



# Microstructure Cluster Analysis with Transfer Learning and Unsupervised Learning

Andrew R. Kitahara<sup>1</sup> · Elizabeth A. Holm<sup>1</sup>

Received: 14 May 2018 / Accepted: 24 July 2018 / Published online: 30 August 2018  
© The Minerals, Metals & Materials Society 2018

## Abstract

We apply computer vision and machine learning methods to analyze two datasets of microstructural images. A transfer learning pipeline utilizes the fully connected layer of a pre-trained convolutional neural network as the image representation. An unsupervised learning method uses the image representations to discover visually distinct clusters of images within two datasets. A minimally supervised clustering approach classifies micrographs into visually similar groups. This approach successfully classifies images both in a dataset of surface defects in steel, where the image classes are visually distinct and in a dataset of fracture surfaces that humans have difficulty classifying. We find that the unsupervised, transfer learning method gives results comparable to fully supervised, custom-built approaches.

**Keywords** Characterization · Computer vision · Machine learning · Microstructure

## Introduction

At its core, the primary initiative of materials science and engineering is to investigate processing-structure-property (PSP) linkages in materials [1]. Recent advances in applied statistics (data mining) and the rapid development of data science bring an opportunity to bridge these disciplines with materials science, recently termed as materials data science (MDS) [1, 2]. Current opportunities for MDS include searching for novel and promising material composition, processing conditions, properties, and performance metrics [1]; but notably, microstructure is not readily amenable to these types of analyses. The history of microstructure science is dominated by qualitative and subjective observations. Microstructure science does include some quantitative measures such as grain size, phase distributions, and shape descriptions of primary and second phase particles [3]; however, these high-level measurements do not comprise a complete microstructural description and are generally applied on a case-by-case basis. Although these measurements provide helpful insight to their niche applications, there is no universal approach for capturing all of the information,

both quantitative and qualitative, that is contained in a microstructural image. MDS offers a new path to describe microstructure image data objectively and comprehensively.

Recent advances in computer vision and machine learning have shown promise in a breadth of applications from panoramic photo stitching [4] to mass surveillance [5]. Convolutional neural networks (CNNs) are especially popular in the computer vision community with recent successes in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC), which has been a benchmark image dataset for computer vision applications since 2010 [6]. In 2012, CNNs became the new standard in image classification with the development of GPU-based AlexNet [7]. In 2014, VGG16 had a spectacular performance in the competition, and is the inspiration of this work [8]. Although the computer vision community is currently celebrating the success of neural networks, these architectures were hypothesized to be well-suited for developing functional models in abstract domains such as defect analysis as early as 1999 [9]. DeCost was among the first to illustrate how computer vision can be used to identify objective linkages between visual microstructure and processing conditions [10]. This work expands on this approach by demonstrating how the ImageNet pre-trained VGG16 CNN can be used in an unsupervised learning mode to generate image feature descriptors of microstructure images on two distinct datasets, to qualitatively group similar micrographs, and to perform a classification task.

---

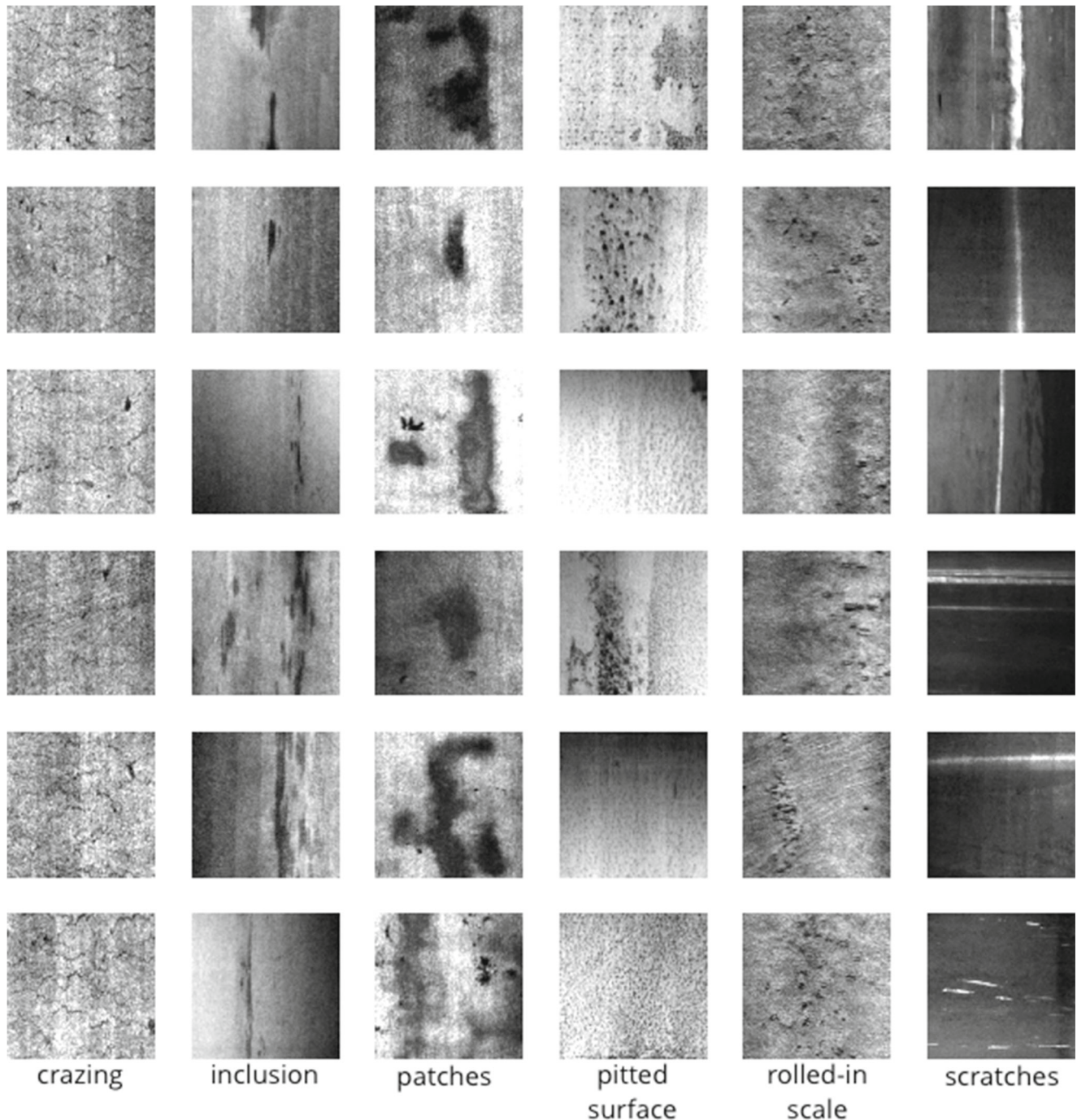
✉ Elizabeth A. Holm  
eaholm@andrew.cmu.edu  
Andrew R. Kitahara  
akitahar@andrew.cmu.edu

<sup>1</sup> Carnegie Mellon University, Pittsburgh, PA, USA

## NEU Surface Defect Database

The proposed pipeline was first tested as a utility for quality control and defect detection applications in steel production. Song and Yan [11] collected and maintain a database of surface defects observed on hot-rolled steel strip. The database contains images of six common surface defects (crazing, inclusions, patches, pitting, rolled-in scale, and

scratches). Each defect class contains 300 sample images providing a total database collection of 1800 8-bit grayscale images. The database's purpose is to be a benchmark for testing realtime surface defect detectors working in hot-rolling manufacturing facilities. The originally published work with the creation of this dataset used supervised learning techniques and reported a detector with 97.89% accuracy [11]. A small subset of the dataset is seen in Fig. 1.



**Fig. 1** Six example images of all six surface defect classes in the NEU Surface Defect Database [11]

## In-718 Charpy Fracture Surface Dataset

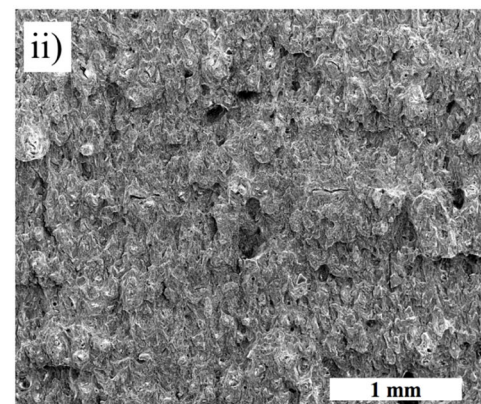
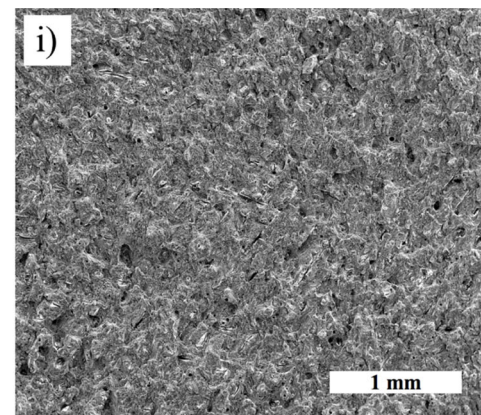
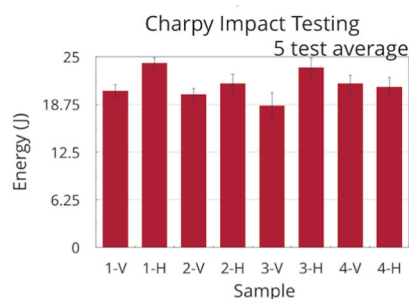
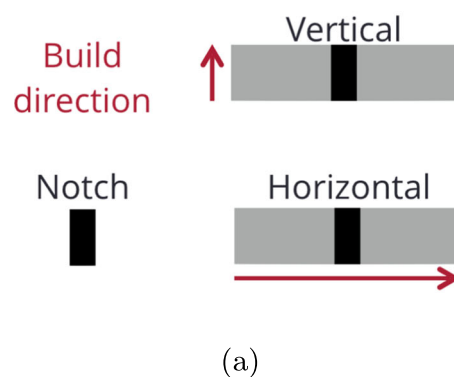
The pipeline was also tested on a more visually challenging dataset. The In-718 Charpy Fracture Surface dataset was developed at the NextManufacturing Center for additive manufacturing research at Carnegie Mellon University. In the creation of this dataset, In-718 Charpy coupons were additively manufactured using selective laser melting. Two build geometries described in Fig. 2a were investigated in the creation of this dataset. The printed coupons then underwent a Charpy impact test and absorbed energies were recorded. It was expected that horizontally and vertically built samples would have different fracture energies, and Fig. 2b confirms that intuition. The data used in our analysis pipeline are SEM images of the fracture surfaces, two examples of which are shown in Fig. 2c; these were indistinguishable to the human eye in field testing among materials science graduate students. Original MDS work on this dataset applied hypercolumn methods described by [10], but the result was also no better than random guessing [12]. Thus, the challenge remained to identify a visual signature in the fracture surface images that can relate to the build geometries.

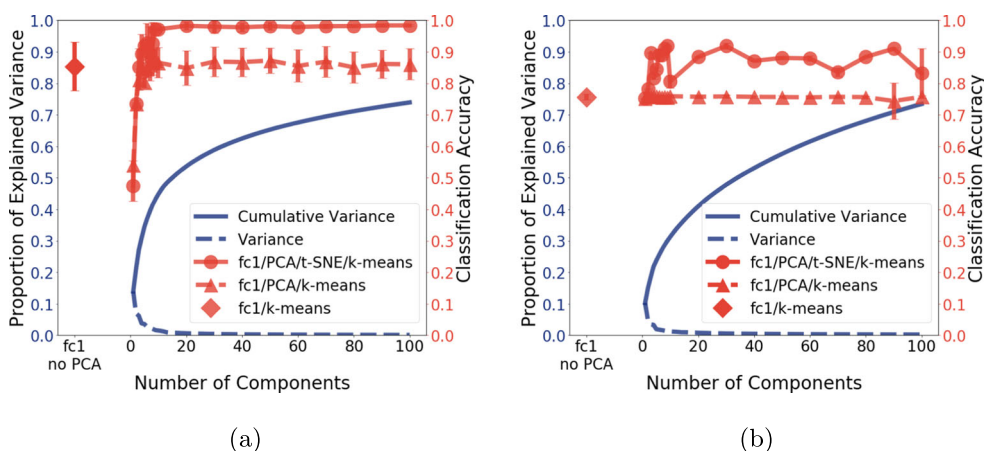
## Analysis Pipeline

### Image Processing

The NEU surface defect and In-718 fracture datasets required slightly different image pre-processing before following an otherwise identical processing pipeline. The NEU dataset is almost perfect for direct implementation with the VGG16 fully connected neural network because it has high interclass diversity, low intraclass variance, and hundreds of samples for each of the six classes. Each image in the NEU dataset can easily be processed to VGG16 input requirements by mapping the grayscale values to an RGB color space and reshaping the image to 224 x 224 pixels from 200 x 200 pixels. The In-718 dataset contains only six fracture surface images of size 2048 x 1768 pixels. These were each sliced into a grid about the center containing 56 patches of size 224 x 224. Although a full resolution patch of this size may not be able to adequately capture the fine information in the fracture surface, this was shown to work and the success is credited to the design of the VGG16 CNN and the diversity of training data from the ImageNet challenge [6]. Finally, adaptive histogram equalization with

**Fig. 2** **a** Charpy impact testing coupons were made using powder bed additive manufacturing techniques in horizontal and vertical build orientations. **b** Experimental collaborators described the differences in measured Charpy impact energy as significant in their domain. **c** Sample fracture micrographs for (i) horizontal and (ii) vertical build directions. Note that fracture images (and not impact energies) were the only data used in this analysis





**Fig. 3** Scree plots for **a** NEU Surface Defect Database and **b** In-718 dataset. The proportion of cumulative variance explained by PCA approaches unity as the number of components reaches the maximum number of principal components, which is  $\min(N, D)$ , where  $N, D$  are the number of samples and feature dimensions, respectively.

scikit-image [13] is applied to normalize global image intensity and contrast; classifying on dark/light would be a failure.<sup>1</sup>

### Feature Extraction

Image feature descriptors were generated with the VGG16 CNN, which has a modular structure with five blocks each containing two or three convolution layers [8]. Each convolution layer consists of a set of optimized weights from training on the ImageNet database [6]. The pretrained network enables users to access deep convolutional filters that were trained to perform well on a large and diverse dataset; this dataset contains over one million images that have visual textures that cover many length scales. The first convolution layer performs a kernel convolution on the input image using the pre-trained weights for that layer. The rectified linear unit (ReLU) of the convolution is taken and the output serves as input for the next layer; this continues for all subsequent layers. Max pooling is used to spatially pool outputs of blocks of prior convolution layer blocks to following blocks. This results in a multi-scale image representation that can detect fine features in shallow layers and coarse structure in deeper layers. VGG16 has a final softmax layer which is used to perform a classification on ImageNet images; however, the interest in this architecture is only to extract the feature descriptors, so the softmax layer is removed. Each image runs through VGG16 and a feature descriptor of the image is computed from the result of the series of convolution and spatial pooling layers.

<sup>1</sup>Selecting an appropriate pre-processing method can be critical in the performance of certain models as discussed by Pal & Sudeep [14].

Performance metrics are shown for the full pipeline (fc1/PCA/t-SNE/k-means) and simplified versions of the pipeline (fc1/PCA/k-means and fc1/k-means); in both experiments, the full pipeline outperforms the simpler versions

Ultimately, this computes a 4096-dimension vector, termed a fully connected layer, where each component describes a multi-scale low-level response from the image. VGG16 produces two fully connected layers (fc1 and fc2); we find significantly better classification performance using fc1 as our image representation. We note that this analysis pipeline is similar to DeCost’s analysis of ultra high carbon steel micrographs [15]; contrary to DeCost’s work, this work is unsupervised, does not employ VLAD encoding, and attempts to do find distinguishing image features that humans cannot identify.

### Dimensionality Reduction

4096 feature descriptors become computationally inefficient to produce visualizations for this analysis. Because of this, we first apply a linear dimensionality reduction method, principal component analysis (PCA) [16, 17]. PCA is one of the most popular and oldest multivariate statistic techniques [16], and it finds significant use here. The general idea of PCA is to reduce the feature descriptors that exist in  $\mathbb{R}^n$  to  $\mathbb{R}^k$  where  $k < n$  while retaining the maximum variance in the data. In the scope of this work, the feature descriptors from VGG16 are reduced from a 4096-dimensional space to a k-dimensional space, keeping k as small as possible for computational efficiency while capturing as much variance in the data as possible for model performance.. The PCA captured variances and the corresponding classification accuracies are shown in Fig. 3a, b for the NEU surface defect database and the In-718 dataset respectively. A rule-of-thumb in computer vision applications is to select  $k=50$  dimensions. For both of our datasets, this retains  $\sim 60\%$  of the total variance and yields classification accuracy well within the upper plateau. While larger  $k$  values can increase

the retained variance, computational costs increase, and classification accuracy does not significantly improve.

### t-distributed Stochastic Neighbor Embedding

t-distributed Stochastic Neighbor Embedding is a state-of-the-art dimensionality reduction technique used in high-dimensionality data explorations [18]. t-SNE is highly regarded because it retains both local and global data structure by determining a probabilistic conditional model of all points within the data [10]. However, this construction can become computationally expensive to calculate pairwise distances in very high dimensional space; because of this, it is best to perform this in a lower-dimension space obtained from PCA [18]. For larger datasets where visualization is the primary use for t-SNE, it is recommended to use the Barnes-Hut implementation of t-SNE [19]. The original t-SNE algorithm has a computational expense that scales with  $\mathcal{O}(N^2)$ , whereas Barnes-Hut t-SNE scales with  $\mathcal{O}(N \log N)$  where  $N$  is the number of images [19]. Moreover, when exploring a new dataset, it is recommended to start by following [18] and use PCA for dimensionality reduction to 50-D for image features of higher dimensionality, as discussed above.

It is important to note that t-SNE does not come without its limitations. Notably, t-SNE projections have no sense of inter-cluster distance. Although data are grouped by similarity, clusters near other clusters are not inherently *more similar* than clusters far apart from one another. Also, t-SNE offers no method of reconstructing original data as PCA does, and t-SNE must be reconstructed as new data are collected. Finally, t-SNE is useful only for *ex situ* characterization, although pretrained CNN features can be used in an *in situ* characterization task.

There are some hyperparameters to adjust with t-SNE such as perplexity, defined as  $2^{H(X)}$  where  $H(X)$  is the Shannon Entropy [18], and can be thought of as a parameter which tunes the sensitivity to local neighbors in the data manifold. There are other t-SNE hyperparameters as well, but they are beyond the focus of this work, and van der Maaten and Hinton already cover this topic in great depth [18].

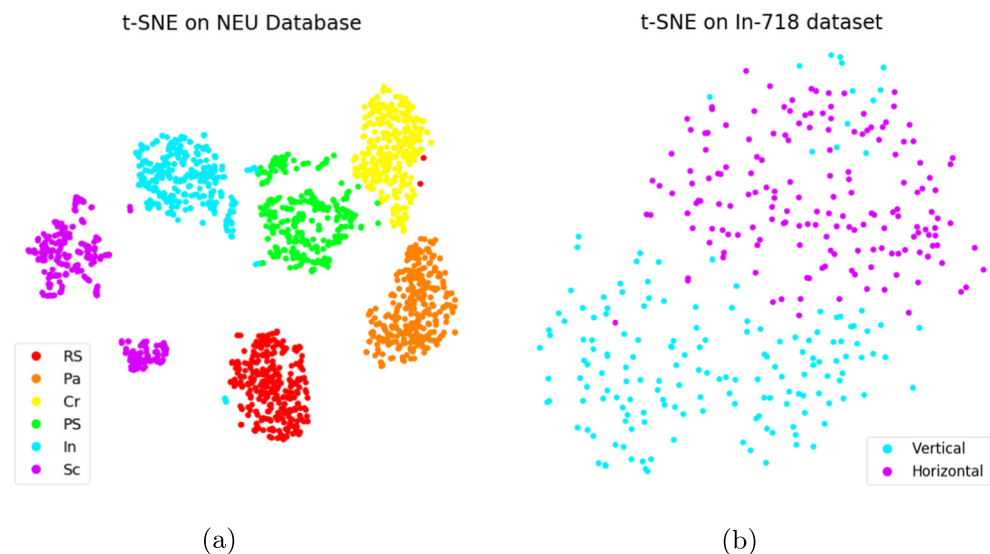
The t-SNE plot in Fig. 4a shows clear clustering of classes for the NEU database, with strong intra-cluster localization and inter-cluster separation. This suggests that the NEU database contains visually distinct features, which is easily confirmed by human inspection. Interestingly, the scratches split into two sub-clusters, which happen to distinguish vertical from horizontal scratches.

When data are colored by their true labels, t-SNE plots for the In-718 data show distinct intra-cluster localization, but inter-cluster separation is minimal (Fig. 4b). However, the unsupervised learning algorithm still separates horizontal and vertical build directions by examining only surface feature data from the fracture surfaces. This is rather surprising, considering that the fracture surface images are not distinguishable by human experts.

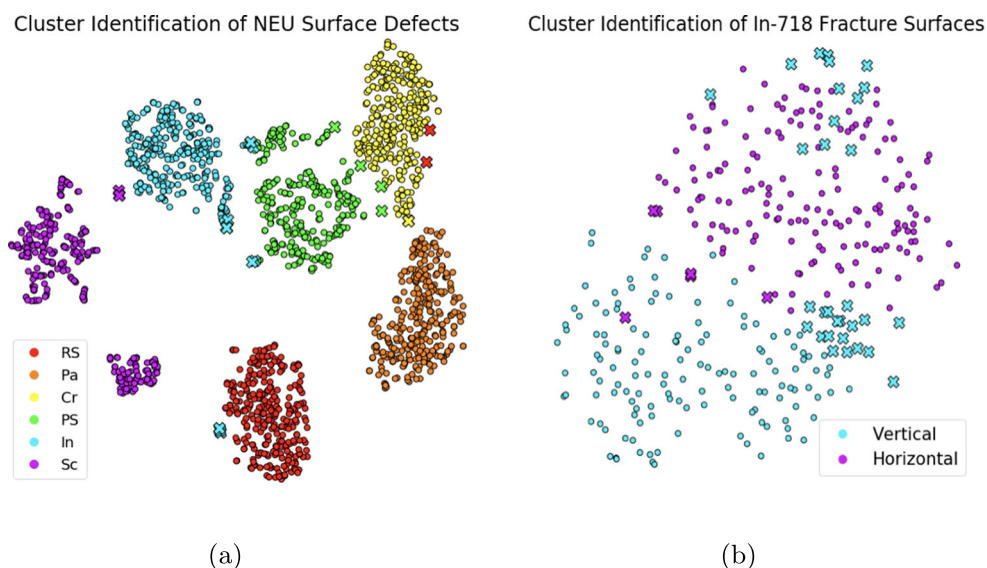
### Classification

This final step is optional but is useful for quantifying the success of the analysis pipeline. All machine learning to this point has either been based on transfer learning (VGG16) or unsupervised learning (t-SNE). However, in this last step, user input is introduced to identify the number of cluster centers (e.g., 6 clusters for the NEU database and 2 clusters for the In-718 dataset); k-means [20, 21] is then applied to find the cluster centers and assign labels to the points

**Fig. 4** t-SNE projection of 50-dimensional PCA encoding to 2-dimensional visual similarity map. Shown for **a** NEU surface defect database, which shows clustering with strong intra-cluster localization and inter-class separation, and **b** In-718 fracture surface dataset, which without coloring by true label, there is no distinguishable clustering apparent. The abbreviations for the labels (top to bottom) in **a** stand for: rolled-in scale, patches, crazing, pitted surfaces, inclusions, and scratches as described in “Introduction”



**Fig. 5** Classification visualization for **a** NEU surface defect database and **b** In-718 fracture surface dataset. Data points are colored according to their true labels. Each point is classified using k-means in the projected t-SNE space. Correctly labeled points are shown as circles. Incorrectly classified points are shown as cross marks



in the projected t-SNE feature space. The k-means labels are categorical, with arity (number of classes) equal to the number of clusters set by the user. Meaning can be assigned to the labels by examining a projection of the data with the original input images at each point, which can be compared to the true class labels to establish an accuracy metric.

The plots shown in Fig. 5 show 6- and 2-class classification tasks for the NEU and In-718 datasets, respectively. The points that represent the correctly and incorrectly labeled points with circles and crosses, respectively. Note that the only supervision in this pipeline is in defining the number of classes that the k-means algorithm should attempt to label.

## Results and Discussion

### Classification Accuracy

The proposed pipeline has proven to be a highly functional model for classification tasks in two distinct microstructural image challenges; fc1/PCA/t-SNE/k-means is an unsupervised learning (USL) method that represents images using fc1, reduces dimensionality via PCA, groups images by visual similarity with t-SNE, and finds image clusters using k-means [8]. We also explore two simpler unsupervised pipelines that employ the same image representation, but use k-means to find image clusters either on the PCA results (fc1/PCA/k-means) or on the fc1 image representation directly (fc1/k-means). As one might expect, all models performs more strongly with the NEU database since the images are more visually distinct than the fracture surfaces in the In-718 dataset. The original classification benchmark for the NEU database was set in the work by Song et al. with the adjacent evaluation completed local

binary patterns (AECLBP) custom-built feature descriptor; AECLBP was used as an input for a support vector machine (SVM) and a nearest neighbor classifier (NNC) with strong performances [11]. Additionally, Zhou, et al. recently reported a supervised CNN-based approach on the NEU database. Table 1 compares the accuracies of the proposed models with accuracy scores from other groups on the NEU database. These results demonstrate that an unsupervised general transfer learning approach can achieve similar classification accuracy to previous supervised and custom-built methods [11, 22]. Furthermore, the full fc1/PCA/t-SNE/k-means pipeline outperforms the simpler fc1-based analyses.

The In-718 dataset is relatively new and unexplored. We first attempted to classify the images in this dataset using a machine learning algorithm learning on VLAD-encoded hypercolumns [23]; however, the result was no better than random guessing. In fact, our initial hypothesis was that it might be impossible to discern useful detail from the images due to the manifestation of entropy from the Charpy impact test on the visual signatures of the fracture surfaces. This,

**Table 1** Table of various classification performances

	NEU accuracy (%)	In-718 accuracy (%)
fc1/PCA/t-SNE/k-means	98.3 ± 1.2	88.4 ± 2.5
fc1/PCA/k-means	87.2 ± 4.0	75.6 ± 0.4
fc1/k-means	85.4 ± 7.8	75.6 ± 0.5
AECLBP-SVM [11]	98.9 ± 0.6	—
AECLBP-NNC [11]	97.8 ± 0.2	—
CNN [22]	99 <sup>a</sup>	—

<sup>a</sup> No error or standard deviation reported

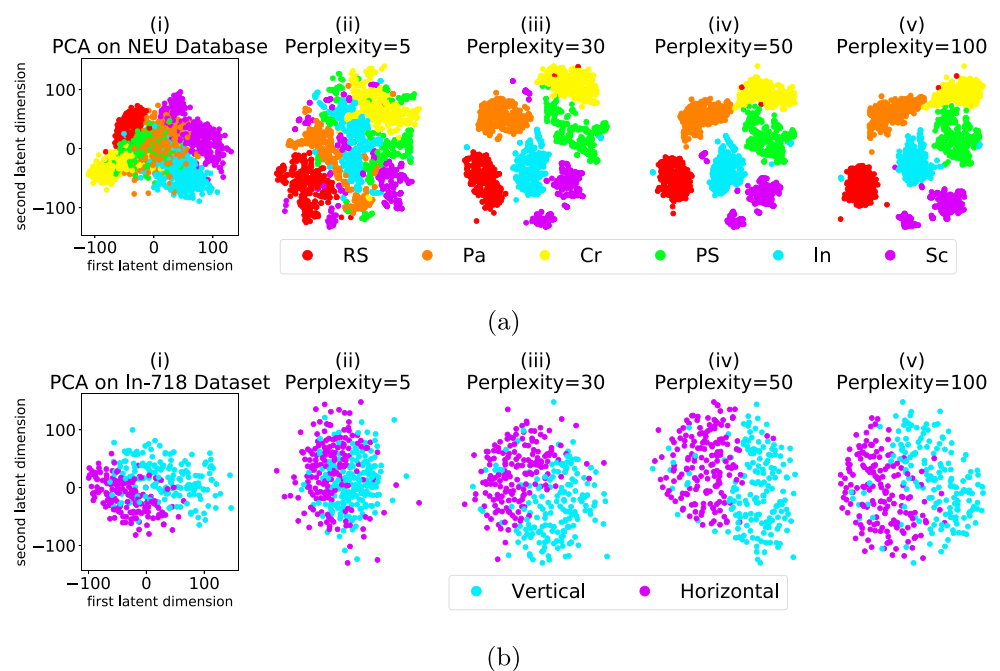
apparently, is not the case. But it does lead one to question what the neural network sees that humans cannot, since in field tests, human classification of the In-718 dataset was around 50% (random guessing). As seen in Table 1, though, the proposed model performs well beyond random classification accuracy.

The primary purpose of the fc1/PCA/t-SNE/k-means pipeline was to simulate some kind of *useful* analysis (e.g., unsupervised classification) that can be performed using pretrained CNNs such as VGG16. Nevertheless, it is worth discussing whether a simpler pipeline might suffice. For example, PCA reduces the original feature space to a lower dimensional space with a linear transformation such that each output dimension is orthogonal (e.g., zero correlation between dimensions). This can sometimes be used to feed a clustering algorithm such as k-means directly, as in Fig. 6(a)(i) and (b)(i). From Fig. 6 and Table 1, it is clear that a pipeline that omits t-SNE grouping (i.e., fc1/PCA/k-means) can successfully cluster and classify the NEU data set. (Note that although the clusters are not unambiguously separated in the 2-D projection, they are separated in the 50-D PCA space.) However, using t-SNE on the PCA data provides clearly distinguishable clusters in 2-D, particularly as the perplexity value increases, and classification accuracy increases. The situation is similar for the In-718 data set: PCA gives some clustering, but high perplexity t-SNE does a better job. Hence, we choose the full fc1/PCA/t-SNE/k-means pipeline to perform classification. Moreover, Fig. 3a, b and Table 1 confirm that the full pipeline outperforms simpler versions for both datasets.

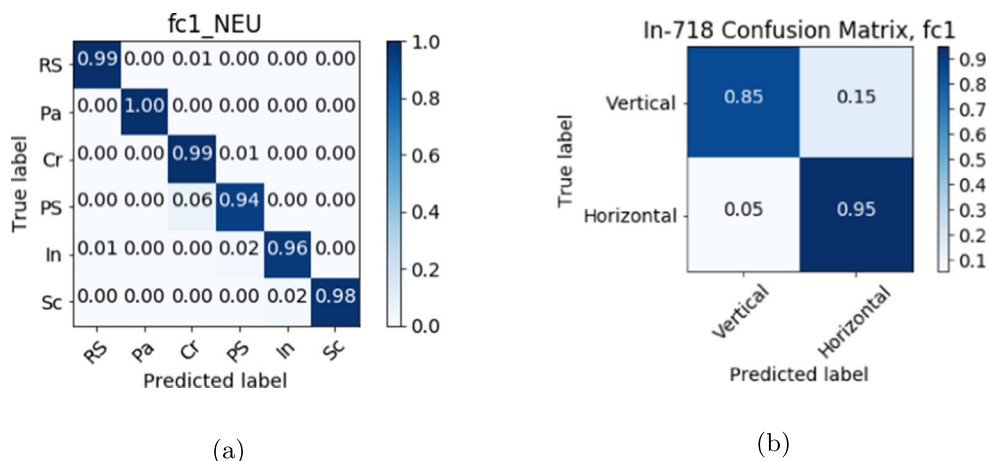
Confusion matrices are commonly used to identify how well an algorithm is performing classification tasks because they provide more insight to system performance than a reported accuracy value. Figure 7 shows the true and predicted labels of the data for each material system explored. Large values associated with a misclassification give a clue to how the system is failing, which is useful for optimizing the system's prediction capabilities. For example, Fig. 7a indicates that although the overall classification accuracy is quite high, the most prevalent misclassification is designating pitted surfaces as crazing. This is understandable, since both classes contain images in which alternating vertical light and dark stripes are a prominent visual feature.

A key limitation of this low-level feature descriptor is its interpretability. Because of black-box nature of a trained CNN, it is difficult to determine which image features the computer uses to describe and classify an image. There are numerous examples in which CNN-based computer vision systems make classification decisions based on information a human might not consider relevant [24] or where the computer is fooled by extraneous data [25]. Understanding how to interpret CNN models is an ongoing challenge in the computer vision community [26–28], but is perhaps especially important to the materials science community. For example, although our method is able to classify fracture surfaces in the In-718 dataset, we do not know if the computer is sensing visual features that are salient to the process (e.g., oriented features that relate to build direction) or property (e.g., surface roughness that correlates with fracture energy) or

**Fig. 6** (i) PCA and (ii–v) t-SNE perplexity plots for **a** NEU Surface Defect Database and **b** In-718 dataset. Although PCA can be directly used for cluster analysis with all principal components (only two shown here), t-SNE with sufficient perplexity has better performance with k-means clustering



**Fig. 7** Confusion matrices for **a** NEU database and **b** In-718 dataset. The vertical axis indicates the true class of an image, and the horizontal axis gives the predicted class. Perfect classification would result in 1's along the diagonal



whether it is keying in on some other, physically irrelevant feature (e.g., background noise characteristic of a particular microscope session). The full utility of the MDS approach to making PSP correlations will not be realized until we can confirm that the computer is learning physically relevant characteristics of the images.

## Conclusion

Micrographs encode subtle clues about processing and expected property behavior; the challenge is to find a robust image representation that captures the visual content in complex images. The key contributions of this work include:

- Developing a transfer learning pipeline that utilizes the fully-connected layer of a pre-trained convolutional neural network (the VGG16 CNN trained on the ImageNet database) as the image representation.
- Applying unsupervised learning (t-distributed Stochastic Neighbor Embedding) to discover visually distinct clusters of images within two microstructural data sets.
- Classifying micrographs using minimally supervised clustering approaches (k-means).
- Demonstrating that this approach successfully classifies images both in a dataset with visually distinctive classes (NEU surface defects) and in a dataset that humans have difficulty classifying (In-718 Charpy fracture surfaces).
- Showing that the unsupervised, transfer learning method gives results comparable to fully supervised, custom built approaches on the NEU dataset.
- Python code for this analysis pipeline can be found at [arkitahara.github.io](http://arkitahara.github.io)

**Acknowledgements** This work was performed at Carnegie Mellon University and has been supported by the US National Science Foundation award number DMR-1507830. The In-718 dataset was generously provided by the NextManufacturing Center for additive manufacturing research.

## References

1. Kalidindi SR, De Graef M (2015) Materials data science: current status and future outlook. *Annu Rev Mater Res* 45(1): 171–193. <https://doi.org/10.1146/annurev-matsci-070214-020844>, <http://www.annualreviews.org/doi/10.1146/annurev-matsci-070214-020844>
2. Potyrailo R, Rajan K, Stoewe K, Takeuchi I, Chisholm B, Lam H (2011) Combinatorial and high-throughput screening of materials libraries: review of state of the art. *ACS Comb Sci* 13(6):579–633. <https://doi.org/10.1021/co200007w>
3. Pickering FB (1976) The basis of quantitative metallography. Institute of Metallurgical Technicians
4. Rublee E, Garage W, Park M (2011) ORB : an efficient alternative to SIFT or SURF. In: *IEEE International Conference on Computer Vision*, pp 2564–2571. <https://doi.org/10.1109/ICCV.2011.6126544>
5. Oliver N, Rosario B, Pentland A (2000) A Bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8):831–843. [https://doi.org/10.1007/3-540-49256-9\\_16](https://doi.org/10.1007/3-540-49256-9_16)
6. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) ImageNet large scale visual recognition challenge. *International Journal of Computer Vision* 115(3):211–252. arXiv:1409.0575, <https://doi.org/10.1007/s11263-015-0816-y>
7. Krizhevsky A, Sutskever I, Geoffrey EH (2012) ImageNet classification with deep convolutional neural networks. *Adv Neural Inf Proces Syst* 25 (NIPS2012) 86:1–9. arXiv:1102.0183, <https://doi.org/10.1109/5.726791>
8. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition, pp 1–14. arXiv:1409.1556, <https://doi.org/10.1016/j.infsof.2008.09.005>
9. Wu X, Wang J Neural and machine learning to the surface defect investigation in sheet metal forming, *Neural Information*. <https://doi.org/10.1109/ICONIP.1999.844687>. [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=844687](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=844687)
10. Decost BL (2016) Microstructure representations microstructure representations, dissertations
11. Song K, Yan Y (2013) A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Appl Surf Sci* 285(PARTB):858–864. <https://doi.org/10.1016/j.apsusc.2013.09.002>
12. Kitahara AR (2017) Automated classification of fracture surfaces with computer vision and machine learning, Poster



13. van der Walt S, Schönberger JL, Nunez-Iglesias J, Boulogne F, Warner JD, Yager N, Gouillart E, Yu T (2014) The scikit-image contributors, scikit-image: image processing in Python. *PeerJ* 2:e453. <https://doi.org/10.7717/peerj.453>
14. Pal KK, Sudeep KS (2016) Preprocessing for image classification by convolutional neural networks. In: 2016 IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT), pp 1778–1781. <https://doi.org/10.1109/RTEICT.2016.7808140>
15. DeCost BL, Francis T, Holm EA (2017) Exploring the microstructure manifold: image texture representations applied to ultra-high carbon steel microstructures. *Acta Mater* 133:30–40. arXiv:1702.01117, <https://doi.org/10.1016/j.actamat.2017.05.014>
16. Abdi H, Williams LJ (2010) Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics* 2(4):433–459. arXiv:1011.1669v3, <https://doi.org/10.1002/wics.101>
17. Minka TP (2001) Automatic choice of dimensionality for PCA. In: *Advances in neural information processing systems*, pp 598–604
18. Van Der Maaten LJP, Hinton GE (2008) Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research* 9:2579–2605. arXiv:1307.1662, <https://doi.org/10.1007/s10479-011-0841-3>
19. van der Maaten L (2013) Barnes-Hut-SNE, pp 1–11. arXiv:1301.3342
20. Hartigan JA, Wong MA (1979) Algorithm as 136: a k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28(1):100–108. <http://www.jstor.org/stable/2346830>
21. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V et al (2011) Scikit-learn: machine learning in python. *The Journal of Machine Learning Research* 12:2825–2830
22. Zhou S, Chen Y, Zhang D, Xie J, Zhou Y (2017) Classification of surface defects on steel sheet using convolutional neural networks. *Materiali in Tehnologije* 51(1):123–131. <https://doi.org/10.17222/mit.2015.335>
23. Bansal A, Chen X, Russell B, Gupta A, Ramanan D PixelNet: Representation of the pixels, by the pixels, and for the pixels, arXiv:1702.06506
24. Fong RC, Vedaldi A (2017) Interpretable explanations of black boxes by meaningful perturbation. In: *Proceedings of the IEEE International Conference on Computer Vision 2017-October*, pp 3449–3457. arXiv:1704.03296, <https://doi.org/10.1109/ICCV.2017.371>
25. Goodfellow IJ, Shlens J, Szegedy C Explaining and harnessing adversarial examples, pp 1–11. arXiv:1412.6572
26. Zhou B, Bau D, Oliva A, Torralba A Interpreting deep visual representations via network dissection, arXiv:1711.05611
27. Zhang Q, Nian Wu Y, Zhu S-C Interpretable convolutional neural networks, arXiv:1710.00935
28. Olah C, Mordvintsev A, Schubert L Feature visualization, *Distill* <https://distill.pub/2017/feature-visualization>, <https://doi.org/10.23915/distill.00007>