Acta Materialia 223 (2022) 117434

Contents lists available at ScienceDirect

Acta Materialia

journal homepage: www.elsevier.com/locate/actamat

Overview article

Microstructure classification in the unsupervised context

Courtney Kunselman[a], Sofia Sheikh[a], Madalyn Mikkelsen[b], Vahid Attari[a,*], Raymundo Arróyave[a,b,c]

[a] Department of Materials Science and Engineering Department, Texas A&M University, College Station, TX 77843, United States
[b] Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843, United States
[c] Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX 77843, United States

ARTICLE INFO

Article history:
Received 18 August 2020
Revised 27 August 2021
Accepted 20 October 2021
Available online 6 November 2021

Keywords:
Microstructure classification
Unsupervised learning
Phase field modeling

ABSTRACT

Traditional microstructure classification requires human annotations provided by a subject matter expert. The requirement of human input is both costly and subjective and cannot keep up with the current volume of experimentally and computationally generated microstructure images. In this work, we develop a framework that is capable of reducing the cost of human annotation in this process by leveraging novel machine learning procedures for class discovery and label assignment. To reduce the penalty of a poor label assignment made by this automated process, labels are only assigned to high-confidence observations while ambiguous data are left unlabeled. Semi-supervised classification is then employed to leverage the high- and low-confidence label assignments, and a novel generalization of an established semi-supervised error estimation technique to the multi-class context is introduced to assess the resulting classifiers. Finally, it is shown that this framework can be used to produce highly accurate classifiers over microstructure image class taxonomies which are discovered solely through data-driven methods and which display consistent structural trends within and distinct morphological differences between classes.

1. Introduction

As evidenced by recent attention given to the Materials Genome [1] and Integrated Computational Materials Engineering (ICME) [2] initiatives and the advent of materials informatics as a well-established sub-field within Materials Science, the new scientific paradigm of (big) data-driven discovery has become a staple of research focusing on materials design [3]. ICME specifically aims to accelerate the discovery and design of new materials through quantitative modeling and *exploitation* of processing-structure-property (PSP) relationships. The recent explosion of available data due to advances in experimental, theoretical, and computational capabilities has necessitated the use of automated frameworks to assist in both discovering and quantitatively establishing these linkages. Indeed, a recent US Government report [4] identified the development of such automated frameworks in the data-driven discovery context as a pressing research interest for accelerating the materials discovery process.

While quantitatively capturing processing conditions and material properties for use in machine learning frameworks tends to be a straight-forward exercise, characterizing microstructure is considerably less trivial. This is due to microstructural information generally being provided in the form of images. These images vary from atomistic to macroscopic length scales and often only contain structural features of interest to a specific application, resulting in inconsistent and incomplete reports of quantitative microstructure characterizations [5]. The challenges that image data presents to uniformity and completeness of microstructure characterizations have led to classification being the automated tool of choice for analyzing and navigating the microstructure space [6–9].

Traditional classification algorithms are supervised learning methods which are trained to map numerical inputs to categorical outputs. The advantage of such predictors is that the method of image characterization can be quite abstract or have very little physical meaning as long as the assigned class label is sufficiently descriptive. Indeed, while physical descriptors tend to be used as features for classification frameworks in the literature [10,11], popular characterization methods also include reduced order representations of statistical functions or pre-trained convolutional neural network (CNN) layers [12–14] that have at best a limited relationship to physical characteristics of the microstructure.

Unfortunately, the categorical targets of classification can also be a source of great disadvantage. This is because expert human annotation required by supervised frameworks is expensive to deploy and can be quite subjective [15]. If the dataset of interest is

* Corresponding author.
E-mail address: attari.v@tamu.edu (V. Attari).

https://doi.org/10.1016/j.actamat.2021.117434
1359-6454/

extremely large or growing at a rate that outpaces the expert's ability to assign class labels, then a portion of the data is unusable to train or validate the model. Furthermore, due to the inherent subjectivity of the labeling exercise, there is no guarantee that the data which is labeled is reliable. These obstacles of expense and subjectivity are further compounded in the face of large and structurally diverse microstructure image datasets collected from material systems with no established class taxonomy.

Although annotation using incomplete and/or potentially unreliable data is a reality in the new paradigm of (big) data science, there is not much work addressing this problem in the context of microstructure classification in materials science. While a recent work by a subset of the present authors demonstrated an approach to semi-supervised classification of ambiguous (to the annotator) microstructures [15], the majority of existing frameworks simply apply supervised classification algorithms to well-studied datasets for purposes of automating tedious analysis tasks. Thus, there are significant opportunities to develop and demonstrate data-driven (i.e. unsupervised) microstructure classification methodologies capable of uncovering underlying relationships and features in complex, unlabeled, microstructure spaces. Looking at how other fields have incorporated unlabeled data into their classification schemes could provide valuable insight as to how to proceed.

For example, semi-supervised learning methods, a category of machine learning tools which use both labeled and unlabeled data to train predictive models, are popular in studies building image classifiers for remote sensing applications [16–18]. However, similar to the traditional supervised case, a limitation of these methods is that they still require an established all-encompassing class taxonomy with labeled examples of each class. Positive unlabeled learning trains binary classifiers using unlabeled data and only labeled examples of the positive class and has been used for disease identification and data stream classification [19,20]. However, the discovery of the "negative" class is difficult to generalize to the discovery of multiple latent classes, and these methods still require prior knowledge of at least one present class. Thus, while these methods may be helpful in addressing this problem, we still need a tool which provides insights into a possible class taxonomy before they can be applied.

Unsupervised learning methods provide a promising answer to the class discovery dilemma. At a high level, unsupervised learning aims to provide inferences into the underlying structure of unlabeled data. In general, these algorithms provide these inferences in a data-driven fashion governed by rule-based assumptions about the ground-truth data distribution. Because these methods operate in an unsupervised environment, it can be difficult to determine if the inferences that they provide are valid or not; thus, caution should be taken to ensure that algorithm assumptions agree with what is known about the data and that drawn inferences about underlying structure are not blindly accepted. There are a few studies concerning image classification which boast success in the unsupervised context [21–23]. However, similar to the shortcomings of many popular clustering algorithms such as K-Means and Self-Organizing Maps, these methods require *a priori* knowledge in the form of a user-specified number of classes. Due to this concern, the bioinformatics community has employed class discovery techniques which only require the user to specify a range of the number of classes to be considered [24,25]. A further appeal of many of these methods is that they provide an objective level of confidence for each class taxonomy discovered (the method in [26] even provides confidence levels for each data point within its assigned class). But, as alluded to earlier, these confidence measures must be accepted with caution because they are based on uncorroborated assumptions made about the underlying structure of the data.

The discussion above highlights the need, opportunities, and challenges associated with applying the data-driven scientific discovery paradigm to the microstructure classification problem. In this work, we aim to develop a general framework of microstructure classification in the unsupervised context which leverages the merits of plentiful data, automated systems, and human reason. We are not yet at the point where we can blindly trust the answers and inferences which automated systems produce, but we can use their suggestions as starting points for our own investigations in order to accelerate the discovery and design process.

2. Problem definition

The problem of microstructure classification in the unsupervised context (i.e. no available *a prori* label information) is schematically represented in Fig. 1. At the most general level, we would like to be able to present an unknown microstructure image dataset to an *"oracle"* capable of: elucidating the class taxonomy of the microstructure set and making reliable predictions about whether a specific image belongs to one of the *discrete* classes in the taxonomy. To build confidence in such an *oracle* it would also be necessary to have a reliable metric for stability of the discovered taxonomy—i.e. how stable is the discovered set of classes—as well as a reliable estimation of the error in the (multi-class) classifier itself. Ultimately, the fundamental problem is to categorize microstructures that have not yet been labeled by humans and the solution should make use, as much as possible, of automated tools for class discovery.

A more granular description of such framework involves the solution to several problems. First, the microstructure space should be featurized and the corresponding microstructure information should be reduced in order to improve computational performance of the classification algorithms and reduce as much as possible redundancy. It is important to note that since we want to make the process generalizable, the microstructure featurization scheme should not rely on microstructural features specific to any one material class. A second element of the framework is the robust discovery of the class taxonomy, that is, the intrinsic categorical organization of the microstructure space. Such a categorical organization should be stable against the choice of training/testing sets and must also be capable of carrying out this class discovery operation with as little *a priori* knowledge of the *number of discrete microstructure classes* in the set as possible. A third element of the framework is the deployment and evaluation of classification schemes capable of accurately predicting, with an objective measure of confidence, the category of a given microstructure instance. Finally, a proper error metric in this multi-class classification problem must be developed, noting that the evaluation of classification error against unlabeled data is a non-trivial task [15]. The ultimate goal is to have a predictor with a low error rate constructed over a class taxonomy with a high class confidence metric. If the error rate is low but our confidence in our class taxonomy's ability to represent the structural diversity of the material system of interest is also low, then the low error rate of the predictor has very little meaning. Likewise, we can be very confident that the discovered class taxonomy is sufficiently descriptive, but if the error rate of the predictor is high, then the usefulness of the framework is, again, questionable.

In order to demonstrate our framework, we require a large curated microstructure dataset. We decided to use a synthetic microstructure dataset generated with an elasto-chemical phase field model [27]—available in the Open Phase-field Microstructure Database (OPMD) [28]—developed by a subset of the co-authors of the present work. Several example microstructures to illustrate the diversity of the microstructures in the OPMD Database with reference to the experimental cases are shown in Fig. 2. The bene-

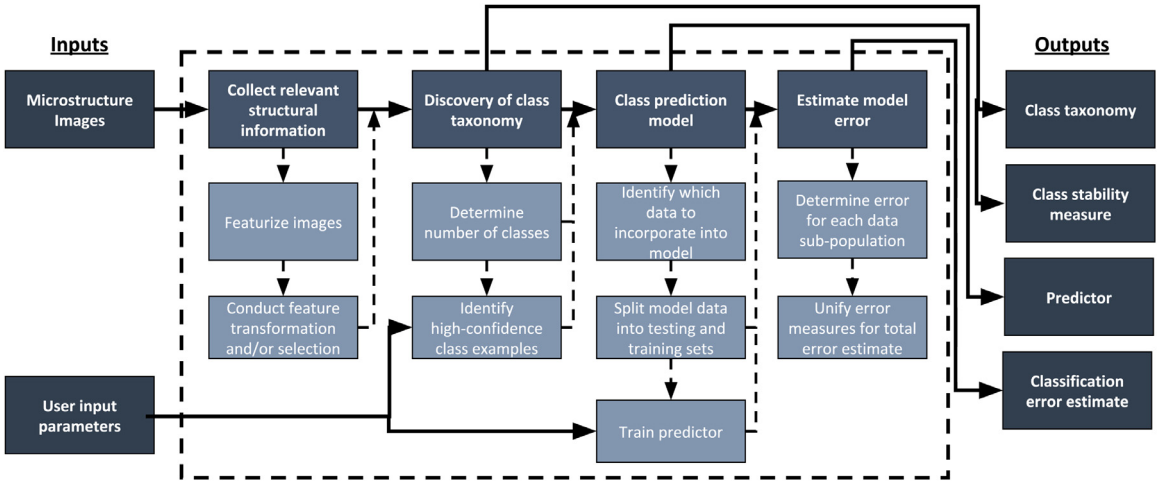


Fig. 1. Schematic representation of the unsupervised microstructure classification problem.

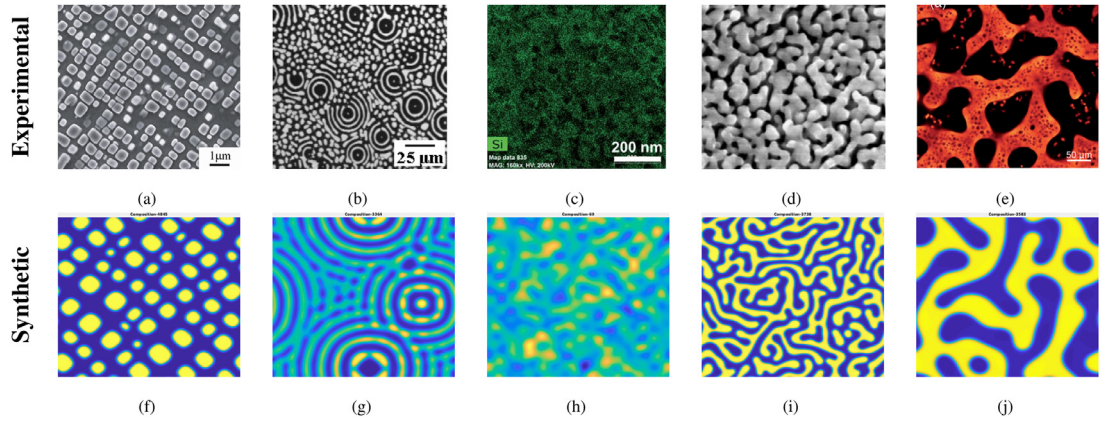


Fig. 2. Display of the structural diversity of the image dataset obtained by means of propagating uncertainty in input parameters of the phase-field equation [27,28]. Experimental microstructures of a) SEM micrograph of $\gamma + \gamma'$ structure of Co-9.2Al-10.2W(at.%) reproduced from [29] with permission, b) ring-shaped pattern and droplets appeared in Polystyrene, reproduced from Ref. [30] with permission, c) $Mg_2Sn_{0.3}Si_{0.7}$ after high energy ball milling and isothermal heat treatment reproduced from Ref. [31] with permission, and d) SEM micrograph of the $Cu_{50}Zr_{45}A_{l5}$ bulk metallic glass immersed in the 0.05 M HF solution for 1 days, reproduced from Ref. [32] with permission. d) CLSM micrograph of phase-separated gel systems composed of whey protein isolate and gellan gum incubated at 5 °C, reproduced from Ref. [33] with permission. e,f,g,h) Selected synthetic microstructures corresponding to the experimental observations (Figure is reprinted with permission from [27].)..

fits of this dataset include its relatively large size, constant length scale, and lack of predetermined labels (prior annotations have the potential to bias our evaluations of the class taxonomies discovered through our framework). Furthermore, these images are structurally diverse, which implies that the suggested class taxonomies have a high chance of containing multiple interesting classes. Each of the microstructure images are associated with a set of material parameters and processing conditions. Additionally, since the images are synthetic, we do not need to worry about noise, instrument limitations, or human error inherent in experimental imaging processes.

The dataset consists of over 200,000 of these images to choose from. Thus, in the interest of computational tractability, we initially narrowed our search to microstructures generated from 10,000 simulations evaluated after a constant time lapse with various combinations of the processing parameters. The list was further screened to only include microstructures which experienced phase separation. After applying the constraints above, we were left with a dataset of 1925 images. A human annotator could potentially discover clusters of similar-looking images in order to start building a class taxonomy, but the complexity of this task increases as the number of considered images and associated structural morphologies increases, causing considerable uncertainty as to which class an image may belong to.

3. Methods

3.1. Featurization and data reduction

In order to convert the generated microstructure images into useful information, the images must undergo the process of featurization. When the catalog of morphological motifs in the microstructural systems is well known, image characterization can be successfully conducted using various physical descriptor metrics such as volume fraction, eccentricity, aspect ratio, etc. [10,34,35]. However, the proposed framework is intended to work in cases in which no *a priori* knowledge of the discriminative structural features present in the dataset is available. In such cases, featurization based on approaches such as statistical metrics, bag of words or pre-trained CNN layers could be used.

Reduced order representations of statistical functions, such as the n-point correlation and lineal path functions, have been shown to be effective methods of featurization for the microstructure classification problem [6,9,36]. However, these methods require *a priori* knowledge about the microstructure (e.g. what phases are represented by what pixel values in the images) which we do not have.

Bag of visual words [37] approaches could be used, as demonstrated by DeCost and Holm in a microstructure classification setting [8]. While this technique does not require *a priori* knowledge

of the features most relevant for characterization, it does get very computationally expensive as it requires considerable user input to tune the hyper-parameters and select the appropriate clustering algorithms and corresponding cluster numbers used to build the visual dictionary's vocabulary.

Another approach to featurization is to use pre-trained CNN architectures, such as VGG [38], ResNet [39], and Inception [40]. In [14], DeCost et al. successfully used the VGG16 architecture to featurize hundreds of images in the UHCSDB database. With VGG16, image featurization is conducted by extracting information from the convolutional or max pool layers of the network. As with bag of visual words, this featurization approach does not require any *a rpriori* knowledge of the most discriminative structural attributes. Furthermore, there are no hyper-parameters to tune since the architecture is pre-trained. One downside of this method, however, is that the convolutional layers can present very high-dimensional feature spaces, which can be problematic for many classification algorithms. Another downside is that it is difficult to know exactly which layer in the pre-trained network would provide the most discriminative feature space, but it has been shown that deep CNNs trained on sufficiently large and diverse datasets can generalize well to categories of images on which they have not been trained [41], with studies in medical imaging [42], pest detection [43], and crack detection [44] successfully employing the last layer of the fifth convolutional block of the VGG16 architecture.

Given its advantages, and considering the success of prior efforts [42–44], in this work we featurize the microstructure image dataset via a pre-trained VGG16 architecture and extract the last layer of the fifth convolutional block. Two different workable and computationally tractable feature spaces are then created by applying two data reduction techniques to this very high-dimensional extracted layer. Data reduction on CNN convolutional layers is often performed through max pooling [45] or average pooling [46] (taking the maximum or average of each filter output, respectively). Wishing to have all elements of the convolutional layer contribute to the final feature space, we choose to use average pooling as our first dimension reduction method. For the second dimension reduction technique, we consider applying Principal Component Analysis (PCA) [47]. PCA coupled with CNN layer featurization is not nearly as common as max or average pooling, but it has been used in recent studies [48,49]. PCA not only constitutes a useful dimensional reduction tool, but its linear character makes it possible to project new data into the new orthogonal feature space, which is not possible with other non-linear dimensional reduction methods such as Multidimensional Scaling [50].

3.2. Class discovery

Once the dataset is featurized and reduced into a workable space, the class taxonomy must be discovered. Clustering methods are often used to find underlying data groupings in the unsupervised context, and can be divided into two broad categories: partitional and hierarchical. Partitional techniques, such as K-means, self organizing maps, and mixture-resolving algorithms require the user to specify the number of clusters (denoted as K) and return a single partition of the dataset into K groups. In contrast, hierarchical methods return nested groupings of the data that allow the user to choose the number of clusters through consideration of the defined distance between clusters or through inspection of the resulting dendrogram, a visual representation of the clustering structure [51]. Unfortunately, both categories of clustering methods have some serious shortcomings for the class discovery problem. For partitional methods, there must be *a priori* knowledge of the value of K, and this information is often not available. Hierarchical clustering sidesteps this issue, but it does make a big assumption that the class structure is hierarchical. Additionally, cutting the dendrogram based on visual inspection alone is a highly subjective exercise which can be very misleading [52].

To address these concerns, Monti et al. developed a technique known as consensus clustering [26]. The main idea behind the consensus clustering method is that the stability of a given number of clusters (K) can be evaluated through iterative resampling and clustering, and the optimal value of K for the data is that whose clusters are most robust to sampling variability. This method first requires the choice of an internal clustering algorithm. The feature space is then sub-sampled a prescribed number of times and clustered for each value of K being considered. For all pairs of data points i, j, the proportion of clustering runs in which i and j are grouped together when also sampled together is recorded, and this information, known as the consensus index for points i, j is stored in element (i, j) of the consensus matrix, M. Hierarchical clustering is then performed on a distance matrix defined as $\mathbf{1} - \boldsymbol{M}$, and the resulting dendrogram is cut at the specified value of K to provide final cluster assignments. The consensus matrix can also be used to calculate confidence measures for each proposed cluster and for each data point assigned to each cluster. For $k = 2, 3, \ldots, K$, the cluster consensus for cluster k is defined as the average consensus index for data point pair i, j both assigned to k. Similarly, the item consensus of data point i for cluster k is defined as the average consensus index of i with all points in k. These additional measures are extremely useful because they provide objective measures of confidence for the proposed class taxonomy and for each data point within each class, respectively.

A consensus matrix containing only 0′s and 1′s would imply perfect consensus and high level of robustness to sampling variability. Real data, however, rarely results in perfect consensus. In these cases, a method to obtain the optimal K number of clusters is necessary. Qualitative methods that rely on the assumption that the cumulative distribution function (CDF) of an ideal consensus matrix is a step function exist. However, they tend to be highly subjective and have issues with dividing unimodal data into apparently stable clusters, as shown by Senbabaoglu et al. [53]. An objective metric developed by Senbabaoglu et al. [53] is the proportion of ambiguously clustered pairs (*PAC*), defined as the fraction of consensus indices in the open interval $(x_1, x_2) \in [0, 1]$, and the optimal K is that with the lowest PAC. However, this method still does not test the null hypothesis of $K = 1$, and in [54], John et al. demonstrated that the PAC is often biased towards higher values of K. To rectify these concerns, John et al. formulated a technique known as Monte Carlo Consensus Clustering (M3C) for choosing the optimal K that includes a Monte Carlo reference procedure to eliminate bias towards higher values of K and to test the null hypothesis $K = 1$. In this work, as will be shown below, we used consensus clustering coupled with the M3C method for class discovery.

3.3. Classification and error estimation

Once the class taxonomy is discovered and confidence measures associated with class membership are assigned to each image, the next step is to build an automated classifier to accurately predict the class label of future images. While there are a number of supervised classification schemes[55], a major drawback is that the entire training set must be labeled and these labels must be "hard" (that is, class membership is mutually exclusive), which means that these algorithms cannot incorporate the item consensus values produced by the class discovery step. While fuzzy classifiers, trained with "soft" labels [56,57] could be used, in general there is the assumption that the sum of the "soft" labels for any training point over all possible classes must add to 1 (i.e. the labels are analogous to probability of class membership). This is a problem, however, because the sum of the item consensus values for a

given point over all clusters does not necessarily equal 1 because item consensus is not probability of cluster membership [26].

Semi-supervised classification methods provide a potential solution. These algorithms use both labeled and unlabeled data to train classification models [58]. Thus, we could use the item consensus information to break the data into two categories: high-confidence (labeled) data and low-confidence (unlabeled) data. In this way, we can still incorporate the item consensus information provided by the class discovery step. However, similar to unsupervised methods, there is always the concern that a poor matching of semi-supervised algorithm assumptions with latent data structure can lead to degraded classifier performance [59]. Furthermore, many semi-supervised methods are transductive, so they need to be paired with a supervised algorithm to train an inductive predictor. Recently, we addressed these shortcomings by constructing a semi-supervised framework that applies a collection of semi-supervised methods to a partially labeled training set, identifies the subset of unlabeled data points which receive a labeling consensus, adds this subset to the labeled data, and then trains a SVM over this appended training set [15]. In that work we also showed that adding this "safe" subset of the unlabeled data to the labeled training set did not deteriorate classification performance on the high-confidence data. Moreover, that work provided a method of semi-supervised error estimation, although in that work the classification problem was binary, and thus needs to be generalized to the multi-class case.

4. Implemented informatics framework

The following discussion delves into the details of and the connections between the computational tools which comprise the final framework for the classification of microstructure images in the unsupervised context. As outlined above, we start with featurizing the data.

4.1. Featurization using the VGG16 architecture

VGG16 is a pre-trained CNN proposed by Simoyan and Zisserman [38]. The full network is used for classification, but any image can be featurized in a very general fashion with no *a priori* knowledge of the important structural motifs by extracting information from internal convolutional or pooling layers. The convolutional layers are the output of employing various filters over the images, and the pooling layers reduce these high-dimensional volumes by performing a transformation on each filter slice. In this work, we input 384 x 384 pixel images and extracted the final layer of the fifth convolutional block using the keras application for python. This led to a 24 x 24 x 512- dimensional feature space, which required further dimensional reduction.

4.2. Dimension reduction

In this work, we used two different dimensional reduction approaches: In the first, *pooling*-based scheme, the 24 x 24 x 512 output of the VGG16 featurization process was down-sampled to a 1 x 512 feature space consisting of the average [46] of each 24 x 24 filter slice in the original feature space. We also used PCA [47] on the feature space generated from the VGG16 using scikit-learn [60]—the resulting scree plot is shown in Fig. 3. Application of PCA over the VGG16-derived feature space shows that even using the 100 principal components only explains about 36% of the variance in the data, but we still used this cutoff of 100 principal components as the maximum dimension size for the PCA-based dimension reduction approach.

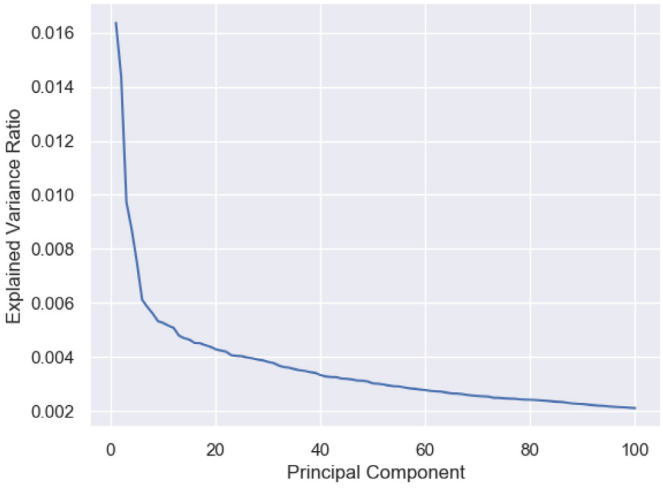


Fig. 3. The ratio of explained variance as a function of principal component.

4.3. Class discovery

As discussed above, consensus clustering [26] was used to provide automated class taxonomy suggestions for both feature spaces. Consensus clustering operates under the premise that the true clusters in the data should be robust to sampling variability; thus, the optimal value of K for a given clustering method should be that which produces the most robust clusters. The pseudo code is given in Algorithm 1.

Algorithm 1: Consensus Clustering.

Input: a set of items $D = \{e_1, e_2, \ldots, e_N\}$
a clustering algorithm *Cluster*
a resampling scheme *Resample*
number of resampling iterations H
set of cluster numbers to try, $\mathcal{K} = \{K_1, \ldots, K_{\max}\}$
for $K \in \mathcal{K}$ **do**
  $\boldsymbol{M} \leftarrow \emptyset$ {set of connectivity matrices, initially empty}
  for $h = 1, 2, \ldots, H$ **do**
    $D^{(h)} \leftarrow Resample(D)$ {generate perturbed versions of D}
    $M^{(h)} \leftarrow Cluster(D^{(h)}, K)$ {cluster $D^{(h)}$ into K clusters}
    $\boldsymbol{M} \leftarrow \boldsymbol{M} \cup M^{(h)}$
  end
  $\mathcal{M}^{(K)} \leftarrow$ compute consensus matrix from $\boldsymbol{M} = \{M^{(1)}, \ldots, M^{(H)}\}$
end
$\hat{K} \leftarrow$ best $K \in \mathcal{K}$ based on consensus distribution of $\mathcal{M}^{(K)}$s
$\boldsymbol{P} \leftarrow$ partition D into $\hat{K}$ clusters based on $\mathcal{M}^{(\hat{K})}$
return $\boldsymbol{P}$ *and* $\{\mathcal{M}^{(K)} : K \in \mathcal{K}\}$

As highlighted in Algorithm 1, the degrees of freedom afforded by this method include choice of internal clustering method, resampling scheme, number of resampling iterations, and the set of cluster numbers (K) to try. The output of this procedure is a set of consensus matrices where each consensus matrix ($\mathcal{M}^{(K)}$) corresponds to a considered cluster number. For observations i, j and cluster number K, the consensus index $\mathcal{M}^{(K)}(i, j)$ is the number of iterations in which i, j are clustered together divided by the number of iterations in which i, j are both sampled. Final cluster assignments are then determined by performing hierarchical clustering on the distance matrix $\mathbf{1} - \mathcal{M}^{(K)}$ and cutting the resulting dendrogram to produce K clusters. This result is often visualized using heatmaps (see Fig. 5 below for examples) where perfect consensus would look like K sharp blocks along the diagonal.

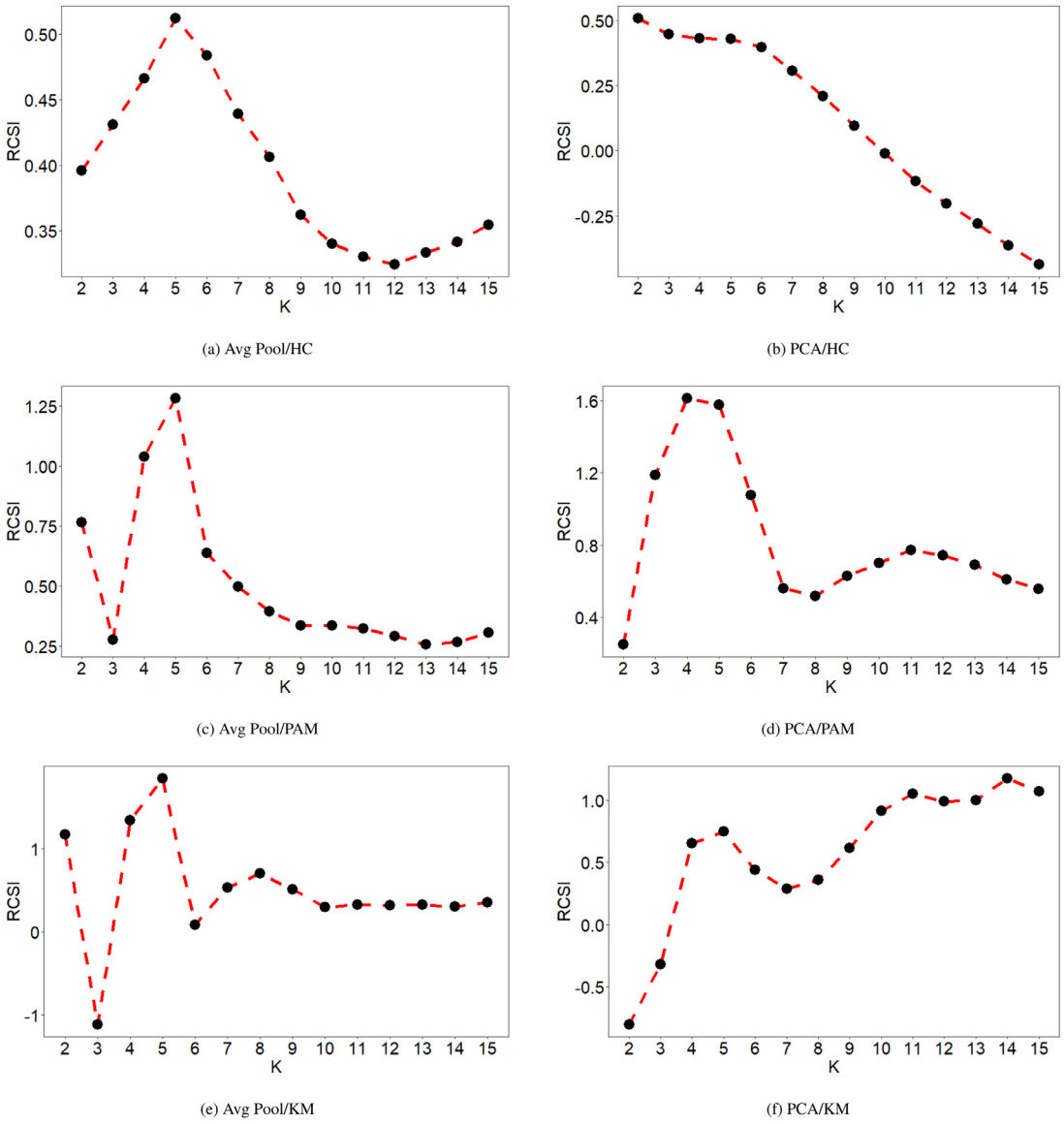


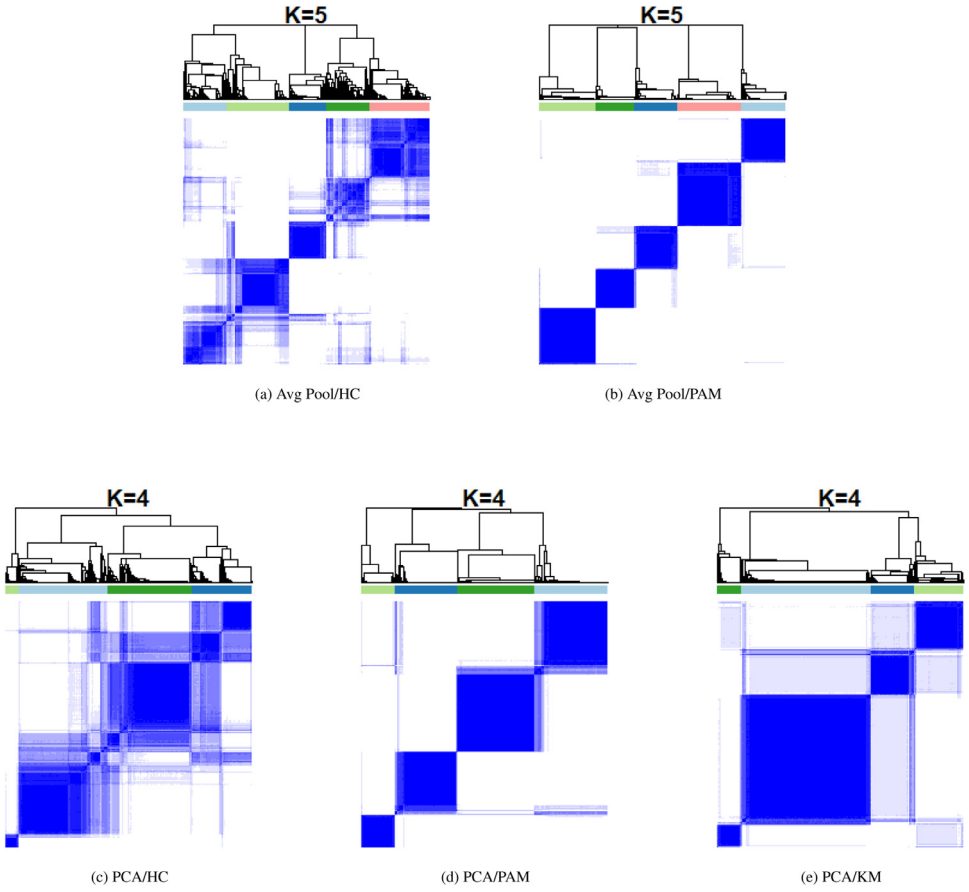
Fig. 4. RCSI plots for each feature space/clustering algorithm combination.

As mentioned previously, objective metrics which assess the stability of each suggested cluster and each observation relative to each cluster can be calculated from $\mathcal{M}^{(K)}$. For $k = 1, 2, \ldots, K$, the cluster consensus for cluster k is defined as the average consensus index between observations both assigned to cluster k. Thus, the closer the cluster consensus is to 1, the more confident we are that this cluster represents true structure in the data. Similarly, as a function of observation i and cluster k, the item consensus is defined as the average consensus index between i and all observations in k (excluding i). Thus, the closer the item consensus is to 1 for a given i and k, the more confident we are that observation i belongs to cluster k. Subsequent sections will provide details to the application of both of these metrics to the problem at hand.

For this study, we decided to employ consensus clustering with an 80% subsampling resampling scheme, 1000 iterations, and a range of cluster numbers from 2 to 15. Once the consensus matrices were computed, average linkage was used for the outer hierarchical clustering step. Three internal clustering algorithms – K-means [51], partitioning around medoids (PAM) [61], and hierarchical clustering with Ward linkage [62] – were used for both feature spaces, resulting in six feature space/clustering algorithm combinations. All consensus clustering calculations were performed using the ConsensusClusterPlus package in R [63].

As mentioned above, consensus clustering works as a meta-approach towards class discovery, which in principle can use different approaches to clustering. In this work, we used three different clustering methods. Each of the three clustering methods make different assumptions about the structure of the data; thus, each feature space/clustering algorithm pair can result in very different suggested class taxonomies. Note that in this study, all of the internal clustering used euclidean distance as the dissimilarity measure.

In order to determine the optimal cluster number K for each feature space/clustering algorithm combination, we used the Monte Carlo Consensus Clustering (M3C) technique [54]. M3C enhances the Monti et al. consensus clustering method [26] by providing statistically rigorous procedures for testing the null hypothesis $K = 1$ and for choosing the optimal cluster number K. This is accomplished by using Monte Carlo methods to generate a collection of null datasets with no clusters ($K = 1$) and with the same feature correlation structure as the real data. Traditional consensus clustering is used on each reference dataset, and the proportion of ambiguously clustered pairs (*PAC*, described above) is calculated for



(a) Avg Pool/HC (b) Avg Pool/PAM

(c) PCA/HC (d) PCA/PAM (e) PCA/KM

Fig. 5. Consensus clustering heat maps for the chosen K for each feature space/clustering algorithm combination which passed the sanity check.

all cluster numbers of interest for each dataset. This creates empirical distributions of PAC values for data with the same correlation structure as the real dataset, but where $K = 1$. Consensus clustering is also performed on the real data and the same PAC metric is calculated. From here, the hypothesis test flows quite naturally: the null hypothesis is that the PAC score comes from data with a single cluster, and the alternative is that the PAC score does not come from data with a single cluster. Thus, extreme values of the PAC score for the real data relative to the empirical distribution created from the null datasets indicate multi-cluster structure.

These reference PAC scores are also used to calculate the Relative Cluster Stability Index (RCSI), an objective metric metric used to rank the clusters returned from the cluster numbers of interest in order to choose the optimal K. For each cluster number K, the RCSI is expressed as

$$\text{RCSI}_K = \log_{10}\left(\frac{1}{B}\sum_{b=1}^{B} \text{P}_{ref_{K,b}}\right) - \log_{10} \text{P}_{real_K} \tag{1}$$

where B is the number of generated reference datasets, $\text{P}_{ref_{K,b}}$ is the PAC score for null dataset b generated for cluster number K, and P_{real_K} is the PAC score for the real data for cluster number K. According to John et al., incorporating the reference PAC scores removes the bias toward higher values of K exhibited by the original PAC metric. In this work, we used the M3C method with 100 reference datasets, 100 consensus clustering iterations, an interval of (0.1,0.9) for calculation of the PAC, and a significance level of 0.05 for the hypothesis test (all other parameters for consensus clustering were identical to those given above). All RCSI values and p-values (both empirical and those derived from beta distribution estimations) were calculated using the M3C package in R [64]. Plots of the RCSI are provided below in Fig. 4.

Once all of the RCSI and p-values were calculated for the six feature space/clustering algorithm combinations, we needed to choose the best K for each combination. While a reasonable decision would be to simply go with the class taxonomy associated with the highest RCSI value, we recognize that blindly trusting the suggestions of this automated process could produce results which appear nonsensical under visual inspection. Furthermore, while very small clusters may truly exist in the data, we cannot train a well-functioning classifier with too few examples of a class. With these concerns in mind, we established the procedure presented in Algorithm 2 to choose the optimal K for each combination.

Note that if none of the suggested class taxonomies corresponding to the cluster numbers in $\mathcal{K}$ pass the sanity check, it is concluded that there is poor agreement between the feature space and clustering algorithm for the application at hand, and this combination is discarded.

Once the optimal cluster numbers were determined for each feature space/clustering algorithm combination, we needed to separate the data into high-confidence and ambiguous categories in preparation for semi-supervised classification. To do so, we used the item consensus corresponding to the cluster that each observation was assigned to. Since item consensus values are not probabilities of cluster membership, using a constant threshold value (i.e. all observations with an item consensus corresponding to their assigned cluster below this value is ambiguous) could completely eliminate some clusters and leave others almost completely intact. Thus, we decided to label the bottom third of each cluster (about 643 observations in total) as ambiguous (this decision also sim-

Algorithm 2: Sanity Check.

Data: set of cluster numbers $\mathcal{K} = \{K_1, \ldots, K_{\max}\}$ and associated RCSI_K, p-value$_K$, and data partition P_K for each $K \in \mathcal{K}$
item consensus values for each cluster $k = 1, \ldots, K$ in P_K for each $K \in \mathcal{K}$

```
check ← False {optimal K not yet found}
while (check == False) or (K == ∅) do
    K̄ ← K ∈ K such that RCSI_K is maximized
    if p-value_K̄ > 0.05 then
        K ← K − {K̄}
    else if any cluster k = 1, ..., K̄ has 50 or fewer members then
        K ← K − {K̄}
    else
        visual ← Visual_Inspection(K̄)
        if visual == False then
            K ← K − {K̄}
            K_opt ← none
        else
            check ← True
            K_opt ← K̄
end
return K_optFunction Visual_Inspection(K):
    visual ← True
    for k = 1, ..., K do
        Visually inspect the images in cluster k with the 10 highest item consensus values
        if the 10 images do not appear to share structural similarities then
            visual ← False
        end
    end
    Further inspect images within and across clusters
    if clusters appear redundant or too diverse then
        visual ← False
    end
    return visual
end
```

plifies the semi-supervised error estimation discussed above). The remaining labeled data was then split into training and validation sets consisting of 75% (about 961 observations) and 25% (about 321 observations) of the high-confidence labeled data, respectively.

4.4. Classification framework

As mentioned in the previous section, we are using the semi-supervised classification framework proposed in [15]. This framework uses both high-confidence labeled and ambiguous unlabeled data to train and assess five classifiers using a variety of semi-supervised learning methods as transductive predictors for the ambiguous data and trains support vector machines (SVM) [65] over these results to produce five inductive classifiers. The four semi-supervised methods are the Modified Yarowsky algorithm (MY) [66], Safe Semi-Supervised Support Vector Machines (S4VM) [67], Label Propagation [68], and COP K-MEANS [69]. Four of the classifiers are trained over the original high-confidence training set along with the output of a single semi-supervised method applied to the ambiguous set, and the fifth is trained over the labeled training set along with the subset of the ambiguous set which receives a labeling consensus from all four semi-supervised methods. This fifth classifier is known as the "updated" classifier. It must be noted that the framework presented in [15] was used in a binary classification problem, although all aspects, except for the error estimation portion are readily generalizable to the multi-class problem. The following section details how the error estimation method can be generalized to the multi-class case.

4.5. Error estimation

The semi-supervised error estimation technique established by Kunselman et al. requires independent estimates of classification error on the labeled and unlabeled sub-populations of the data. These independent estimates are then combined into a total error estimate through a convex combination in which the associated weights are the probabilities of a given feature vector being sampled from the corresponding sub-population. That is,

$$\hat{\epsilon} = \hat{P}(\boldsymbol{X} \in \pi_U)\hat{\epsilon}_U + \hat{P}(\boldsymbol{X} \in \pi_L)\hat{\epsilon}_L \tag{2}$$

where $\hat{\epsilon}$ is the total error rate estimate, $\hat{P}(\boldsymbol{X} \in \pi_U)$ is the estimate of the probability of a given feature vector X belonging to the unlabeled sub-population π_U, $\hat{P}(\boldsymbol{X} \in \pi_L)$ is the estimate of the probability of X belonging to the labeled sub-population π_L, and $\hat{\epsilon}_U$, $\hat{\epsilon}_L$ are error rate estimates of the unlabeled and labeled sub-populations, respectively. Since we decided that one third of the data would be ambiguous, we have $\hat{P}(\boldsymbol{X} \in \pi_U) = 1/3$ and $\hat{P}(\boldsymbol{X} \in \pi_L) = 2/3$.

In this work, the labeled error estimate was determined for each feature space/clustering algorithm/classifier combination using the corresponding labeled validation set derived from the data with matching feature space and clustering algorithm. We cannot claim that this error estimate is completely unbiased since each validation set was used for the five classifiers trained on each feature space/clustering algorithm combination, but it should be less biased than estimates which employ the training data. Furthermore, all validation sets contain approximately 321 observations, so the variance should be small.

While the labeled sub-population error estimates were quite straight-forward to compute, the unlabeled error estimates required a more complicated approach. In accordance with [15], in this work the basis of the unlabeled sub-population error estimation was the constrained optimization approach introduced by Platanios et al. in [70], but generalized to the multi-class context, using the argument below:

As in [70], let A be a set of classifiers, a_A be the probability that all of the classifiers in A assign the same label (i.e. the agreement rate) and let e_A be the probability that all of the classifiers in A make an error (not necessarily the same error). Additionally, we will introduce C_A, the event that all classifiers in A assign the correct label. Through the rules of probability, we note that

$$1 = P(C_A) + P(\bar{C}_A). \tag{3}$$

That is, either all of the classifiers assign the correct label or at least one of them assigns the wrong label. Through application of the inclusion-exclusion principle, we see that

$$P(\bar{C}_A) = -\sum_{k=1}^{|A|}\left[(-1)^k \sum_{\substack{I \subset A \\ |I|=k}} e_I\right]. \tag{4}$$

Now, let p_A be the probability that all of the classifiers in A make the same error. Then

$$P(C_A) = a_A - p_A. \tag{5}$$

Substituting Eqs. 4 and 5 into 3, we have

$$1 = a_A - p_A - \sum_{k=1}^{|A|}\left[(-1)^k \sum_{\substack{I \subset A \\ |I|=k}} e_I\right], \tag{6}$$

Table 1
Master table of cluster consensus, optimal cluster number, beta distribution p-values, and classifier total error rates for all six feature space/clustering algorithm combinations. The minimum cluster consensus and minimum classifier total error rate will be used as objectives to make the final design choice.

Feature Space	Avg Pool	PCA	Avg Pool	PCA	PCA
Cluster Alg	HC	HC	PAM	PAM	KM
Optimal K	5	4	5	4	4
Optimal K p-value	5.30e-33	1.35e-67	8.74e-10	3.17e-46	1.19e-18
Cluster 1 Consensus	0.7207	0.7391	0.9439	0.8710	0.9604
Cluster 2 Consensus	0.9006	0.7133	0.9138	0.9067	0.9162
Cluster 3 Consensus	0.7132	0.8869	0.9734	0.9606	0.9287
Cluster 4 Consensus	0.6534	0.8297	0.9891	0.9869	0.8908
Cluster 5 Consensus	0.7963	N/A	0.9511	N/A	N/A
Min Cluster Consensus	**0.6534**	**0.7133**	**0.9138**	**0.8710**	**0.8908**
MY Total Error	0.0170	0.0648	0.0191	0.0288	0.0382
S4VM Total Error	0.0137	0.0632	0.0181	0.0283	0.0382
LP Total Error	0.0578	0.1635	0.0668	0.0983	0.1039
CKM Total Error	0.1001	0.0747	0.0611	0.1253	0.0334
Updated Total Error	0.0394	0.0485	0.0403	0.0210	0.0225
Min Total Error	**0.0137**	**0.0485**	**0.0181**	**0.0210**	**0.0225**
Classifier	**S4VM**	**Updated**	**S4VM**	**Updated**	**Updated**

Table 2
Classifier labeled sub-population error rate estimates for each feature space/clustering algorithm combination.

Feature Space	Avg Pool	PCA	Avg Pool	PCA	PCA
Cluster Alg	HC	HC	PAM	PAM	KM
MY Labeled Error	0	0.0312	0.0062	0.0031	0.0062
S4VM Labeled	0	0.0312	0.0062	0.0031	0.0062
LP Labeled Error	0	0.0592	0.0187	0.0374	0.0312
CKM Labeled Error	0.0125	0.0405	0.0093	0.0405	0.0125
Updated Labeled Error	0	0.0374	0.0062	0.0093	0.0093

which reduces to the agreement rate constraints in [70] when $e_A = p_A$ (i.e. the binary class case). Since p_A is a probability of all classifiers in A making an error in a specific fashion, we can include the additional inequality constraints

$$p_A \leq e_A. \tag{7}$$

In this work, we used the method of Platanios et al. (including the objective which aims to minimize error rate dependence), but we replaced the agreement rate constraints with Eq. 6, added the constraints given in Eq. 7, and added the appropriate p_A variables to the design vector. Furthermore, in line with Kunselman et al., we added the constraint that at least one of the individual error rates must be less than 0.5. The optimization was carried out using sequential quadratic programming in MATLAB with 3000 different starting points to avoid local minima.

We note that a collection of scripts giving an example of this workflow on an abridged dataset can be found in [71].

5. Results and discussion

Table 1 provides the optimal cluster number, cluster consensus values, p-values associated with optimal cluster numbers, and semi-supervised classification error estimates for each feature space/clustering algorithm combination which had a suggested class taxonomy that passed the sanity check (for a breakdown of the error estimates for the labeled and unlabeled subpopulations, see Table 2 and Table 3, respectively). All of the suggested class taxonomies for the Avg Pool/KM combination failed the sanity check ($K = 2, 3$ generated clusters with too much structural diversity; $K = 4, 5, 6, 8$ generated clusters in which the 10 microstructures with the highest item consensus values looked very different; $K = 7$ generated very redundant clusters; and $K = 9, 10, 11, 12, 13, 14, 15$ generated at least one class with fewer than 50 members). It is worth noting that the consensus clustering implementation that we used gave a warning that the K-means algorithm failed to converge for the data in the feature space produced through average pooling, and no such warning was given for any other combination. While it is difficult to determine precisely why unsupervised methods perform poorly, this struggle to converge is potential evidence of poor agreement between latent data structure and clustering algorithm assumptions (e.g. the higher dimensional data could have been an obstacle for K-means). This poor agreement could have caused the class discovery process to terminate prematurely, resulting in no class taxonomies which could pass the sanity check.

Interestingly, the two remaining combinations with the feature space produced through average pooling have optimal $K = 5$, and all three combinations with the feature space produced through PCA have optimal $K = 4$. On the surface, it would appear that the feature space has a much greater influence on the cluster number than the clustering algorithm. However, as illustrated in the discussion above, we must remember that the sanity check can be quite restrictive.

Fig. 4 provides plots of RCSI as a function of K for all six feature space/clustering algorithm combinations. Remember that the RCSI is the objective metric which ranks the cluster numbers and guides the order of our sanity check. We see that for all three clustering algorithms operating in the feature space made from average pooling, $K = 5$ had the highest RCSI. Even though the final class taxonomy corresponding to $K = 5$ for the K-means combination did not pass the sanity check, the consistency in the RCSI across three clustering algorithms provides some evidence that, for the average pool feature space, five robust clusters could exist. Note that $K = 5$ passed the sanity check for both the hierarchical and partitioning around medoids clustering algorithms in this feature space; thus, no other cluster numbers had to be tested.

The RCSI plots for the PCA feature space tell a much different story. We see that each clustering algorithm produced a maximum RCSI at a different value of K and that the partitioning around medoids clustering method was the only one whose K corresponding to the highest RCSI passed the sanity check. When hierarchical clustering was used, smaller cluster numbers had higher RCSI values, but $K = 2$ and $K = 3$ had microstructures that looked very different in the same cluster with very high item consensus values. In contrast, when K-means was used, larger cluster numbers had

Table 3
Classifier unlabeled sub-population error rate estimates for each feature space/clustering algorithm combination.

Feature Space Cluster Alg	Avg Pool HC	PCA HC	Avg Pool PAM	PCA PAM	PCA KM
MY Unlabeled Error	0.0511	0.1321	0.0449	0.0800	0.1021
S4VM Unlabeled	0.0412	0.1274	0.0418	0.0786	0.1021
LP Unlabeled Error	0.1734	0.3721	0.1631	0.2201	0.2495
CKM Unlabeled Error	0.2755	0.1431	0.1647	0.2948	0.0754
Updated Unlabeled Error	0.1181	0.0707	0.1086	0.0443	0.0489

the highest RCSI index values, but at least one of the clusters produced for $K = 5, 10, 11, 12, 13, 14, 15$ (all cluster numbers with RCSI values greater than that at $K = 4$) contained 50 or fewer members. However, visual inspection showed that the proposed clusters with a small number of members were not unreasonable or overly redundant, implying that there could be small groups of outliers in this dataset. However, as stated above, clusters which are too small could create a very large class imbalance, and it is very difficult to train well-functioning classifiers under this condition. Note that $K = 4$ was the cluster number with the largest RCSI that passed the sanity check for all clustering algorithms in this feature space, making it the optimal cluster number.

The fourth row of Table 1 shows that all of the p-values associated with the optimal cluster numbers chosen through the sanity check are significantly less than 0.05. Thus, all proposed class taxonomies which passed our sanity check also resulted in a rejection of the null hypothesis that the data comes from data with a single cluster. Note that all of these p-values were calculated using the estimated beta distribution method detailed in [54] rather than the empirical method. This is because, for all optimal clusters, there were no reference PAC scores less than or equal to the real PAC score, and the empirical p-value calculation relies on at least one reference PAC score being less than or equal to the real score.

As we continue to move down the rows in Table 1, we see that both sets of clusters found with hierarchical clustering produced clusters with relatively low cluster consensus values. This observation is supported graphically in Fig. 5 where the heat maps corresponding to the hierarchical clustering method have more noise than those heat maps associated with the other two clustering methods. This implies that the objective level of confidence for at least one of the clusters found with hierarchical clustering is relatively low. In contrast, the cluster consensus values for the cluster sets produced by partitioning around medoids and K-means are quite high. In further contrast to the hierarchical clustering combinations, the average pool feature space has a higher minimum (worst case) value than the PCA feature space for partitioning around medoids combinations.

The last important conclusions to draw from Table 1 come from the information on classifier performance. As a general trend, the SVMs trained using the labeling results from the MY and S4VM methods and from the consensus of all four semi-supervised methods tended to have lower total error rate estimates than those classifiers trained using the results from LP and CKM (the one exception is CKM for the PCA/KM combination). Another noteworthy trend is that the classifiers trained with the results from MY and S4VM consistently had lower error rates for the average pool feature space (compared to the PCA feature space) for the partitioning around medoids and hierachical clustering combinations. Furthermore, the classifiers trained on the S4VM results in the average pool feature space all had the lowest total error rate estimates while the classifiers trained on the labeling consensus had the lowest rates for the PCA feature space. However, for all feature space/clustering algorithm combinations, the updated classifiers never had a total error rate estimate above 0.0485 whereas a classifier trained on S4VM results did reach an estimated error rate of 0.0632.

As stressed by Kunselman et al., the above discussion highlights the importance of considering multiple semi-supervised methods for classification. Some methods may be particularly suited to a given problem (e.g. S4VM for the average pool feature space combinations) while some methods may be quite unsuitable for the problem at hand (e.g. LP for the PCA/HC combination). On the other hand, as with the PCA feature space combinations, the consensus among multiple methods could lead to the best performance.

Lastly, we see from Tables 2 and 3 that most of the total classification error for all feature space/clustering algorithm combinations comes from the unlabeled sub-population. On the surface this may seem troubling, but we must remember that most methods of classification error estimation do not take high- and low-confidence label assignments into account, and it makes sense that the low-confidence or ambiguous data would be the source of a large portion of the error relative to the high-confidence data.

6. Choosing the final design of unsupervised microstructure classifier

The design choices consist of all feature space/clustering algorithm/classifier combinations. As mentioned above, the final design choice is based on the multi-objective approach of optimizing both some metric of confidence in the discovered class taxonomy and some measure of classifier performance. For the class confidence metric, we decided to use the cluster consensus values produced through the consensus clustering process. Because the optimal number of clusters varies between feature space/clustering algorithm combinations and because the numbering of clusters within a suggested class taxonomy is arbitrary (i.e. it makes no sense to compare the cluster associated with $k = 1$ across feature space/clustering algorithm combinations), some statistic must be calculated over the cluster consensus values associated with each combination before any comparisons can be made. Deciding that we wanted to optimize a worst case measure, we took the minimum of the cluster consensus distribution corresponding to each feature space/clustering algorithm combination. Whereas a weighted average could mask one or two low-confidence clusters, using the minimum cluster consensus as an objective to maximize can provide a clear warning that at least one questionable cluster could exist.

For the classifier performance objective, we chose to minimize an error metric. As displayed in Table 1, the semi-supervised error estimation method produced an overall error estimate for five different classifiers. Since the final design requires only one prediction model, we chose to consider only the classifier with the lowest total error estimate for each feature space/clustering algorithm combination. This narrowed down our design choices to five feature space/clustering algorithm/classifier combinations.

The two objectives associated with each of these five designs are plotted in Fig. 6. To further assist in our decision-making process, Table 1 is summarized graphically in a radar chart in Fig. 7. We see from Fig. 6 that there are only two non-dominated solutions (emphasized in the plot with the blue ellipse), and the information in Table 1 tells us that, moving from low to high er-

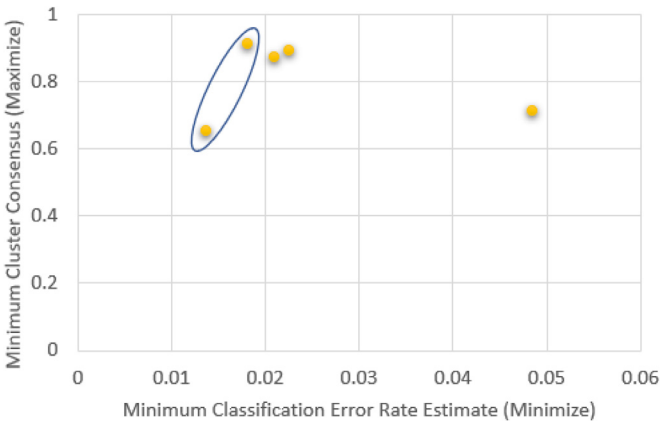


Fig. 6. Plot of both objectives for each of the considered designs. From the information in Table 1, we see that the Avg Pool/HC/S4VM and Avg Pool/PAM/S4VM combinations are the non-dominated solutions.

ror estimates, these solutions are the Avg Pool/HC/S4VM and Avg Pool/PAM/S4VM combinations, respectively. We now must choose between these two designs. Both options have quite low total error rate estimates (less than 2%), but the error estimate of the Avg Pool/HC/S4VM combination is more optimal. Furthermore, inspection of Fig. 7 shows us that classifiers trained on the suggested class taxonomy for the Avg Pool/HC combination tend to have higher accuracies than those trained on the Avg Pool/PAM class taxonomy. This could suggest that classifiers trained on the Avg Pool/HC class taxonomy are more robust to label changes (or even incorrect label assignments) for the more ambiguous microstructures in the training set.

However, we cannot ignore the fact that the minimum cluster consensus value for the Avg Pool/HC/S4VM solution is significantly lower than that of the Avg Pool/PAM/S4VM design. Indeed, the Avg Pool/HC/S4VM design actually has the lowest minimum cluster consensus value of all considered designs displayed in Fig. 6. As mentioned above, this is a warning to us that at least one of the proposed clusters could be questionable. To explore this concern, we go back to the high-confidence examples of each class and scrutinize them more closely. Fig. 8 displays the ten microstructures for each cluster with the highest ten item consensus values corresponding to that cluster. We note that Cluster 4 has the lowest cluster consensus value. From Fig. 8, we can see that these images show very distinct structural patterns upon which a class could be defined. That is, Cluster 1 contains bicontinuous or branched, tortuous motifs; Cluster 2 boasts a darker matrix with a plethora of small, light particles; the images representing Cluster 3 have a scattering of light and/or dark particles in a mid-gray matrix; Cluster 4 consists of mid-sized, light, circular and ellipsoidal particles in a dark matrix; and finally, the images of Cluster 5 show an assortment of highly circular light particles in a dark matrix where the contrast between particle and matrix is relatively high.

From this limited examination, it appears that the warning provided by the low cluster consensus values could be a false alarm. To be thorough, we examine the Avg Pool/PAM/S4VM design in a similar fashion and compare. The example microstructure images are given in Fig. 9. Clusters 1 and 3 share structural similarities with their counterparts in Fig. 8; the structural elements in the images representing Cluster 2 consist mainly of small-to-mid-sized light, circular or ellipsoidal particles in a dark matrix; Cluster 4 has light particles of a variety of sizes and shapes which are very space-filling; and Cluster 5 contains larger, more regularly shaped light particles in a dark matrix. Although these general trends are discernible (which is why this class taxonomy passed the sanity check) there are images which look somewhat out of place. For example, the seventh image from the left in Cluster 1 does have elongated domains, but they appear as striations within the particle-like structures (see Fig. 10), and it could be argued that the first image from the left in Cluster 4 is more similar to the images in Cluster 1.

From this visual comparison, it is difficult to say why the cluster consensus values for the Avg Pool/HC/S4VM design are so low. It could be that our visual inspection was too superficial or that cluster consensus is not an extremely helpful method of comparison across different clustering algorithms. It is also possible that there is a complex level of structural features present in this dataset that our objective framework is finding that is not necessarily detectable by human inspection. Regardless, in choosing our

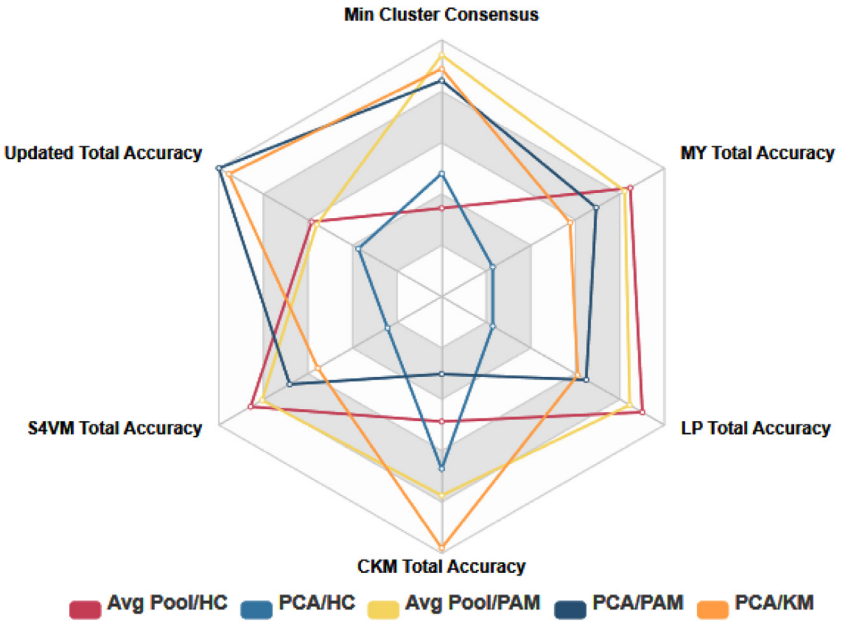


Fig. 7. Radar chart of metrics of interest for each feature space/clustering algorithm combination which passed the sanity check. Note that for ease of visualization and interpretation, accuracy estimates are displayed instead of error estimates, and the data along each axis has been scaled to the same mean and variance.

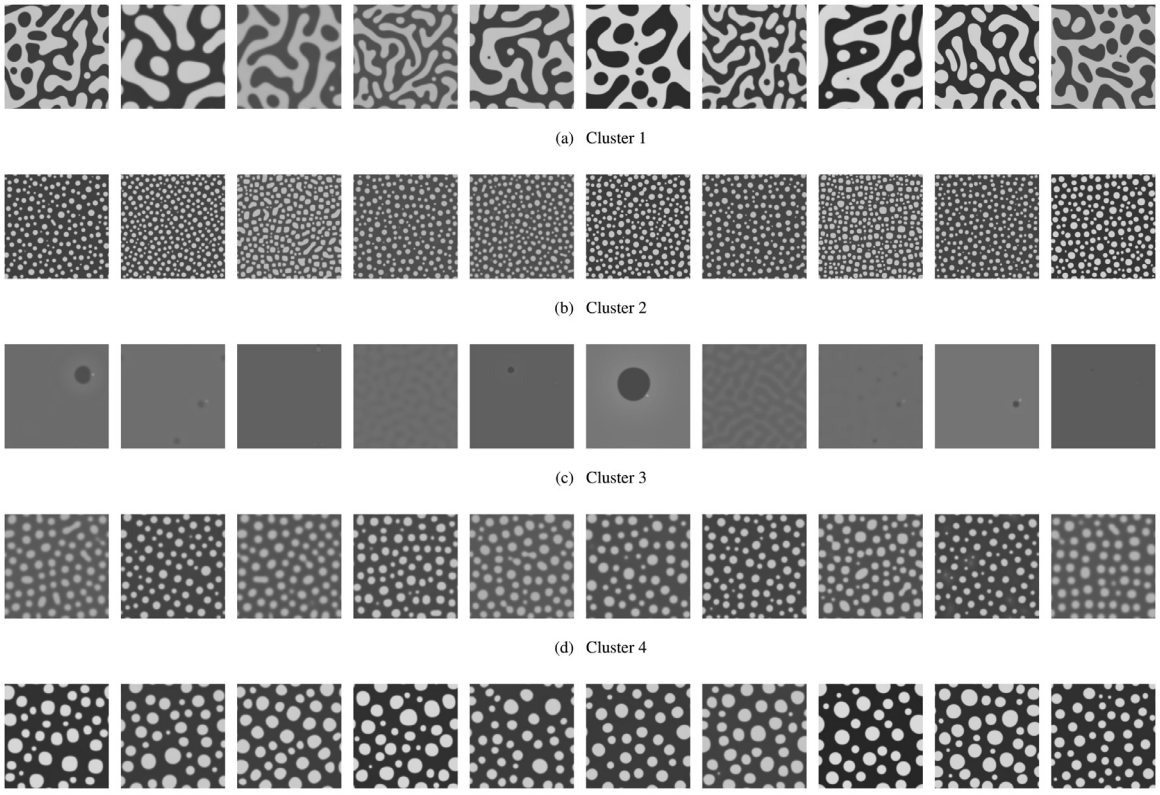


Fig. 8. Microstructure images with the ten highest item consensus values for each of the five clusters found for the Avg Pool/HC combination.

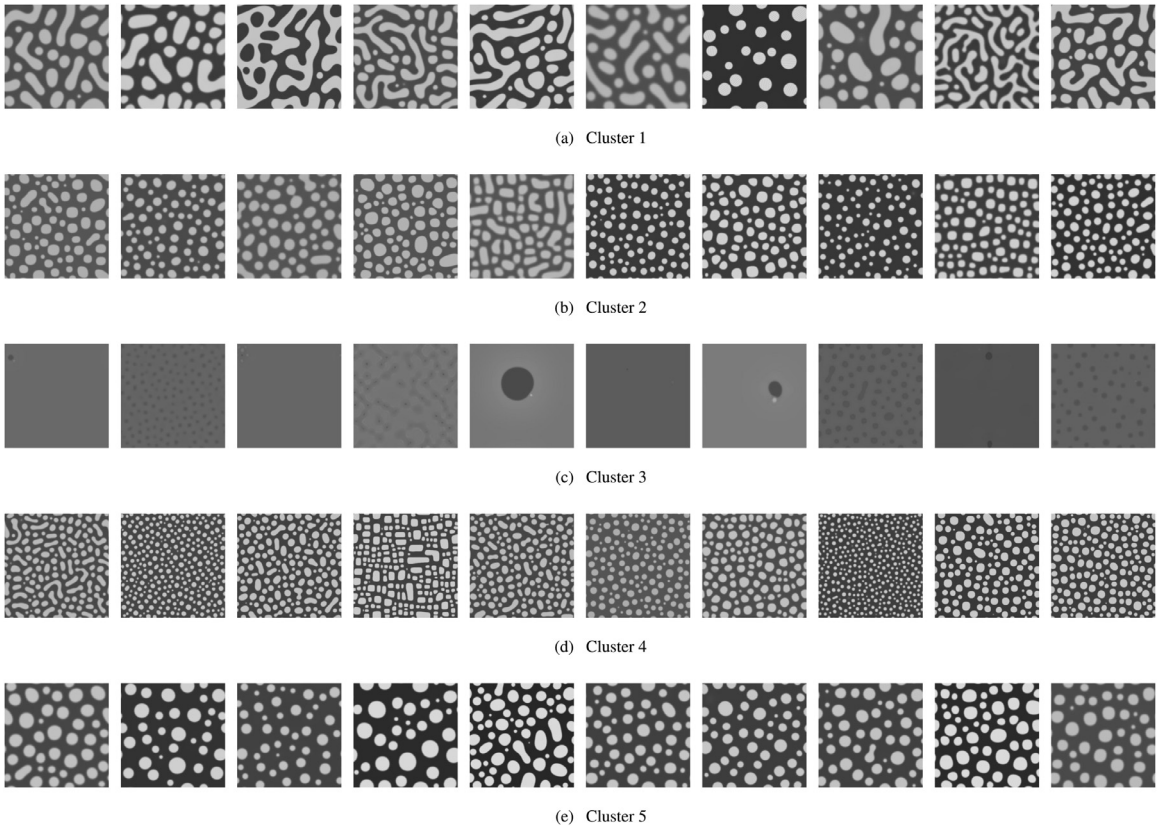


Fig. 9. Microstructure images with the ten highest item consensus values for each of the five clusters found for the Avg Pool/PAM combination.

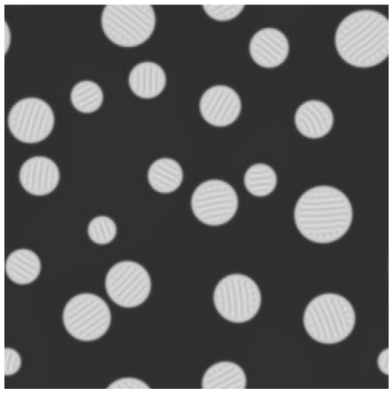


Fig. 10. A zoomed-in view of the suspicious microstructure image in Cluster 1 of Fig. 9.

final design we must remember that the premise of this study is to use data-driven methods to suggest class taxonomies and to guide our inspection and classification of these suggestions. That is, the automated framework leads us to a small number of acceptable solutions in an objective and timely manner, but in the end we still want to choose a design with a class taxonomy that could have been discovered through visual inspection (i.e. the structural similarities that define a class have to make sense to a human). Since the Avg Pool/HC/S4VM design has the lowest classification error rate and high-confidence class representatives which show more consistent morphological trends than those of the Avg Pool/PAM/S4VM solution, we choose the Avg Pool/HC/S4VM framework as our final design, *for this specific microstructure set*

7. Conclusions and future work

The aim of this work was to develop a data-driven framework for microstructure classification in the unsupervised context. While we are not yet at the point that we can blindly trust the decisions made by automated systems for completely unannotated data, we can certainly use their suggestions as starting points in our own investigations in order to accelerate the process of materials discovery and design. In this work, we rigorously established the forward mapping from raw microstructure images to class taxonomies to trained classifiers and finally to performance metrics. Pre-trained CNN architectures allow for informative image featurization which requires no *a priori* knowledge of important structural motifs, and dimension reduction techniques such as average pooling and PCA transform the massive outputs of CNN architectures into workable spaces. Once the data is transformed, consensus clustering offers a method of class discovery which does not require any prior knowledge of the class taxonomy. Furthermore, this method provides valuable objective confidence measures for each suggested cluster and for each assigned data point. Choosing the optimal K from the results of consensus clustering has historically been more of an art than a science, but we are confident that the RCSI ranking produced through M3C provides a mathematically rigorous mechanism for guiding class taxonomy decision-making. Semi-supervised classification allowed us to leverage the confidence measures for each assigned data point in order to produce more accurate classifiers for these suggested class taxonomies, and we were able to assess the accuracy of these classifiers through a novel generalization of an established semi-supervised error estimation technique to the multi-class problem. Lastly, the class discovery and classification processes produced performance metrics which guided our exploitation of the design space by narrowing down the candidate pool to two non-dominated solutions, and our choice of final design corresponded to that solution with both a low classification error and a class taxonomy which shows consistent morphological trends within and distinct structural differences between classes.

Although the present results are promising, we concede that our sanity check was quite constrained by requiring all classes in any passing class taxonomy to be of a certain size, and we would like to find a method of incorporating extremely small classes into the classifier training process. We postulate that this could be achieved through some sort of rare event simulation [72,73] which could create more examples of these smaller classes for training. Alternatively, resampling schemes for improved class unbalancing could be used. Additionally, finding a more mathematically rigorous metric for class confidence which has been proven to be appropriate for comparing suggested class taxonomies across clustering algorithms would help make the design space exploitation and final design choice processes more robust and less subjective, respectively.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

CK acknowledges the support of the National Science Foundation's Research Traineeship (NRT) program, D^{3}EM, under Grant No. NSF-DGE-1545403. RA would like to acknowledge grant No. NSF-CMMI-1462255, NSF-CISE-1835690 and NSF-CDSE-2001333. VA also acknowledge the support of Lawrence Livermore National Laboratory under Collaborative R&D in Support of LLNL Missions, Task Order No. B623252 and Master Task Agmt. B575363 as well as the Texas A&M Institute for Data Science (TAMIDS) through the Data Resource Development Program for partially supporting this effort. MM also acknowledges the support of the AFRL through the AFRL-MLP program, under contract UTC-165852-19F5830-19-02-C1.

References

[1] A. Jain, S.P. Ong, G. Hautier, W. Chen, W.D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, et al., Commentary: the materials project: a materials genome approach to accelerating materials innovation, APL Mater 1 (1) (2013) 011002.
[2] J. Allison, D. Backman, L. Christodoulou, Integrated computational materials engineering: a new paradigm for the global materials profession, Jom 58 (11) (2006) 25–27.
[3] A. Agrawal, A. Choudhary, Perspective: materials informatics and big data: realization of the 'fourth paradigm' of science in materials science, APL Mater 4 (5) (2016) 053208.
[4] N. Baker, F. Alexander, T. Bremer, A. Hagberg, Y. Kevrekidis, H. Najm, M. Parashar, A. Patra, J. Sethian, S. Wild, et al., Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence, Technical Report, USDOE Office of Science (SC), Washington, DC (United States), 2019.
[5] S.R. Kalidindi, M. De Graef, Materials data science: current status and future outlook, Annu Rev Mater Res 45 (2015) 171–193.
[6] S.R. Niezgoda, A.K. Kanjarla, S.R. Kalidindi, Novel microstructure quantification framework for databasing, visualization, and analysis of microstructure data, Integrating Materials and Manufacturing Innovation 2 (1) (2013) 54–80.
[7] V. Sundararaghavan, N. Zabaras, Classification and reconstruction of three-dimensional microstructures using support vector machines, Comput. Mater. Sci 32 (2) (2005) 223–239.
[8] B.L. DeCost, E.A. Holm, A computer vision approach for automated analysis and classification of microstructural image data, Comput. Mater. Sci 110 (2015) 126–133.
[9] R. Bostanabad, A.T. Bui, W. Xie, D.W. Apley, W. Chen, Stochastic microstructure characterization and reconstruction via supervised learning, Acta Mater 103 (2016) 89–102.
[10] J. Gola, D. Britz, T. Staudt, M. Winter, A.S. Schneider, M. Ludovici, F. Mücklich, Advanced microstructure classification by data mining methods, Comput. Mater. Sci 148 (2018) 324–335.
[11] P. Prakash, V. Mytri, P. Hiremath, Fuzzy rule based classification and quantification of graphite inclusions from microstructure images of cast iron, Microsc. Microanal. 17 (6) (2011) 896–902.

[12] A. Choudhury, Y.C. Yabansu, S.R. Kalidindi, A. Dennstedt, Quantification and classification of microstructures in ternary eutectic alloys using 2-point spatial correlations and principal component analyses, Acta Mater 110 (2016) 131–141.
[13] S. Niezgoda, D. Fullwood, S. Kalidindi, Delineation of the space of 2-point correlations in a composite material system, Acta Mater 56 (18) (2008) 5285–5292.
[14] B.L. DeCost, T. Francis, E.A. Holm, Exploring the microstructure manifold: image texture representations applied to ultrahigh carbon steel microstructures, Acta Mater 133 (2017) 30–40.
[15] C. Kunselman, V. Attari, L. McClenny, U. Braga-Neto, R. Arroyave, Semi-supervised learning approaches to class assignment in ambiguous microstructures, Acta Mater 188 (2020) 49–62.
[16] L. Gómez-Chova, G. Camps-Valls, J. Munoz-Mari, J. Calpe, Semisupervised image classification with laplacian support vector machines, IEEE Geosci. Remote Sens. Lett. 5 (3) (2008) 336–340.
[17] C. Gong, D. Tao, S.J. Maybank, W. Liu, G. Kang, J. Yang, Multi-modal curriculum learning for semi-supervised image classification, IEEE Trans. Image Process. 25 (7) (2016) 3249–3260.
[18] M. Guillaumin, J. Verbeek, C. Schmid, Multimodal semi-supervised learning for image classification, in: 2010 IEEE Computer society conference on computer vision and pattern recognition, IEEE, 2010, pp. 902–909.
[19] P. Yang, X.-L. Li, J.-P. Mei, C.-K. Kwoh, S.-K. Ng, Positive-unlabeled learning for disease gene identification, Bioinformatics 28 (20) (2012) 2640–2647.
[20] X.-L. Li, P.S. Yu, B. Liu, S.-K. Ng, Positive unlabeled learning for data stream classification, in: Proceedings of the 2009 SIAM International Conference on Data Mining, SIAM, 2009, pp. 259–270.
[21] M.G. Omran, A.P. Engelbrecht, A. Salman, Differential evolution methods for unsupervised image classification, in: 2005 IEEE Congress on Evolutionary Computation, volume 2, IEEE, 2005, pp. 966–973.
[22] M. Omran, A. Salman, A. Engelbrecht, Dynamic clustering using particle swarm optimization with application in unsupervised image classification, in: Fifth World Enformatika Conference (ICCI 2005), Prague, Czech Republic, 2005, pp. 199–204.
[23] T.-W. Lee, M.S. Lewicki, Unsupervised image classification, segmentation, and enhancement using ica mixture models, IEEE Trans. Image Process. 11 (3) (2002) 270–279.
[24] Z. Yu, H.-S. Wong, Class discovery from gene expression data based on perturbation and cluster ensemble, IEEE Trans Nanobioscience 8 (2) (2009) 147–160.
[25] A.L. Hsu, S.-L. Tang, S.K. Halgamuge, An unsupervised hierarchical dynamic self-organizing approach to cancer class discovery and marker gene identification in microarray data, Bioinformatics 19 (16) (2003) 2131–2140.
[26] S. Monti, P. Tamayo, J. Mesirov, T. Golub, Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data, Mach Learn 52 (1–2) (2003) 91–118.
[27] V. Attari, P. Honarmandi, T. Duong, D.J. Sauceda, D. Allaire, R. Arroyave, Uncertainty propagation in a multiscale calphad-reinforced elastochemical phase–field model, Acta Mater 183 (2020) 452–470.
[28] V. Attari, Open Phase-field Microstructure Database (OPMD), 2019. http://microstructures.net.
[29] K. Ishida, Intermetallic compounds in co-base alloys–phase stability and application to superalloys, MRS Online Proceedings Library Archive 1128 (2008).
[30] J. Peng, R. Xing, Y. Wu, B. Li, Y. Han, W. Knoll, D.H. Kim, Dewetting of thin polystyrene films under confinement, Langmuir 23 (5) (2007) 2326–2329.
[31] S.-i. Yi, V. Attari, M. Jeong, J. Jian, S. Xue, H. Wang, R. Arroyave, C. Yu, Strain-induced suppression of the miscibility gap in nanostructured mg_2si–mg_2sn solid solutions, Journal of Materials Chemistry A 6 (36) (2018) 17559–17570.
[32] C. Wang, M. Li, M. Zhu, H. Wang, C. Qin, W. Zhao, Z. Wang, Controlling the mechanical properties of bulk metallic glasses by superficial dealloyed layer, Nanomaterials 7 (11) (2017) 352.
[33] S. Wassén, R. Bordes, T. Gebäck, D. Bernin, E. Schuster, N. Lorén, A.-M. Hermansson, Probe diffusion in phase-separated bicontinuous biopolymer gels, Soft Matter 10 (41) (2014) 8276–8287.
[34] J.P. MacSleyne, J.P. Simmons, M. De Graef, On the use of 2-d moment invariants for the automated classification of particle shapes, Acta Mater 56 (3) (2008) 427–437.
[35] R. Bostanabad, Y. Zhang, X. Li, T. Kearney, L.C. Brinson, D.W. Apley, W.K. Liu, W. Chen, Computational microstructure characterization and reconstruction: review of the state-of-the-art techniques, Prog Mater Sci 95 (2018) 1–41.
[36] S.R. Kalidindi, S.R. Niezgoda, A.A. Salem, Microstructure informatics using higher-order statistics and efficient data-mining protocols, JOM 63 (4) (2011) 34–41.
[37] P. Tirilly, V. Claveau, P. Gros, Language modeling for bag-of-visual words image categorization, in: Proceedings of the 2008 international conference on Content-based image and video retrieval, 2008, pp. 249–258.
[38] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556 [cs] (2015). ArXiv: 1409.1556
[39] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[40] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.
[41] S. Tammina, Transfer learning using vgg-16 with deep convolutional neural network for classifying images, International Journal of Scientific and Research Publications 9 (10) (2019) 143–150.
[42] K.-S. Lee, S.-K. Jung, J.-J. Ryu, S.-W. Shin, J. Choi, Evaluation of transfer learning with deep convolutional neural networks for screening osteoporosis in dental panoramic radiographs, J Clin Med 9 (2) (2020) 392.
[43] D.I. Swasono, H. Tjandrasa, C. Fathicah, Classification of tobacco leaf pests using vgg16 transfer learning, in: 2019 12th International Conference on Information & Communication Technology and System (ICTS), IEEE, 2019, pp. 176–181.
[44] K. Gopalakrishnan, S.K. Khaitan, A. Choudhary, A. Agrawal, Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection, Constr. Build. Mater. 157 (2017) 322–330.
[45] J. Brownlee, A Gentle Introduction to Pooling Layers for Convolutional Neural Networks, 2019.
[46] Z. Xu, Y. Yang, A.G. Hauptmann, A discriminative cnn video representation for event detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1798–1807.
[47] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, Chemometrics and Intelligent Laboratory Systems 2 (1) (1987) 37–52.
[48] Z. Cao, M. Shaomin, X. Yongyu, M. Dong, Image retrieval method based on cnn and dimension reduction, in: 2018 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), IEEE, 2018, pp. 441–445.
[49] C. Yuan, X. Li, Q.J. Wu, J. Li, X. Sun, Fingerprint liveness detection from different fingerprint materials using convolutional neural network and principal component analysis, Computers, Materials & Continua 53 (3) (2017) 357–371.
[50] I. Borg, P. Groenen, Modern multidimensional scaling: theory and applications, J Educ Meas 40 (3) (2003) 277–280.
[51] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM computing surveys (CSUR) 31 (3) (1999) 264–323.
[52] T. CaliŃski, Dendrogram, Wiley StatsRef: Statistics Reference Online (2014).
[53] Y. Senbabaoglu, G. Michailidis, J.Z. Li, Critical limitations of consensus clustering in class discovery, Sci Rep 4 (1) (2014) 1–13. Number: 1 Publisher: Nature Publishing Group
[54] C.R. John, D. Watson, D. Russ, K. Goldmann, M. Ehrenstein, C. Pitzalis, M. Lewis, M. Barnes, M3C: Monte carlo reference-based consensus clustering, Sci Rep 10 (1) (2020) 1–14. Number: 1 Publisher: Nature Publishing Group
[55] S.B. Kotsiantis, I. Zaharakis, P. Pintelas, Supervised machine learning: a review of classification techniques, Emerging artificial intelligence applications in computer engineering 160 (2007) 3–24.
[56] J.C. Bezdek, A review of probabilistic, fuzzy, and neural models for pattern recognition, Journal of Intelligent & Fuzzy Systems 1 (1) (1993) 1–25.
[57] R. Jain, A. Abraham, A comparative study of fuzzy classification methods on breast cancer data, Australasian Physics & Engineering Sciences in Medicine 27 (4) (2004) 213–218.
[58] X. Zhu, A.B. Goldberg, Introduction to semi-supervised learning, Synthesis lectures on artificial intelligence and machine learning 3 (1) (2009) 1–130.
[59] T. Yang, C.E. Priebe, The effect of model misspecification on semi-supervised classification, IEEE Trans Pattern Anal Mach Intell 33 (10) (2011) 2093–2103.
[60] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in python, Journal of Machine Learning Research 12 (2011) 2825–2830.
[61] M. Van der Laan, K. Pollard, J. Bryan, A new partitioning around medoids algorithm, J Stat Comput Simul 73 (8) (2003) 575–584.
[62] F. Nielsen, Hierarchical Clustering, in: Introduction to HPC with MPI for Data Science, Springer, 2016, pp. 195–211.
[63] M.D. Wilkerson, D.N. Hayes, Consensusclusterplus: a class discovery tool with confidence assessments and item tracking, Bioinformatics 26 (12) (2010) 1572–1573.
[64] C.R. John, D. Watson, D. Russ, K. Goldmann, M. Ehrenstein, C. Pitzalis, M. Lewis, M. Barnes, M3c: Monte carlo reference-based consensus clustering, Sci Rep 10 (1) (2020) 1–14.
[65] B. Scholkopf, A.J. Smola, Learning with kernels: Support vector machines, regularization, optimization, and beyond, MIT Press, Cambridge, MA, USA, 2001.
[66] S. Abney, Understanding the yarowsky algorithm, Computational Linguistics 30 (3) (2004) 365–395.
[67] Y.-F. Li, Z.-H. Zhou, Towards making unlabeled data never hurt, IEEE Trans Pattern Anal Mach Intell 37 (1) (2014) 175–188.
[68] X. Zhu, Z. Ghahramani, Learning from Labeled and Unlabeled Data with Label Propagation, Technical Report, Carnegie Mellon University, 2002.
[69] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al., Constrained k-means clustering with background knowledge, in: Icml, volume 1, 2001, pp. 577–584.
[70] E.A. Platanios, A. Blum, T. Mitchell, Estimating accuracy from unlabeled data, in: Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, 2014, pp. 682–691.
[71] C. Kunselman, Microstructure Classification in the Unsupervised Context, 2020. https://github.com/cjkunselman18/Microstructure-Classification-Unsupervised-Context.
[72] A. Agarwal, S. De Marco, E. Gobet, G. Liu, Study of new rare event simulation schemes and their application to extreme scenario generation, Math Comput Simul 143 (2018) 89–98.
[73] S.S. Kubatur, Stochastic Modeling and Rare Event Simulation for Gibbs Distributions with Applications in Materials Engineering, Purdue University, 2017 Ph.D. thesis.