A transfer learning approach to multi-label classification with uncorrelated property classes for data-efficient quantitative metallography

Anshuman Senapati
(MM17B010)

IIT Madras

Bachelor's Thesis, Part 1
Nov, 2020

Contents **Page No.**

For a quick summary please refer to the ‘Conclusion’ section.
All the code for the experiments is available at https://github.com/Anshu1245/dl-microstructures.

Introduction

With the boom in compute power towards the early 2000s, the sub-field of *deep learning* has been dominating the larger umbrella field of AI[1] by increasingly becoming a part of our day-to-day lives and is finding itself as a major component of a wide range of user applications being developed by major tech giants. Since it's advent from simplified ideas of mathematically modelling ensembles of individual neurons in the 1940s[2], the field has come a long way, suffering several lows along the path, and finally into its present form, where it has diversified itself into modelling distinct functional areas of human intelligence as we understand it today. The basic building block of any deep learning model is called a neuron, and an ensemble of layers of such units is called a *deep neural network* (DNN). A neural network, in its very essence, is one class among others, of nonlinear function approximators. And any learning that these networks do is essentially a form of statistical fitting of the model parameters so as to be as close to the observed data distribution as possible, through a gradient based optimization[3] process.

As it turns out, larger and larger neural networks owing to their high degree of nonlinearity are able to learn more and more complex representations of the data, in turn delivering incredibly powerful data-driven solutions. But as a caveat, as the model complexity increases, the network tends to overfit or remember the data that is being seen rather than learning useful representations from the data. It leads to a drop in the generalization ability of the model, that is, performance of the model on previously unseen data drops drastically. Hence, DNN models are 'data-hungry' and it is important for these models to feed on huge chunks of data to deliver fairly generalizable results on new data.

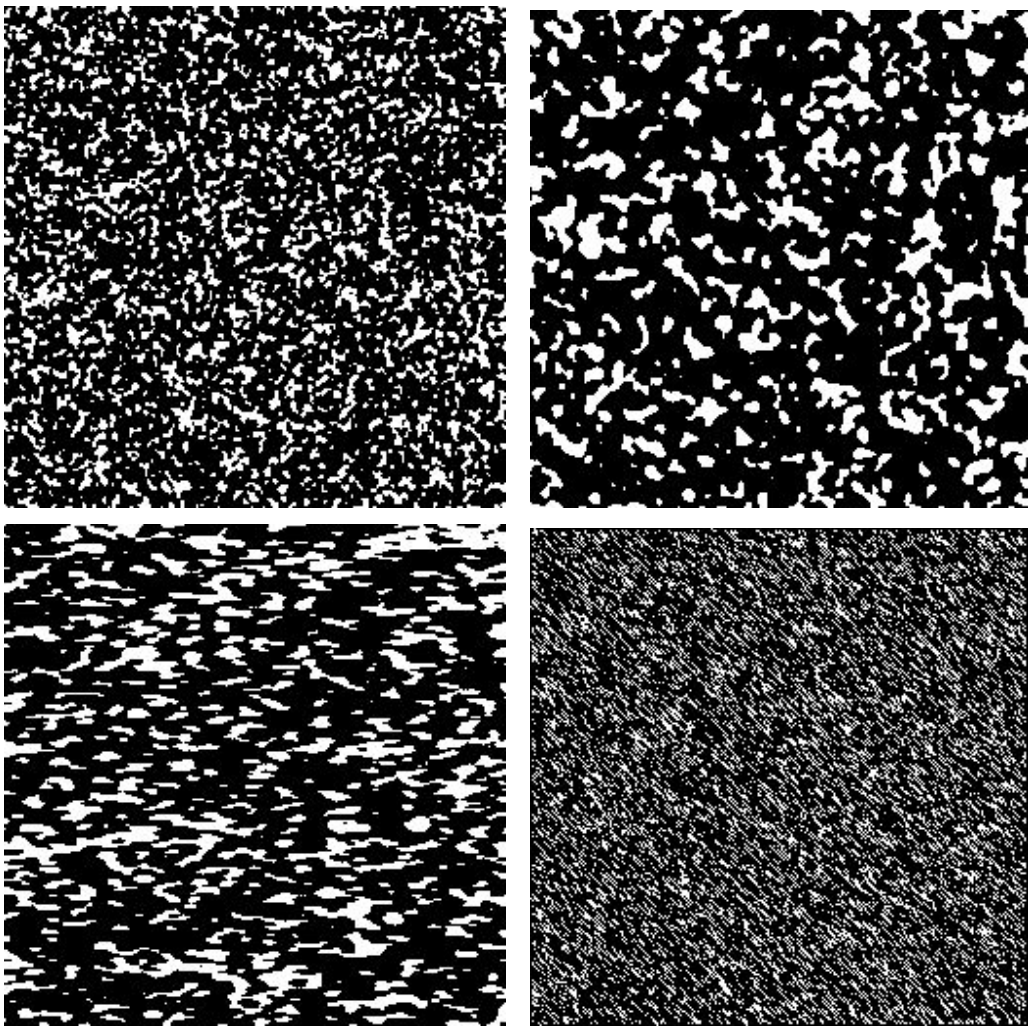
Fundamentally different functionalities of the intelligence system like speech, vision, memory, decision-making, etc call for the need of differently designed neurons in the model. Deep Learning has revolutionized the field of Computer Vision starting with the successful practical demonstration of the Convolutional Neural Networks (CNNs)[4, 5] in combination with the backpropagation optimization algorithm on handwritten digits by LeCun in '89[6, 7]. In 2009, the Stanford AI Lab annotated a dataset of 14 million images called the ImageNet[8] to act as a standard grounds for facilitating the development and evaluating the performances of novel CNN architectures. The field has gone on to expand ever since.

Machine Learning has just recently ventured into the field of Material Science and has been instrumental in accelerating both fundamental as well as applied research in the field. With the increased availability of huge raw datasets, fundamental algorithmic developments and presence of cheap compute power, data-driven automation pipelines are becoming more and more easier to develop. Another reason why deep neural network models are suitable for analysis of data in these engineering fields is, despite our relatively incomplete understanding of the inner workings of such networks or in other words, the black-box behaviour of the nets, they learn remarkable features from the raw or relatively unpreprocessed data automatically, that is, the models are end-to-end, getting rid of the need for the intermediate tedious manual efforts and human intuition required earlier for feature engineering in kernel based algorithms[9, 10] and subsequent inference. New grounds have been broken in predictive stable material discovery and

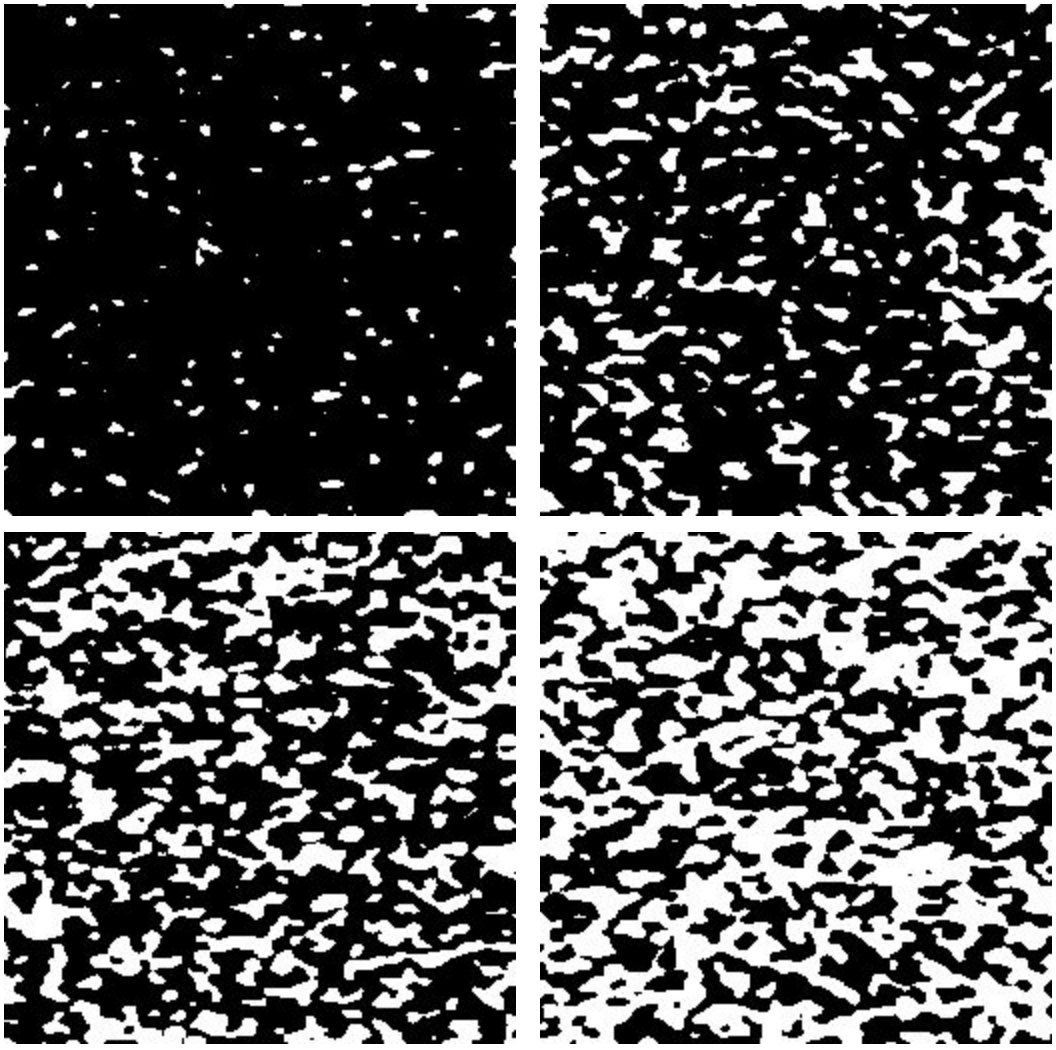
structure prediction, quantitative metallography, analysis and calculation of fundamental properties of materials, aiding first-principle calculations among others.

Motivation and Related Work

The dataset is a central node in the pipeline of any data-driven experiment. Although ideally deep learning outperforms alternatives when trained on a considerably large volume of data, it is not always possible to collect such an amount of raw data, especially in fields like the Material Sciences owing to a variety of constraints. An alternative would be to simulate the required data synthetically while it might hold risks of modelling errors and missing out on essential features that might act adversarially when inferred on real data inducing brittleness into the model. Although this cannot be generalized to all tasks in the domain and certain tasks can be fairly learned through such generated data. In this work, we use a dataset that faces both such challenges of volume and synthetic generation. We focus on solving the first challenge, while it serves us well to relax the second, as for our purposes, the degree of modelling is enough to represent the geometrical features learnt by our model. Some sample images are shown in Figure 1.



(a)



(b)

Figure 1. (a) sample images of 4 different morphologies with 25% phase by volume of the white phase. (b) sample images with the same morphology with phase volumes of white phase as 5%, 20%, 35% and 50%.

The size of our dataset does not permit us the luxury of training a network from randomized weight matrices and forces us to rely on a set of methods that intend to learn from scarce datasets. The paradigm of *transfer learning*[11] has picked up much interest recently with the development of large robust models trained on massive public datasets. Transfer learning refers to an open problem in the field of machine learning whereby 'knowledge' gained through learning a certain task in one (source) domain is re-used or transferred to solve the task at hand in some related (target) domain. This has been especially effective in computer vision tasks and several state-of-the-art results have been proposed in image classification[12, 13, 14], eliminating the need for high computational costs and enabling exploitation of smaller datasets.

Novel CNN architectures are trained and evaluated on standard datasets such as ImageNet[8] and these trained models are available to be repurposed as pre-trained models for subsequent transfer learning usage. The ConvNets are essentially composed of two different types of layers of neurons. The initial layers are the convolution layers performing the convolution operation and extracting 'visual' features from the raw inputs, that is, mapping arrays of pixels to feature vectors. The subsequent and the final layers are fully connected and act as a nonlinear transformation from the feature vector space to the desired output. It turns out that the learnt feature representations are hierarchical, that is, the initial base layers extract generic, problem-independent and higher level features while the higher layers that are closer to the output extract task-specific or specialized features that help the network perform the current task better. So, the convnet is broadly divided into feature extractor and a classifier as shown in Figure 2. Hence, somewhere along the network there must exist a point of bifurcation where the transition must be occurring[15]. Although this point cannot be isolated, it is upto the investigator's imagination to decide how much of the pre-trained network has to be declared useful and retained for the target task. Depending on the similarity of the source and the target dataset and the size of the target dataset the two extremities are either the entire model can be trained from scratch or only the task-specific classifier is replaced and retrained according to the new data keeping the old feature extractors frozen. We have a small dataset leading to a risk of overfitting hence, forbidding us to allow for too many unfrozen parameters in the model. Furthermore, there have been studies on the effect of different classifiers on top of frozen feature extractors[5, 16] which we will not be investigating.

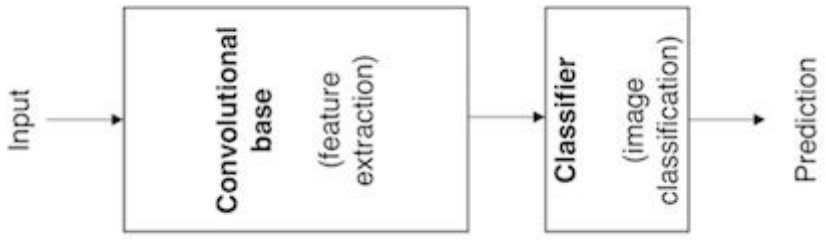


Figure 2. A broad schematic breakdown of the major components of any CNN model.

Quantitative metallography is central to the field of materials engineering and design as it binds together the structural properties of microstructures to their physical counterparts. Traditional approaches in this field have largely involved manual but careful quantitative measurements of volume fractions, size distributions, shape descriptors and other geometric properties for 3-dimensional microstructural features such as grains or suspended phase particles, and constitute what is called a quantitative microstructural state. These states of microstructure are connected to models of material properties. Although standard computerized methods to

analyse images of microstructures like morphology quantification by relative elongation of phases or feature segmentation used to employ heavy image processing techniques, such pipelines were very specialized and brittle to generalization, that is, required manual tuning of experts for previously unobserved microstructural systems. Furthermore, the complexity of microstructures bounds the ability for subjective analysis, and laborious annotation becomes infeasible. This called for the need of an automated generalizable pipeline and deep learning has provided just that. CNNs have aided material scientists in application of computer vision techniques to the context of microstructures since deep learning gained popularity. There have been studies for classification of microstructures[17, 18] using these techniques, using pre-trained CNNs for learning relationships between preprocessing conditions and microstructures through some unsupervised methods such as dimensionality reduction[19] and visualization techniques[20], semantic segmentation to separate out constituent phases in a microstructure[21] and combine these models to calculate denuded zone width distributions and empirical particle sizes[22]. This intersection of quantitative metallography and deep learning has been more or less dominated by some form of supervised learning, which requires manual annotation of data. Our work provides an alternative to a semantic segmentation approach to classification which is a pixel-wise classification problem and naturally involves a heavy and tedious annotation process whereas our approach works on top of labels assigned to images as a whole, going easier on the annotation process.

Multi-label classification is still a relatively new and open challenge in the space of classification problems. Nevertheless, many tasks are inherently multiple labelled. The area comes with a specific set of obstacles that need to be tackled for usable predictions. The correlation among the labels have been exploited using graph neural networks[23]. With a large set of labels comes the problem of class imbalance, that is, a greater number of possible negative labels than the possible positive labels injecting inaccuracies into the predictions. This has been addressed by designing an asymmetric focal loss, focusing more on positive samples leading to a more balanced network and the current state-of-the-art results[24]. Quantitative metallography naturally falls into the multilabel classification domain as a single image of a microstructural system consists of multiple distinct features. This appears highly convenient for situations where the required measurements can be discretized into classes and annotated accordingly.

Later we propose a bifurcated network architecture which shall facilitate the extension of this problem into a regression space catering to the need of continuous real values as demanded by common quantitative metallography routine.

Methodology

Data

The data is an artificially generated experimental set of binary phase microstructure images. The phases were generated through thresholded Gaussian filtered noise. The shape of the phases was controlled by a Gaussian filter with different variances along the 3 directions. The volume fraction is also controlled by thresholding[25]. A dataset of total 3500 images consisting of 7

morphology classes and 10 discrete volume fraction classes was synthesized. Each image is mapped to exactly two labels.

$$\mathcal{D} = \{(x_1, (y_1^{(1)}, y_1^{(2)})), (x_2, (y_2^{(1)}, y_2^{(2)})), \ldots, (x_m, (y_m^{(1)}, y_m^{(2)}))\}$$

Model Architecture

The basic pretrained feature extractor chosen for our purpose was a deep residual network[14], ResNet50, a residual network with 50 layers. These types of networks are proven to be more easily trainable than conventional CNNs and their skip connections solve the degrading accuracy problem. A simplified representation is shown below in Figure 3. After the choice of the model, the classifier of the pretrained model is replaced with a task specific classifier. Depending on the mapping of data to the label space there can be a variety of tail end architectures and loss functions.

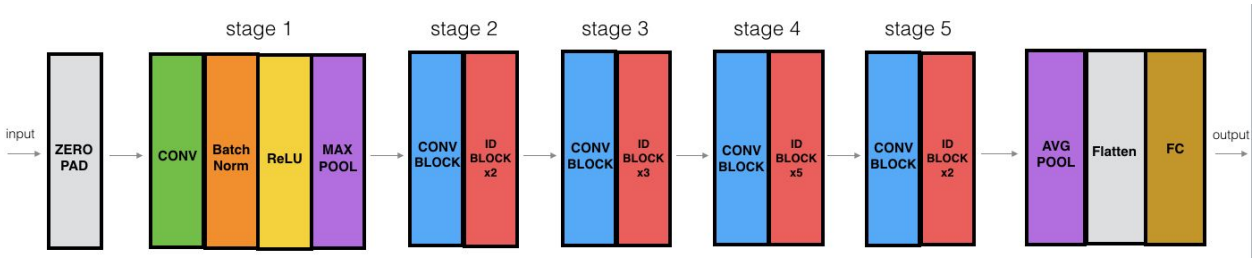


Figure 3. A simplified schematic representation of ResNet50. Each ‘conv’ and ‘id’ block consists of 3 convolutional layers. The model has a total of around 23 million trainable parameters.

Application to quantitative metallography allows us to exploit the design choice of the problem. In contrast to problems like movie genre prediction, where the assigned labels are chosen from a common pool, our problem allows us to separate the annotated labels into distinct property classes such that one label from each class has to be assigned to any given input. A common pool of labels might have gone with a solution where the size of the output layer has ‘ℓ’ neurons where ‘ℓ’ is the total number of labels present, and use a sigmoid-type activation function to indicate whether a label has been assigned or not and a binary cross entropy loss. Whereas our problem design allows us to reduce the problem into several simultaneous classification problems.

$$\mathscr{L} = \sum_{i=1}^{n} \ell_i$$

where, $\mathscr{L}$ = total number of annotated labels,
n = total number of distinct property classes, and
ℓ_i = number of labels for the i^{th} property class

A trivial but probably the ideal solution to our reduced problem would be to maintain n separate networks for n separate predictions. However, it will have the highest computational inefficiency. As the properties form distinct classes and are uncorrelated, instead, we propose a shared network for handling the several classification problems. The features learnt will be shared for all the classification tasks and only the respective replaced classifiers will be tuned. This results in a n-furcated classifier on top of the feature extractor. We show that the representations learnt by a common feature extractor suffices to serve multiple independent classification problems.

$$\begin{aligned} \text{output, } \mathbf{o} &= \mathscr{MM}(\mathbf{i}) \\ &= [(\mathscr{C}_1 \circ \mathscr{FE})(\mathbf{i}), (\mathscr{C}_2 \circ \mathscr{FE})(\mathbf{i}), \ldots, (\mathscr{C}_n \circ \mathscr{FE})(\mathbf{i})] \\ &= [o_1, o_2, \ldots, o_n] \end{aligned}$$

Thus, the i^{th} output, $\mathbf{o_i} = (\mathscr{C}_i \circ \mathscr{FE})(\mathbf{i})$

$\mathscr{MM}$ = the proposed model,
$\mathscr{C}_i$ = the i^{th} fully-connected module, and
$\mathscr{FE}$ = the pretrained feature extractor.

Such an architecture, as mentioned earlier, can be extended to generalize to a mixture of classification and regression problems. Properties which take continuous real values can have a fully-connected layer output a real value whereas in the same model a certain different property class could be classified into some discrete labels. This architecture then serves the purpose of both an automated qualitative and a quantitative pipeline for microstructural analysis.

Experiments and Results

Single label classification

Initially the datasets are marginalised and the pretrained network is retrained to learn single classification problems. We have two classes of labels, the morphology class and the discrete phase volume percentage class. The fully-connected layer is replaced with layers finally outputting ℓ_i values. Depending on the size of the data further training could have been carried out into the final convolutional layers but as it turns out retraining only the classifier provides significant results. The parameters of the layers below the classifier are frozen and the optimizer is passed only the trainable weights.

The amount data split is varied and the network is allowed to train for a maximum of 1024 epochs over the dataset. The results are shown below in Table 1 & 2 and the corresponding graphs are shown in Figure 4. This is a very application specific work and hence no data is available for a comparative study.

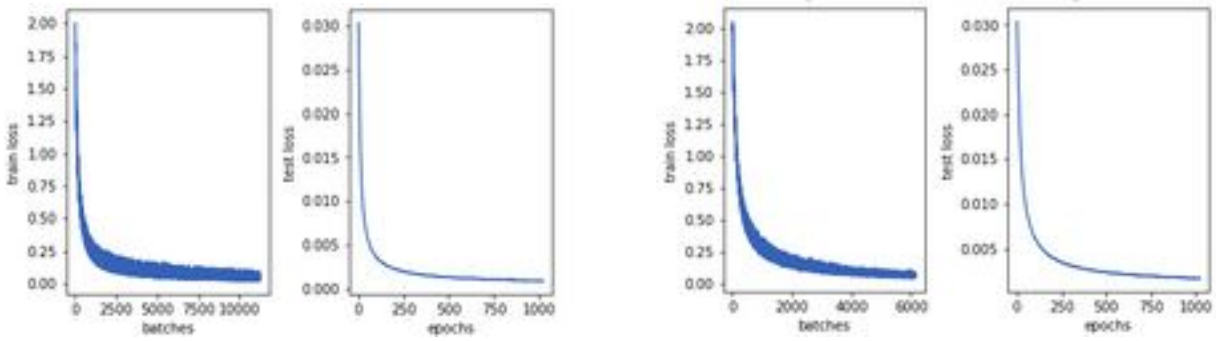
Morphology class prediction

	Test-train split	Error rate	epochs
1.	0.1-0.9	**0.0000%**	38
2.	0.5-0.5	**0.0000%**	698
3.	0.9-0.1	0.3175%	1024
4.	0.95-0.05	2.6466%	1024

Table 1. Results for the morphology class prediction. The test-train split of the data, the error rate of classification on holdout data after the commencement of training and the number epochs required to reach the reported result.

As can be inferred from Table 1, the network is able to perfectly classify morphology of microstructures even when trained on 50% of a small dataset, that is, just 1750 images in our case. This is because the pretrained network has already learnt a very rich set of representations when it was trained on millions of images of the ImageNet dataset and we leverage those representations or ‘transfer’ that knowledge for our task. Even when only 10% of the data, or just 350 images (50 per class) are available to the network, the error is still lower than 0.32% which is a remarkable result. The error rates shoot up as the test-train approaches extreme values.

In comparison, a not-so-deep traditionally designed CNN consisting of only convolution, pooling and linear layers, even when trained on 90% of the data still shows an error rate of around 2% on morphology classification and a rate of 5% on volume fraction classification!



(a) (b)

Figure 4. Loss graphs. (a) for entry 3 of Table 1. (b) for entry 4 of Table 1.

A similar set of results are reported for phase volume fraction classification below in Table 2 and Figure 5.

Volume fraction prediction

	Test-train split	Error rate	epochs
1.	0.1-0.9	0.4011%	212
2.	0.5-0.5	0.9882%	902
3.	0.9-0.1	2.3556%	1024
4.	0.95-0.05	6.2126%	1024

Table 2. Results for the morphology class prediction. The test-train split of the data, the error rate of classification on holdout data after the commencement of training and the number epochs required to reach the reported result.

The error rates and number of epochs here are slightly higher than in the previous morphology classification case probably because there are more number of classes for the same amount of data and hence less amount of data per class.

Multi-label classification

For the multilabel case, we use the bifurcated model architecture proposed earlier.

The equivalence argument

One of the key arguments for why such a bifurcated classifier network with frozen feature extractors should work is that in the single label experiments mentioned earlier, the only difference in the network architecture between different experiments were the separate classifiers. Hence, if separate classifiers are designed, and their input is provided from the fixed block of neurons, there seems to be an equivalence between maintaining separate network for all the tasks and a state of superposition where a common feature extractor is made to provide input to the classifiers that are part of the same structure of networks.

The resnet50 model has a fully-connected layer of dimensions (4096, 1000) which is removed and replaced by a fc_layer with (4096, 100) and then bifurcated to *n* (=2) blocks of fully-connected classifiers (100, 7) and (100, 10). The classifiers can be made deeper if needed and are solely dependent on the designer's choice.

We use separate cross entropy loss functions for separate classifiers aligned with our view of the problem as superposition of uncorrelated classification tasks. Hence the individual classification accuracies should approximately mirror simultaneous classification accuracies. The only point of commonality that keeps us from assuming that these problems are uncorrelated and independent in the truest and strictest sense is the fc_layer which is shared by the path of forward propagation for all the *n* tasks. The gradients flowing back through the classifiers train them for that specific task and all the backpropagating gradients meet, sum up and update fc_layer assuming that is the maximum depth till which the pretrained network weights are kept unfrozen. The results are reported below for this experiment in Table 3.

	fc_layer o/p size	cumulative error	saturation epochs
1.	100	8.1429%	320
2.	1000	8.0119%	521

Table 3. Results for the multi-label classification experiments. The input size to the bifurcated classifiers, cumulative of the individual classification error rates and the epochs at which saturation of accuracy occurs.

A higher rate of error (of around 8%) than that expected is probably due to the common fc_layer which prevents the network to function as multiple independent classifiers. Considering the size of the dataset these results are fairly good.

Conclusion

We proposed to frame quantitative metallography as a natural multi-label classification problem as an alternative to the intra-image classification approaches that are popular in the literature for ML in metallography such as semantic segmentation which would demand a tedious pixel-by-pixel annotation of the data. In comparison, an annotation over images could be fairly simple and aided by traditional metallographic techniques. Furthermore, we worked with a small dataset as would be the case with investigators carrying out their personal experiments and without access to a large labelled dataset based on their specific problem statement. Constrained by this, we decided to reuse the features learnt by state-of-the-art pre trained networks and showed that it performs well on such small datasets without retraining too many of the parameters. Then we exploited the structure of the quantitative microstructural analysis task to observe the presence of distinct uncorrelated classes of properties associated with the microstructure images. To facilitate this we argued and showed that such a structure could use a parted graph for a network architecture with a fixed feature extraction mechanism and training this *n*-furcated network would be equivalent to performing simultaneous classification or a mixture of classification and regression tasks with separately maintained networks.

Further work

There could be many directions to further this work. Some immediate work could involve devising a novel loss function taking into account the error for simultaneous classification of all the labels, eliminating common paths while forward propagation to ensure true independence among tasks and focusing on optimizing the current accuracies of the model. Addition of a contrastive and other unsupervised losses to differentiate between the images could help. Another extension of this work could be addressing the extremities of the amount of labelled data available, that is, to a situation where considerable data is available but a small fraction is labelled, in other words, a semi-supervised approach to multi-label classification. This could address the situation where material scientists are able to generate a lot of raw unlabelled data (such as images of microstructures) but annotation is tedious and hence annotated data is limited in number. Finally, any representation method that combines the best of sample-efficiency or annotated data complexity with lower error rates will stand superior and prove itself to be most useful for applications to fields such as materials science and engineering.

References

[1]LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. nature, 521(7553), p.436.

[2]Chakraverty S., Sahoo D.M., Mahato N.R. (2019) McCulloch–Pitts Neural Network Model. In: Concepts of Soft Computing. Springer, Singapore. https://doi.org/10.1007/978-981-13-7430-2_11

[3]Hadamard, J. (1908). Mémoire sur le problème d'analyse relatif à l 'équilibre des plaques `` élastiques'' encastrées . Mémoires pr ´ esent´ es par divers savants `` à l ` emie des sciences de l'Institut de´ France: Extrait. Imprimerie nationale.

[4]Bengio, Y., 2009. Learning deep architectures for AI. Foundations and trends in Machine Learning, 2(1), pp.1–127.

[5]Rawat, W. and Wang, Z., 2017. Deep convolutional neural networks for image classification: A comprehensive review. Neural computation, 29(9), pp.2352–2449.

[6]LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Back-propagation applied to handwritten zip code recognition. Neural Computation, 1(4):541–551.

[7]LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1990a). Handwritten digit recognition with a back-propagation network. In Touretzky, D. S., editor, Advances in Neural Information Processing Systems 2, pages 396–404. Morgan Kaufmann.

[8]Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. and Fei-Fei, L., 2009, June. Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (pp. 248–255).

[9]Vapnik, V. (1995). The Nature of Statistical Learning Theory. Springer, New York.

[10]Scholkopf, B., Burges, C. J. C., and Smola, A. J., editors (1998). ¨ Advances in Kernel Methods - Support Vector Learning. MIT Press, Cambridge, MA.

[11]Pan, S.J. and Yang, Q., 2010. A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), pp.1345–1359.

[12]Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097–1105).

[13]Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[14]He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770–778).

[15]Yosinski, J., Clune, J., Bengio, Y. and Lipson, H., 2014. How transferable are features in deep neural networks?. In Advances in neural information processing systems (pp. 3320–3328).

[16]Tang, Y., 2013. Deep learning using linear support vector machines. arXiv preprint arXiv:1306.0239.

[17]Brian L. DeCost and Elizabeth A. Holm. "A computer vision approach for automated analysis and classification of microstructural image data". In: Computational Materials Science 110 (2015), pp. 126–133.

[18]Aritra Chowdhury et al. "Image driven machine learning methods for microstructure recognition". In: Computational Materials Science 123 (2016), pp. 176–187 (cit. on p. 2).

[19]Nicholas Lubbers, Turab Lookman, and Kipton Barros. "Inferring LowDimensional Microstructure Representations Using Convolutional Neural Networks". In: CoRR (2016).

[20]Brian L. DeCost, Toby Francis, and Elizabeth A. Holm. "Exploring the microstructure manifold: image texture representations applied to ultrahigh carbon.

[21]Seyed Majid Azimi et al. "Advanced Steel Microstructural Classification by Deep Learning Methods". In: Scientific Reports 8.1 (2018). issn: 2045-2322.

[22]Brian L. DeCost et al. High throughput quantitative metallography for complex microstructures using deep learning: A case study in ultrahigh carbon steel. Microscopy and Microanalysis Aug. 21, 2018.

[23]Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5177– 5186, 2019.

[24]Emanuel Ben-Baruch et al. Asymmetric Loss For Multi-Label Classification. arXiv preprint arXiv:2009.14119

[25]Hyman J. & Winter L., Stochastic generation of explicit pore structures by thresholding Gaussian random fields, Journal of Computational Physics, 277, 2014