Master in High Performance Computing

Representation Learning and Hierarchical Clustering for microscopy images

Supervisor(s):
Dr. Stefano Cozzini Supervisor,
Dr. Alessio Ansuini Supervisor

Candidate:
Alberto Cazzaniga

6th edition
2019–2020

Contents

Chapter 1

Introduction

The Nano Foundries and Fine Analysis (NFFA) Europe project has been conceived with the objective of creating a platform to promote multidisciplinary research at the nanoscales connecting several specialised European institutions and laboratory facilities. Within the project particular effort has been focused on the creation of an Information and Data management Repository Platform (IDRP) to collect and maintain diversified data resources, with particular focus on meeting open and FAIR best practices.

The analysis of images collected via scanning electron microscope (SEM) has been one of the main case studies. During the last five years, continuous collaboration with the nanoscientists at the CNR-IOM facilities led to a careful analysis of the metadata of SEM images, and to the creation of several publicly available datasets of images humanly classified by content into 10 categories. Training the weights of a convolutional neural network on the labelled images enabled to create a service able to predict the belonging of newly acquired images to one of the 10 NFFA categories.

The classification into the 10 NFFA categories provides an extremely valuable proof of concept of how modern deep learning techniques can serve as a tool for supporting and stimulating scientific discoveries. Nonetheless, the classification is too coarse and cannot possibly cover the entire spectrum of content of the experiments collected at the various facilities. This motivates the study of state-of-the art techniques that allow at the same time to refine the NFFA classification while adapting to the continuous data acquisition. This work aims at a combined investigation of supervised and unsupervised deep leaning techniques that have the potential of meeting these requirements.

Thesis overview

This project has been focusing on learning representations of SEM images which are sensitive to their semantic content, with the objective of defining a fine grain hierarchical structure on the existing dataset without requiring further human effort.

Chapter 2 is devoted to the description of the SEM datasets and of the format and characteristics of their images. We discuss the role of the scale metadatum, and we describe the SEM_Hierarchical dataset, a hand-crafted dataset containing a only few images but that constitute a prototype of dataset with a fine grain classification.

Chapter 3 is divided in two parts. The first part motivates the choice of the deep learning model we employ for extracting representations. The quality and characteristics of the representations clearly depend on the training procedure, that we describe in the second part of the Chapter. In particular, we discuss the details of four training procedures, and report their statistics on the SEM datasets: the fine-tuning and triplet loss methods that leverage on the labels of the SEM_dataset, the "pure" transfer learning procedure that is based only on weights learned on the ImageNet dataset, and the deep clustering algorithm which allows to learn semantic features of the SEM images in a fully unsupervised manner.

Chapter 4 describes methodologies and results of the hierarchical clustering procedure we employ on the SEM datasets. The representation we extract synthesize the content of the images, and the dataset lies on a nonlinear manifold of substantially smaller dimension than the embedding space. After introducing the 2-NN algorithm, we study the intrinsic dimension (ID) of the datasets in Section 4.1. The knowledge if the ID is crucial for employing the advanced density peaks algorithm (ADP), that we describe in Section 4.2. Working on the embedded manifold where the data lies, the ADP algorithm avoids loss of information introduced by projecting onto smaller dimensional spaces, and allows us to define the hierarchical partition of the dataset(s) that motivated this project. Both quantitative and qualitative properties of the results of the clustering procedure are discussed in the final Section 4.2.3.

Chapter 2

The dataset(s)

During the last years the Information and Data management Repository Platform (IDRP) has been created as data-platform for the nanoscience community, and in particular to face the needs of the scientific groups and laboratories participating in the NFFA-EUROPE project. Large effort has been addressed to develop a framework where the acquisition and the organisation of the data meets the FAIR principles (Wilkinson et al., 2016).

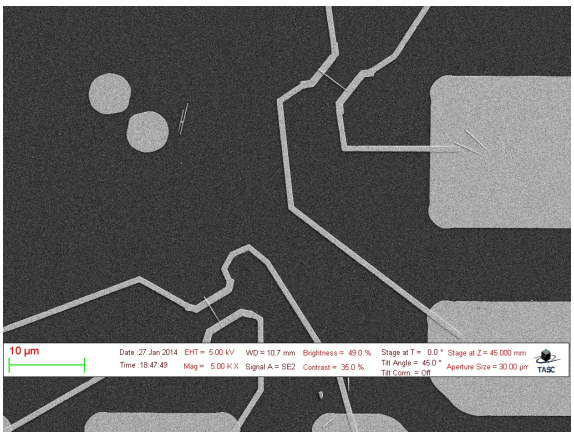


Figure 2.1: An image in the SEM_full dataset.

Particular attention has been focused on images collected via scanning electron microscope (SEM) collected at several European synchrotron facilities, and that will be the main object of study of this thesis. Previous work led to the construction of several publicly available datasets, among which (Aversa et al., 2018c), and a SQL database describing the location, unique identifier,

and metadata of the images has been created to handle continuous data collection and complex queries (Khalil, 2019).

The specific focus of this work concerns the analysis of the visual content of the SEM_full dataset collecting 146917 RGB images, with size of 1024×768 pixels and varying image depth, stored both in TIFF and JPG format. The dataset contains images with unique JPG content, and each element can be retrieved by its unique MD5 hash associated to the JPG or TIFF content, thanks to the analysis developed in Coronica (2018).

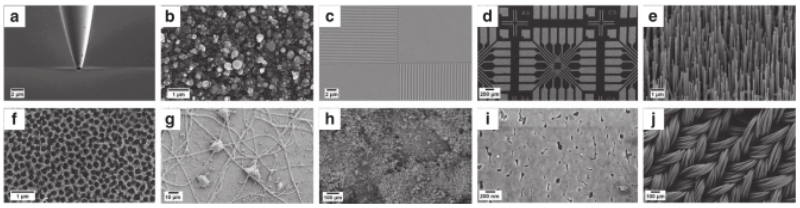


Figure 2.2: Representatives of the NFFA categories of the SEM_dataset: Tips (a), Particles (b), Patterned surfaces (c), MEMS devices and electrodes (d), Nanowires (e), Porous sponge (f), Biological (g), Powder (h), Films and coated surfaces (i), Fibres (j) (Aversa et al., 2018a).

A portion of the images in the SEM_full dataset has been manually classified by content into 10 categories, referred to as the NFFA categories, by previous collaboration with CNR-IOM scientists, see Figure 2.2. All the labelled images collected by Aversa et al. (2018c) and carrying unique JPG content are included in the SEM_full dataset, and this subset of annotated images is denoted by SEM_dataset.

The SEM_dataset contains 18261 images, and it corresponds to a coarse classification in 10 macro-families. A part from being an extremely valuable scientific resource, this dataset can be considered as a prototype for a real life dataset of images:

- the class distribution is highly unbalanced, as shown in Figure 2.3;
- majority classes have large within-class content variability;
- the dataset contains images taken at variable scales.

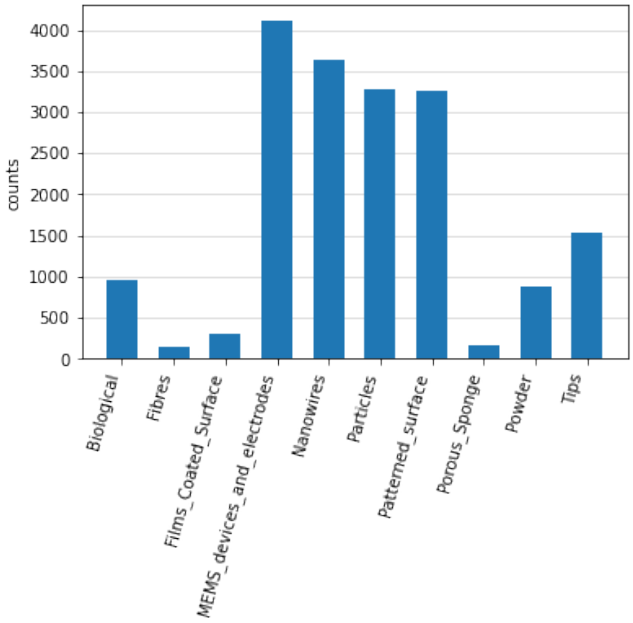


Figure 2.3: Distribution of images in SEM_dataset by NFFA category.

One of the most important information on the images, from a scientific perspective, is the microscope resolution at which the image has been taken. Only in some cases this metadatum can be retrieved from the TIFF file of the images, but it is always hard coded in the image as a segment with the correspondent unit of measure, as in bottom left of Figure 2.1. A procedure involving the use of the OpenCV library and the Tesseract engine for Optical Character Recognition, developed by Coronica (2018) and Khalil (2019), allows to measure the length in pixels of the ruler and to read the corresponding number and unit of measure.

Once the hard coded content on the scale has been detected, one can compute the size in μm corresponding to a single pixel. Figure 2.4 shows that a coherent subdivision of the dataset by scale can be obtained by considering images with the same hard coded unity of measure (μm or nm) and having bar length of the same order. As reported in Table 2.1, the most representative scale group corresponds to hard coded bar with reported annotation 1 μm or 2 μm. The group includes a total of 52682 images of which 7557 are labelled according to the NFFA category subdivision, and it will be denoted by SEM_1u2u in the rest of the thesis. Considering only images in the majority scale group reduces the amount of both labelled and unlabelled images by approximately 75%. For this reason, in order not to lose a significant

Scale	Labelled	Tot
100 nm	1197	16531
200 nm	3426	37052
1 μm	3959	30107
2 μm	3598	22575
10 μm	2840	14729
20 μm	1316	9475
100 μm	790	5941
200 μm	530	3265

Table 2.1: Number of images of most represented scales

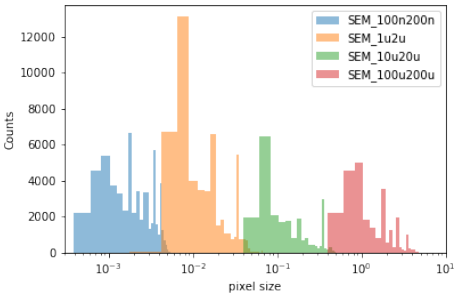


Figure 2.4: Pixel size (in μm) distribution of most represented scales

amount of training labels, it was decided to train our CNN models on images regardless of their scale, and to compare along the way the results obtained on the whole dataset SEM_full and on the majority scale group SEM_1u2u.

From a scientific perspective the NFFA partition in 10 categories is too coarse. This, together with the need for a more balanced class distribution, led to the creation of the SEM_Hier dataset (Aversa et al., 2018b). The images are labelled in 26 classes organised hierarchically in a tree structure of maximal depth 3, see Figure 2.5 (left). The SEM_Hier dataset includes only 1038 images of which just 138 at the scale $1\mu m$-$2\mu m$, and its distribution both in terms of scale and macro-categories does not reflect the one of the SEM_dataset. The main motivation of the thesis is to develop a pipeline for creating a large and fine grain dataset of SEM images organised in a hierarchical structure, and as aligned as possible w.r.t. NFFA classification, without requiring further human labelling.

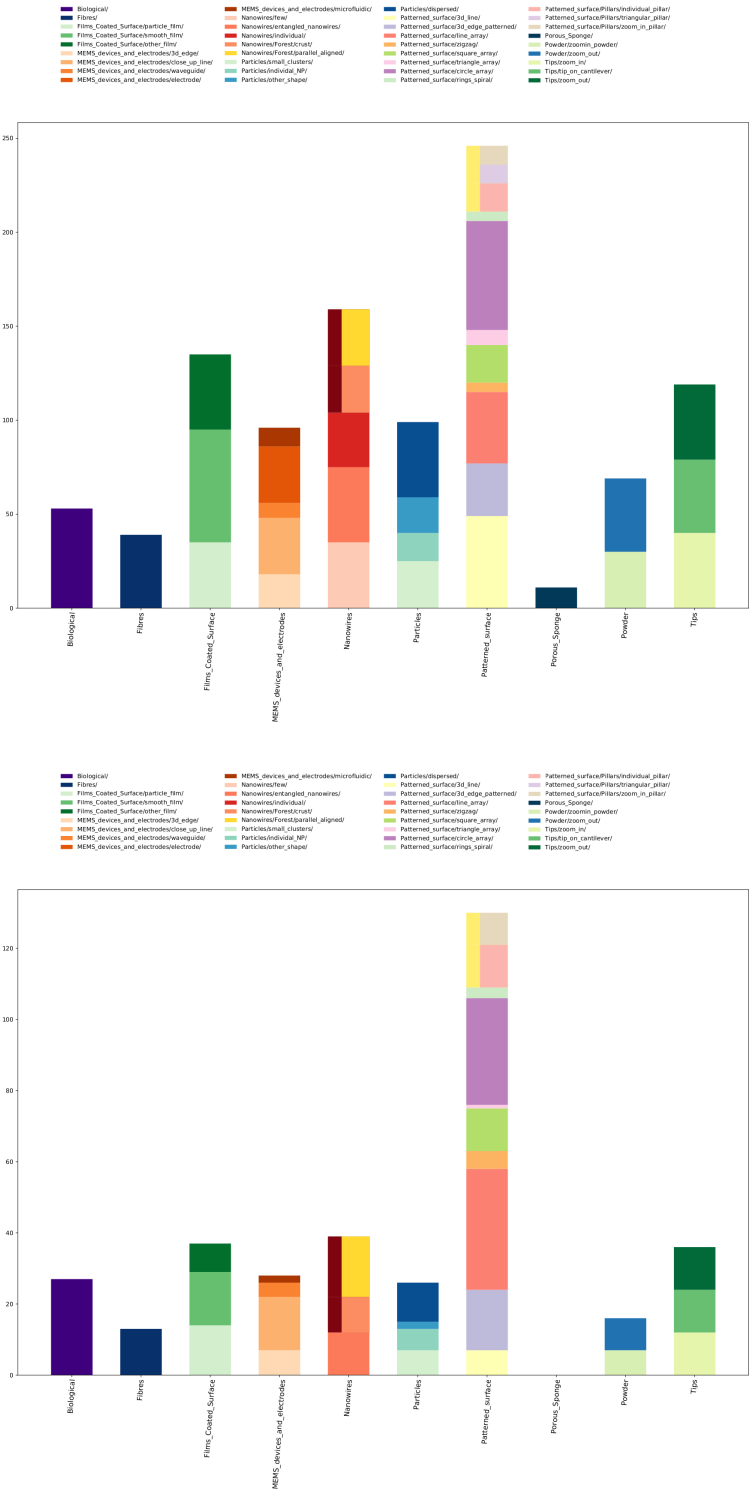


Figure 2.5: Distribution of SEM_Hier (left) and SEM_hier_1u2u (right) datasets.

Chapter 3

Representation learning via convolutional neural networks

With the objective of constructing representations of SEM images that reflect their semantic content, we investigate embeddings of the SEM_full dataset obtained by training a convolutional neural network with different strategies.

The SEM_full dataset can be thought as a point-cloud in the vector space $\mathbb{R}^D$, where each of the 1024×768 coordinates represents the brightness intensity of a pixel. Although natural, this embedding is not suitable for describing the relevant information of an image, since most of the local brightness properties are irrelevant for determining its actual content.

In fact, the necessary information for understanding an image is thought to lie on a manifold of much smaller dimension. Modern deep learning techniques define non-linear transformations to combine the original features of an image into a synthetic encoding that describes its content. The Euclidean space in which this representations are embedded has substantially smaller dimension, and distances between the transformed images describe the similarity of their semantic information.

In principle one would like the representation to contain all the relevant information of our dataset, but the characteristics of the embedding are in practice affected both by the architecture of the neural network, determining its prior and expressive power, and by the training strategy. Being our final objective the automatic construction of a hierarchical partition of the SEM_full dataset, we empirically evaluate the quality of the embedding by studying the correlation between class membership and Euclidean distances on the labelled part of the dataset.

This Chapter is devoted to the study of different techniques for constructing representations of the SEM_full dataset. In Section 3.1 we briefly introduce convolutional neural networks, and describe more in depth the ResNet-50 architecture that was used for extracting representations. We investigate both supervised and unsupervised learning strategies. The supervised approaches of fine-tuning and triplet-based method are introduced respectively in Section 3.2.1 and in Section 3.2.2. A dataset independent strategy where we leverage entirely on weights learned on the ImageNet dataset is discussed in 3.2.1, and a fully unsupervised learning strategy based on the deep clustering algorithm is presented in Section 3.2.3.

3.1 The choice of architecture

3.1.1 Convolutional layers, a snapshot

The *convolution* operation has become ubiquitous in pure and applied sciences. Given a signal f and a kernel g the convolution $f * g$ is defined as

$$(f * g)(s) = \int f(t)g(t-s)dt.$$

The core operation at the foundation of convolutional neural networks (CNN), also denoted as convolution, consists in a discretisation of the classical technique. Convolutions have largely been used before the rise of deep learning for describing hand-crafted *features* of images by means of *kernels* (or *filters*), see Figure 3.1 for an example. The pioneering work of LeCun et al. (1989) signed a shift of paradigm: the most suitable filters to describe the relevant features of a dataset of images to solve a (classification) task are learned by means of the stochastic gradient descent algorithm.

We describe in some detail the simple example when a 2×2 kernel K, encoded by a 2×2 real valued matrix, is applied with stride 2 and no padding on a 4×4 input image X, represented by a 4×4 real valued matrix where each entry denotes pixel intensity. Writing *input* X in block form as

$$\begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{23} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{pmatrix} = \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix},$$

and denoting by $X_{ij} \bigodot K$ the inner product between the vectors obtained by flattening the matrices X_{ij} and K, the *feature map* of the convolution is

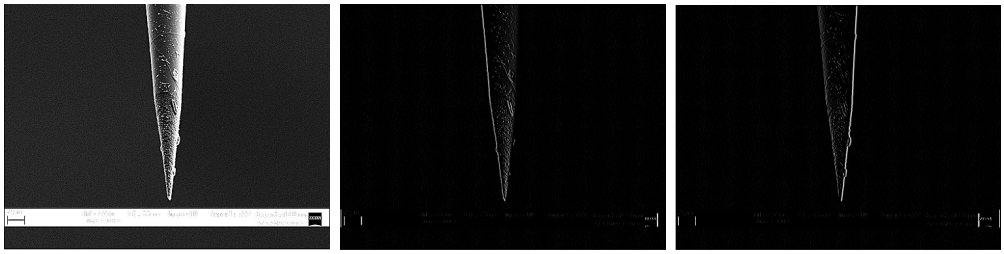


Figure 3.1: An image of a Tip (left). Convolving by handcrafted 3×3 filter obtain left-edge (center) and right-edge (right) detection.

the 2×2 matrix

$$\begin{pmatrix} X_{11} \bigodot K & X_{12} \bigodot K \\ X_{21} \bigodot K & X_{22} \bigodot K \end{pmatrix}.$$

It is worth noticing that, upon flattening the input and the features, the above operation can be expressed as a multiplication by a 4×16 matrix W depending only on the 2×2 parameters of the filter K.

In general, as represented in Figure 3.2, a $(k \times k)$ filter K can be applied sliding with a certain stride s on a size (h, w) input X possibly padded with p zeroes along the edges to obtain an output of size

$$\left(\left\lfloor \frac{h + 2p - k}{s} + 1 \right\rfloor, \left\lfloor \frac{w + 2p - k}{s} + 1 \right\rfloor \right).$$

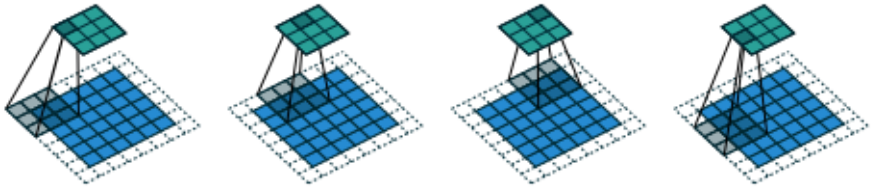


Figure 3.2: Sliding of a 3×3 filter (grey) on a 5×5 input (dark blue) with padding $p = 1$ (dashed) to obtain a 3×3 output (green).

In the case when the input has C_{in} channels, for instance $C_{in} = 3$ when considering RGB images, a filter acts as a $3d$ filter of depth C_{in} by stacking independent $2d$ filters of size $k \times k$ along the input channel.

A (2d)-*convolutional layer* (CL) is a (linear) function f between two vector spaces of dimension respectively $C_{in} \times H_{in} \times W_{in}$ and $C_{out} \times H_{out} \times W_{out}$ obtained by applying C_{out} convolutions by independent filters as above.

Together with *maxpooling* layers, convolutional layers are the main ingredient of the strong prior provided by modern NN architectures employed for computer vision, as we will discuss in Section 3.1.2. Their success can be mainly be traced back to the following properties of CL:

- locality: the response at a neuron in the output is determined only by a local subset of the input with same size as the kernel. This improves statistical efficiency allowing detection of detailed features, such as edges, and, in the case when the filter is of significantly smaller size than the input, it reduces computational complexity;

- translation equivariance: applying a translation T on the input results in a (related but possibly different) traslation T' of the output. This property is fundamental in classification and detection tasks in image recognition as it ensures that the relative spatial position of the relevant features is preserved;

- parameter sharing: since, at least in the case of images, the importance of a feature is essentially independent of its position, the same filter is applied to different portions of the input. Given input and output respectively of size n and m the memory requirement is reduced from $O(n \times m)$ to $O(k^2)$.

3.1.2 Model selection: ResNet-50 architecture

As discussed in the introduction of the Chapter, we aim at extracting features of images in the SEM_full dataset that model the probability distribution $P(X)$ of the image content. The strategy will be to extract representations from a CNN trained with different levels and type of supervision on (a part of) the SEM_full dataset.

The choice of the CNN to employ for this purpose is motivated by the results of the following preliminary experiment comparing the main state-of-the-art CNN designed for computer vision tasks. We test a very coarse approximation of our final pipeline for evaluating the expressive power of the different architectures. Given a model:

- load model weights pretrained on the ImageNet dataset (Deng et al., 2009) and remove the classification head (see Section 3.2.1 for a more detailed description of the procedure);
- extract representations of images in the SEM_dataset, and apply principal component analysis (PCA) to reduce the dimensionality from 2048 to 256;
- perform K-means clustering with $k = 100$ clusters. We consider as a measure the normalised mutual information (NMI), defined in Section 4.2.2, between the clustering of the SEM_dataset obtained via K-means and the one induced by the NFFA labels.

The following architectures have been considered: AlexNet (Krizhevsky et al., 2012), InceptionV3 (Szegedy et al., 2016), several models in the ResNet family (He et al., 2016), and ResNetXt-101 (Xie et al., 2017). The results of the experiment are reported in Table 3.1.

Table 3.1: Experiment results, model selection.

Model	**NMI**
AlexNet	0.384
InceptionV3	0.408
ResNet-34	0.415
ResNet-50	0.420
ResNet-101	0.421
ResNet-152	0.432
ResNeXt-101	0.424

The results of our experiment highlight a superiority in the expressive power of ResNet-like models, also when compared to the InceptionV3 architecture employed in previous supervised and semi-supervised approaches on the SEM dataset (Modarres et al., 2017; Aversa et al., 2020). The final choice of the ResNet-50 architecture, that we describe in some detail in the remaining part of this Section, aims at combining high expressivity, and reduced computational and memory requirements with respect to deeper models.

The main novelty introduced by ResNet-like architectures is the presence of *skip connections* between convolutional layers. We describe in detail how a skip connection between two convolutional layers is constructed, and delegate to Figure 3.3 and (He et al., 2016) the general case. Given an input x, let $\mathcal{F}(x, W_i)$ be the image of the composition $W_2 \circ \sigma \circ W_1$, where W_1 and W_2 are convolutions and σ is the ReLu activation function. Adding a skip connection between the two convolutional layers amounts in considering as output y the result obtained applying σ to $\mathcal{F}(x, W_i) + x$. In the case when the output $\mathcal{F}(x\,, W_i)$ has dimension different from the input x, a 1×1 convolution W_s with suitable stride and number of feature maps is applied to x for matching dimensions, resulting in $\mathcal{F}(x, W_i) + W_s x$.

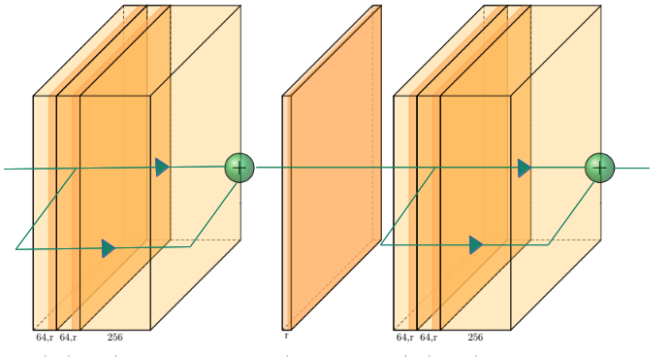


Figure 3.3: Residual connection between two blocks in the conv2_x ResNet-50 layer. Light orange: convolutional layers; dark orange: ReLu layers.

Increasing the depth of a network increases its expressive power. Skip connections are introduced to address the problem of saturation of accuracy and degradation in performance when training models of considerable depth. In particular, adding a residual block to an already deep network makes particularly easy for the model to learn the identity function on the last block; this drastically reduces the possibility of a degradation of performance introduced by an increase of the depth. Furthermore, skip connections contribute to reducing the effect of vanishing gradients that slows convergence. Notice also that this construction does not change significantly the number of weights, and does not alter computational complexity.

Excluding an initial convolutional layer conv1, a maxpooling operation in conv2_x, and the pooling+classification head in the last layer, the ResNet-50 architecture described in Table 3.2 has a modular structure made of four blocks with repeated residual units. Each unit is constructed as follows:

two convolutional+ReLu layers are applied to the input x, the result of this operation is passed through a further convolutional layer, a skip connection adds the result of the last convolution to the input x, and finally a ReLu activation function is applied to obtain the output of the residual unit.

Table 3.2: Layer structure of ResNet-50 architecture.

Layer name	**Output size**	
conv1	112×112	$7 \times 7, 64$, stride 2
conv2_x	56×56	3×3 max pool, stride 2 $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 4$
conv3_x	28×28	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
conv4_x	14×14	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$
conv5_x	7×7	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	avg pool, 1000-d fc, softmax

The modularity of the architecture allows to easily increase depth by modifying the number of residual units in the various blocks. For instance, the ResNet-152 architecture is just obtained by considering 8, resp. 32, units in the layer conv3_x, resp. conv4_x, blocks.

3.2 Learning representations of SEM images

As discussed in the introduction to the Chapter, the quality of the embedding of a dataset can be discussed only in terms of a successive task. Since we want to construct representations of the SEM_full dataset in order to perform clustering as a downstream task, we want the Euclidean distances of the learned representations to reflect content similarity of the images. We

use membership to the NFFA categories as a proxy for content similarity.

Table 3.3: Number of sample per category, SEM_full and SEM_1u2u datasets

NFFA category	**full**	**1u2u**
Porous_Sponge	20	20
Patterned_surface	81	76
Particles	81	46
Films_Coated_Surface	20	20
Powder	22	20
Tips	38	35
Nanowires	90	94
Biological	23	29
MEMS_devices_and_electrodes	102	78
Tips	20	19

Throughout rest of the Section we consider the following empirical test. We sample a fixed randomly extracted selection $\{x_i\}$ of labelled images in SEM_full (resp. SEM_1u2u) dataset as in Table 3.3, making sure that minority classes are sufficiently represented. Class membership in the NFFA categories $\{y_i\}$ naturally defines a discrete distance matrix

$$d_{disc}(x_i, x_j) = \delta_{y_i, y_j}.$$

Ordering the images for increasing value of the NFFA label one obtains a heatmap with blocks on the diagonal that simply describes class membership of the selection $\{x_i\}$, Figure 3.4. Given an embedding f_θ defined by forward-pass through a suitably trained ResNet-50 model, we will study the distance matrix

$$d_\theta(x_i, x_j) = d_{eucl}(f_\theta(x_i), f_\theta(x_i))$$

discussing the properties of the corresponding heatmap, and we will consider the Pearson correlation coefficient between d_{disc} and d_θ as a preliminary measure of the quality of the representations that will be considered for clustering in Section 4.2.

3.2.1 Representations from transfer-learning

Suppose we are given a set of data points X with a certain probability distribution $P(X)$, and suppose a model f_θ, depending on some parameters θ,

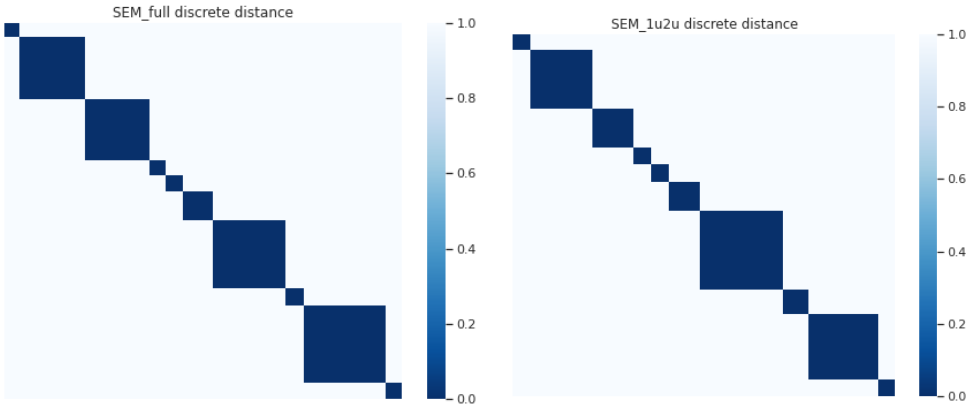


Figure 3.4: Heatmap of discrete distance d_{disc} for selection of SEM_full (left) and SEM_1u2u (right) datasets.

has been trained on data pairs $\{x_i, y_i\}$ corresponding to a set of labels Y. Assume now we are given a different but compatible dataset $(X', P(X'))$, and we are required to solve a task requiring to minimise a loss function f'_ω with respect to a given set of labels Y'. *Transfer learning* consists of using the weights θ learned by solving the *source* task f_θ on (X, Y) to simplify the solution of the *target* task f'_ω on the dataset (X', Y').

In our applications, the source domain X and the set of labels Y consist of the ILSVRC2012 dataset, a subset of the Imagenet dataset (Deng et al., 2009) containing 1000 mutually exclusive classes. In particular, we consider the weights θ obtained by training of a ResNet-50 model on the Imagenet classification task, as provided by the PyTorch library.

The fist application consists of a *"pure" transfer learning* procedure. We consider SEM_full (resp. SEM_1u2u) as target dataset and we think of our unsupervised clustering procedure of Section 4.2 as the target task. We leverage on the fact that the very general and complex classification task posed by the ILSVRC2012 dataset forces the model to learn weights which are useful also for extracting representations of the SEM_full (resp. SEM_1u2u) dataset that are sensitive to the content of the SEM images. We extract SEM_full (resp. SEM_1u2u) representations of the target dataset X': after loading pre-trained weights on the ResNet-50 model, we remove the classification head, and we perform a forward-pass on the SEM_full (resp. SEM_1u2u) dataset to obtain representations $f_\theta(X') \subset \mathbb{R}^{2048}$.

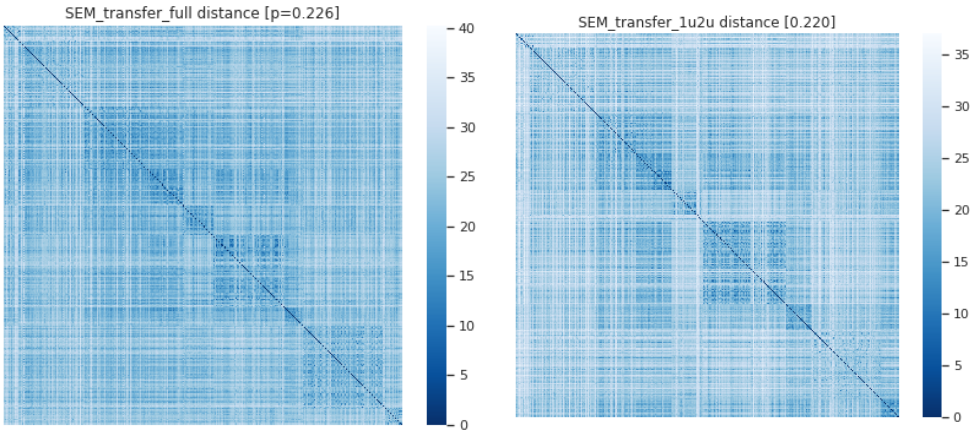


Figure 3.5: Euclidean distances of SEM_full (left) and SEM_1u2u (right) representations by transfer learning. p measures Pearson correlation coefficient with d_{disc}.

The correlation coefficient between the distance matrix $dist_{eucl}(f_\theta(x_i), f_\theta(x_j))$ and the discrete distance d_{disc}, reported in Figure 3.5, is low. Nonetheless, having a non-trivial signal proves that the weights learned classifying images of the Imagenet dataset are relevant to discriminate the semantic content of SEM images. Comparison with analogous results obtained by means of the InceptionV3 architecture (Coronica, 2018, Figure 3.10) highlight the higher expressive power of the ResNet-50 model.

The second application consists of extracting representations by means of a *fine-tuning* procedure. In this case the target dataset is the SEM_dataset, and the target task is the classification into the 10 NFFA categories. After loading on the ResNet-50 model weights θ pretrained on the ILSVRC2012 dataset, we replace the classification head with a randomly initialised fully connected layer with 10 outputs followed by a softmax layer. We train the model on the SEM dataset minimising cross-entropy loss function with stochastic gradient descent (SGD) optimisation and batch size 32. As a result of a grid search procedure, we set learning hyperparameters as follows: learning rate 0.001, momentum 0.9, and weight decay 10^{-4}. Following the best practices in the literature, and the results of some further experiments, we apply standard pre-processing during the training procedure: we resize the images to 256×256 pixels, we perform center crop of size 224×224 and we normalize by mean and standard deviation of the ILSVRC2012 dataset.

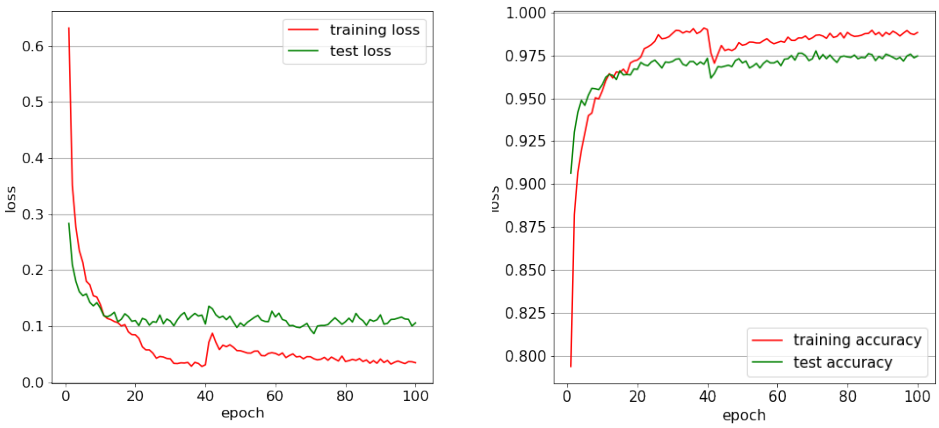


Figure 3.6: Training and test statistics for fine-tuning on the SEM dataset. Left: training and test loss; right: training and test accuracy.

We train the model for 100 epochs on 80% of the images in the SEM_dataset, and validate our results on the remaining 20%. As can be observed in Figure 3.6, the model saturates at an accuracy of 97.5% at epoch 70. The result only slightly improves the state-of-the-art performance of the DNet-121 architecture (De Nobili, 2017; Aversa et al., 2020) on the SEM dataset, reaching 97.3% accuracy, but it largely reduces the time-to-solution reaching convergence after only 70 epochs.

As expected, the Pearson correlation coefficient, reported in Figure 3.7, between the Euclidean distance of representations extracted by fine-tuning and the discrete distance matrix increases significantly. Notably, the correlation reached by the features extracted by the fine-tuned ResNet-50 is even higher than the one measured for InceptionV3 features after dimensional reduction (Coronica, 2018, Figure 3.9).

3.2.2 Representations from triplet-loss function

Let us consider a labelled dataset $\{x_i, y_i\}$ with data-points $x_i \in \mathbb{R}^D$. The *triplet loss function* has been introduced by Weinberger et al. (2006) to learn a linear transformation $L: \mathbb{R}^D \longrightarrow \mathbb{R}^D$ that ensure large margin separability among data-points belonging to different classes, while preserving the structure within each class. The modern approach introduced by Schroff et al. (2015) builds upon this idea to train a convolutional neural network to learn a

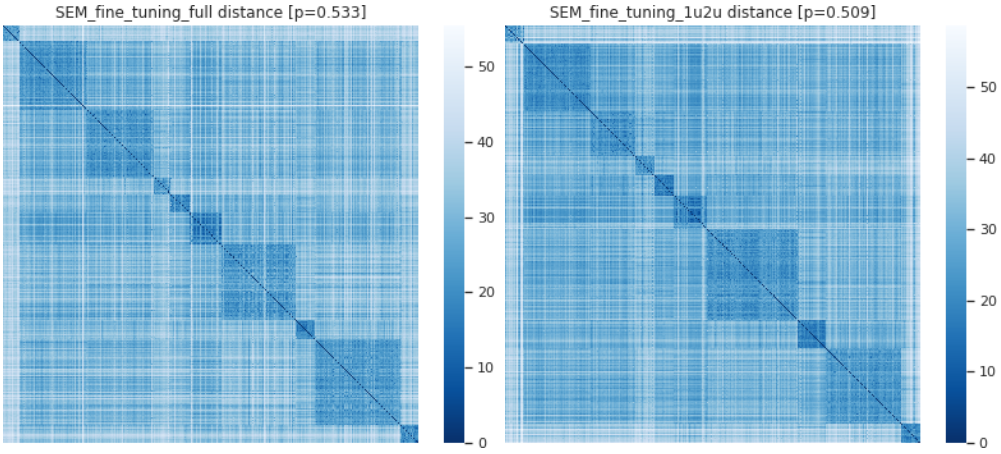


Figure 3.7: Heatmap Euclidean distances of of SEM_full (left) and SEM_1u2u (right) representations by fine tuning. In the title Pearson correlation index with d_{disc}

non-linear embedding of $\{x_i\}$ satisfying analogous separability assumptions. In local citation the authors show that this technique gives outstanding results when used in combination with a clustering algorithm on the resulting representations.

We adapt the construction of Schroff et al. (2015) to the case of the SEM dataset with NFFA labels, and where the convolutional neural network is the ResNet-50 architecture. Removing the classification head from the ResNet model and initialising from Imagenet pre-trained weights, we start from an embedding map

$$f_\theta \colon \mathbb{R}^D \longrightarrow \mathbb{R}^{2048}, D = 1024 \times 768.$$

Given an ordered collection of three data-points $t = (x, x_p, x_n)$ in the SEM dataset, we say that t is a *triplet* if x and x_p belong to the same NFFA class, and x and x_n belong to different NFFA classes. In order for our embedding to enforce class separation with a threshold $\alpha > 0$, we aim at finding parameters $\overline{\theta}$ such that for any triplet the following inequality on the distances of the corresponding representations is satisfied:

$$\|f_{\overline{\theta}}(x) - f_{\overline{\theta}}(x_p)\|_2^2 + \alpha \leq \|f_{\overline{\theta}}(x) - f_{\overline{\theta}}(x_n)\|_2^2 ,$$

where $\|\_\|_2$ denotes the Euclidean L^2 norm. The naive idea would be then to train f_θ with SGD minimising the loss

$$\sum \max \left(\|f_\theta(x) - f_\theta(x_p)\|_2^2 + \alpha - \|f_\theta(x) - f_\theta(x_n)\|_2^2 , 0 \right) ,$$

where the sum is taken over all the possible triplets. The naive implementation is computationally unfeasible since most of the triplets automatically satisfy the condition slowing the training procedure, and since SGD requires to work on batches. Following Schroff et al. (2015), the contribution to the loss function of a batch B, denoted in the literature as *hard batch triplet loss*, is defined considering the triplets that are difficult to separate:

- for each $x \in B$ select the element $x_p \in B$ maximising $\|f_\theta(x) - f_\theta(x_p)\|_2^2$;
- for each $x \in B$ select the element $x_n \in B$ minimising $\|f_\theta(x) - f_\theta(x_n)\|_2^2$;
- average the values

$$\max\left(\|f_\theta(x) - f_\theta(x_p)\|_2^2 + \alpha - \|f_\theta(x) - f_\theta(x_n)\|_2^2, 0\right)$$

  over the number of elements in B giving strictly positive contribution.

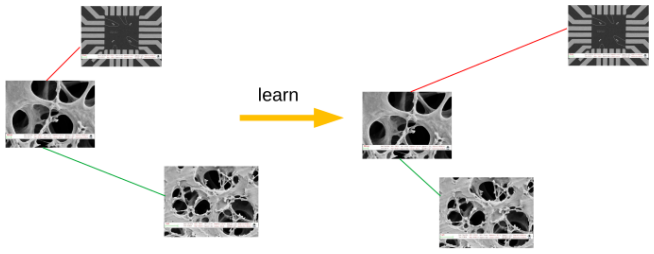


Figure 3.8: Triplet loss enforces minimisation of distances of images in the same class, and maximisation distance of images in different classes.

We train our model for 90 epochs with SGD minimising hard batch triplet loss with margin $\alpha = 1$. The training hyper-parameters are set as follows: learning rate 0.001, weight dacay 10^{-5}, and momentum 0.9. To favour generalisation, we perform substantial data augmentation: we randomly modify brightness and contrast by a factor 0.1, we randomly perform horizontal flip, we randomly apply rotation of maximal angle 10°, and we randomly crop and resize the image to size 224×224. The validation statistics are reported in Figure 3.9.

We use the learned weights $\overline{\theta}$ to extract representations of the SEM images by means of the embedding map $f_{\overline{\theta}}$. Given the predisposition of the algorithm to separate images in different NFFA categories, it is not surprising that we reach very high Pearson correlation index between the matrix of Euclidean distances of the SEM_full (resp. SEM_1u2u) representations by triplet loss

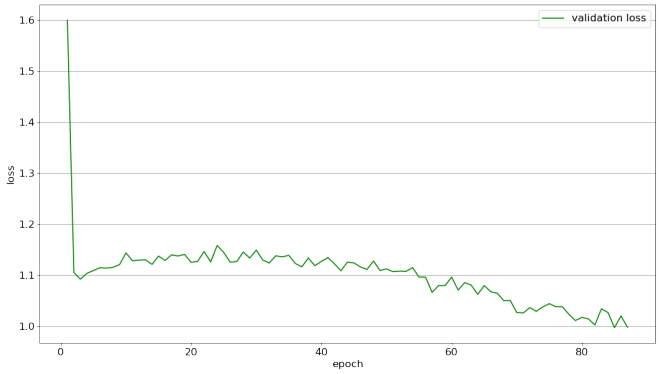


Figure 3.9: Validation curve for hard batch triplet loss statistics.

and the discrete distance, as shown in Figure 3.10. Nonetheless, it is worth noticing that representations of images in smaller classes, such as Porous Sponges, are at a very small distance reflecting the small within class variability, while for majority classes, such as MEMS Devices and Electrodes, at least some of the internal structure is preserved. This is in agreement with the observation that representations learned via the triplet loss function embed images in the same class on a manifold reflecting the semantic content (Schroff et al., 2015).

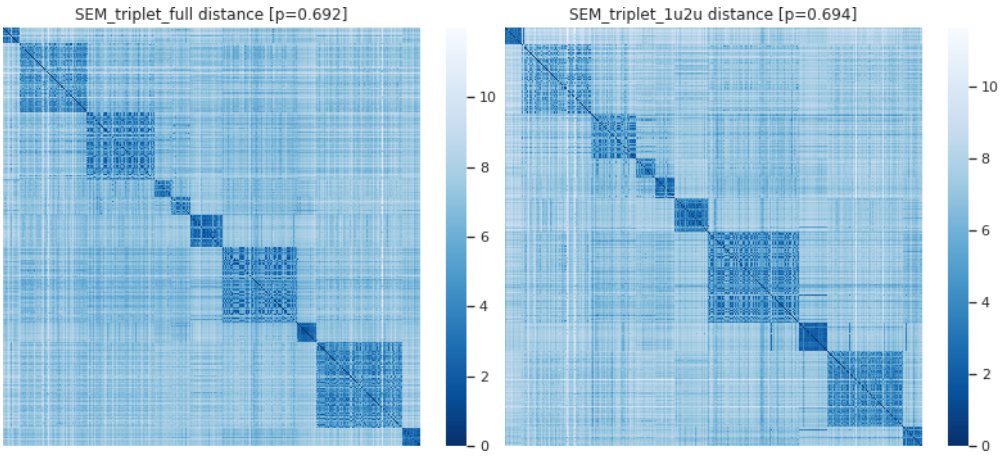


Figure 3.10: Heatmap Euclidean distances of SEM_full (left) and SEM_1u2u (right) representations by triplet loss function. p measures Pearson correlation coefficient with d_{disc}.

3.2.3 Representations from deep clustering

Deep clustering has been recently proposed as an end-to-end unsupervised learning technique for computer vision (Caron et al., 2018). The algorithm has been originally designed to learn simultaneously weights of a convolutional neural network and clustering on the extracted features by k-means clustering. Using deep cluster one can extract representations of a dataset whose neighboring properties reflect the content similarity of the images, as proven by the competitive results obtained on image retrieval tasks (Caron et al., 2018, Section 5.3). We will focus on a two-step approach: we use the deep clustering algorithm for learning representations of the SEM datasets, and we perform Advanced Density Peaks on the resulting features 4.2.3. From a pragmatic point of view, this relieves us from deciding a priori the number of clusters in which partitioning the SEM dataset, and it allows to define a hierarchical subdivision that is not naturally induced by k-means clustering. From a more theoretical perspective, recent observations show that representations of image datasets in the hidden layers of CNN lie in the proximity of highly curved manifolds with possibly complicated topology (Ansuini et al., 2019, Section 3.3), so that density based clustering algorithms are expected to outperform classical clustering techniques.

We describe in more detail a training cycle of our adaptation of the deep clustering procedure to the SEM_full dataset X. Consistently with the previous Sections, we consider a ResNet model f_θ with removed classification head, and with randomly initialised weights. Given a fixed hyper-parameter k we perform k-means clustering on $f_\theta(X) \subset \mathbb{R}^{2048}$, or more precisely on its dimensional reduction to $d = 256$ dimensions obtained by means of PCA. Thus, we learn a $d \times k$ dimensional matrix C whose rows are cluster centroids, and a cluster assignment $y_i \in \{0,1\}^k$ for each data-point, that solve the minimisation problem

$$\min_{C \in \mathbb{R}^{d \times k}} \frac{1}{|X|} \sum_{i=1}^{|X|} \|f_\theta(x_i) - C y_i\|^2 \text{ , s.t. } y_i^T \mathbb{1}_k = 1 \text{ for all } i.$$

We attach a fully connected layer $g_W \colon \mathbb{R}^{2048} \longrightarrow \mathbb{R}^k$ to the output of the ResNet. Using the cluster assignment $Y = \{y_i\}$ as set of (pseudo-)labels we train the model for one epoch using SGD on the optimisation problem

$$\min_{\theta, W} \frac{1}{N} l(g_W(f_\theta(x_i)) - y_i),$$

where l is the cross-entropy loss function. Once updated the weights θ, the

training cycle just described is repeated for the desired number of epochs.

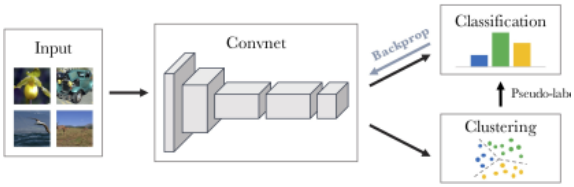


Figure 3.11: Deep clustering training cycle in an image (Caron et al., 2018)

At a first sight it might seem that this method could not possibly learn meaningful features as the fist label assignation is based on mutual distances of the representations extracted with random weights. Notice, though, that the feature extraction is performed using a CNN that imposes a very strong prior on images, as testified by the fact that randomly initialised weights achieve more 10% accuracy on the ImageNet classification task for which we expect a 0.1 accuracy at chance level. Deep clustering training builds upon this weak signal to construct semantically meaningful representations.

We discuss now some more technical details of the training loop implementation. In order to choose the number of clusters k, we run deep clustering as an end-to-end method on the labelled SEM_dataset and compute NMI w.r.t. the NFFA category. We set $k = 300$ in light of the results reported in Table 3.4, observing that considering k larger than the expected number of clusters can be beneficial, as observed by Caron et al. (2018). In order to make training computationally more efficient both PCA reduction and k-means clustering assignment are performed on the GPU by means of the FAISS library (Johnson et al., 2017) for fast similarity search. Furthermore, in order to avoid trivial solutions caused by a possible disparity in the size of the classes, we rescale the loss function l by weighting the contribution of each element by the inverse of the size of its corresponding cluster.
We train the ResNet-50 architecture by deep clustering setting $k = 300$ on the SEM_full dataset for 150 epochs, choosing batch size 32 for the self-supervised part of the training loop. Following Caron et al. (2018), after the usual pre-processing consisting in a resize and crop of the images, we apply Sobel filtering to remove colors and enhance local contrast. After some preliminary exploration of the hyper-parameters, the SGD of the self-supervised part of the training loop is performed with learning rate 0.001, weight decay 10^{-4}, and momentum 0.9. Together with training loss, we monitor during training the NMI between the clusters produced by the algorithm at each step and the

Table 3.4: Experiment results, k-selection.

k	NMI
30	0.18
100	0.311
300	0.345
1000	0.317

partition given by the NFFA category, restricting both only to the labelled part of the dataset (Figure 3.12).

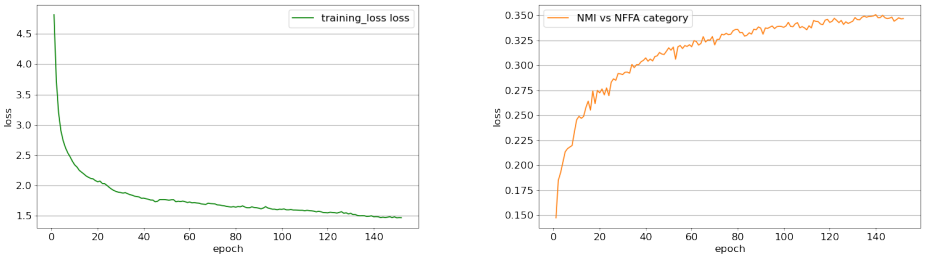


Figure 3.12: Deep clustering loss function during training (left), and NMI of deep clustering and NFFA category on SEM_dataset.

Using the weights learned by the deep clustering algorithm we extract representations $f_\theta(X)$ of the SEM_full (resp. SEM_1u2u) dataset. The heatmaps in Figure 3.13 constructed from the Euclidean distances, and the corresponding Pearson correlation coefficients with respect to d_{disc}, show that deep clustering is able to learn at least some of the semantic features of the dataset in a fully unsupervised manner. It is also worth noticing that information on the NFFA minority classes, which present a lower semantic variability, is reflected by the neighboring properties of the representations, while representations of majority classes are characterised by a larger dispersion.

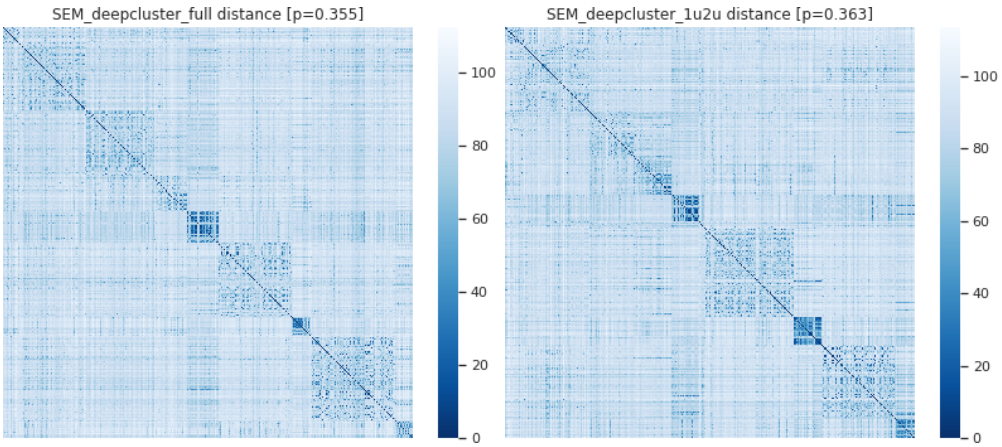


Figure 3.13: Heatmap Euclidean distances of SEM_full and SEM_1u2u representations from deep clustering. p measures Pearson correlation coefficient with d_{disc}.

Chapter 4

Hierarchical clustering in high dimensions

In Chapter 3 we studied representation of the SEM_full dataset by means of supervised and unsupervised learning techniques. Thus, to each vector in $\mathbb{R}^D$ corresponding to a SEM image has been associated the synthetic information represented by a vector in $\mathbb{R}^{2048}$, whose coordinates describe the features that are relevant for detecting image content.

In particular, the transformation defined by a forward-pass through the ResNet-50 architecture reduces the dimensionality of the embedding space by two orders of magnitude. Nonetheless, the manifolds where the datasets approximately lie have been generally observed to be of significantly lower dimension, and highly curved.

Motivated by this line of thought, a thorough study of the intrinsic dimension of the representations of the SEM datasets is presented in Section 4.1.2. Since representations of the SEM dataset do not lie on a linear subspace of smaller dimension, the computation is performed by means of the 2-NN algorithm that we describe in Section 4.1.1.

Both embeddings by supervised and unsupervised methods of the SEM_full dataset lie in the proximity of manifolds of dimension two order of magnitude less than the one of the embedding space $\mathbb{R}^{2048}$. Employing a recently developed version of the advanced density peaks algorithm, it is possible to study the peaks of density and saddles working directly on the manifold of data, thus avoiding any loss of information caused by further projections.

After discussing the Advanced Density Peaks algorithm in Section 4.2.1, and

introducing the necessary tools for the evaluation of our clustering procedure in Section 4.2.2, we discuss the final results of the pipeline consisting in a hierarchical clustering of the SEM datasets.

4.1 The dimension of SEM-representations

4.1.1 Intrinsic dimension

Representations of image datasets obtained from the inner layers of CNN lie in very high dimensional vector spaces. Without any further assumption, it would be extremely complicated drawing any conclusion on the probability distribution of datapoints' representations, essentially due to the curse of dimensionality. It has been widely appreciated, though, that this models tend to be largely overparameterized both in terms of weights and activation neurons. This led to the belief, motivated by empirical observations (Ansuini et al., 2019), that representations of a structured dataset, such as a dataset of images, in the inner layers of CNN lie in the proximity of a manifold of substantially smaller dimension than the ambient space, denoted in the literature by *intrinsic dimension* (ID). Even from an intuitive perspective it is

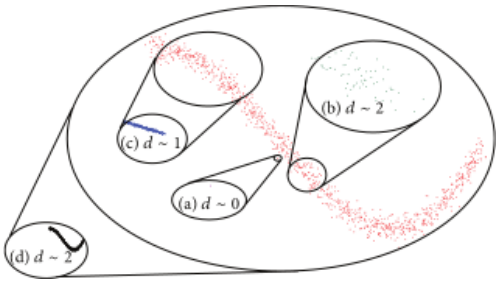


Figure 4.1: The scale problem: looking at the same dataset at different can lead to different ID estimations.

complicated to determine the dimension of a finite collection of points, since for instance considering a point cloud from different perspectives might lead to contradicting conclusions, as described in Figure 4.1. Even so more it is non-trivial to define such a concept from the theoretical perspective, as any point-cloud can be interpolated by a curved enough smooth manifold, and only recent developments discuss a (non-computationally feasible) algorithm to establishing the existence of a suitably defined approximating manifold (Fefferman et al., 2013). Nonetheless, in the spirit of the *manifold hypotesis*, we will assume that our representations lie, possibly up to noise, in the

neighborood of a subvariety $M \subset \mathbb{R}^D$, and we will focus on the estimation of $dim(M)$.

In principle, the large difference between the dimension D of the embedding space and the dimension $dim(M)$ could be due to M lying inside a linear sub-space $\mathbb{R}^k \subset \mathbb{R}^D$ with $k \ll D$. The absence in Figure 4.2 of a clear gap in the eigenvalues spectrum when performing linear dimensional reduction (PCA) on the representations of the SEM_1u2u dataset, analogous to recent observations in Ansuini et al. (2019, Section 3.3) for the ImageNet dataset, show that this is not the case.

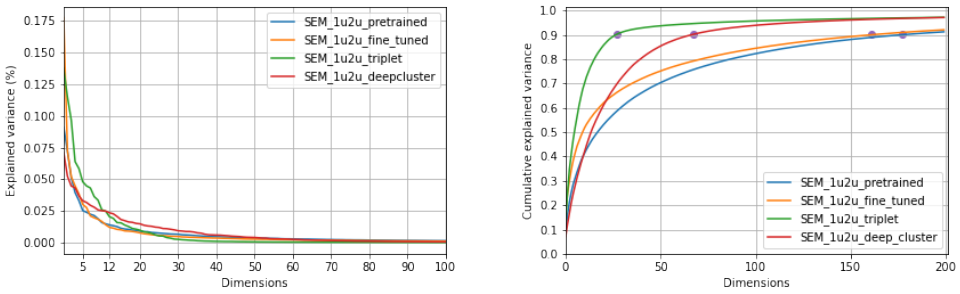


Figure 4.2: Singular values of the covariance matrix for SEM_1u2u representations do not present a gap (left). Number of singular values explaining 90% of the variance overestimate the dimension (right).

For this reason we will compute the ID of SEM representations using the Two Nearest Neighbor (2-NN) estimator developed by Facco et al. (2017). Relying only on the distances r_1 and r_2 of the elements in the dataset from their first and second nearest neighbor, it guarantees reliable results also when the data-cloud lies on a manifold with complex topology, and with possibly varying curvature and density.

To fix notation consider a dataset $X \subset \mathbb{R}^D$ and denote by μ the distribution of the ratio r_2/r_1. Under the mild assumption that the density varies smoothly at the level of the distances to the second neighbors, the authors prove that the ID of X can be computed as a quotient of an explicit expression in the cumulative distribution $F(\mu)$ and the logarithm of μ. Thus, replacing $F(\mu)$ with its empirical distribution, the ID of X can be estimated by a linear fit:

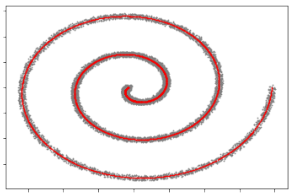
1. $\forall x_i \in X$ find distances $r_1(x_i)$ and $r_2(x_i)$ from first and second neighbor;

2. $\forall x_i \in X$ compute $\mu(x_i) = \frac{r_2(x_i)}{r_1(x_i)}$;

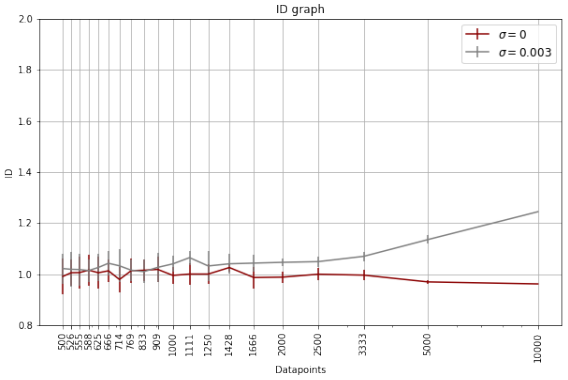
3. find permutation σ indices $\{1, \ldots, N\}$ so that $\mu(x_{\sigma(i)})$ is sorted in ascending order, so that $F^{emp}(\mu_\sigma(i)) = \frac{i}{N}$;

4. the ID is the coefficient of the linear fit passing through the origin of $\{(\log(\mu(x_i), -\log(1 - F^{emp}(x_i)))\} \subset \mathbb{R}^2$.

The linear regression estimating the ID might be heavily influenced by the presence of outliers, for which $r_1 \ll r_2$. The authors suggest discarding the 10% of the points with higher μ during the linear fit. Furthermore, in order to address the scale problem described in Figures 4.1 and 4.3 the authors suggest to compute the ID for different sub-samples of the dataset: successive decimation of the dataset progressively decreases the number of data-points increasing in turn the average distance r_2 from the second neighbor and thus increasing the scale. An eventual plateau in the ID graph obtained from the block analysis provides a reliable estimation of the intrinsic dimension.



(a)



(b)

Figure 4.3: (a) A sampling of 10^4 points from a spiral (red), and its scatter around the original value by $\sigma = 0.003$. (b) The corresponding ID graphs.

4.1.2 Intrinsic dimension of SEM-representations

As discussed in Section 4.1.1, we expect that representations of the SEM datasets lie on a curved sub-manifold of $\mathbb{R}^{2048}$ of substantially smaller dimension. Notice, though, that the extracted representations model the probability distribution of our data only up to a certain level of noise. For this reason, as described in Figure 4.3, in order to obtain a reliable estimation

of the intrinsic dimension via the 2-NN algorithm, it is crucial to perform a thorough block analysis looking for a plateau in the ID graph.

More specifically, given a set of representations $X \in \mathbb{R}^{2048}$ with $|X| = N$, for each $k \in \{1, \dots 20\}$ we perform a random partitioning of X in k disjoint parts $\{X_{k_j}\}_{j=1}^{20}$, and define the ID at "scale $1/k$" to be the average of the results of the 2-NN algorithm on the X_{k_j}'s. As suggested in Facco et al. (2017), and as discussed in Section 4.1.1, the 10% of datapoints in X_{k_j} maximising the ratio r_2/r_1 is discarded during the estimation.

We adopt the following strategy to perform the block analysis. After computing the distance matrix d_{ij} for the whole dataset X, we estimate the ID of each X_{k_j} as follows: we select an appropriate sub-matrix $dist_{k_j}$, we find distances r_1 and r_2 of each point from its two nearest neighbors by sorting $dist_{k_j}$ along a direction, and apply (2-4) in Section 4.1.1. Given the necessity of storing the distance matrix, the algorithm requires large RAM availability, and for this reason it could be sub-optimal when run in a restricted memory setting. The choice of this technique is justified by the fact that we run our experiments at the Orfeo cluster facility at AREA Science Park where the infrastructure has been designed also with focus on applications requiring large amounts of memory. All the experiments have been performed on a *thin node* of the cluster which has 800GB RAM availability.

We report in Figure 4.4 the graphs describing the block analysis for the representations of the SEM_dataset (resp. the SEM_1u2u datset, and the SEM_full datset) obtained by forward-pass through the ResNet-50 model with weights learned as in Section 3.2. In Table 4.1 we report the final value emerging after the analysis of the plateaux, approximating by excess in case of uncertainty. Even if the ID graphs do not always exhibit a clear plateau, it is evident that the measured ID is significantly lower than the estimation from linear methods described in Figure 4.2, further confirming that the extracted representations lie on curved submanifolds of the embedding space.

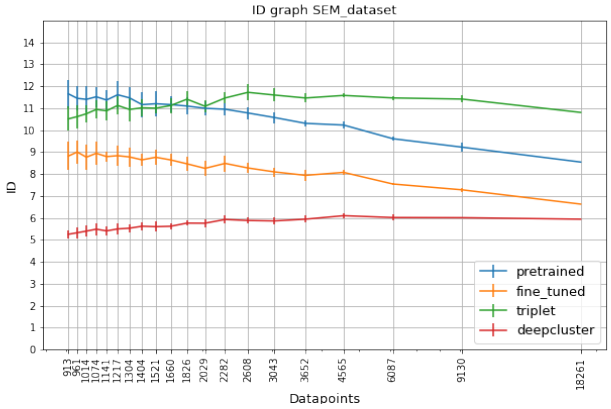
Let us discuss qualitatively the results of Figure 4.4 starting from the representations obtained by pure transfer learning (blue) and fine-tuning (blue). The ID graph obtained for the SEM_dataset exhibits a rather well defined plateau in both cases. On the contrary, in the case of representations obtained by pure transfer learning, the results emerged for the representations of the SEM_1u2u and SEM_full datasets report a quite steady decrease. It is not simple to determine the root cause of this behaviour, but we suspect it could be caused by representations lying on a manifold only up to a rather

Table 4.1: Estimated ID for varying extraction type and datset.

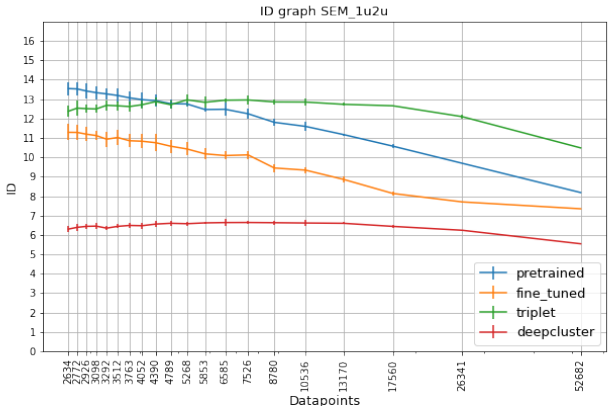
	SEM_dataset	**SEM_1u2u**	**SEM_full**
Pretrained	11	13	15
Fine-tuned	8	10	12
Triplet	12	13	15
Deepcluster	6	7	8

high level of noise. In order to avoid loss of information, we consider early plateaux as a reliable upper bound for the estimation of the ID. The curve of the estimated ID of fine-tuning representations (yellow) shows a less steep decrease when increasing the number of datapoints, and rather well defined plateaus. The lower value of the estimated ID in the case of fine-tuning representations is in line with observations in Ansuini et al. (2019), where the authors sustain that compression of information corresponding to a lower ID value corresponds to higher performances of the model.

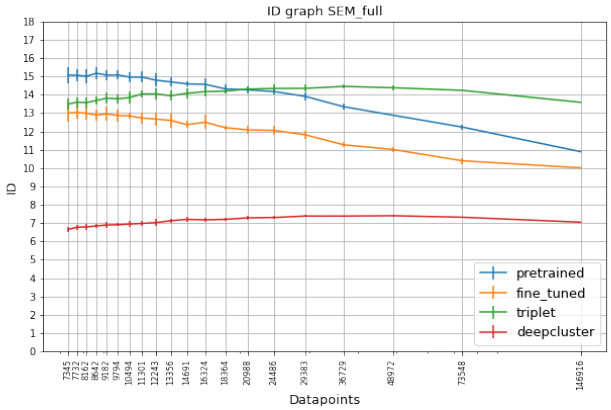
Both the ID graphs of representations obtained from triplet loss function (green) and deepcluster (red) show well defined plateaux for all datasets, but the two manifest an opposite behaviour. In the case of triplet representations the estimated ID is high, similar to the one of representations by pure transfer learning, while in the case of deepcluster representations the measured ID is even lower than the one measured for fine-tuning representations. In particular, the observed ID values for the deepcluster representations might be a warning signal that not all the distinctive features of the SEM datsets have been detected by training the weights with this procedure. Further more systematic investigation on the behaviour of the ID in the layers of CNN trained with strategies different from classification would be needed in order to draw more precise conclusions.



(a) ID graph SEM_dataset



(b) ID graph SEM_1u2u



(c) ID graph SEM_full

Figure 4.4: ID graphs of the SEM datasets.

4.2 Hierarchical clustering of SEM images

4.2.1 Advanced Density Peaks clustering

The density peaks (DP) algorithm (Rodriguez & Laio, 2014) is a (density-based) clustering algorithm founded on the following idea: the centers of the clusters are local maxima of the density function, and it is likely that centers are separated by a relatively high distance from elements of higher density. One of its strengths, shared by other density-based algorithms, is the ability to detect clusters underlying both convex and non-convex geometries. Furthermore, it does not require a priori knowledge of the number of clusters, and it depends only on the relative distance of the datapoints.

Given a dataset $X = \{x_i\}$ and the distance matrix $d_{ij} = dist(x_i, x_j)$ between its elements, the algorithm can be summarised as follows:

- estimate the local density ρ_i at each point x_i;
- for each x_i find the distance d_i from the closest point with density greater than ρ_i;
- the cluster centers $\{c_j\}_{j=1}^{K}$ are points x_i with high density ρ_i and high d_i, i.e. they are outliers in the diagram (ρ_i, d_i) (see Figure 4.5);
- for j in $\{1, \dots, K\}$, assign label j to the cluster center c_j. Each other point x_i is assigned the same label of its closest point of higher density.

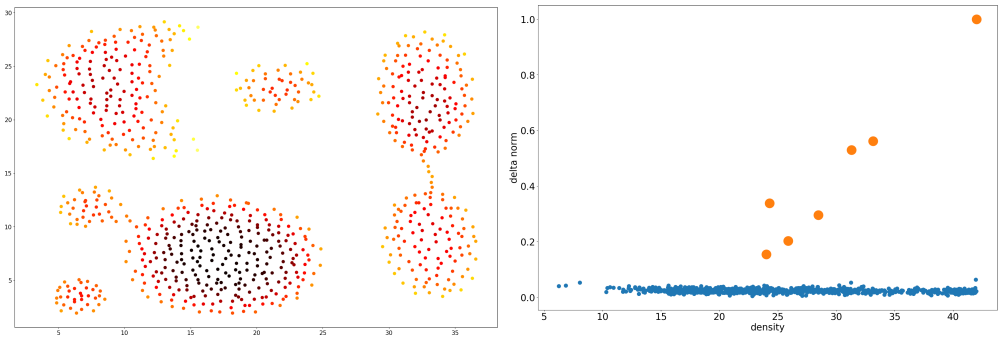


Figure 4.5: *Left*: heatmap of local densities, points color coded so that darker points have higher density. *Right*: density peaks diagram of dataset in left figure, orange circles correspond to cluster centers, and the dots to remaining points of the dataset.

In d'Errico et al. (2018) the authors propose an improved variant of the DP algorithm, which we will denote by advanced density peaks (ADP), to address some crucial aspects of the procedure among which:

- obtain a reliable estimation of the density even when the dataset X lies in a high dimensional vector space $\mathbb{R}^D$;
- define a selection method for the cluster centers not relying on the visual inspection of the density-distance graph, and only subject to hyper-parameters with a direct statistical interpretation;
- density peaks should not define distinct cluster centers when they are connected by a path of points of almost uniform density, even when lying far apart;
- describe an organisation of the density peaks in a hierarchical structure.

The general strategy of the algorithm consists in finding not only the peaks of density but also the saddle points connecting them. A remarkable by-product of the procedure is a two-dimensional reconstruction of the *topography* of the the cluster centers and their mutual relation described by the saddles. In the remaining part of the Section we describe the ADP algorithm in some detail.

First of all, one needs to estimate the local density at each point x_i of the dataset X. Let d be the intrinsic dimension of X, assume that the density is approximately constant for the first k neighbors of x_i, and denote by $r_{i,k}$ the distance of x_i from its k-th neighbor. Using a maximum likelihood argument one can estimate the density as

$$\rho_i = \frac{k}{V_{i,k}}, \quad V_{i,k} = \omega_d \cdot r_{i,k}^d,$$

where ω_d is the volume of a d dimensional sphere of radius 1. It is clear that the choice of the number of neighbors k for performing the estimation could be crucial for obtaining a reliable performance. In Rodriguez et al. (2018), the authors develop a sophisticated method, denoted as Point Adaptive k nearest neighbor (PAk), to estimate for each x_i the maximum number of neighbors k_i for which the density can be considered constant. Nonetheless, recent results on the representations extracted from the ImageNet (Doimo et al., 2020) dataset show that even choosing a constant (relatively small) k the clustering procedure is effective. For both strategies of k selection, it is also possible to compute the error ϵ_i on the estimation of $\log(\rho_i)$ by measuring the variance of the likelihood.

Once estimated the local densities, a preliminary search of the peaks of density is performed finding the set $C_{prel} = \{c_1, \dots, c_K\}$ of local maxima of

$$g_i = \log(\rho_i) + \epsilon_i.$$

Denoting by $N_{k_i}(i)$ the indices of the k_i nearest neighbors of a point x_i, this is realised in two steps:

- find the set C_{prel} of x_i's for which $g_i > g_j$ for all $j \in N_{k_i}(i)$;
- remove x_i from C_{prel} if $i \in N_{k_j}(j)$ and $g_j > g_i$.

A preliminary label assignation is performed considering the elements in C_{prel} as cluster centers and proceeding as in the original DP algorithm, thus obtaining a subdivision of X in K disjoint subsets C_j.

The statistical reliability of the clusters is assessed by studying the saddle points between probability peaks. A point $x_i \in C_\alpha$ belongs to the boundary $\partial_{\alpha,\beta}$ between cluster C_α and cluster C_β if there is $x_j \in N_{k_i}(x_i) \cap C_\beta$ such that

$$dist(x_j, x_i) = \min_{x_l \in C_\alpha} (dist(x_j, x_l)).$$

A point x in the boundary $\partial_{\alpha,\beta}$ is a saddle between C_α and C_β if it is the point maximising $\log(\rho) + \epsilon$ on $\partial_{\alpha,\beta}$. We will denote by $\rho_{\alpha,\beta}$ its density, and by $\epsilon_{\alpha,\beta}$ the corresponding error.

The preliminary clustering C_{prlim} is coarsened by removing clusters corresponding to probability peaks that are not statistically significant. Cluster C_α, with center c_α of density ρ_α, is merged with cluster C_β if they are connected by a saddle such that

$$\log(\rho_\alpha) - \log(\rho_{\alpha,\beta}) < Z \cdot (\epsilon_\alpha + \epsilon_{\alpha,\beta}).$$

The crucial hyperparamter Z is responsible both for the level of sensitivity of the algorithm on density variation and for the statistical reliability of the clusters, as extensively discussed in d'Errico et al. (2018, Section II.6). The labelling $C_{ADP} = \{C_\alpha\}$ obtained after merging clusters in C_{prelim} is the result of the ADP algorithm.

The structure of the peaks and saddles of the density induces a natural hierarchical structure on the clusters $\{C_\alpha\}$. Two peaks are considered near if they are connected by a saddle of relatively high density, so that one is led to consider the distance

$$d_{clust}(C_\alpha, C_\beta) = \min(\log(\rho_\alpha), \log(\rho_\beta)) - \log(\rho_{\alpha,\beta}). \tag{4.1}$$

The dendogram describing the hierarchical structure of the clusters can be obtained by applying the single-linkage agglomerative clustering on the resulting distance matrix.

We implement the algorithm defining a Python class *ADP* with methods performing the various required steps. The most demanding operation consists in nearest neighbors lookup, specially since the PAk density estimation method requires finding in advance a large number of NN. The NN search is performed using the Faiss library (Johnson et al., 2017), implementing an exact NN search fully exploiting intra-node parallelism. The class ADP allows to optionally select between density estimation with fixed number of NN k or PAk. The PAk version calls a Fortran routine implementing the Newton-Raphson method required for the maximim likelihood estimation, that was generously shared by A. Rodriguez with the author in private correspondence. The computationally intensive part of the clustering method is delegated to a Cython function performing preliminary clustering and merging of the clusters, which was developed by the authors of (Doimo et al., 2020). The class contains method for automatic qualitative and quantitative evaluations that will be discussed in Sections 4.2.2 and 4.2.3.

4.2.2 Measure of cluster significance: NMI, AMI, ARI

The combination of learning representations of a SEM dataset X by one of the methods in Section 3.2 and clustering by ADP produces a partition $\mathcal{E} = \{E_1, \dots, E_L\}$ of X into disjoint subsets. Let $Y \subset X$ be the part of the dataset that has been humanly labelled into the NFFA categories. The restrictions $E_i \cap Y$ define a partition $\mathcal{C} = \{C_1, \dots, C_K\}$ of the labelled dataset. On the other hand, the NFFA labels define a partition $\mathcal{D} = \{D_1, \dots, D_{10}\}$ of Y just by considering class subdivision. In order to evaluate the results of our pipeline we will compare $\mathcal{C}$ and $\mathcal{D}$: a good alignment of the final clustering with the NFFA subdivision indicates that our procedure is sensitive to the content of the SEM images, thus refining and extending the subdivision of the dataset obtained by human effort.

Since the number of clusters $|\mathcal{C}|$ is chosen optimally by the ADP algorithm, our evaluation should be as less subject as possible to the number of subsets of the two partitions. Furthermore, since $\mathcal{C}$ consists of a partition and not a labelling of the dataset Y, we should consider evaluation metrics which are invariant under permutations of the subsets defining the partition. In order to obtain a robust comparison we will consider three metrics constructed for these purposes, and introduced in the rest of the Section: normalized mutual

information (NMI), adjusted mutual information (AMI), and adjusted Rand index (ARI).

Given a partition $\mathcal{C} = \{C_1, \ldots, C_K\}$ of a set Y, the information of an element y belonging to a cluster C_i is measured by the Shannon entropy of the random variable

$$\mathcal{C}\colon Y \longrightarrow \{1, \ldots, K\}, \;\; \mathcal{C}\colon y \longmapsto i \text{ if } y \in C_i,$$

defined as

$$H(\mathcal{C}) = -\sum_{i=1}^{K} P(C_i)\log(P(C_i)), \;\; P(C_i) := \frac{|C_i|}{|X|}.$$

Given a second partition $\mathcal{D} = \{D_1, \ldots, D_N\}$ of Y, the information $H(\mathcal{C}, \mathcal{D})$ of the common refinement of the two partitions $\mathcal{C}$ and $\mathcal{D}$ is given by the entropy of the joint probability distribution

$$(\mathcal{C}, \mathcal{D})\colon Y \longrightarrow \{1, \ldots, K\} \times \{1, \ldots, N\},$$
$$(\mathcal{C}, \mathcal{D})\colon y \longmapsto (i, j) \text{ if } y \in C_i \text{ and } y \in D_j.$$

On the other hand, the conditional entropy

$$H(\mathcal{C}|\mathcal{D}) = \sum_{i=1}^{K} P(C_i)\left[\sum_{j=1}^{N} P(C_i \cap D_j)\log(P(C_i \cap D_j))\right]$$

is the weighted average of the information of the refined partition induced from $\mathcal{D}$ on each cluster of $\mathcal{C}$. Mutual information is defined by the expression

$$MI(\mathcal{C}, \mathcal{D}) = H(\mathcal{C}, \mathcal{D}) - H(\mathcal{C}|\mathcal{D}) - H(\mathcal{D}|\mathcal{C}),$$

and simultaneously measure the mutual dependence of the two random variables defined by $\mathcal{C}$ and $\mathcal{D}$. In particular, if $\mathcal{C}$ and $\mathcal{D}$ are independent their mutual information is 0, and MI is maximal when $\mathcal{C}$ and $\mathcal{D}$ define the same partition of Y. Furthermore, MI is symmetric, and it is invariant by permutations of the sets in the partition $\mathcal{C}$ (resp. $\mathcal{D}$).

Increasing the number of clusters, the value of the MI can increase unboundedly. For this reason we standardise the MI measure by the average entropy of the partitions

$$NMI(\mathcal{C}, \mathcal{D}) = \frac{2MI(\mathcal{C}, \mathcal{D})}{H(\mathcal{C}) + H(\mathcal{D})}.$$

It can be readily checked that $NMI(\mathcal{C}, \mathcal{D}) \in [0, 1]$, where the extremes are realised respectively when the partitions $\mathcal{C}$ and $\mathcal{D}$ are statistically independent, and when they agree up to permutation.

Despite the normalisation, when the number of clusters increases the value of NMI is likely to increase independently from the mutual information content of the partitions. In order to address this problem, one can consider the adjusted mutual information

$$AMI(\mathcal{C}, \mathcal{D}) = 2\frac{MI(\mathcal{C}, \mathcal{D}) - \mathbb{E}(MI(\mathcal{C}, \mathcal{D}))}{H(\mathcal{C}) + H(\mathcal{D}) - \mathbb{E}(MI(\mathcal{C}, \mathcal{D})},$$

where the expected mutual information is computed over all the possible partitions of Y with the same shapes as $\mathcal{C}$ and $\mathcal{D}$.

Another classical measure for comparing clustering of a given dataset Y is the Rand index. Given two partitions $\mathcal{C}$ and $\mathcal{D}$, denote by $a(\mathcal{C}, \mathcal{D})$ the number of pairs lying in the same cluster both in partition $\mathcal{C}$ and $\mathcal{D}$, and denote by $n(\mathcal{C}, \mathcal{D})$ the number of pairs lying in the distinct cluster for both partitions. The Rand index

$$RI(\mathcal{C}, \mathcal{D}) = \frac{a(\mathcal{C}, \mathcal{D}) + n(\mathcal{C}, \mathcal{D})}{\binom{N}{2}}, \quad N = |Y|$$

normalises the count over the number of all possible pair of elements in Y. Also this measure is symmetric, and invariant by permutations of the sets in the partition $\mathcal{C}$ (resp. $\mathcal{D}$). In order to normalise the measure, and ensure that a pair of random partitions returns an approximately 0 score, one performs the following standardisation:

$$ARI(\mathcal{C}, \mathcal{D}) = \frac{RI(\mathcal{C}, \mathcal{D}) - \mathbb{E}(RI(\mathcal{C}, \mathcal{D}))}{\max(RI(\mathcal{C}, \mathcal{D})) - \mathbb{E}(RI(\mathcal{C}, \mathcal{D}))),}$$

where $\max(RI)$ denotes the maximum achievable RI score given the cluster configurations $\mathcal{C}$ and $\mathcal{D}$, and $\mathbb{E}(RI)$ is the expected RI for randomly extracted partitions given $\mathcal{C}$ and $\mathcal{D}$ configuration. It can be checked that $ARI(\mathcal{C}, \mathcal{D}) \in [-1, 1]$, where the value 1 is attained when $\mathcal{C} = \mathcal{D}$.

4.2.3 Clustering of the SEM datsets

The ADP clustering algorithm has been applied to the representations extracted in Section 3.2 on the SEM datsets, with intrinsic dimension selected according to Table 4.1. The results reported and discussed in the rest of the Section have been obtained from a grid-search on the density estimation technique, and on the value of the hyper-parameter Z. In particular, the ADP algorithm was performed after density estimation with the PAk and fixed k-NN density estimation methods, where we let k vary in $[10, 15, 30, 50, 100, 200]$. The parameter Z was chosen instead among the values $[1.0, 1.2, 1.4, 1.65, 2.0, 2.3]$. Given one of the SEM datsets, for each choice of technique for extracting representations, we select the hyper-parameters that resulted in the clustering with the highest NMI, AMI, and ARI scores against the NFFA classification on the labelled part of the dataset.

For each choice of dataset, once selected the representations that led to the highest score, we present the results of further analysis on the peaks and saddles detected by the ADP algorithm. From Equation 4.1 we can define a distance between the clusters, so that one can obtain a dendogram describing the hierarchical structure of the peaks of density by applying single linkage. Furthermore, using this distance one can obtain a $2D$-representations of the peaks of density by multidimensional-scaling, and describe proximity of the various clusters by an adjacency matrix.

Table 4.2: Scores of results of the ADP clustering on the SEM_dataset

	NN	**Z**	**NMI**	**AMI**	**ARI**
Pretrained	PAk	2.0	0.431	0.428	0.173
Fine-tuned	k=30	1.65	0.749	0.749	0.503
Triplet	k=50	1.65	0.740	0,740	0.504
Deepcluster	k=15	1.65	0.443	0.439	0.179

The scores for the SEM_datset are presented in Table 4.2. Both clustering obtained from representations extracted by fine-tuning and by triplet loss reach remarkable NMI scores, denoting that the clustering of the extracted features is extremely well aligned with the NFFA categories. Clustering of the representations extracted by deepcluster weights reach an NMI value

of 0.443. Even if this result might not seem impressive when compared to the NMI obtained by fine-tuning and triplet loss representations, one has to take in account that this result has been obtained without any use of the labels. In particular, the experiments show that the training of the ResNet-50 architecture on the SEM_full dataset with the deep clustering algorithm produced weights that are comparable, if not superior, to the representations obtained by "pure" transfer learning from the ImageNet dataset.

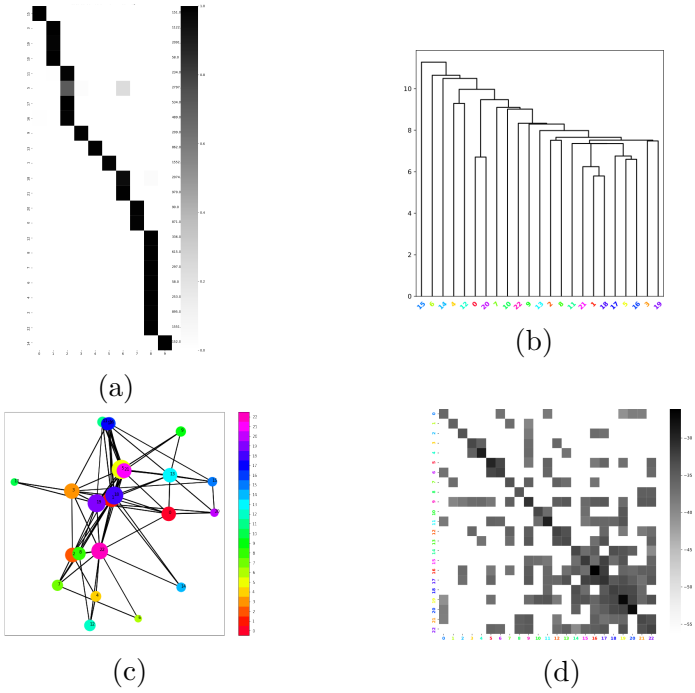


Figure 4.6: Further analysis of ADP clustering on SEM_dataset (representations from fine-tuning). (a) Heatmap of confusion matrix between NFFA category (row) and ADP clustering (column), where darker denotes more aligned. (b) Dendogram of the peaks of density. (c) $2D$-embedding of the density peaks, size of circles proportional to cluster population, width of connecting lines proportional to saddle height. (d) Heatmap of adjacency matrix of clusters with distance as in Equation 4.1.

The weights leading to the clusters achieving the best performance on the SEM_dataset are obtained by fine-tuning of the ResNet-50 architecture on the NFFA labels. In this case, we obtain a partition of the SEM_datset in 22 clusters which are almost perfectly aligned with the NFFA categories, see Figure 4.6 (a). The level of refinement is inferior than the one obtained by human labelling on the much smaller SEM_Hierarchical datsaset. At the price

of slightly lowering the level of alignment, a higher number of clusters can be obtained reducing the parameter Z. This result highlights the possibility of constructing a much larger hierarchical dataset of SEM images with a finer classification compared to the NFFA categories. For achieving this, it would be necessary to obtain only a labelling of the clusters by CNR-IOM scientist, and a further analysis of the hierarchical structure described in Figure 4.6 (b)-(d).

Table 4.3: Scores of results of the ADP clustering on the SEM_1u2u dataset

	NN	**Z**	**NMI**	**AMI**	**ARI**
Pretrained	k=10	2.0	0.423	0.420	0.181
Fine-tuned	k=30	1.65	0.568	0.566	0.335
Triplet	k=50	1.65	0.648	0.643	0.424
Deepcluster	k=30	1.2	0.427	0.425	0.187

The scores for the SEM_1u2u dataset are presented in Table 4.3. The results reflect the general properties observed on the SEM_datasets, except for a lowering in the results for representation extracted by the fine-tuning. The results obtained for representations from deepcluster confirm that it is possible to obtain a (partially) meaningful clustering without leveraging on any human effort. This shows the potential of self-supervised learning techniques as an instrument for reducing labelling time.

Training with triplet loss function defines the embedding resulting in the clusters with a higher NMI score. In this setting, our pipeline produces 34 clusters, essentially the same number of the classes of the SEM_hierarchical dataset. The observed value of NMI for these representations outperforms by a significant margin the ones obtained by other training techniques. The clusters whose labelled part is not well aligned with the NFFA classification (Figure 4.7) contain only a handful of images in the SEM_datatet. It can be observed that poorly aligned clusters correspond to images whose content is not represented in the 10 NFFA categories, so that the wrong grouping could just correspond to labelled images at the boundary of the newly emerged clusters. Also for this reason, further analysis of the clusters in collaboration with domain scientists will be required to indentify the clusters and interpret

the emerging hierarchy 4.7 (b). For the SEM_1u2u dataset the interpretation of the adjancency matrix and the $2D$-embedding becomes more subtle since more peaks are connected by relatively high saddles, and will require further investigation.

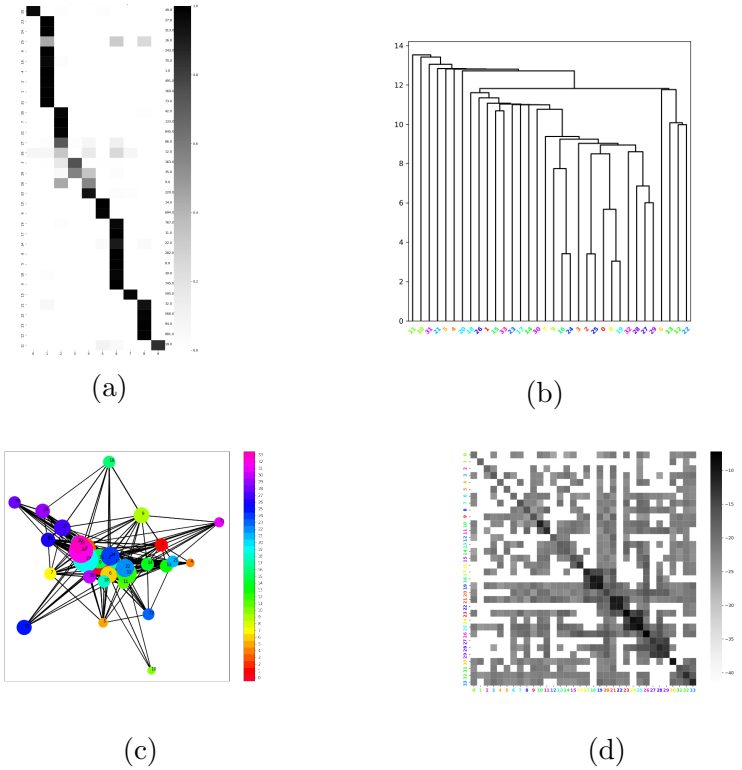


Figure 4.7: Further analysis of ADP clustering on SEM_1u2u (representations from triplet loss). Description of (a)-(d) as in Figure 4.6.

The scores obtained by the ADP algorithm on the representations of the SEM_full dataset are reported in Table 4.4.

Table 4.4: Scores of results of the ADP clustering on the SEM_full dataset

	NN	Z	NMI	AMI	ARI
Pretrained	PAk	1.65	0.418	0.416	0.156
Fine-tuned	k=30	1.65	0.595	0.595	0.323
Triplet	k=50	1.65	0.659	0.657	0.416
Deepcluster	k=50	2.0	0.422	0.420	0.173

In order to maximise the number of labels, the training of the ResNet-50 weights described in Section 3.2 has been performed on the labelled part of the SEM_full dataset, and on the whole dataset in the case of the deep clustering algorithm. Likely, this could be the root cause for the fact that the scores resulting from the clustering of the SEM_full dataset do not present significant differences w.r.t. the results obtained on the SEM_1u2u dataset. Even so more, it is possible to observe the emergence of some clusters forming automatically when considering the same type of object at different scale, and this is reflected in the proximity of the corresponding peaks. Notably, in the case of representations obtained from fine-tuning the results improve significantly when the scale is not considered. It is worth noticing that also for the SEM_full dataset the clusters obtained from representations from the triplet loss function outperform the others. Furthermore, the relatively high NMI measured when considering representations from deepcluster confirms the potential of unsupervised techniques also for detecting relevant features of a dataset containing images of objects taken at different scales. It remains an open problem to determine if it is beneficial restricting to images of objects at similar scale during the training procedure at the cost of sacrificing a significant amount of (pseudo-)labels.

Chapter 5

Conclusions

In this work we focused on the study of representations of scanning electron microscope images learned by means of deep convolutional networks, with the final intent of defining a hierarchical structure on the SEM datasets.

The pipeline we employ for achieving this object consists of three main components: train the weights of a ResNet-50 architecture to learn the semantic content of SEM images, extract representations to obtain an embedding of the dataset in $\mathbb{R}^{2048}$, and perform hierarchical clustering in high dimensions by means of the advanced density peaks algorithm. The result of the final clustering procedure are intimately related to the strategy employed for training, and thus can be exploited for serving different purposes.

As shown in Section 4.2.3, both of the supervised techniques we investigated, where training is performed by fine-tuning or by minimising the triplet loss function, lead to a partition that is extremely well aligned with the NFFA classification. As a first application this allows to detect a finer grain hierarchical subdivision of the labelled part of the dataset, thus enabling to create in automatic fashion a new dataset with similar characteristics to SEM_Hierarchical but containing an order of magnitude more samples. Furthermore, the remarkable NMI results obtained when clustering representations extracted by triplet loss both on the SEM_1u2u and SEM_full datasets, suggest another possible application in combination with the SEM classifier: whenever a newly collected image is predicted to belong with sufficiently high probability to one of the 10 NFFA categories, it could be added to the new hierarchical dataset simply by analysing the distance of the corresponding representation by triplet weights from the peaks of density.

Despite being less aligned with the human labelling of the NFFA classes,

the results obtained by the pipeline when considering representations from weights learned by the deep clustering algorithm should not be underestimated. As shown in Figure 3.13, this procedure is able to detect some of the relevant characteristics of the SEM images in a fully unsupervised manner. Furthermore, from visual inspection it can be observed that the clusters generically correspond to a coherent partition of the datasets by content. This strategy thus is the most promising both for unveiling the emergence of new classes, and for analysing other type of images collected within the NFFA framework where human labelling is scarce or non-existent.

No significant differences in the quality of the results have been observed when clustering only representations coming from images at the same scale. This is clearly related to our choices of training datasets, so that it could be worth exploring in furure work if restricting the whole pipeline to the scales of 1-2μm and 0.1-0.2μm can provide substantial benefit.

Bibliography

Ansuini, A., Laio, A., Macke, J. H., & Zoccolan, D. 2019, Intrinsic dimension of data representations in deep neural networks, in Advances in Neural Information Processing Systems, Vol. 32, 6111–6122

Aversa, R., Coronica, P., De Nobili, C., & Cozzini, S. 2020, Deep Learning, Feature Learning, and Clustering Analysis for SEM Image Classification, Data Intelligence, 2, 513

Aversa, R., Modarres, M., Cozzini, S., & al. 2018a, The first annotated set of scanning electron microscopy images for nanoscience, Scientific Data, 5

Aversa, R., Modarres, M. H., Cozzini, S., & Ciancio, R. 2018b, NFFA-EUROPE – Hierarchical SEM Dataset, https://b2share.eudat.eu/records/b9abc4a997f8452aa6de4f4b7335e582

Aversa, R., Modarres, M. H., Cozzini, S., & Ciancio, R. 2018c, NFFA-EUROPE – SEM Dataset, https://b2share.eudat.eu/records/19cc2afd23e34b92b36a1dfd0113a89f

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. 2018, Deep Clustering for Unsupervised Learning of Visual Features, in European Conference on Computer Vision

Coronica, P. 2018, Feature learning and clustering analysis for images classification, MHPC Thesis, International School for Advanced Studies (SISSA)

De Nobili, C. 2017, Deep Learning for NanoscienceScanning Electron MicroscopeImage Recognition, MHPC Thesis, International School for Advanced Studies (SISSA)

Deng, J., Dong, W., Socher, R., et al. 2009, Imagenet: A large-scale hierarchical image database, in 2009 IEEE conference on computer vision and pattern recognition, Ieee, 248–255

d'Errico, M., Facco, E., Laio, A., & Rodriguez, A. 2018, Automatic topography of high-dimensional data sets by non-parametric Density Peak clustering, arXiv e-prints, arXiv:1802.10549

Doimo, D., Glielmo, A., Ansuini, A., & Laio, A. 2020, Hierarchical nucleation in deep neural networks, arXiv e-prints, arXiv:2007.03506

Facco, E., d'Errico, M., Rodriguez, A., & Laio, A. 2017, Estimating the intrinsic dimension of datasets by a minimal neighborhood information, Scientific Reports, 7

Fefferman, C., Mitter, S., & Narayanan, H. 2013, Testing the Manifold Hypothesis, Journal of the American Mathematical Society, 29

He, K., Zhang, X., Ren, S., & Sun, J. 2016, Deep Residual Learning for Image Recognition, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778

Johnson, J., Douze, M., & Jégou, H. 2017, Billion-scale similarity search with GPUs, arXiv e-prints arXiv:1702.08734

Khalil, A. 2019, Data Management Tools for NFFA-EUROPE Project, MHPC Thesis, International School for Advanced Studies (SISSA)

Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, ImageNet Classification with Deep Convolutional Neural Networks, in Advances in Neural Information Processing Systems, Vol. 25 (Curran Associates, Inc.), 1097–1105

LeCun, Y., Jackel, L. D., Boser, B., et al. 1989, Handwritten digit recognition: Applications of neural network chips and automatic learning., IEEE Communications Magazine, 368, 41–46

Modarres, M., Aversa, R., Cozzini, S., et al. 2017, Neural Network for Nanoscience Scanning Electron Microscope Image Recognition, Scientific Reports, 7

Rodriguez, A., d'Errico, M., Facco, E., & Laio, A. 2018, Computing the Free Energy without Collective Variables, Journal of Chemical Theory and Computation, 14, 1206, pMID: 29401379

Rodriguez, A. & Laio, A. 2014, Clustering by fast search and find of density peaks, Science, 344, 1492

Schroff, F., Kalenichenko, D., & Philbin, J. 2015, FaceNet: A unified embedding for face recognition and clustering, in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 815–823

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. 2016, Rethinking the Inception Architecture for Computer Vision, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2818–2826

Weinberger, K. Q., Blitzer, J., & Saul, L. K. 2006, Distance metric learning for large margin nearest neighbor classification, in In NIPS (MIT Press)

Wilkinson, M., Dumontier, M., Aalbersberg, I., & et al. 2016, The FAIR Guiding Principles for scientific data management and stewardship, Scientific Data, 3

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. 2017, Aggregated Residual Transformations for Deep Neural Networks, in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 5987–5995