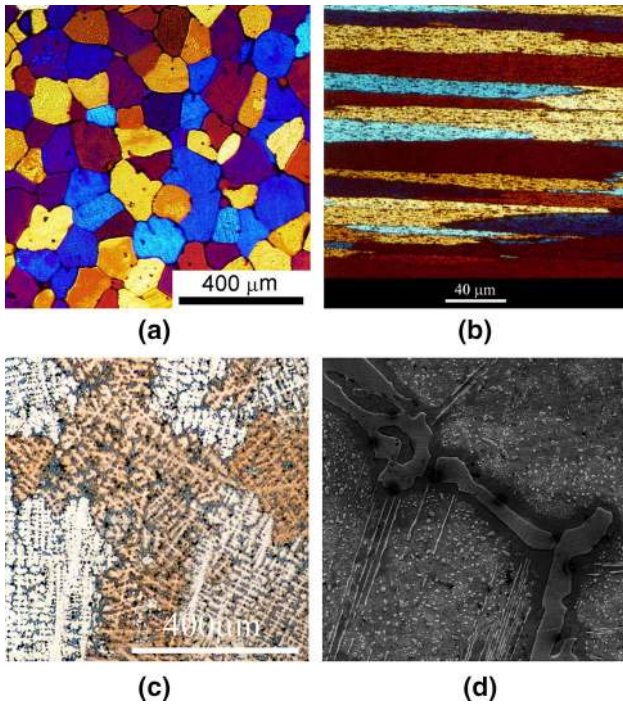


Fig. 1—Microstructural diversity. (a) Equiaxed grains in an Al alloy seen in polarized light. (b) Elongated grains in extruded Al, polarized light micrograph. (c) Dendritic colonies in cast bronze, true color light micrograph. [Figures (a) through (c) reprinted from [4] under the terms of Creative Commons CC BY-NC-SA 2.0 UK license.] (d) Carbide structure in ultrahigh carbon steel, SEM image.^[5]

diameter d with yield strength σ_y via a proportionality constant k .^[10,11] In an equiaxed, single-phase polycrystal such as Figure 1(a), it is a simple matter to measure d . But if the grains are elongated as in Figure 1(b), is d still the relevant metric, or is the grain aspect ratio more important?

As microstructures become more complex, so does metric selection. Which is the most influential scale in Figure 1(c): dendrite arm spacing, dendrite colony size, or porosity distribution? Of course, there is the very real possibility that multiple metrics may arise from a given process history or contribute to a given property. Ultimately, deciding what microstructural feature(s) to measure requires human judgment informed by knowledge and experience.

2. How to measure it: While some microstructural metrics are relatively straightforward to measure, others are not. Consider the steel microstructure in Figure 1(d). A feature that affects this alloy's mechanical properties is the size of the carbon denuded zone (the dark area bordering the network carbides). Although it is simple in principle to measure its average dimension, it requires locating the boundaries of the denuded zone *via* segmentation. (Segmentation is defined as determining feature membership for each pixel in an image). Because the denuded zone boundary is fuzzy, segmentation is problematic and usually is left to

the subjective judgment of a human expert. The outcome can be uncertain or even irreproducible results.

Segmentation issues may affect even apparently uncomplicated measurements. For example, in polycrystalline microstructures like Figure 1(a), some grain boundaries may not etch deeply enough to create a dark line in an optical image. While the human eye is able to locate these “missing boundaries” based on visual clues, automating this process has remained stubbornly intractable.^[12,13] Thus, even the simple metric of grain size can harbor complex measurement challenges.

Even now, 150 years after Sorby noted the iridescent sheen of “the pearly constituent” (pearlite), microstructural analysis often relies on qualitative, rather than quantitative, descriptions. Terms like equiaxed, aligned, rough, dispersed, ordered, columnar, *etc.* are often used without quantification.

Computer vision (CV) is the computer science field that focuses on quantifying the visual information content of digital images.^[14] A digital image is a numeric representation where each pixel has an integer gray value or a short vector color value (*i.e.*, RGB) associated with it; the individual pixels provide little information about the content of the image. The goal of CV is to aggregate pixels to represent an image as a high-dimensional tensor of visual information. While CV may be familiar from facial recognition and self-driving cars, it is also used in applications that are strikingly similar to some microstructural quantification tasks, such as determining how many people appear in a picture of a crowd^[15] or what proportion of a satellite photograph is comprised of farmland.^[16]

In order to extract abstract information from the image representation produced by a CV scheme, we can use methods for finding correlations in high-dimensional data space, many of which fall under the general classification of machine learning (ML).^[17] ML approaches may be supervised or unsupervised. Supervised ML involves training a system based on human-determined ground-truths. For example, given a set of photographs with metadata noting the presence or absence of a cat, a supervised ML system can learn to identify images of cats. Unsupervised learning algorithms find relationships between image representations without ground-truth data or human intervention, typically by generating clusters of related images. These approaches are complementary, and each is applicable to different problems.

A number of recent studies have applied CV and ML approaches to develop a more general approach to quantitative microstructural analysis.^[18–32] This is a promising direction for creating tools that capture rich and complete microstructural information (what to measure) in a quantitative, objective, and general manner (how to measure it). In this paper, we present applications of CV and ML for a variety of microstructural image analysis tasks.^[33–38] We show that the CV/ML approach can assist, improve, or even replace traditional, *ad hoc* microstructural characterization methods.

II. METHODS

Accurate microstructural measurements are fundamental to making structure-processing-property connections. However, there are many microstructural analysis tasks that require human judgment and so have an element of subjectivity that makes them difficult to automate and susceptible to bias. Because artificial intelligence methods, including CV and ML, can learn to replicate human visual judgments, these tools are good candidates to perform these tasks in an objective, autonomous, and efficient manner.

The fields of CV and ML are rapidly evolving. There is no reason to believe that the particular methods presented here are optimal or even will remain competitive as new methods are developed. However, we expect that the concepts and approaches of CV and ML will continue to be valuable tools for microstructural science. For that reason, this section will present the foundational bases of CV and ML, which are common across methods. Details of the specific computational codes used in the examples are available in the supplementary material, Table SI.

A. Computer Vision: Create a Numerical Representation

Computer vision encompasses an array of methods for creating a numerical representation of a visual image, termed the feature vector; most of these methods are optimized for specific tasks (facial recognition, object identification, texture analysis, *etc.*).^[14,39–57] Across application domains, however, there are two basic approaches to CV: feature-based representations and representations based on convolutional neural networks (CNNs). Feature-based methods create an image representation that is in essence a statistical representation of the visual features in the image.^[34,35,37,38,57,58] Filters that activate when they encounter a feature^[46,49] (typically an edge, corner, or blob) are applied to the original image. Each feature is then numerically encoded with a descriptor,^[43,49] and the image representation is some aggregation of the feature descriptors.^[59–62] CNNs also use filter activations as visual features.^[63,64] The primary difference between feature-based and CNN image representations is that while the filters for feature-based methods are selected by human experts, the CNN filters are learned during the training and optimization of the CNN. Moreover, in typical CNN tasks, the filter activations are not abstracted into an image representation; instead, they are used directly as the feature map for the image.^[65] The result is that, since the advent of the AlexNet CNN in 2012,^[66] CNNs have outperformed feature-based methods (and often human experts^[67]) for essentially all CV tasks.^[68] Thus, we will restrict our discussion to CNN-based feature vectors.

The fundamental objective of CV is to represent the visual content of an image in numerical form, and there are numerous methods to accomplish this *via* CNNs. We will focus on two methods that are particularly suitable for microstructural images: CNN layers and hypercolumn pixels.

Convolutional neural networks take an image (or image-like data) as input, apply a variety of signal processing operations to it in order to encode it as a vector, and then utilize an artificial neural network^[68–70] or other ML method to draw a conclusion about the visual content of the image. The first part of the CNN pipeline—encoding the image as a feature vector—is termed the feature learning stage, and the second part—drawing a conclusion—is the classification stage.

In the feature learning stage, shown in Figure 2, the CNN rasters (convolves) sets of filter patches pixel-by-pixel and records the filter activation values. It does this hierarchically, first to the image (generating the first convolution layer) and then to the subsequent layers (generating the second through n th convolution layers). After several convolutions, activations are rectified (typically *via* a rectified linear unit, ReLU, filter that converts negative values to zero). Pooling (downsampling) then combines multiple pixels in one layer into a single pixel in the next layer, and another set of convolutions are performed. After a number of convolution and pooling iterations, the final layer (the fully connected layer) is flattened (written as a vector), forming the representation used for decision-making in the classification stage.^[65,70,71] It is significant to note that the convolution process results in CNN features that are translation invariant but are not rotation or scale invariant. This means that, for example, a feature may require multiple representations to capture its various geometric orientations within a microstructure.

Designing and training a CNN requires deep expertise and a large data set (typically millions of images), making it impractical for most microstructural data sets. However, CNNs that have been optimized and trained on a large set of natural images have been successfully used with other kinds of images, including microstructures. This transferability of results is likely due to the fact that images of very different things share common visual features, including edges, blobs, and visual textures. Thus, transfer learning^[59] enables us to use pre-trained CNNs (such as the VGG16 network^[65] trained on the ImageNet data set^[72]) for microstructural representation. However, since we are not interested in classifying microstructural images into the ImageNet categories (broccoli, bucket, bassoon, *etc.*), we truncate the network before the classification stage. Instead, we use the CNN layers themselves as the image representations for ML tasks, as shown in Figure 2, often using a dimensionality-reducing encoding such as principal component analysis (PCA)^[73] or vector of locally aggregated descriptors (VLAD)^[62] to decrease the length of the feature vector for efficient computing.^[34]

Which layer of a CNN representation is the best choice for a feature vector depends on the characteristics of the micrographs.^[74] Because pooling operations cause pixels in deep layers to represent large areas of the original image, deep layers capture features at large length scales. Conversely, filter activations in shallow layers represent local environments. Thus, visually simple micrographs, like the equiaxed polycrystal in Figure 1(a), may be better represented by shallow layers, and complex microstructures, such as the carbide structure in Figure 1(d), may be

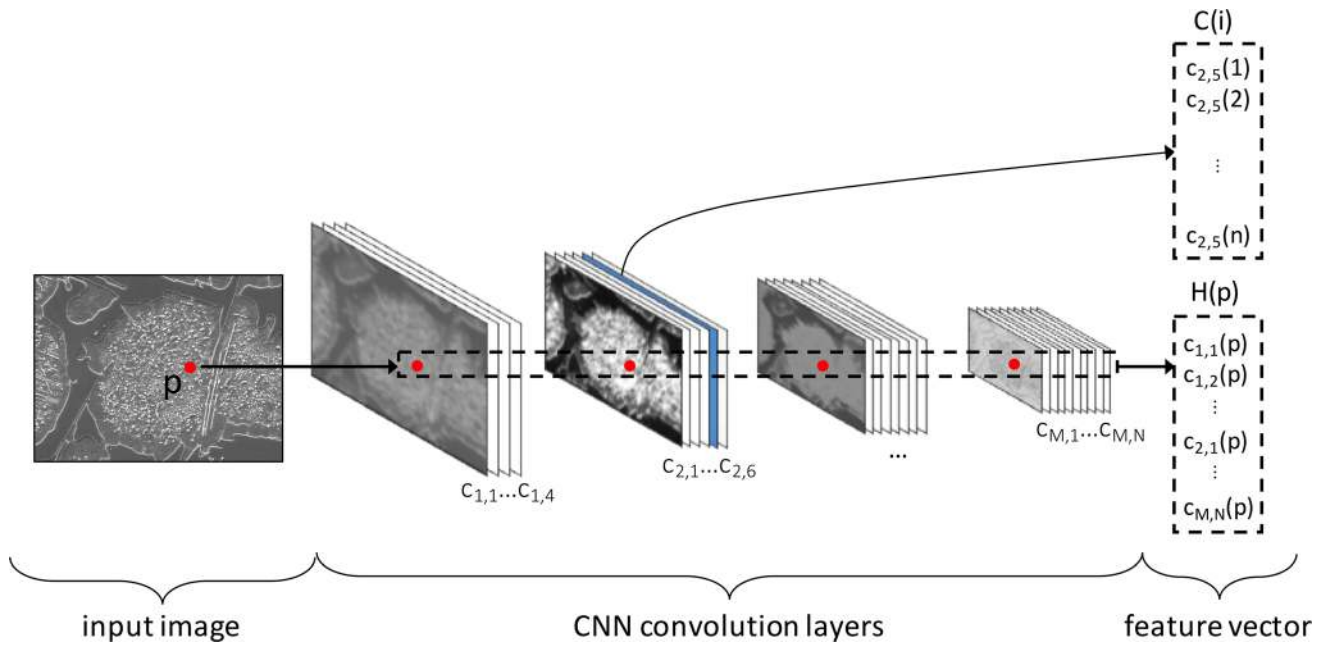


Fig. 2—The feature learning stage of a CNN. The input image i (left) is processed through layers $c_{1,1}$ through $c_{M,N}$. Pooling decreases image resolution between groups (or blocks) of layers. A feature vector for the image as a whole, $C(i)$, may be constructed by flattening any of the layers (here, the $c_{2,5}$ convolution layer shown in blue). Alternatively, a feature vector for each pixel p may be assembled from filter activations in all the layers, giving the hypercolumn pixel vector, $H(p)$ (Color figure online).

better represented by deep layers, which capture multi-scale structures more completely. However, there is not yet a mechanism for determining the optimum layer to represent a given type of microstructure *a priori*; instead, the decision is made by trial and error.^[34,74]

For a more complete image representation, we might wish to combine the information from all of the convolution layers; this is the basis for the hypercolumn pixel feature vector,^[40] as shown in Figure 2. We begin with an image and its convolution layers in a pre-trained CNN. For each of the selected pixels in the original image, the hypercolumn is built by stacking the activations of the convolution layers at the pixel's real-space position; the result is an image representation where each hypercolumn stores information about the pixel's feature membership at multiple length scales. Representing an image by the hypercolumns of every pixel is memory intensive, so typically we select a sparse subset of pixels. Hypercolumn features are primarily used for image segmentation tasks, which we discuss below.^[75]

Finally, it is worth noting that CNNs can operate in reverse (*i.e.*, with transposed convolutions), generating images from feature vectors. This is the basis for variational autoencoders (VAEs)^[76] and generative adversarial networks (GANs).^[77] These methods may be used to generate realistic synthetic images^[78–80] and to enhance image resolution,^[81] among other applications.

B. Machine Learning: Extract Quantitative Visual Information

The goal of ML is to extract quantitative visual information from the high-dimensional feature vector. This information might comprise a classification (*e.g.*,

ferritic, austenitic, martensitic, *etc.*), an association with a metadata value (*e.g.*, yield strength), a measurement (*e.g.*, grain size), presence of a particular feature (*e.g.*, surface defect), or any other value that might be contained in the feature vector. ML methods are either supervised (trained using known correct answers, termed ground truth) or unsupervised (finding patterns without knowledge of a ground truth), and there are important roles for each approach.

There is a wide array of supervised ML methods,^[68,69,82] and the choice of method depends on the application. Supervised ML systems that are widely used for visual image feature vector data include support vectors machines (SVM),^[83,84] random forest (RF)^[85] classifiers, and deep learning methods such as artificial neural networks (ANNs).^[65,68–71] SVMs operate by learning the set of hyperplanes that best separates feature vectors into groups according to their ground truth type or class. Once the separating planes are known, additional vectors can be associated with the appropriate group, *i.e.*, classified. Advantages of SVM are flexibility, generality, and performance. However, the success of an SVM model depends on whether the high-dimensional data structure is amenable to planar separation.

RF classifiers begin by constructing a set of decision trees to predict the class of an image; within each tree, decisions are based on values of feature vector elements. To classify the image, the trees each offer a prediction, and majority rules. In the training phase of an RF classifier, the feature vector elements that populate the trees and their decision values are optimized to give the best match with the known ground truth. Once the trees are optimized, they can be applied to classify additional

vectors. An advantage of RFs is interpretability, since the basis for decision is easily verified.^[86] However, RFs are not always easy to apply to complex image representations.

ANNs process the feature vector through the hidden layers of a neural network^[82] in order to make a prediction regarding the image. The structure and connectivity of the ANN can take many forms, and its architecture is selected to maximize performance, often by trial and error. During the learning process, the weights of the connections between neurons in the input, hidden, and output layers are optimized to give the best match with the known ground truth. Once the weights have been determined, the ANN can make predictions about previously unseen vectors. ANNs as a class are immensely flexible and scale to handle both large amounts of data and very high-dimensional data. However, they are black box models, where the basis for decision can be difficult to parse.

Unsupervised ML algorithms find relationships between image representations without ground-truth data or human intervention, typically by generating clusters of related images. *k*-means is one example of an unsupervised clustering method.^[87] For a set of *N*-dimensional feature vectors, *k*-means groups them into a user-specified number of *N*-dimensional clusters that minimize a cost function that is some measure of “cluster goodness,” such as cluster compactness. Finding a globally optimal clustering is an np hard computational problem (*i.e.*, requires exponential computing time), so *k*-means uses various computing strategies to find good solutions. An implication of this is that for a given set of vectors, *k*-means results may vary depending on computational parameters. An advantage of *k*-means clustering is that because it identifies a set of cluster centroids, additional vectors can be associated with clusters straightforwardly. Thus, it can be used as a basis for classification. A disadvantage is that, as a full *N*-dimensional representation, it may not be easy to visualize the results in 2D or 3D.

There are a number of tools to visualize high-dimensional image representations in two or three dimensions in order to look for clusters based on image similarity.^[88] In the examples to follow, we use t-distributed stochastic neighbor embedding (t-SNE),^[89,90] which weights image similarity on a nonlinear scale that diminishes quickly as image similarity decreases; thus t-SNE highly favors grouping similar images but does not capture relationships between dissimilar images. An advantage of t-SNE is that it often is able to discover and visualize low-dimensional clusters. A disadvantage is that since t-SNE is constructed based on pairwise comparisons of feature vectors, it is not possible to add additional data to a t-SNE map without recomputing it entirely, so t-SNE cannot be used for classification.

The choice of ML modality and model depends on the nature of the input data and the desired outcome. Often, multiple approaches are attempted and evaluated for performance and/or other advantages and disadvantages. In this process, it is helpful to include a domain expert in ML algorithms, since the best-in-class solutions are ever-evolving.

C. Data: The Basis for Data Science

Perhaps the greatest challenge to the creation of CV/ML approach for microstructural science is the need for microstructural image data suitable for training and utilizing such systems. Data size is usually assumed to be the limiting factor, and in some cases it is. However, as discussed below, we have achieved excellent results in some complex image analysis tasks with very small numbers of original micrographs (sometimes fewer than 10). We believe that this has to do with the data-richness of microstructural images compared to the natural images used in typical CV studies. For instance, where a natural image might contain a cat, a micrograph might contain 100 precipitates. In fact, microscopists strive to capture microstructural images that are statistically representative of the material they portray. Furthermore, the spatial relationships and length scales implicit in a microstructural image are physically-based and meaningful. Finally, the entire image typically constitutes the microstructure; there is no “meaningless” background to subtract or ignore. Thus, microstructural images tend to be rich in relevant information.

Similarly, concerns about image quality are often unfounded. ML methods learn to extract the information contained in the data as it is presented to them; there are no judgments about the quality of focus, resolution, field of view, magnification, *etc.* In fact, exposing the ML system to the full scope of the data space results in a more robust outcome. This is not a recommendation to ignore good microscopy practices, but rather a suggestion not to assume that images must be vetted or weeded out based on human standards of quality.

Data collection practices that do increase the performance of CV/ML systems include image redundancy, standardization, and augmentation. Taking redundant images of a sample with non-overlapping fields of view increases the amount of visual data for a relatively low additional cost, compared to taking one “representative” image. This allows the ML system to learn the full data space, while avoiding overfitting and human bias. Standardizing imaging conditions (such as instrument, settings, magnification, orientation, *etc.*) and/or performing judicious image preprocessing helps prevent the CV/ML system from learning the wrong thing. That said, there are data sets in which imaging differences are inevitable; in those cases, verifying that the system is learning the relevant information is paramount.^[37] Data augmentation involves manipulating existing images, usually *via* subsampling, translation, rotation, or affine transformation, to create additional image patches for training and testing.^[91,92] For example, for micrographs that do not possess a standard orientation, augmenting the original data with rotated versions can more fully sample the visual data space represented by the data set, potentially improving ML outcomes. It is important to recognize that all of the physical data is contained in the original images; additional images generated by data augmentation methods may facilitate the training of the models, but they do not create new information.

The most valuable microstructural data sets include metadata that enriches their information content. Metadata may include multiple imaging modalities (*e.g.*, EBSD and backscatter data for the same field of view), as well as information on material system, composition, imaging information, processing history, property measurements, and any other data available related to the image. Capitalizing on multimodal information and metadata is an emerging area of computer vision. Now, more than ever, aggregating and cross-referencing data from multiple sources is the key to discovery.

III. RESULTS: A TAXONOMY OF MICROSTRUCTURAL ANALYSIS

From a given image, a microstructural scientist may extract different kinds of information from various visual signals. This information can be arranged in a hierarchy, or taxonomy, of microstructural analysis tasks, which includes:

- Image classification—identifying the content of an image
- Semantic segmentation—associating each pixel in an image with a constituent of the image
- Object detection—locating individual objects in an image
- Instance segmentation—assigning pixels to individual objects

Figure 3 shows this microstructural taxonomy applied to a scanning electron microscopy (SEM) micrograph where sample preparation yielded an image of dense, overlapping metal powder particles. Image classification identifies the image as belonging to the ‘powder’ class. Semantic segmentation associates pixels with either powder particles or background and so cannot distinguish individual particles when particles overlap. Object detection finds each individual powder particle, and instance segmentation assigns pixels to particles. In the following sections, we will demonstrate and discuss the CV and ML applications for tasks in each component of the microstructural analysis taxonomy.

A. Image Classification and the Feature Vector

Image classification may not seem important, since we usually know what our microstructures are. However, classification of images underlies a host of critical archiving and analysis tasks. A CV approach to image classification begins with defining a feature vector, as discussed above.

Since the feature vector numerically encodes the visual information contained in an image, the visual similarity between two images should be related to the numerical similarity between their feature vectors. Thus, a quantitative measure of overall image similarity can be defined as the distance between feature vectors; there are multiple distance metrics, but an easily computed and interpreted one is the Euclidean distance (L^2 norm). The

visual similarity metric then forms the basis for visual search, visual clustering, and classification. (However, note that because CNN representations are not rotation or scale invariant, the distance metric between two particular features may not be unique.) For example, feature vectors were computed using a CNN layer representation for a database of 961 microstructures of ultrahigh carbon steels.^[5] Figure 4(a) shows the three images with feature vectors closest to that of a given target image; clearly, feature vector similarity is reflected in visual similarity. Notably, while the first two matches contain microconstituents of similar size, fraction, contrast, and structure to the target image, the third is the target image at a lower magnification, demonstrating that the visual search encompasses multiple aspects of visual similarity. Figure 4(b) shows a t-SNE image cluster map where each point represents an image, and the distance between points scales with the distance between feature vectors; point color corresponds to the primary microstructural constituent in each micrograph. Clearly, similar images cluster, which illustrates the visual structure of the data set. Thus, the feature vector enables the primary database functions of search and sort to be performed based on visual, rather than language-based, information.

The fact that the visual clusters in Figure 4(b) correspond to microstructural constituents implies that the feature vector can be utilized to detect the presence of specific microstructural features. In fact, a linear support vector machine (SVM) was trained to classify the primary microconstituent in each image in the UHCS database (supervised machine learning), achieving 99 ± 1 pct accuracy (defined as fraction of correct classifications).^[5] Other studies have used unsupervised and supervised ML to detect and classify defects,^[93–95] microstructural constituents,^[96] atomic structures,^[97] and damage.^[98] Thus, image classification contributes to a wide variety of image processing tasks including image analysis, keyword identification, and quality control.

While the feature vector is often used for qualitative tasks, it is worth remembering that it contains information about pixel membership in the various microstructural features and as such can potentially be used to quantify structural metrics directly, without image segmentation. To do so, we need a data set sufficiently large to train an ML system and with unambiguous ground-truth values of the microstructural metric to be measured. For example, to address the problem of missing boundaries in grain size measurement,^[12,13] we generated 15,213 synthetic optical microstructures of pure isotropic polycrystals with known grain sizes, using the SPPARKS kinetic Monte Carlo grain growth simulation code.^[99] (While synthetic images do not capture the full richness of actual micrographs, they do permit the development of large data sets. This is also a good opportunity for automated image collection systems.) These were used to train a CNN system^[65,100] with a fully connected regression layer^[101] to learn grain size, both for perfect structures and for polycrystals where some of the boundaries were arbitrarily erased. As shown in Figure 5(a), for perfect polycrystals the

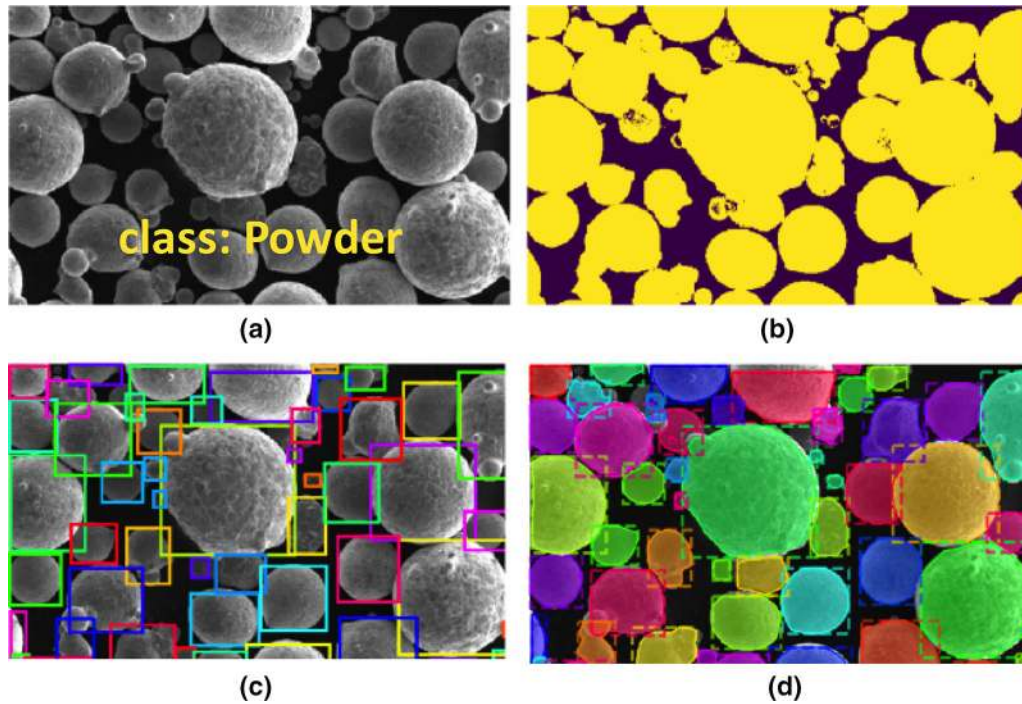


Fig. 3—A taxonomy of microstructural analysis applied to an SEM image of metal powder particles. (a) Image classification identifies the image as belonging to the ‘powder’ class. (b) Semantic segmentation associates pixels with one of two constituents: powder particles (yellow) or background (purple). (c) Object detection finds each individual powder particle (boxes). (d) Instance segmentation assigns pixels to particular particles (colors). [Original image courtesy of I. Anderson, Ames Lab.] (Color figure online).

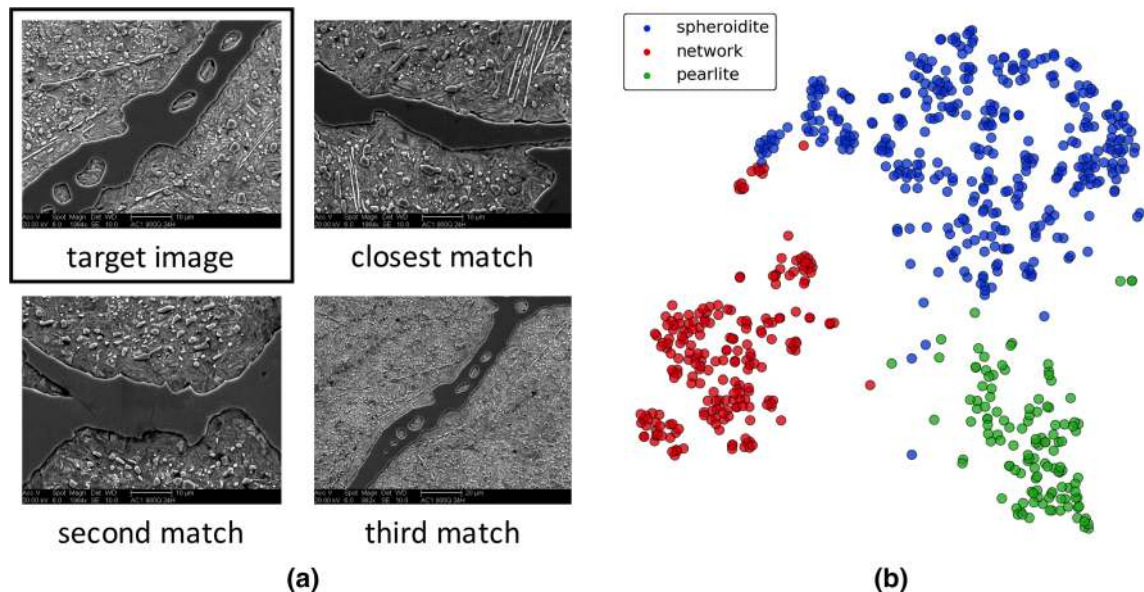


Fig. 4—Applications of the feature vector for image similarity. (a) Visual search for images similar to the target image in a database of about 900 microstructures of ultrahigh carbon steel.^[5] The closest matching images share a number of visual features in common with the target. (b) In this t-SNE plot, micrographs (points) cluster according to their visual similarity, which also corresponds to their primary microconstituent: spheroidite (blue), network carbide (red), or pearlite (green) (Color figure online).

system predicts grain size with a standard error of 2.3 pct; we also found that the standard error increases linearly as the fraction of missing boundaries increases, and reaches 3.9 pct for a missing boundary fraction of 0.4. These errors are well within an expected measurement fidelity for grain size. For instance, the ASTM standard claims a

precision of ± 0.5 grain size units for linear intercept methods, which this model meets for all missing boundary fractions (assuming circular grains and converting to a linear grain size metric).^[102] The ability to measure without segmentation creates opportunities for fault-tolerant, high-throughput microstructural evaluation.

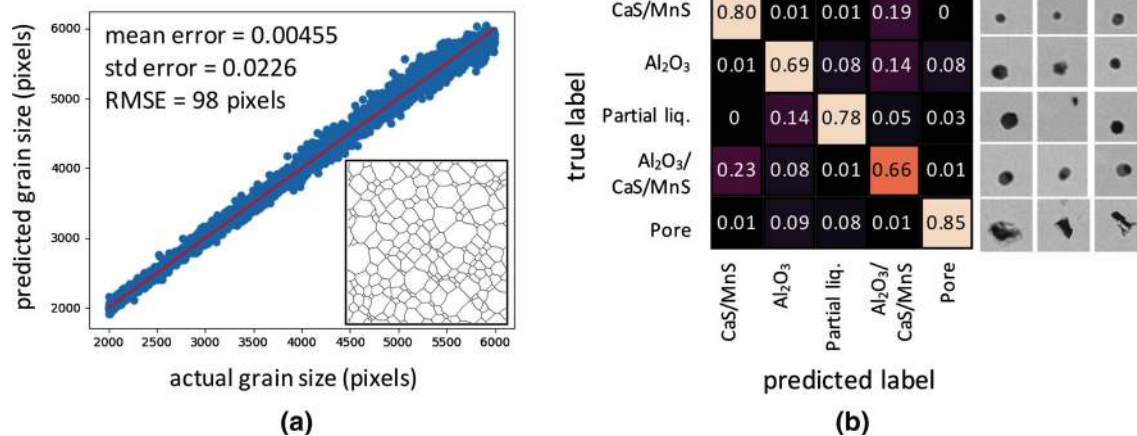


Fig. 5—Advanced applications of the microstructural feature vector. (a) Determining grain size from simulated polycrystalline micrographs without segmentation or direct measurement using deep regression. Blue points indicate previously unseen test images, and the red line corresponds to perfect accuracy (predicted = actual). Inset shows an example polycrystal. (b) Determining steel inclusion composition from SEM image patches *via* a CNN classifier. The confusion matrix gives the fraction of inclusions of each type that are classified with each predicted label; perfect accuracy would result in 1's down the diagonal. Sample image patches for each inclusion type are shown on the right (Color figure online).

Finally, we note that the feature vector encodes the full range of visual information, some of which is not readily perceptible by human vision. For instance, chemical composition is not usually measured visually, but rather with specialized tools such as energy dispersive spectroscopy (EDS). However, chemical information is contained in the grayscale values of backscattered SEM images, albeit subtly. During steelmaking, EDS is used to determine the chemical composition of unintentional inclusions (slag, *etc.*), and SEM images of these inclusions are collected at the same time.^[103] In order to determine whether SEM data alone contains sufficient information to determine inclusion composition, we utilized a pre-trained CNN^[65,72] and retrained the classifier using a data set of 2543 SEM inclusion images, balanced among five inclusion types. When tested on 509 previously unseen inclusion images, the system achieved 76 pct overall classification accuracy, considerably better than random chance accuracy of 20 pct, as shown in Figure 5(b). Furthermore, the confusion matrix confirms that the predominant misclassifications are among the sulfide inclusion types, which include significant compositional overlap. We note that these inclusions are virtually impossible to classify by eye (with the exception of the pores, which tend to be less circular than the other types). To perform this task, the CNN is presumably sensing inclusion shape, size, contrast, and color distribution with a fidelity that exceeds human perception, emphasizing the ability of CV and ML to augment and extend our ability to extract useful information from image data.

B. Semantic Segmentation

Structural metrics, such as feature size, volume fraction, aspect ratio, *etc.*, are the traditional quantities extracted from microstructural images using direct measurement tools. Typically, these measurements require a

segmented image, where each pixel in the image is assigned to a microstructural constituent. Conventional automatic image segmentation algorithms, such as those incorporated in ImageJ,^[104] generally operate by finding blobs of constant contrast or edges where contrast changes. While these approaches can work well on suitable microstructures, complex or non-ideal images often require considerable human intervention or even manual segmentation, resulting in a slow, material-specific, and subjective workflow.

Image segmentation is a foundational CV task, with important applications in robotics and medical imaging among others.^[105] Thus, there is considerable research activity in developing segmentation methods. Because micrographs share features (*e.g.*, edges, blobs, and visual textures) in common with natural images, we can adopt these methods to microstructural images *via* transfer learning. For example, we can use the PixelNet CNN^[40] trained on the ImageNet database of natural images^[72] to compute a hypercolumn feature vector^[40] for each pixel in each micrograph. We then train PixelNet's pixel classification layers to classify pixels according to their microstructural constituent. Two examples are shown in Figure 6. In Figure 6(a) the system was trained using 20 hand-annotated images from the UHCS micrograph database,^[106] and the results for one of the four previously unseen test image are shown. In Figure 6(b) the system was trained on 30 hand-annotated images from a set of tomographic slices of an Al-Zn solidification dendrite,^[107] again the results for one of the 10 previously unseen image are given.

In both cases, segmentation accuracy is excellent: 93 pct for the steel microstructures and 99.6 pct for the Al-Zn. (Measures of precision and recall, which weight false positives and false negatives differently, do not indicate significant systematic errors, such as over- or under-prediction.) These segmentation maps are arguably equal in quality to the human annotations, and

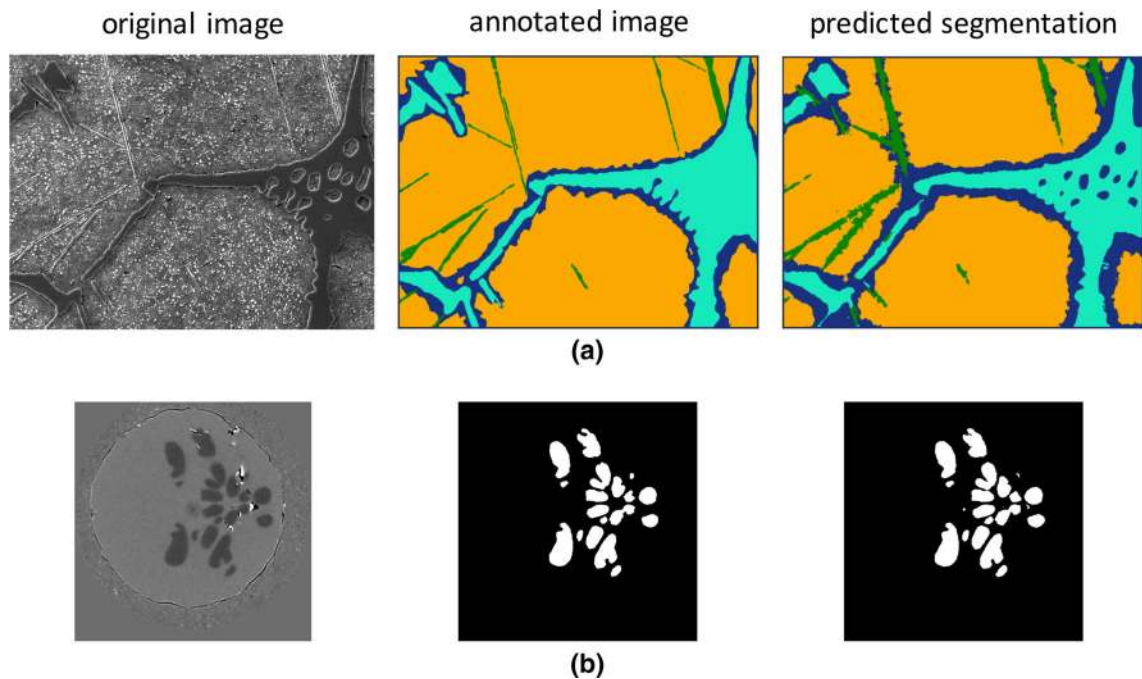


Fig. 6—Semantic segmentation of microstructural images using a pre-trained CNN. (a) Semantic segmentation of microstructural constituents in SEM micrograph of ultrahigh carbon steel. Constituents include network carbide (light blue), ferritic denuded zone (dark blue), Widmanstatten carbide (green), and spheroidite matrix (gold). Note that the CNN segmentation captures “holes” in the network carbide that were unintentionally omitted by the human annotator. (b) Segmentation of a tomographic section of an Al-Zn alloy. The solidification dendrite is shown in white on a black background. Note that the CNN segmentation ignores prominent but irrelevant visual artifacts, including the sample edge, polishing defects, and the beam spot (Color figure online).

certainly adequate for quantitative analysis of the UHCS images (as demonstrated by DeCost *et al.*^[106]) or tomographic reconstruction of the Al-Zn dendrite.^[107] A significant benefit of this approach is that once the system is trained, subsequent image segmentations are calculated very quickly (near real time), and are autonomous, objective, and repeatable, enabling the high throughput necessary for applications such as 3D reconstruction or quality control.

An additional benefit of CNN-based image segmentation is the ability to capture human-like judgments about image features. For instance, in Figure 6(a), the spheroidite matrix constituent, comprised of spheroidite particles in a ferrite matrix, is segmented as a single constituent (orange). Conventional segmentation systems would be challenged to ignore the particles, which show up as distinct bright spots. Likewise, in Figure 6(b), the system learns to ignore sample preparation artifacts such as the sample edge, pores, and the circular beam spot at the center of the image. Again, these features are difficult to remove from conventional segmentation results. It is this capacity for learning what to look for and what to ignore that distinguishes the CV/ML approach to semantic segmentation.

Although the CV/ML system simulates some aspects of human visual judgment, it does not replicate human reasoning. Therefore, it is important to understand the strengths and limitations of CNN-based image segmentation in order to design the most effective tools. A small database of 17 SEM images (15 for training and 2 for testing) of nickel-based superalloy microstructures

deformed in creep^[108] provides an illustrative case study. These micrographs contain two prominent constituents: the oriented γ' cuboidal precipitates and dislocation lines [Figure 7(a)]. Both constituents are demarcated by narrow, linear features. Conventional image analysis [Figure 7(b)] is able to find these linear features, although it tends to overpredict them, creating many short, detached line segment artifacts that are not evident in the original image. A CNN-based image segmentation system (in this case, UNet^[109]) can be trained to replicate the conventional analysis, and it has the advantage of being more resistant to line segment artifacts, as shown in Figure 7(c). However, the goal is to identify the dislocation segments only, ignoring the $\gamma - \gamma'$ phase boundaries.

We initially attempted to modify the feature annotation training images in two ways: In one set, we manually eliminated the the $\gamma - \gamma'$ phase boundaries, leaving only the dislocation annotations. However, the retrained system continued to capture many of the the $\gamma - \gamma'$ boundaries. It was apparently unable to learn which of the linear features to ignore, at least for this small dataset. In a second set, we manually annotated $\gamma - \gamma'$ phase boundaries as a separate constituent from dislocations. In this case, the system classified all near-horizontal and near-vertical linear features as $\gamma - \gamma'$ boundaries, which resulted in an under-prediction of dislocations. In both cases, the system was challenged to differentiate two microstructural constituents that are represented in the feature annotations as lines of single-pixel width.

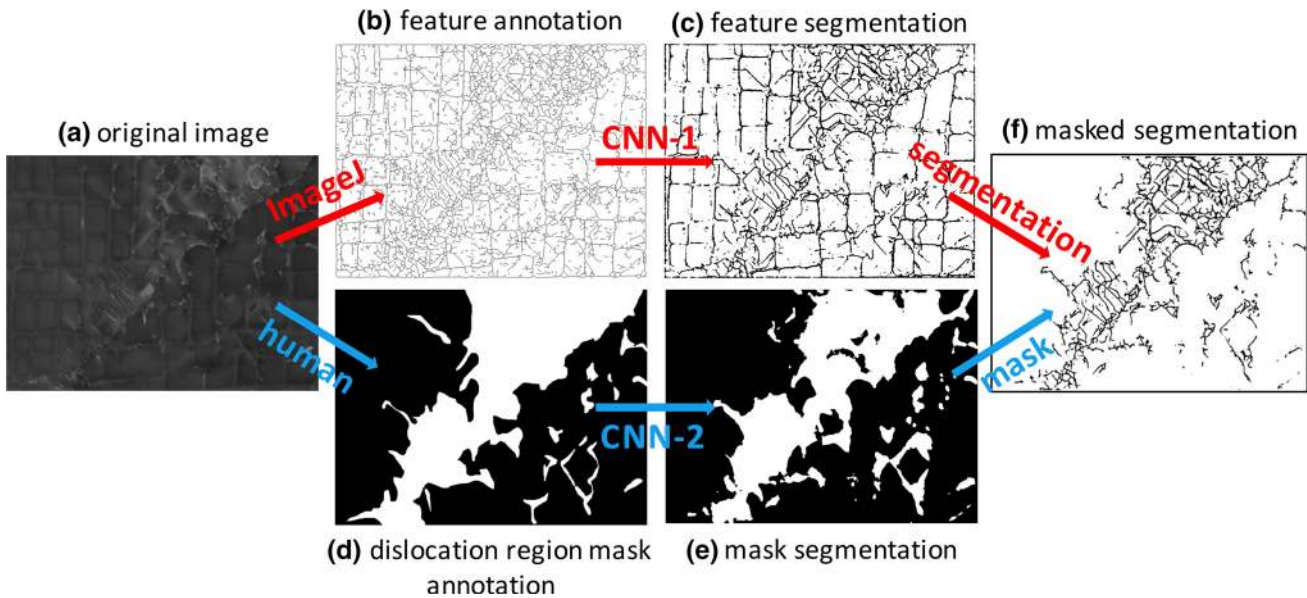


Fig. 7—Multipart segmentation of dislocation structures in SEM micrographs of nickel-based superalloys imaged with electron channeling contrast imaging. (a) An original image showing the $\gamma-\gamma'$ precipitates and dislocation traces. (b) ImageJ analysis annotates linear features. (c) CNN-based image segmentation finds linear features. (d) A human expert annotates the deformed and undeformed regions of the micrograph. (e) A second CNN-based image segmentation finds the dislocation mask. (f) Combining the feature segmentation (red arrows) and mask segmentation (blue arrows) yields a segmentation of the dislocation structure (Color figure online).

High contrast gradients (lines or edges) are one kind of feature that the filter patches typically included in CNN architectures are able to sense. Other filter patches are optimized for uniform contrast areas (blobs), and still others for larger-scale visual textures (*e.g.*, the lamellar structure of pearlite). Examining the superalloy micrographs, it becomes apparent that one way humans distinguish the dislocation lines from the $\gamma-\gamma'$ boundaries is that the dislocations appear in regions of different visual texture. In fact, it is easy for a human to differentiate the deformed and undeformed regions of these micrographs. Using that insight, we manually annotated each image to segment these regions, as shown in Figure 7(d), and trained a separate CNN-based segmentation system to predict these dislocation region masks (Figure 7(e)). By superposing the edge-based linear feature segmentation (red arrows in Figure 7) with the texture-based dislocation mask segmentation (blue arrows in Figure 7), we reconstitute an image segmentation that captures the dislocation lines and omits the $\gamma-\gamma'$ boundaries. While the lack of an unambiguous ground truth makes it hard to evaluate the accuracy of these results, the quality of these segmentations was deemed sufficient to achieve the goal of comparing deformation states. As an added benefit, these segmentations are objective, repeatable, and self-consistent among images, so that the relative differences between images are quantitatively meaningful. Overall, achieving a useful segmentation on this challenging data set requires domain expertise in both microstructural images (materials science) and CNN-based segmentation systems (computational science). The payoff is the ability to expand the reach of traditional quantitative microstructure

characterization to more complex microstructural features that have until now been difficult to treat in an automated fashion.

C. Object Detection

Object detection entails locating each unique object of its kind in an image, *i.e.*, finding each individual precipitate in a micrograph. When objects are spatially separated, object detection can be performed on a semantic segmentation using a conventional image analysis approach, such as a watershed algorithm^[38,110] or connected-component labeling.^[111,112] However, when objects overlap or occlude one another, as in the metal powder particles in Figure 8(a), automatic object detection becomes considerably more complex. Fortunately, object detection is another application of great interest in important CV applications such as self-driving vehicles, which must detect and account for each individual vehicle/pedestrian/tree in the field of view.^[113]

Specialized CNNs, notably Faster R-CNN,^[114] have been developed for object detection and the related task of instance segmentation. As in the case of semantic segmentation, discussed above, transfer learning allows these systems to be trained on natural images (such as the COCO detection data set^[115]) and applied successfully to microstructural images. For example, in order to assess the prevalence of small satellite particles in a gas-atomized metal powder, the first task is to identify the discrete powder particles in a set of SEM images^[116] (Figure 8(a)). Tedious manual annotation yielded 5 images, each with several hundred powder particles outlined (Figure 8(b)); three images were used to train a CNN instance segmentation system, and two were reserved to test the results in a

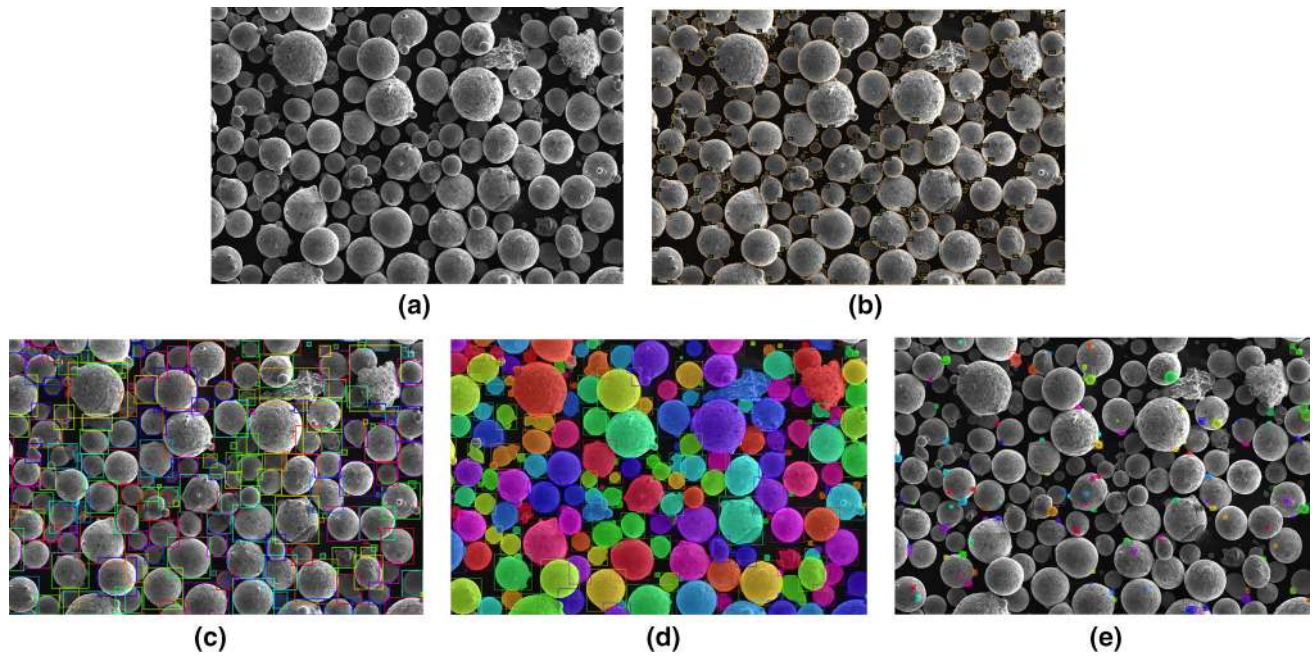


Fig. 8—Object identification and instance segmentation for SEM micrographs of gas-atomized metal powders. (a) The original image contains dense and overlapping particles. (b) Manual annotation identifies and delineates individual particles in each image. (c) A CNN-based object identification system identifies particles and draws a bounding box around each one. (d) The same system segments (colors) each particle instance separately. (e) Following a similar workflow, satellite particles are identified and segmented. By overlaying on the particle segmentation, satellites can be associated with host particles (Color figure online).

cross-validation scheme. As shown in Figure 8(c), the system was successful at locating and delineating bounding boxes around individual particles, achieving an average detection recall (correct particle identifications divided by all actual particles) of 80 pct and a detection precision (correct particle identifications divided by the total number of particles identified) of about 94 pct. Loss of recall occurs when particles are missed, and unsurprisingly the system misses some small, irregularly shaped, and largely occluded particles. Although they comprise about 20 pct of the particles, they represent a very small fraction of the total particle volume. Loss of precision occurs when a particle is identified where none is present. However, the system is not finding particles where the image contains empty space. Instead, it is predicting that there are multiple particles where a single, agglomerated particle is identified in the manual annotation. In many of these cases, it is arguable whether the computer or human is correct. For this object detection task, the results of the CNN-based system are of good quality for subsequent analysis, and the errors that do occur are sensible and to some extent unavoidable.

Object detection is essential for calculating metrics based on counting objects, such as a number density or a population. However, in image analysis, it is more commonly a stepping stone to instance segmentation, as discussed in the next section.

D. Instance Segmentation

Once an object has been identified and bounded, instance segmentation is simply a matter of segmenting the pixels that belong to the object, as described above.

In fact, object detection and instance segmentation are often combined in CNN implementations such as Mask R-CNN.^[117] Figure 8(d) shows the instance segmentation, with each particle segmented (colored) as a separate object of particle type. The particle segmentations achieve a cross-validation average precision of 97.5 pct and recall of 95.4 pct; these high values indicate minimal over- or under-prediction and in fact are comparable to human performance, since there is some subjectivity in locating particle edges. While this application of instance segmentation found every object in the image, the same approach can differentiate and segment objects of particular types. Figure 8(e) shows instance segmentation of satellite particles, defined as small particles adhered to the surface of much larger particles. There is considerably more subjectivity in identifying satellites, which results in more disagreement between human and computer, thus lower detection precision (69.2 pct) and recall (54.5 pct). However, this performance is still adequate to estimate satellite metrics such as satellite content as a function of particle size, and the autonomous nature of the CNN-based system allows much more throughput than would be feasible using manual segmentation.

IV. NEXT STEPS: METRICS, APPLICATIONS, AND INTERPRETATION

A. Novel and Advanced Microstructural Metrics

Beyond facilitating traditional microstructural measurements, CV image representations offer entirely new ways to characterize microstructures. For example, for a

set of SEM micrographs of an Inconel-718 gas-atomized metal powder^[38] we use connected-component labeling^[111,112] to perform instance segmentation of each individual powder particle and then encode each particle patch *via* a CNN layer feature vector. After reducing the representation dimension *via* PCA,^[73] we apply *k*-means unsupervised ML^[87] to identify 8 visual clusters of particles. Figure 9(a) shows examples of particles belonging to each of the clusters. It is clear that this method is able to sort powder particles into visually similar groups. When the individual particle images are plotted in a t-SNE map (Figure 9(b)), particles cluster according to visual characteristics such as surface roughness, shape, and size. The statistics of the *k*-means clustering (*i.e.*, particles per cluster) and the structure of the t-SNE plot (*i.e.*, particle density map) act as fingerprints of the powder material, containing information about not only particle size, but also shape, roughness, agglomeration, *etc.* These representations can be applied to develop new, quantitative metrics to characterize powder materials that capture considerably more information than traditional powder size distributions.^[118] We note that it is not feasible for humans to sort thousands of image patches, nor would we expect an objective and repeatable result. These new metrics are only accessible using a CV/ML approach, which leverages the ability of computers to perform repetitive tasks with the human-like visual judgment imparted by CV/ML methods.

When image characterization is performed as a supervised task, it requires training and validation/test data that has been annotated with the ground truth (*e.g.*, grain size, inclusion composition, microconstituent, *etc.*). Because annotation is often tedious and sometimes impossible, unsupervised image analysis is a goal in microstructural science and in CV more generally. The *k*-means clustering analysis described above suggests one route to an image metric (particle cluster statistics) using unsupervised ML. In this case, rather than designing an unsupervised ML method to measure a particular metric, we use the result of a standard, unsupervised ML method as the metric. Likewise, one can envision new microstructural metrics that arise directly from aspects of the CV/ML system, such as the feature vector or CNN filter activations.^[119]

Finally, both microstructural science and CV share the goal of inferring 3D structure from 2D images.^[120] Natural images are usually 2D projections of 3D scenes, so inferring the third dimension is at least in part an optics problem. In contrast, micrographs are typically direct representations of 2D cross sections, so inferring the third dimension is the purview of quantitative stereology.^[121] Because of this, the types of data and the CV approaches most promising for inferring 3D structure from 2D images remain open questions in CV/ML methods for microstructural science.

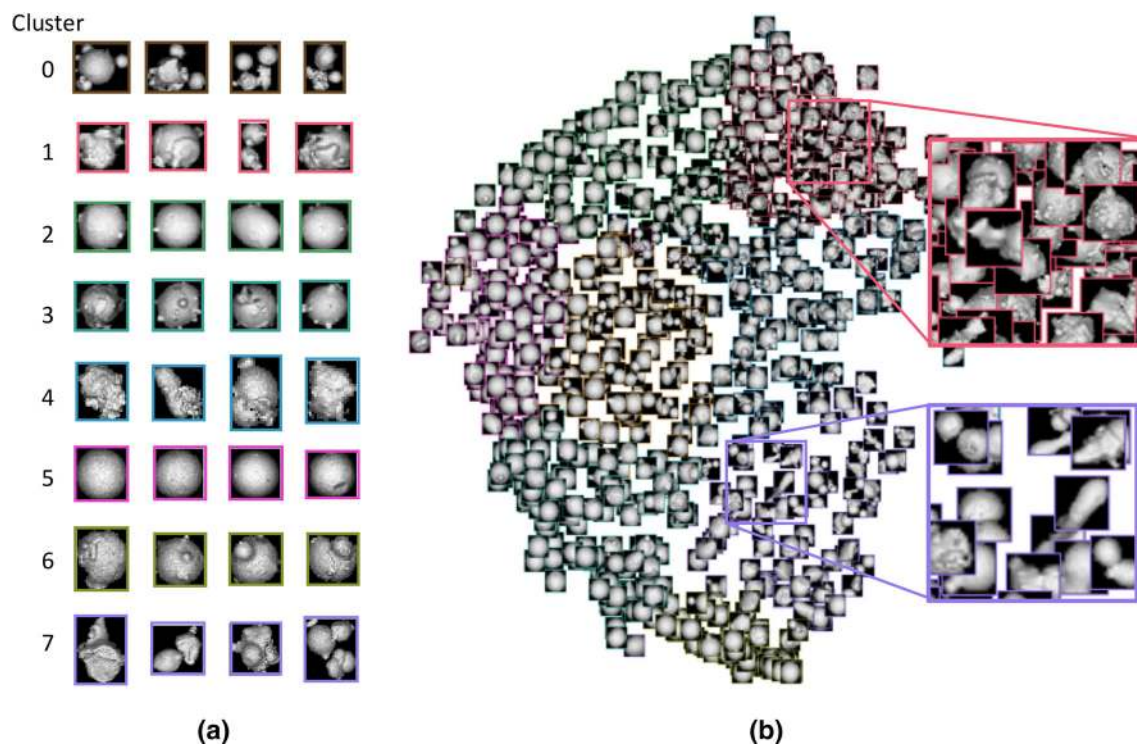


Fig. 9—The visual density map for a metal powder. (a) Individual particles are clustered visually; here an 8-cluster *k*-means analysis showing four example particles per cluster. (b) A t-SNE visual similarity map for about 1100 Inconel-718 particles. Insets show how visually similar particles cluster (red = rough, agglomerated, purple = elongated). Box color corresponds to *k*-means cluster number. The *k*-means cluster statistics and image density map are quantitative fingerprints of the powder material (Color figure online).

B. CV/ML for Processing–Structure–Property Links

A primary objective of microstructural analysis is to discover processing-microstructure-property (PSP) relationships for systems of scientific and technological interest. While the application of CV/ML systems to PSP problems is beyond the scope of this overview, we will note that it is an active area of research. For instance, to make the processing/structure connection, CV/ML studies have examined micrograph databases to correlate microstructure with annealing history.^[34,122] Likewise, CV/ML systems have been applied to relate microstructure with outcome properties including stress hot spot^[123,124] and damage formation,^[98] fatigue failure initiation,^[125] fracture energy,^[93] ionic conductivity,^[100] and fatigue strength.^[122] Using the ability of CNNs to generate structures, several recent studies have also made strides toward the inverse problem of designing microstructures with target properties.^[122,126] Limiting factors in making robust PSP connects is the scarcity of large data sets that link specific microstructural images to processing history and/or property outcomes and the need to capture the property-limiting features, which are often three dimensional, in 2D micrographs.

C. Interpretability: Opening the Black Box

Scientists and engineers can be hesitant to rely on “black box” algorithms, where the basis for a decision or prediction is unknown.^[127] While this may not be a significant consideration for characterization tools such as semantic segmentation where performance can be assessed straightforwardly (and where the human version is not particularly interpretable either), it is critical for analysis tasks where understanding the basis for arriving at the conclusion is essential, such as making PSP connections. How black box CV systems make decisions is a significant, open question in computer science with many partial solutions, but none that are generally applicable across data sets and methods. Approaches include associating feature vector characteristics with microstructural length,^[74] examining filter activations and characteristic textures,^[74] and locating the image region most salient to decision-making.^[128] In order to extract abstract scientific information from concrete visual information, the critical step is to identify the visual signatures of underlying physical processes. Discovering that signature in the feature vector how it is processed in the CV/ML pipeline is a grand challenge in AI-supported microstructural characterization and analysis.

V. CONCLUSIONS

The quantitative representation of microstructure is the foundational tool of microstructural science, connecting the materials structure to its composition, process history, and properties. Microstructural quantification traditionally involves a human deciding *a priori* what to measure and then devising a purpose-built

method for doing so. However, recent advances in data science, including computer vision (CV) and machine learning (ML) offer new approaches to extracting information from microstructural images.

The key function of CV is to numerically encode the visual information contained in a microstructural image into a feature vector. The feature vector then provides input to ML algorithms that find associations and trends in the high-dimensional image representation. CV/ML systems for microstructural characterization and analysis span the taxonomy of image analysis tasks, including image classification, semantic segmentation, object detection, and instance segmentation. Applications include:

- Visual search, sort, and classification of micrographs *via* feature vector similarity.
- Extracting information not readily visible to humans, such as chemical composition in SEM micrographs, by using latent information in the feature vector.
- Performing semantic segmentation of microstructural constituents with a high accuracy and human-like judgment about what to look for and what to ignore.
- Combining segmentations based on different feature types to segment complex structures.
- Finding and bounding all instances of individual objects, even when they impinge and overlap.
- Segmenting individual objects to enable new capabilities in microstructural image analysis.

A common characteristic among all of these applications is that they capitalize on the ability of computational systems to produce accurate, autonomous, objective, repeatable results in an indefatigable and permanently available manner. These tools enable new approaches to microstructural analysis, including the development of new, rich visual metrics and the discovery of processing-microstructure-property relationships.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under grant CMMI-1826218 and the Air Force D³OM²S Center of Excellence under agreement FA8650-19-2-5209. The authors appreciate data sets, collaborations, and helpful conversations provided by Dr. Iver Anderson (Ames National Laboratory), Dr. Brian DeCost (NIST), Anna Smith (Merck), Dr. Sabin Sulzer (Oxford), Prof. Peter Voorhees and Dr. Tiberiu Stan (Northwestern University), and Prof. Brian Webler (CMU).

ELECTRONIC SUPPLEMENTARY MATERIAL

The online version of this article (<https://doi.org/10.1007/s11661-020-06008-4>) contains supplementary material, which is available to authorized users.

REFERENCES

- C.S. Smith: *A History of Metallography: The Development of Ideas on the Structure of Metals before 1890*, MIT Press, Cambridge, MA, 1988, pp. 1–314.
- H.C. Sorby: *J. Iron Steel Inst.*, 1887, vol. 37, pp. 255–88.
- G.F. Vander Voort: *Standard News*, 1991, vol. 19, pp. 42–47.
- University of Cambridge DoITPoMS: Micrograph Library <http://www.doitpoms.ac.uk/miclib/index.php> (2004–2013), <http://www.doitpoms.ac.uk/miclib/index.php>. Accessed 9 Aug 2020.
- B.L. DeCost, M.D. Hecht, T. Francis, Y.N. Picard, B.A. Webler, and E.A. Holm: *Integr. Mater. Manuf. Innov.*, 2017, vol. 6, pp. 197–205.
- G.F. VanderVoort: *Metallography: Principles and Practice*, ASM International, Metals Park, OH, 1999, pp. 1–752.
- E.E. Underwood: *Quantitative Stereology*, Addison-Wesley, New York, 1970, pp. 1–274.
- D. Dingley: *J. Microscopy*, 2004, vol. 213, pp. 214–24.
- H.F. Poulsen, S.F. Nielsen, E.M. Lauridsen, S. Schmidt, R.M. Suter, U. Lienert, L. Margulies, T. Lorentzen, and D. Juul-Jensen: *J. Appl. Crystallogr.*, 2001, vol. 34, pp. 751–56.
- E.O. Hall: *Phys. Soc. Lond.*, 1951, vol. 64, pp. 747–53.
- N.J. Petch: *J. Iron Steel Inst.*, 1953, vol. 174, pp. 25–28.
- M.-N. Feng, Y.-C. Wang, H. Wang, G.-Q. Liu, and W.-H. Xue: *Int. J. Miner. Metall. Mater.*, 2017, vol. 24, pp. 257–63.
- J. J. Friel, E. B. Prestridge, and F. Glazer, in *MiCon 90: Advances in Video Technology for Microstructural Control*, G. Vander Voort, ed., ASTM International: Philadelphia, PA, 1991, pp. 170–184.
- R. Szeliski: *Computer Vision: Algorithms and Applications*, Springer, New York, 2010, pp. 1–812.
- S. Yoshinaga, A. Shimada, and R.-i. Taniguchi: *Procedia Soc. Behav. Sci.*, 2010, vol. 2, pp. 143–52.
- A.P. Carleer, O. Debeir, and E. Wolff: *Photogramm. Eng. Remote Sens.*, 2005, vol. 71, pp. 1285–94.
- P. Flach: *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*, Cambridge University Press, Cambridge, UK, 2012, pp. 1–409.
- O.B. Abouelatta: *J. Am. Sci.*, 2013, vol. 9, pp. 213–23.
- A. Cecen, T. Fast, E.C. Kumbur, and S.R. Kalidindi: *J. Power Sources*, 2014, vol. 245, pp. 144–53.
- A. Chowdhury, E. Kautz, B. Yener, and D. Lewis: *Comput. Mater. Sci.*, 2016, vol. 123, pp. 176–87.
- V.H.C. de Albuquerque, P.C. Cortez, A.R. de Alexandria, and J.M.R.S. Tavares: *Nondestruct. Test. Eval.*, 2008, vol. 23, pp. 273–83.
- S.R. Kalidindi and D.T. Fullwood: *JOM*, 2007, vol. 59, pp. 26–31.
- S.R. Kalidindi, S.R. Niezgoda, and A.A. Salem: *JOM*, 2011, vol. 63, pp. 34–41.
- A.C. Lewis, C. Suh, M. Stukowski, A.B. Geltmacher, G. Spanos, and K. Rajan: *JOM*, 2006, vol. 58, pp. 52–56.
- S.R. Niezgoda, D.T. Fullwood, and S.R. Kalidindi: *Acta Mater.*, 2008, vol. 56, pp. 5285–92.
- S.R. Niezgoda and S.R. Kalidindi: *CMC Comput. Mater. Contin.*, 2009, vol. 14, pp. 79–97.
- S.R. Niezgoda, S.R. Kalidindi, X. Hu, G.A. Cingara, D.S. Wilkinson, M. Jain, P. Wu, R.K. Mishra, M. Arafin, and J. Szpunar: *Comput. Mater. Continua*, 2010, vol. 14, pp. 79–98.
- S.R. Niezgoda, A.K. Kanjarla, and S.R. Kalidindi: *Integr. Mater. Manuf. Innov.*, 2013, vol. 2, pp. 1–27.
- S.R. Niezgoda, Y.C. Yabansu, and S.R. Kalidindi: *Acta Mater.*, 2011, vol. 59, pp. 6387–6400.
- G. Saheli, H. Garmestani, and B.L. Adams: *J. Comput. Aided Mater. Des.*, 2004, vol. 11, pp. 103–15.
- J.P. Simmons, P. Chuang, M. Comer, J.E. Spowart, M.D. Uchic, and M. De Graef: *Modell. Simul. Mater. Sci. Eng.*, 2009, vol. 17, pp. 0250021–22.
- A. Velichko, C. Holzapfel, A. Siefers, K. Schladitz, and F. Mucklich: *Acta Mater.*, 2008, vol. 56, pp. 1981–90.
- B.L. Decost: *Materials Science and Engineering*, Carnegie Mellon University, Pittsburgh, PA, 2016, pp. 1–169.
- B.L. DeCost, T. Francis, and E.A. Holm: *Acta Mater.*, 2017, vol. 133, pp. 30–40.
- B.L. DeCost and E. Holm: *Comput. Mater. Sci.*, 2017, vol. 126, pp. 438–45.
- B.L. DeCost and E. Holm, in *Statistical Methods for Materials Science: Data Analytics in Microstructure Characterization*, J. Simmons, C. Bouman, L. Drummy, and M. de Graef, ed., CRC Press, Boca Raton, FL, 2017, pp. 73–93.
- B.L. DeCost and E.A. Holm: *Comput. Mater. Sci.*, 2015, vol. 110, pp. 126–33.
- B.L. DeCost, H. Jain, E. Holm, and A. Rollett: *JOM*, 2017, vol. 69, pp. 456–65.
- H.D. Ballard: *Pattern Recognit.*, 1981, vol. 13, pp. 111–22.
- A. Bansal, X. Chen, B. Russell, A. Gupta, and D. Ramanan, *arXiv preprint arXiv:1702.06506* [cs.CV], 2016, pp. 1–17.
- X. Chen, A. Shrivastava, and A. Gupta, *2013 IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 1409–16.
- M. Comer, C.A. Bouman, M. De Graef, and J.P. Simmons: *JOM*, 2011, vol. 63, pp. 55–57.
- N. Dalal and B. Triggs, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005 (CVPR)*, (2005), pp. 886–93.
- P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan: *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010, vol. 32, pp. 1627–45.
- R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell, *arXiv arXiv:1704.02895*, 2017.
- C. Harris and M. Stephens, *Proceedings of the Fourth Alvey Vision Conference (Machester, UK)*, 1988, pp. 147–151.
- Y.-G. Jiang, J. Yang, C.-W. Ngo, and A.G. Hauptmann: *IEEE Trans. Multimedia*, 2010, vol. 12, pp. 42–53.
- T.-Y. Lin, A. RoyChowdhury, and S. Maji, *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1449–57.
- D.G. Lowe, *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, pp. 1150–57.
- D.G. Lowe: *Int. J. Comput. Vis.*, 2004, vol. 60, pp. 91–110.
- A. Oliva and A. Torralba: *Prog. Brain Res.*, 2006, vol. 155, pp. 23–36.
- H. Peng, F. Long, and C. Ding: *IEEE Trans. Pattern Anal. Mach. Intell.*, 2005, vol. 27, pp. 1226–38.
- F. Perronnin: *IEEE Trans. Pattern Anal. Mach. Intell.*, 2008, vol. 30, pp. 1243–56.
- P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica-Perez, and T. Tuytelaars: *IEEE Trans. Pattern Anal. Mach. Intell.*, 2007, vol. 29, pp. 1575–89.
- F.S. Khan, R.M. Anwer, J. van de Weijer, A.D. Bagdanov, M. Vanrell, and A.M. Lopez, *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3306–13.
- J.C. van Gemert, C.J. Veenman, A.W.M. Smeulders, and J.-M. Geusebroek: *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010, vol. 32, pp. 1271–83.
- J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid: *Int. J. Comput. Vis.*, 2007, vol. 73, pp. 213–38.
- J. Li and N.M. Allinson: *Neurocomputing*, 2008, vol. 71, pp. 1771–87.
- G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, vol. 1, pp. 1–16.
- J. Sivic and A. Zisserman, *Proceedings Ninth IEEE International Conference on Computer Vision*, 2003, pp. 1470–77.
- T.S. Guzella and W.M. Caminhas: *Expert Syst. Appl.*, 2009, vol. 36, pp. 10206–22.
- H. Jégou, M. Douze, C. Schmid, and P. Pérez, *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3304–11.
- R.M. Cichy, A. Khosla, D. Pantazis, A. Torralba, and A. Oliva: *Sci. Rep.*, 2016, vol. 6, pp. 277551–13.
- M.D. Zeiler, G.W. Taylor, and R. Fergus, *2011 International Conference on Computer Vision*, 2011, pp. 2018–25.
- K. Simonyan and A. Zisserman, *International Conference on Learning Representations 2015*, 2015, [arXiv:1409.1556](https://arxiv.org/abs/1409.1556), pp. 1–14.
- A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, Curran Associates Inc., Lake Tahoe, Nevada, 2012, pp. 1097–1105.
- A. Jalalian, S.B.T. Mashohor, H.R. Mahmud, M. Iqbal, B. Saripan, A. Rahman, B. Ramli, and B. Karafi: *Clin. Imaging*, 2013, vol. 37, pp. 420–26.

68. I. Goodfellow, Y. Bengio, and A. Courville: *Deep Learning*, MIT Press, Cambridge, MA, 2016, pp. 1–800.
69. Y. LeCun, Y. Bengio, and G. Hinton: *Nature*, 2015, vol. 521, pp. 436–44.
70. J. Schmidhuber: *Neural Netw.*, 2015, vol. 61, pp. 85–117.
71. Yu. Yanming Guo, A.O. Liu, S. Lao, W. Song, and M.S. Lew: *Neurocomputing*, 2015, vol. 187, pp. 27–48.
72. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei: *Int. J. Comput. Vis.*, 2015, vol. 115, pp. 211–52.
73. H. Abdi and L.J. Williams: *Wiley Interdiscip. Rev. Comput. Stat.*, 2010, vol. 2, pp. 433–59.
74. J. Ling, M. Hutchinson, E. Antono, Brian L. DeCost, E. Holm and B. Meredig, *Mater. Discov.*, 2017, [arXiv:1711.00404](https://arxiv.org/abs/1711.00404).
75. B. Hariharan, J. Malik, and D. Ramanan: *Computer Vision—ECCV 2012*, Springer, New York, 2012, pp. 459–72.
76. D.P. Kingma and M. Welling, [arXiv:1312.6114](https://arxiv.org/abs/1312.6114) [stat.ML], 2013.
77. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014)*, 2014, pp. 2672–80.
78. Z. Yang, X. Li, L.C. Brinson, A.N. Choudhary, W. Chen, and A. Agrawal: *J. Mech. Des.*, 2018, vol. 140, p. 110301.
79. S. Chun, S. Roy, Y.T. Nguyen, J.B. Choi, H.S. Udaykumar, and S.S. Baek: *Sci. Rep.*, 2020, vol. 10, p. 13307.
80. A. Gayon-Lombardo, L. Mosser, N.P. Brandon, and S.J. Cooper: *NPJ Comput. Mater.*, 2020, vol. 6, p. 82.
81. H. Chen, X. He, Q. Teng, R.E. Sheriff, J. Feng, and S. Xiong: *Phys. Rev. E*, 2020, vol. 101, p. 023305.
82. T.M. Mitchell: *Machine Learning*, McGraw-Hill, New York, 1997, pp. 1–414.
83. C. Cortes and V. Vapnik: *Machine Learning*, 1995, vol. 20, pp. 273–97.
84. B. Scholkopf and A.J. Smola: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge, MA, 2001, pp. 1–632.
85. A. Liaw and M. Wiener: *R News*, 2002, vol. 2 (3), pp. 18–22.
86. A. Mangal and E.A. Holm: *Integr. Mater. Manuf. Innov.*, 2018, vol. 7, pp. 87–95.
87. S. Lloyd: *IEEE Trans. Inf. Theory*, 1982, vol. 28, pp. 129–37.
88. L. van der Maaten, E. Postma, and H. Herik: *J. Mach. Learn. Res. JMLR*, 2007, vol. 10, pp. 301–48.
89. G. Hinton and S. Roweis, *Proceedings of Advances in Neural Information Processing Systems*, 2002, vol. 15, pp. 833–841.
90. L. van der Maaten and G. Hinton: *J. Mach. Learn. Res.*, 2008, vol. 9, pp. 2579–2605.
91. L. Perez and J. Wang, [arXiv:1712.04621](https://arxiv.org/abs/1712.04621) [cs.CV], 2017.
92. C. Shorten and T.M. Khoshgoftaar: *J. Big Data*, 2019, vol. 6, p. 60.
93. A.R. Kitahara and E.A. Holm: *Integr. Mater. Manuf. Innov.*, 2018, vol. 7, pp. 148–56.
94. J. Masci, U. Meier, D. Ciresan, J. Schmidhuber, and G. Fricout, *The 2012 International Joint Conference on Neural Networks (IJCNN)*, 2012, pp. 1–6.
95. W. Li, K.G. Field, D. Morgan, and N.P.J. Comput: *Mater.*, 2018, vol. 4, p. 36.
96. S.M. Azimi, D. Britz, M. Engstler, M. Fritz, and F. Mücklich: *Sci. Rep.*, 2018, vol. 8, p. 2128.
97. J. Madsen, P. Liu, J. Kling, J.B. Wagner, T.W. Hansen, O. Winther, and J. Schiøtz: *Adv. Theory Simul.*, 2018, vol. 1, p. 1800037.
98. C. Kusche, T. Reclik, M. Freund, T. Al-Samman, U. Kerzel, and S. Korte-Kerzel: *PLoS ONE*, 2019, vol. 14, p. e0216493.
99. S.J. Plimpton, A. Thompson, and A. Slepoy, *SPPARKS Kinetic Monte Carlo Simulator*, Sandia National Laboratories, 2009, <http://www.sandia.gov/~sjplimp/spparks.html>. Accessed 9 Aug 2020.
100. R. Kondo, S. Yamakawa, Y. Masuoka, S. Tajima, and R. Asahi: *Acta Mater.*, 2017, vol. 141, pp. 29–38.
101. S. Lathuilière, P. Mesejo, X. Alameda-Pineda, and R. Horaud: *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019, vol. 36, p. 1.
102. ASTM: *E112-13 Standard Test Methods for Determining Average Grain Size*, ASTM International, West Conshohocken, PA, 2013, pp. 1–28.
103. H.V. Atkinson and G. Shi: *Prog. Mater. Sci.*, 2003, vol. 48, pp. 457–520.
104. C.A. Schneider, W.S. Rasband, and K.W. Eliceiri: *Nat. Methods*, 2012, vol. 9, pp. 671–75.
105. A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez: *Appl. Soft Comput.*, 2018, vol. 70, pp. 41–65.
106. B.L. DeCost, T. Francis, and E.A. Holm: *Microsc. Microanal.*, 2019, vol. 25, pp. 21–29.
107. T. Stan, Z. Thompson, and P. Voorhees: *Mater. Charact.*, 2020, vol. 160, p. 110119.
108. S. Sulzer, Z. Li, S. Zaefferer, M.H. Haghighat, A. Wilkinson, D. Raabe, and R. Reed: *Acta Mater.*, 2020, vol. 185, pp. 13–27.
109. O. Ronneberger, P. Fischer, and T. Brox: *U-Net: Convolutional Networks for Biomedical Image Segmentation*, Springer, Cham, 2015, pp. 234–41.
110. J. Cousty, G. Bertrand, L. Najman, and M. Couprie: *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009, vol. 31, pp. 1362–74.
111. C. Fiorio and J. Gustedt: *Theor. Comput. Sci.*, 1996, vol. 154, pp. 165–81.
112. W. Kesheng, E. Otoo, and K. Suzuki: *Pattern Anal. Appl.*, 2009, vol. 12, pp. 117–35.
113. L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen: *Int. J. Comput. Vis.*, 2020, vol. 128, pp. 261–318.
114. S. Ren, K. He, R. Girshick, and J. Sun: *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017, vol. 39, pp. 1137–49.
115. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Lawrence Zitnick, *European Conference on Computer vision ECCV 2014*, 2014, pp. 740–55.
116. I. Anderson, *Personal communication*, 2019.
117. K. He, G. Gkioxari, P. Dollár, and R. Girshick, *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–88.
118. A. Strondl, O. Lyckfeldt, H. Brodin, and U. Ackelid: *JOM*, 2015, vol. 67, pp. 549–54.
119. N. Lubbers, T. Lookman, and K. Barros: *Phys. Rev. E*, 2017, vol. 96, p. 052111.
120. X. Han, H. Laga, and M. Bennamoun: *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019, vol. 40, pp. 1224–44.
121. E.E. Underwood, in *Microstructural Analysis: Tools and Techniques*, J.L. McCall and W.M. Mueller, ed., Springer, Boston, MA, 1973, pp. 35–66.
122. R. Noraas, N. Somanath, M. Giering, and O.O. Olusegun, *AIAA Scitech 2019 Forum*.
123. A. Mangal and E.A. Holm: *Int. J. Plast.*, 2018, vol. 111, pp. 122–34.
124. A. Mangal and E.A. Holm: *Int. J. Plast.*, 2019, vol. 114, pp. 1–14.
125. C. Kantzos, J. Lao, and A. Rollett: *Mater. Charact.*, 2019, vol. 158, p. 109961.
126. H. Xu, R. Liu, A. Choudhary, and W. Chen: *J. Mech. Des.*, 2015, vol. 137, p. 050301.
127. E. Holm: *Science*, 2019, vol. 363, pp. 3–4.
128. R.C. Fong and A. Vedaldi, *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3449–57.

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