Contents lists available at ScienceDirect

Information Fusion

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

A tutorial on the segmentation of metallographic images: Taxonomy, new MetalDAM dataset, deep learning-based ensemble model, experimental analysis and challenges

Julián Luengo [a,*], Raúl Moreno [b], Iván Sevillano [a], David Charte [a], Adrián Peláez-Vegas [a], Marta Fernández-Moreno [b], Pablo Mesejo [a], Francisco Herrera [a]

[a] *Department of Computer Science and Artificial Intelligence, Andalusian Research Institute in Data Science and Computational Intelligence, DaSCI, University of Granada, 18071, Granada, Spain*

[b] *ArcelorMittal Global R&D, New Frontier, Spain*

ARTICLE INFO

Keywords:
Image segmentation
Metallography
Machine learning
Deep learning
Microscopy images
Computer vision

ABSTRACT

Image segmentation is an important issue in many industrial processes, with high potential to enhance the manufacturing process derived from raw material imaging. For example, metal phases contained in microstructures yield information on the physical properties of the steel. Existing prior literature has been devoted to develop specific computer vision techniques able to tackle a single problem involving a particular type of metallographic image. However, the field lacks a comprehensive tutorial on the different types of techniques, methodologies, their generalizations and the algorithms that can be applied in each scenario. This paper aims to fill this gap. First, the typologies of computer vision techniques to perform the segmentation of metallographic images are reviewed and categorized in a taxonomy. Second, the potential utilization of pixel similarity is discussed by introducing novel deep learning-based ensemble techniques that exploit this information. Third, a thorough comparison of the reviewed techniques is carried out in two openly available real-world datasets, one of them being a newly published dataset directly provided by ArcelorMittal, which opens up the discussion on the strengths and weaknesses of each technique and the appropriate application framework for each one. Finally, the open challenges in the topic are discussed, aiming to provide guidance in future research to cover the existing gaps.

1. Introduction

The popularization of Artificial Intelligence (AI) has deeply permeated industry manufacturing processes [1]. Usually, such processes require expert knowledge about the available materials, their characteristics and manufacturing procedures. The aim of AI is to support the experts in their decisions, encoding a significant portion of knowledge in a certain model [2], constituting a key part of the next Industry 4.0 revolution [3].

Image segmentation of metallographic images is a challenging task, where features of the material that will have an impact on the final product are grouped and identified together in structures at the microscopic level imaging. Hence, having more knowledge about the inputs (i.e. the materials employed) will result in a better understanding of the output of the manufacturing process (i.e. the final product), assessing the quality of the product for the client or even easing the development of new products. Images of the input materials contain plenty of knowledge related to the mechanical properties of the material and relevant information about the manufacturing process [4,5], and there is a wide range of application fields: additive manufacturing (for powder generation or printed samples), biomedical applications or electrochemistry, just to name a few. Commonly, qualitative and quantitative analyses of segmented microstructures are manually performed by materials scientists.

The rise of Deep Learning (DL) [6] has created new opportunities, enabling the experts to directly utilize images in their AI applications. Many AI applications were based on traditional computer vision techniques and classical AI methods in order to help scientists on their quantification analysis [7–9], but the use of DL has broadened the spectrum of possibilities and yielded good results in complex problems that were unfeasible before.

* Corresponding author.
E-mail addresses: julianlm@decsai.ugr.es (J. Luengo), raul.moreno1@arcelormittal.com (R. Moreno), isevillano@ugr.es (I. Sevillano), fdavidcl@ugr.es (D. Charte), adrianpelaez@ugr.es (A. Peláez-Vegas), marta.fernandezmoreno@arcelormittal.com (M. Fernández-Moreno), pmesejo@decsai.ugr.es (P. Mesejo), herrera@decsai.ugr.es (F. Herrera).

https://doi.org/10.1016/j.inffus.2021.09.018
Received 10 June 2021; Received in revised form 9 September 2021; Accepted 12 September 2021
Available online 30 September 2021

Among the many applications in industry manufacturing that greatly benefit from the potentialities of DL, we focus on the segmentation of metallographic images. Image segmentation is the task of partitioning an image into non-overlapping regions according to some criteria. In other words, it is the task of finding groups of pixels that "go together" [10]. It represents one of the oldest and most widely studied computer vision problems [11–13], and it is involved in many essential applications, like medical image analysis [14,15], autonomous driving [16], and robotics [17], just to name a few.

Although quantification tasks provide more accurate information, pixel-wise annotation of previous segmentation is a very time-consuming task. Therefore, automatizing microstructure characterization is one of the industrial applications that are actively adopting DL-based image segmentation to describe manufactured materials, helping the experts in such a demanding task. Since microstructure segmentation is too complex, is hard to fully automatize the characterization but in general automatic methods act as an assistant tool for experts. In addition to this, the scarcity of annotated samples favors the application of unsupervised or semi-supervised methodologies that are able to leverage unlabeled images.

The variety of microstructural images, AI techniques, methodologies and expert labeling availability create an important challenge when facing a new metallographic segmentation task. The literature on this topic focuses on the particularities of a single dataset, with a limited generalization ability to related metallographic segmentation tasks. This paper focuses on the semantic segmentation of metallographic images with the objective of providing scientific readers with a comprehensive overview of the available AI techniques. Practical users will also find interesting methodological aspects that will lead to a successful approach to any related metallographic problem, with real-world scenarios as an example. The main contributions of this work are both theoretical and technical and are summarized as follows:

- We create a completely new metallography dataset from additive manufacturing of steels (MetalDAM), with 42 labeled steel micrographs.
- We provide an updated taxonomy of semantic segmentation methods applied to metallographic images, including many DL-based models which represent the current state of the art, as well as a description of the most relevant approaches in the field.
- We propose a new DL-based ensemble proposal specialized in the semantic segmentation task.
- We develop an extensive experimental comparison of state-of-the-art models and the newly proposed ensembles with two case studies, the Ultrahigh Carbon Steel Micrographs dataset (UHCS) and the new MetalDAM dataset, covering several approaches, such as supervised and semi-supervised scenarios.
- We present a thorough analysis of the current difficulties that arise when tackling microstructure segmentation problems using AI techniques, and some challenges are discussed.

The paper is organized as follows. Firstly, Section 2 describes the task of semantic segmentation applied to metallographic images. Section 3 continues with the taxonomy and descriptions of the state-of-the-art approaches, whereas Section 4 introduces the proposed ensemble-based solutions. Section 5 explains the experimentation framework and provides analyses on the obtained results. Next, Section 6 discusses the main difficulties which can be found when dealing with this task in a real world scenario and the main challenges in the topic. Section 7 finishes with the conclusions of the work.

2. Segmentation of metallographic images

This section introduces the main characteristics of the semantic segmentation problem when applied to microstructures, as well as the available metallography datasets and the newly released MetalDAM dataset.

2.1. Semantic segmentation fundamentals

The objective of this task is to be able to correctly categorize each pixel X_{ij} in a w pixels wide by h pixels tall image $X \in \mathcal{M}_{w\times h}(\{0,\dots,255\})$ (assuming the image is grayscale), as its class $Y_{ij} \in \mathcal{Y}$. The nature of the problem suggests to consider the image as a whole rather than a pixel-by-pixel problem. Therefore, a solution to the task consists in a function $f : \mathcal{M}_{w\times h}(\{0,\dots,255\}) \to \mathcal{M}_{w\times h}(\mathcal{Y})$ that assigns images to label matrices.

According to the available data and, more specifically, whether it is labeled or not, a dataset S will consist of either image-label pairs $(X,Y) \in \mathcal{M}_{w\times h}(\{0,\dots,255\}) \times \mathcal{M}_{w\times h}(\mathcal{Y})$ or just simply images.

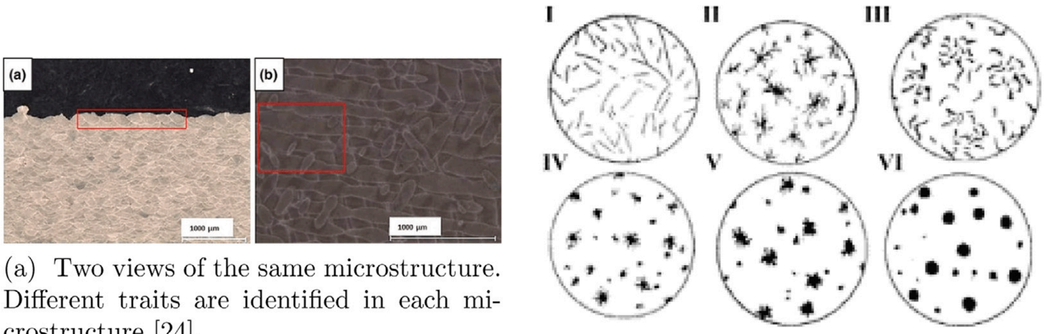
There are numerous works that address image segmentation from a more general perspective [18–21], but in this work, in order to create a more compact and well-posed study, we exclusively focus on the segmentation of metallographic images and associated methods.

2.2. Distinctive aspects of microstructure segmentation

Analysis of microstructures is essential to materials engineering and design. This field covers the discovery and study of new materials, usually solids. This analysis is traditionally based on features carefully measured by experts, and usually includes segmenting each microstructure, that is, separating it into regions according to the different phases or components of the material. The analysis and this region detection in particular are often carried out manually, which makes it a costly and time-consuming process. Several other features, such as volume fractions, size distributions and shape descriptors, some of which can be appreciated in Fig. 1, can be extracted more easily from the segmentation of a microstructure [24], thus the interest in automatizing the segmentation step itself. Those features are related to theoretical and/or empirical models developed by experts, such as dislocation theories [25], tensile properties [22] or simply characterization of certain materials [23].

The characteristic features of microstructures differentiate this type of segmentation from many other real-world situations, for example, object segmentation in regular photographs. In a normal photo taken with a camera, there is usually some portion of the image that does not contain elements of interest, and one or more objects of a certain size and a relatively closed shape that can be categorized according to a given list of classes. Objects may be repeated but they are usually countable. A micrograph, however, may show dozens or even hundreds of small regions of the same class, often connected to one another, and completely interleaved with other regions of a different class. To a certain extent, they could be interpreted as large-scale aerial images of complex networks of water and land masses where the objective is to identify rivers, lakes and different land types.

Accurate segmentation of microstructures is currently achieved by specialized applications, for example, ImageJ [26]. With this kind of application, experts can perform the segmentation task manually. Nevertheless, they require expert fine-tuning and a lot of time to achieve good performance on a specific dataset. There are several proposals that aim to automate this process by using machine learning with supervised, semi-supervised and unsupervised approaches. Although there are proposals based on classical techniques such as support vector machines [27], most of them propose methods based on deep learning, in particular, on convolutional neural networks (CNNs) [28–30] for the segmentation task, especially in a supervised way [31,32]. These techniques, however, need pixel-wise labels as a basis for their optimization, that is, each pixel in each microstructure image needs to be assigned a class according to the phase or component it belongs to. In the specific case of microstructures, this kind of labeling needs strong expert supervision in order to be reliable. This places some limits as to the size of the available datasets that can be used to train models, as will be explained later.



(a) Two views of the same microstructure. Different traits are identified in each microstructure [24].

(b) Different graphite particle patterns characterized by shape [25].

Fig. 1. Different studies of microstructures shapes. These studies are crucial in determining the behavior of each material.

Traditionally, color has been used to differentiate phases in metallography [33]. Due to the increase of microstructural complexity and the development of new advanced steels and their processes, optical microscopy is limited since spatial resolution is not enough to analyze the material substructure. Scanning Electron Microscope (SEM) images provide information in grayscale values based on relief phase-contrast, where phase identification is based on morphology, size, distribution and relative location to other phases. Furthermore, it is usually applied complementary sensitive crystallographic or chemical experimental methods to confirm the identification [34].

MetalDAM is cataloged as an Advanced High Strength Steel with an multi-phase ultra thin micro structure, sometimes enriched with precipitates. These facts increase the complexity compared to traditional steels (for example austenitic or ferritic inox, Fig. 10 and 13 in [35] or biphase steels like duplex, Fig. 14 in [35]) in terms of: number of phases, size and methods to prepare samples in the microscope.

Although it may seem that microstructure segmentation is a very particular problem, it can also be related to other types of segmentation. For example, it shares some commonalities with medical image segmentation [14]: both usually show very small but relevant regions, regions of the same class can have complex shapes and be connected, and are tasks that require a high effort from experts to be performed by hand. Similar reasoning may apply to remote sensing [36] and astronomical images [37]. This shows that learning from the obstacles that micrographs present can also help design new approaches for these tasks from very different fields. Fig. 2 shows two examples of medical and astronomical images where the segmentation task is used. As a result, any findings in the microstructure scenario may hint to general ideas on how to solve many other problems as well.

2.3. Datasets

In this section we analyze the current availability of metallography datasets and introduce details about the ones we use in the experimental section of this work, one of them being a novel public dataset that constitutes one of the main contributions of this paper.

We can find extensive datasets of metallographic images from the materials industry, both private, such as the aluminum alloy micrograph dataset provided by Nanshan Aviation Materials Industrial Park used in [40] for unsupervised segmentation, and public, such as Ultra-High Carbon Steel Micrograph DataBase [41]. The problem with these datasets is the absence of pixel-level labels, which is the main obstacle when addressing the segmentation problem.

In some studies, the problem of supervised segmentation of metallographs is addressed by labeling some metallographic images themselves with the help of materials science experts, this is the case in [27,42,43]. But, in most cases, these datasets are not made public to the scientific community. In other cases [44], labels are proposed at the image classification level, so they are not useful for the segmentation task.

Up to the authors' knowledge, the only public and pixel-wise annotated metallographic image dataset is the one presented in [45]. This dataset consists of a subset of labeled metallographs belonging to the previously introduced UHCS database. In fact, due to the current scarcity of labeled datasets, this paper proposes a new one for the semantic segmentation of metallographs, which will be described in more detail in the following subsections. Table 1 presents a summary of the main characteristics of the existing datasets.

2.3.1. UHCS

The openly available Ultrahigh Carbon Steel (UHCS) microstructure dataset was originally introduced in [41]. Subsequently, a subset of this dataset[1] was updated in [45] enabling the supervised semantic segmentation task by the addition of segmentation labels, and becoming the first public benchmark for semantic segmentation of metallographs.

This dataset is composed of 24 grayscale metallographic images. These images correspond to a single magnification level and the annotations have been obtained through a partially-automated edge-based segmentation workflow where the final particle segmentations are verified and retouched manually [45]. An example of these metallographs can be seen in Fig. 4. These images have a size of 645 × 484 pixels and the labels annotated distinguish between four classes of microstructures: proeutectoid cementite network, fields of spheroidite particles, the ferritic matrix in the particle-free denuded zone near the network, and Widmanstätten laths. More details about class distribution are provided in Table 2 and the class distribution per image is shown in Fig. 3.

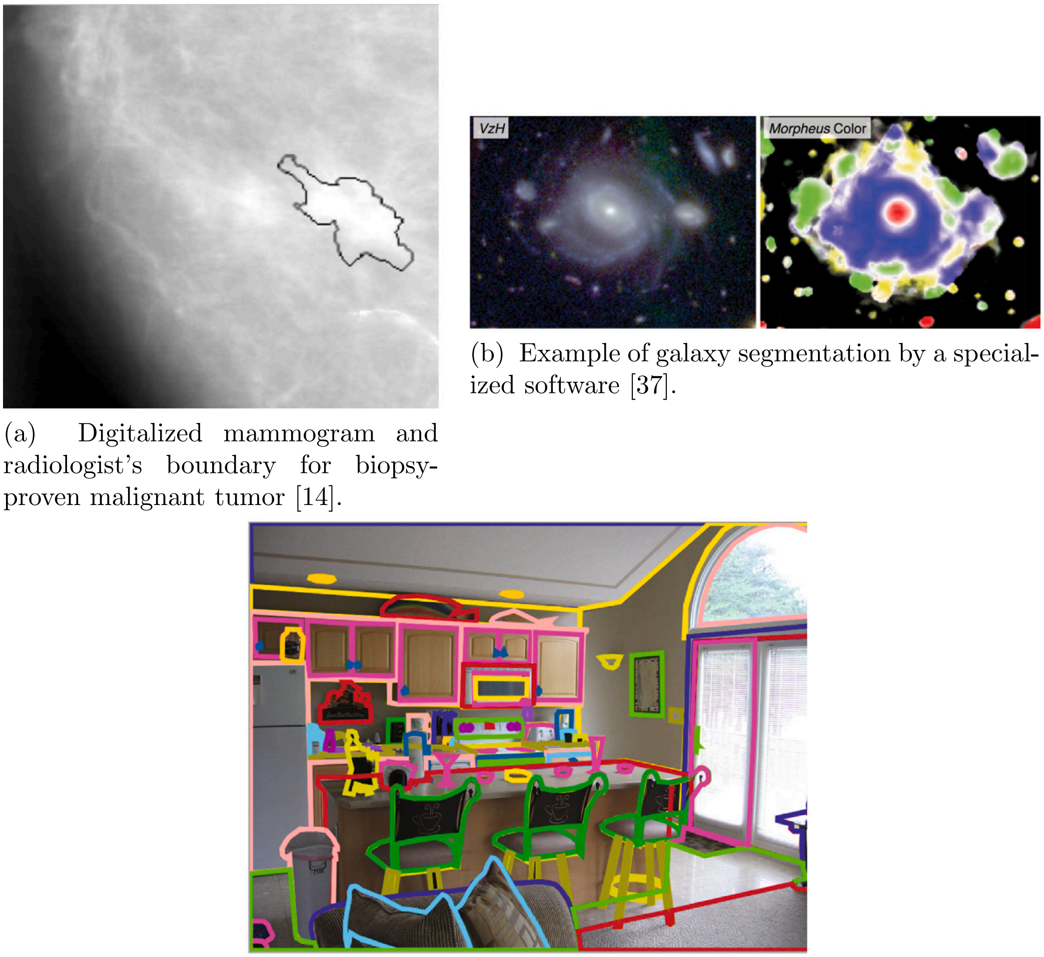
The main difficulties present in this dataset, according to the authors who have addressed the segmentation problem on these metallographs [45], are:

- The presence of areas where the cementite network phase is very fine or the contrast between this phase and the ferritic matrix is poor can lead to confusion between both phases.
- The low presence and the fine and broken shape of Widmanstätten lath areas.
- Areas with a very low density of spheroidite particles can be confused with ferritic matrix.
- The fact that the segmentation of microstructures can sometimes be ambiguous to human experts, which can lead to noisy labeling.

2.3.2. MetalDAM

In this paper, we present MetalDAM (Metallography Dataset from Additive Manufacturing), a new public benchmark dataset for semantic segmentation of metallographic images. This represents one of the main

[1] https://materialsdata.nist.gov/handle/11256/964.



(a) Digitalized mammogram and radiologist's boundary for biopsy-proven malignant tumor [14].

(b) Example of galaxy segmentation by a specialized software [37].

(c) Example of the ADE20K segmentation dataset [38].

Fig. 2. Medical and astronomical study with direct application of the semantic segmentation task. These images and metallographs have in common a local characteristic. This differentiates them from typical segmentation benchmark datasets such as Cityscapes [39] or ADE20K [38] (on Fig. 2c), where a global scene gives structure to the image. Moreover, unlike the usual benchmark datasets, the annotation can be performed exclusively by experts in the area. This adds an additional difficulty to the problem.

Table 1
Summary of metallography datasets.

Dataset	Images	Resolution	Labeled	Availability
Zhang et al. [40]	5086	2560 × 1920	No	Private
UHCSDB [41]	961	645 × 484	No	Public
Roberts et al. [42]	2	2048 × 2048	Yes	Private
Zhang et al. [40] (subset) [27,43]	30	1024 × 768	Yes	Private
UHCSDB (subset) [45]	24	645 × 484	Yes	Public
MetalDAM	42	1280 × 895, 1024 × 703	Yes	Public
MetalDAM (unlabeled)	164	1280 × 895, 1024 × 703	No	Public

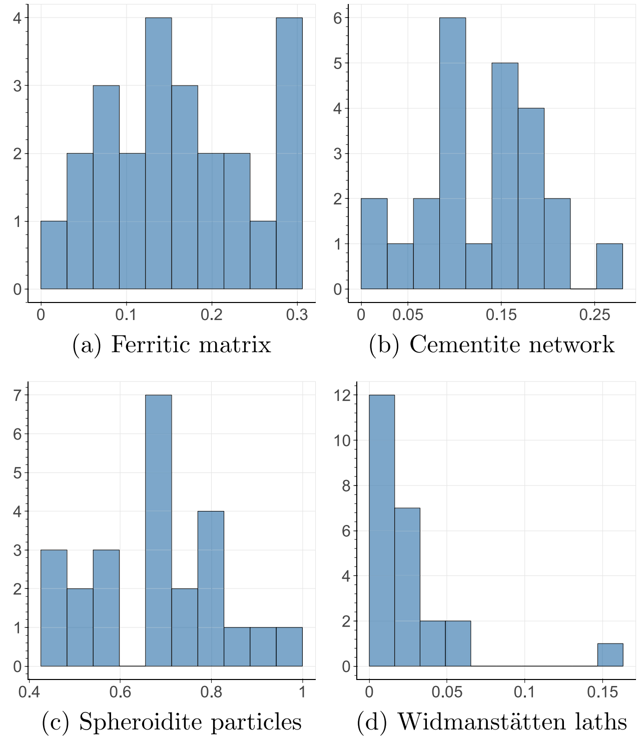
Table 2
Class ratio in UHCS.

Class	Ratio (%)
0. Ferritic matrix	16.12
1. Cementite network	12.98
2. Spheroidite particles	68.23
3. Widmanstätten laths	2.67

contributions of this study, due to the scarcity of public benchmark datasets for this task. MetalDAM has several advantages over existing ones, such as a larger number of annotated images and a higher image resolution. The MetalDAM dataset is available for download by the scientific community at https://dasci.es/transferencia/open-data/metal-dam/.[2]

This dataset contains 42 grayscale images taken from a SEM with resolutions 1280 × 895 and 1024 × 703. Such images are micrographs of steels that have been generated employing additive manufacturing techniques, and contain relevant information that can be used for quantitative and qualitative analysis of the material. An additional set of 164 unlabeled images obtained from the same materials is also provided at the same repository.

[2] Direct download of the images, both labeled and unlabeled, is available at https://github.com/ari-dasci/OD-MetalDAM/releases/tag/1.0.



(a) Ferritic matrix (b) Cementite network

(c) Spheroidite particles (d) Widmanstätten laths

Fig. 3. Histograms of class distribution per image in the UHCS dataset.

Table 3
Class ratio in MetalDAM.

Class	Ratio (%)
0. Matrix	31.86
1. Austenite	58.26
2. Martensite/Austenite (MA)	8.96
3. Precipitate	0.24
4. Defect	0.68

All the images in the labeled dataset have been annotated pixel-wise according to the 5 microconstituents that are present. Three out of five are different phases named as matrix, austenite and martensite/austenite (MA), and the two remaining are precipitates and defects. More details about class distribution are provided in Table 3 and the class distribution per image is shown in Fig. 5.

As mentioned before, the qualitative analysis of microstructures is really time-consuming. The case of MetalDAM contain images with higher resolution than UHCS, higher resolution will increase the time and effort of labeling process by domain experts. Regarding all of this, a process was designed to enhance the pixel-wise accuracy of the annotations and reduce the time spent by the annotators. The first step of this process is providing a pre-annotation instead of letting the experts annotate each pixel of the image from scratch. These pre-annotations are masks where each pixel of the image belongs to a certain class. Taking into account that 2 out of 5 MetalDAM microconstituents (matrix and austenite) are present on every single image and both represent more than 90% of pixels, binary segmentation of matrix and austenite phases was selected as pre-annotation. With the purpose of generating these masks, an unsupervised segmentation method [46] was employed. Even if this approach has obtained promising results on micrographs before [47], on MetalDAM it tends to under-segment the images by generating a binary mask instead of returning a multiclass segmentation. After optimizing this method for each image by tuning the hyperparameters, the pre-annotations were obtained through selecting the most accurate matrix-austenite segmentations. The second step is a manual refinement of the pre-annotations done by the material science experts through an image segmentation labeling tool. This refinement task basically consisted of adding the annotations of MA and defects, as well as, some minor fixes of the matrix-austenite pre-annotation in some of the cases. The reduced size of the precipitates increased notably the effort of annotating these microconstituents, as precipitates do not provide much meaningful knowledge while analyzing these kinds of microstructures, most of them were ignored during the annotation process. Due to this fact, the precipitates are ignored while measuring the performance of every image segmentation model.

Finally, a few images were manually labeled from scratch and compared to the ones generated from the pre-annotations. In terms of time, as the classes manually included during the refinement process represent less than 10% of the pixels on the dataset, the labeling effort of the full dataset was highly reduced. Additionally, in terms of accuracy, the unsupervised segmentation method was able to do a better segmentation of complex boundaries between matrix and austenite phases. A sample of an image from MetalDAM together with its annotation (i.e., its ground truth segmentation) is shown in Fig. 6.

3. Taxonomy of metallography segmentation methods

Several techniques for segmentation of metal microstructures have been proposed in the literature, with varying degrees of complexity and different ways of utilizing the available data. This section reviews existing proposals as well as some other approaches to segmentation that can be applied to this specific problem, organizing them into a taxonomy.

3.1. Taxonomy

The techniques used in the literature to segment metallographs and microstructures can be broadly divided into two large methodological

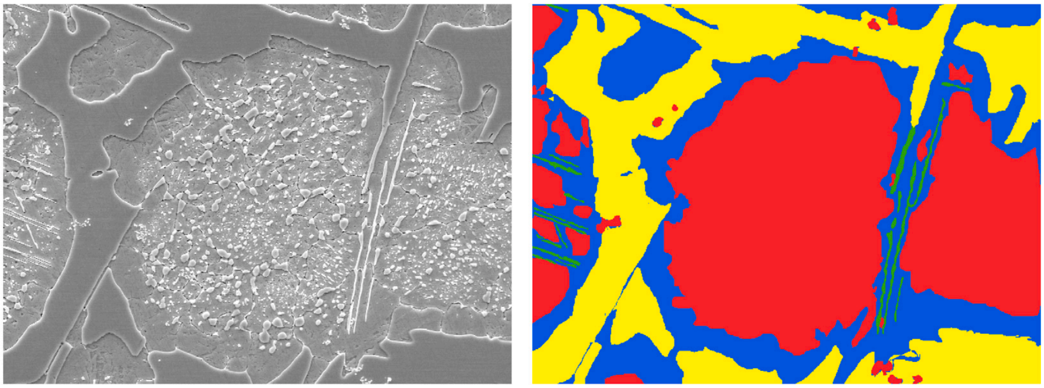


Fig. 4. Example of metallograph extracted from the UHCS dataset. Left: original image. Right: ground truth segmentation.

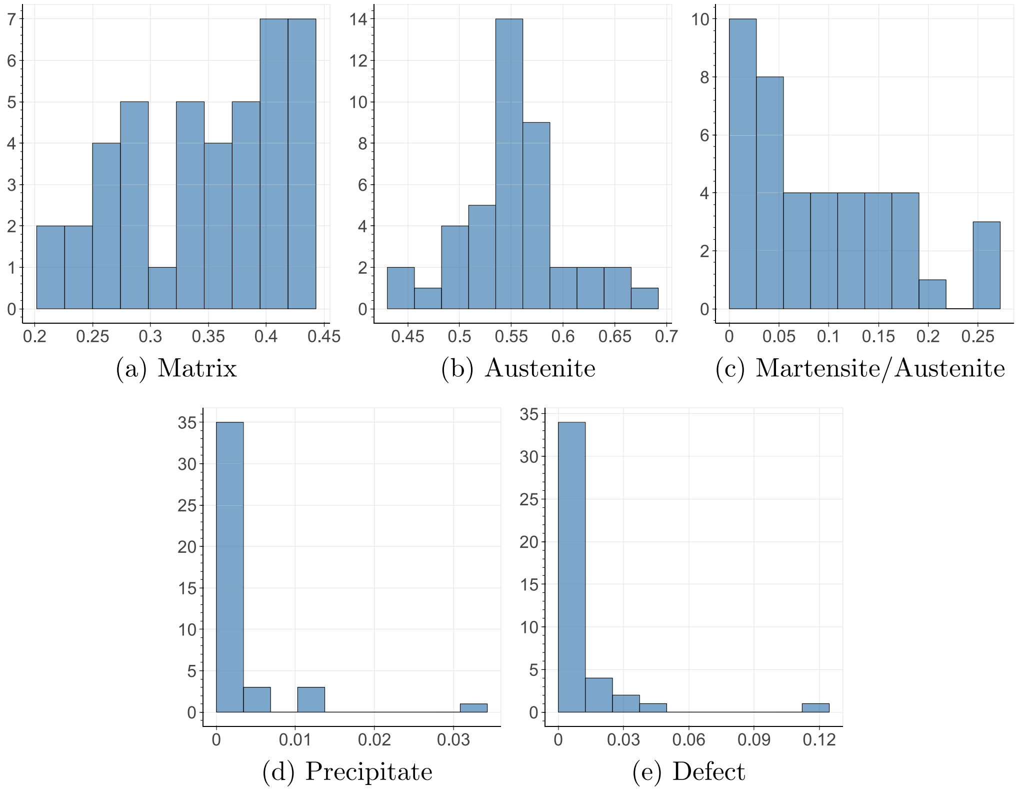


Fig. 5. Histograms of class distribution per image in the MetalDAM dataset.

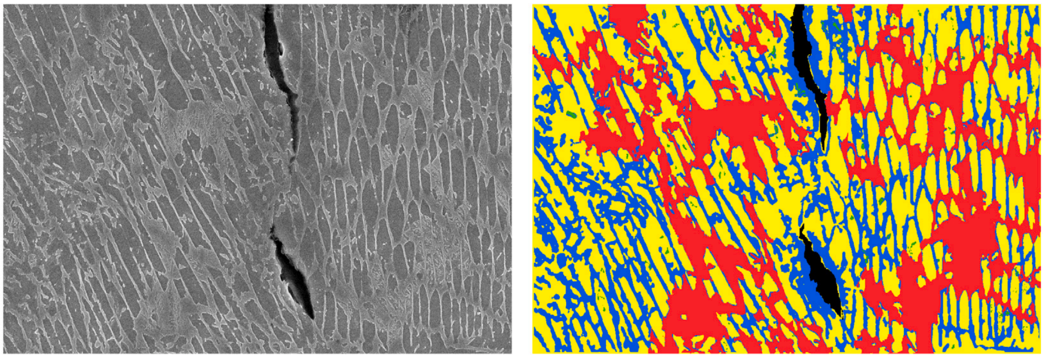


Fig. 6. Example of metallograph extracted from the MetalDAM dataset. Left: original image. Right: ground truth segmentation.

categories. On the one hand, those techniques that do not use any form of machine learning, and that segment the images based solely on pixel intensity or on the shape of the structures to be segmented. Within this category there are classical image processing and computer vision techniques, such as thresholding [48,49], region growing [50] and deformable models [51]. On the other hand, there are learning-based segmentation techniques, where available data (either annotated or not) guide the segmentation task through a training process. Within this category, most works presented in the literature are of supervised nature [31,32,42,45,52–56], relegating unsupervised approaches to a limited number of works [47,52]. The vast majority of learning-based methods employ some form of deep learning [6], but more classical approaches, where feature extraction and classification are performed in two separate stages, are also present in the literature [27,57,58].

In particular, the supervised approaches employed in the experimental section of this paper, as well as most of the ones employed in the literature, can be classified into the following sub-categories: (a) those presenting an encoder–decoder architecture where upsampled higher-level feature maps are fused with higher-resolution, lower-level feature maps [56,59–61]; (b) methods based on the extraction of a hypercolumn of descriptors that are then fed to a multilayer perceptron [62]; and, finally, (c) approaches that present some sort of pyramidal structure to perform the segmentation using multi-scale feature maps, rather than only utilizing a single and final upsampled feature map [63–68].

Supervised learning techniques generally outperform image processing methods. However, this remarkable performance comes at the cost of employing data hungry approaches. This is an issue since, in microstructural characterization and analysis, due to the high cost of labeling images, the available datasets are very limited in the number of training images. Therefore, there is a great interest in developing segmentation methods requiring a lower annotation effort (either in terms of label detail, e.g. providing just a weak supervision instead of pixel-level annotation, or in the number of training images). In this sense, and despite their evident utility, semi- and unsupervised metallographic image segmentation algorithms are still in their infancy. In fact, up to the authors' knowledge, only one semi-supervised segmentation method has been applied to this problem so far [43]. In this paper, we introduce one semi-supervised method [69] in the experimental comparison, and explore the possibilities of such approaches in this context.

Regarding unsupervised learning methods, most of them perform a quite straightforward application of classical clustering algorithms, like k-means [70], DBSCAN [40] or fuzzy c-means [71]. Neural-based approaches have also been employed, including self-organizing maps [52], superpixels and convolutional neural networks [47], and conditional generative adversarial networks [72]. However, it is important to notice that all these unsupervised methods, even if relevant and promising, have not yet been thoroughly investigated and, so far, do not produce results that are good enough to be actually applied in real scenarios.

Fig. 7 shows a taxonomy of methods used in the segmentation of metallograhic images, including methods from the literature and those employed in this paper. These latter are described more in detail in next subsections.

3.2. Image processing techniques

In this section we give a brief overview of the image processing techniques that have been applied to metallographic or microstructural images, both in the literature and in this paper. This section deals with techniques that do not employ any learning procedure from data, and that rely on the inherent properties of each image (like gray-level intensity or shape). These techniques are amongst the simplest and fastest ones to tackle this task. Although they are not expected to perform as well as more sophisticated methods (e.g. deep learning-based approaches), their simplicity, efficiency and accessibility make them ideal candidates to be part of the baselines of any study.

The image processing techniques applied to metallographic image segmentation can be mainly grouped into three families. Image thresholding techniques are the most widely applied in this problem, among which approaches based on Otsu's thresholding method and minimum cross-entropy thresholding stand out. Another proposed way to segment these images are techniques based on region growing, such as the classic Watershed algorithm. Finally, deformable models, such as active contour models, have also been applied to metallographic images. An overview of these techniques is provided below:

- Image thresholding. This kind of techniques, widely used in image segmentation problems [73,74], are based on the selection of specific histogram threshold values, either manually or automatically, which will define the intensity ranges corresponding to each class. Thresholding techniques are used when, apparently, the mere gray-level intensities could allow one to segment the structures of interest. These are possibly the simplest techniques, where only the histogram of the image is used to carry out the segmentation. However, these methods present serious limitations when the complexity of the image to be segmented increases. For an image whose histogram is not bimodal, assuming that we want to perform binary segmentation, the results obtained by thresholding techniques could be poor, and the need for more sophisticated methods arises. Due to the weaknesses of these simple techniques to address a complex problem such as metallographic image segmentation, it is sometimes necessary to combine them with other methods, in order to obtain better performance than with the mere application of thresholding techniques.

  In this sense, computer vision techniques such as geometric transformations, lighting correction or denoising, can be applied in combination with thresholding algorithms to improve their performance. In particular, an existing approach applies these types of techniques in combination with a multi-Otsu global thresholding method specifically designed for this segmentation problem. This method is compared with five other general-purpose thresholding and clustering methods, namely multi-Otsu [75], fuzzy c-means [71], fuzzy-Tsallis entropy with differential evolution [76], multilevel thresholding-based on fractional-order Darwinian particle swarm optimization [77], and thresholding based on harmony search optimization [78]. The results of this experimental study show a better performance of the proposed method according to the authors' visual inspection [49].

  The minimum cross-entropy thresholding algorithm [79] yields remarkable results in another case study of metallographic images, where an experimental comparison of twelve classical thresholding algorithms is carried out. Those algorithms are divided into point-dependent [79–86] and region-dependent [87–89] algorithms, depending on if they are based solely on the pixel value or if they incorporate additional information from neighboring pixels. In this experimental framework, minimum cross-entropy thresholding showed the most robust results [48].
- Region Growing. This is another family of image segmentation methods widely used in many different application domains [90,91]. These techniques are based on the initial selection of seed pixels, and the subsequent iterative expansion of these seeds into larger regions, through the association of neighboring pixels, or regions of pixels, that meet certain similarity conditions. These techniques are simple and effective in many cases, and they generally overcome thresholding methods in noisy images, where the region edges are not well-defined. However, the performance of these methods is very dependent on the initial selection of seeds, as well as the criteria to expand the segmented regions. Within this methodological framework, the classic Watershed algorithm [92] is applied on metallographic images in an iterative way, in combination with a thresholding based seeds selection and a ridge detection algorithm [93]. To overcome the over-segmentation problem that the Watershed algorithm presents, the authors propose a refinement stage based on the Bayes rule [50].
- Deformable models. Deformable models [94] are segmentation techniques that adapt a curve with the goal of maximizing its overlap with the actual contour of an object of interest within an image. Specifically, active contour models [95], also called "snakes", are parametric deformable models where the deformation procedure is driven by the minimization of an energy function, until the deformable model coincides with the object boundary. Active contour models, in combination with a geometric deformable model called level set [96], is applied to the problem of segmentation of microstructural images in three dimensions. By means of numerical approximations to partial differential equations, and guided by force vectors extracted from the image data, this method evolves a 3D surface to fit the boundaries of the phases. A visual comparison shows how the proposed method outperforms a traditional thresholding approach on a metallographic image [51].

As a baseline for our study, we include the widely known Multi-Otsu thresholding method, introduced in [75] as a multilevel extension of the classic Otsu's method [80], in combination with superpixel algorithm [97]. The core idea of Otsu's approach is to find the optimal thresholds by maximizing the variance between classes. This choice is motivated by the fact that multi-Otsu, as well as other methods based on Otsu's algorithm, are amongst the most commonly employed methods on image segmentation. In addition, it has attractive characteristics for our case study, such as multilevel segmentation or automatic threshold selection, and has been largely employed in the segmentation of metallographs [48,49].

3.3. Learning-based approaches

Although image processing techniques are varied and applicable to different situations, many problems require more specific and flexible solutions, and the expert knowledge needed to select appropriate filters

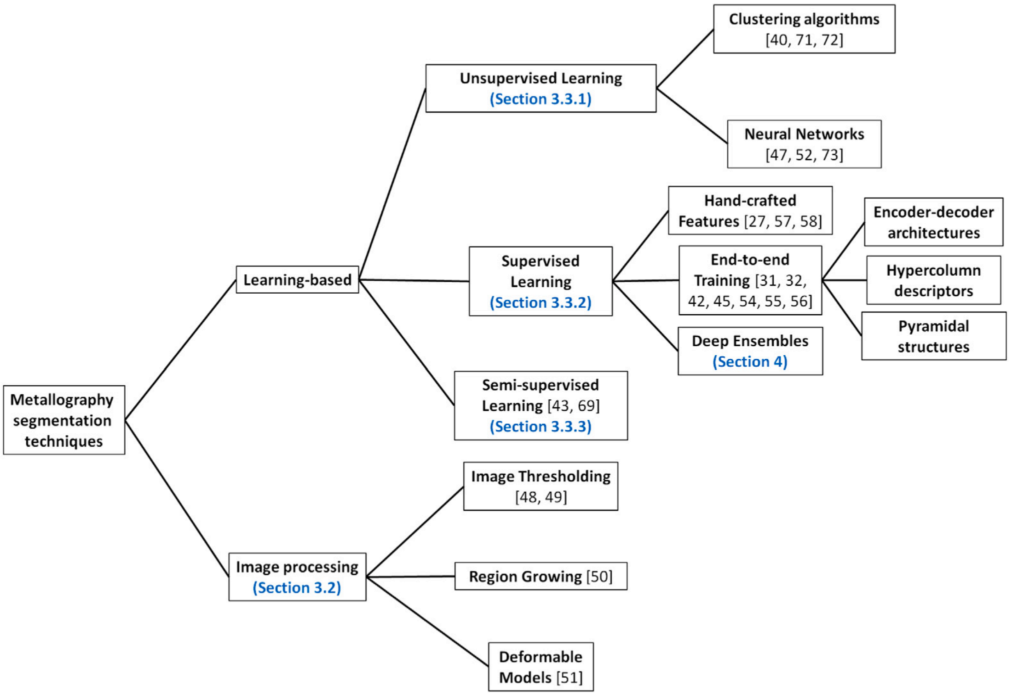

Metallography segmentation techniques
Learning-based
Unsupervised Learning (Section 3.3.1)
Clustering algorithms [40, 71, 72]
Neural Networks [47, 52, 73]
Supervised Learning (Section 3.3.2)
Hand-crafted Features [27, 57, 58]
End-to-end Training [31, 32, 42, 45, 54, 55, 56]
Deep Ensembles (Section 4)
Encoder-decoder architectures
Hypercolumn descriptors
Pyramidal structures
Semi-supervised Learning [43, 69] (Section 3.3.3)
Image processing (Section 3.2)
Image Thresholding [48, 49]
Region Growing [50]
Deformable Models [51]

Fig. 7. Dendrogram displaying the methodological taxonomy employed in this paper. Representative references for each category are located next to each leaf node. The only exception is represented by the Deep Ensembles, which are a contribution of this work, with no previous works in the field employing them. The section where the description of each methodological family can be found is shown in blue within brackets.

may not be available. When the solution needs to adapt to the concrete traits of the images in order to improve performance, a learning technique may be adequate. Learning a model usually depends more on available data than expert knowledge, so it will be easier to apply in many cases.

These methods can be characterized according to whether they are provided with the expected outcomes for the examples of the problem they learn from, that is, the type of supervision they receive. As such, they can be unsupervised, if no solutions are provided; supervised, if each example is paired with its solution; or semi-supervised, when only a certain portion of examples are solved. The following sections describe the main approaches to semantic segmentation for each of these categories, focusing on specific methods that are also used within the experimentation.

3.3.1. *Unsupervised methods*

As indicated in Section 2.3, the two most massive metallographic image datasets do not have their corresponding labeling, due to the complexity and high annotation cost of these images, only small subsets of these have been manually annotated by experts. Semi-supervised learning approaches allow to take advantage of the few annotated examples that metallography segmentation datasets have (see Section 3.3.2), but in the complete absence of labels we can only tackle this problem from the unsupervised point of view. This fact motivates research in this line and several unsupervised approaches have been proposed. In addition, this type of methods can also help the expert in the annotation process, being part of the image labeling pipeline, reducing costs and speeding up the process, as has been the case of the MetalDAM dataset (see Section 2.3.2). In this subsection, we introduce the existing unsupervised approaches applied in metallographic image segmentation.

Unsupervised techniques applied to metallography segmentation can be divided into two groups. On the one hand, clustering techniques, such as k-means, DBSCAN and fuzzy c-means have been applied to metallographic images. However, due to the complexity of the problem, in some cases the application of classic clustering algorithms may not be sufficient. In order to achieve a better performance, more sophisticated techniques based on neural networks such as self-organizing maps, CNNs in combination with superpixel algorithms, and conditional generative adversarial networks (CGANs), which are an extension of the classic GANs with the ability to receive an input information that conditions the subsequent image generation, have also been proposed to address the problem. A brief description of these methods is provided below:

- Clustering algorithms. We can find different applications of classical clustering methods in metallography and microstructural segmentation problems. Although these methods are designed for clustering tasks, they can be used directly in segmentation either individually [98] (taking as data points the pixels of the image, which will be 1D data points in the case of grayscale images, and 3D data points in the case of RGB images) or in combination with other methods. As with image processing techniques, the simplicity of classical clustering algorithms may imply the need to be supported by other auxiliary techniques when faced with a complex problem, as is the case here. In this sense, there are several pipelines proposed for metallographic image segmentation, with clustering algorithms as main segmentation technique and supported by other algorithms. For instance, the use of clustering algorithms such as Mean Shift [99], k-means [100] or DBSCAN [101] (depending on if the desired number of clusters is known) together with image processing techniques [102], is proposed as a pipeline to address the metallographic segmentation problem [70]. Another common approach is to perform dimensionality reduction in the image using superpixel algorithms and the subsequent use of clustering techniques over the superpixels generated. An improved version of the SLIC superpixel algorithm [103], together with the use of the DBSCAN clustering algorithm to perform the segmentation, and a final refinement stage using the k-means algorithm, form another segmentation pipeline applied to metallographic images [40]. Another strategy is to redesign a clustering algorithm to obtain a better performance on a specific type of images. In particular, the fuzzy

c-means algorithm [104] is modified to improve its performance when segmenting metallographic images, the basic idea being the incorporation of spatial information related to the gray level of each pixel neighborhood [71].
- Neural Networks. Due to the high complexity of the images, we are trying to segment, as well as the recent success of deep learning in many computer vision problems, more advanced approaches based on unsupervised neural networks have also been explored. The first neural approach to this problem was the application of self-organizing maps [105], a type of unsupervised neural network able to produce a low-dimensional representation of the input space. Unlike most neural networks, it does not learn by error correction through backpropagation, but it uses competitive learning, where the nearest neuron for each input data is selected, and its weights and those of its neighbors are updated, producing the aforementioned low-dimensional representation. This method is directly applicable to the segmentation task, by including a neuron for each class that we want to segment [52]. Another approach is to try to mimic the manual labeling process carried out by metallography experts. With this purpose, the SLIC superpixel algorithm [103] is applied to the image producing small coherent regions, and then a CNN is used to classify different spatially separated superpixels with similar characteristics under the same label [47]. Recently, deep generative approaches, like CGANs, have also been applied to metallographic image segmentation [72] but their effectiveness, benefits and applicability to this problem has to be confirmed yet.

In this paper, we use k-means combined with superpixels [97] as unsupervised approach. In opposition to other more complex techniques, it is a simple, effective and popular method, that has already been successfully used in the literature. Both unsupervised and image processing techniques are referred to as baselines in the experimental section. This is a natural consequence of the algorithmic nature of Otsu's thresholding method and k-means clustering algorithm, since it is well-known that the objective function of Otsu's method is equivalent to that of k-means in multilevel thresholding [106]. Furthermore, they are both based on the same criterion that minimizes the within-class variance.

3.3.2. Semi-supervised methods

A common obstacle in many real-world scenarios, especially present when applying machine learning to industrial problems, is the unavailability or high cost of labeled instances. As explained before, this is also the case for metallographs. As a result, unlabeled images can be extracted at a faster pace than they can be annotated. In order not to waste potentially useful information, semi-supervised learning [107] can take advantage of unlabeled samples as well as labeled ones. This section introduces some recent techniques for semi-supervised segmentation which could be applied to microstructures, and describes an existing proposal as well as the method used later in the experimentation.

There exist several ways of drawing information from unlabeled instances. A very common approach is self-supervision [108], which focuses on withholding or creating some information that the model will need to predict in order to be optimized. A different perspective is provided by information theory, with measures that can provide objectives to be optimized using unlabeled data. The following is a brief description of the main approaches to semi-supervised segmentation:

- Self-supervision. This kind of methodologies either obscure some information from the images in order for the model to predict and be evaluated with respect to it, or create an output which serves to assess the adequacy of the predicted labels. For instance, a common approach is eliminating color from images, using only grayscale data as inputs, and predicting color versions. Another one is pseudo-labeling, which means predicting labels over unlabeled images using a supervised model and taking them as true labels to continue training [43,109].
- Information theory and probability. Different approaches can be made from the perspective of information theoretical measures and probability theory techniques. For example, a general framework for entropy-based semi-supervised learning has been proposed [110]. Later in this paper, we test the potential of the proposal in [69] when it comes to segmenting microstructures.
- Adversarial training. Some approaches utilize a generative adversarial network (GAN), which is a convolutional network with two main components, a generator and a discriminator. The generator usually takes some random noise and provides fake inputs to the discriminator, which must learn to discern them from the real inputs sampled from the dataset. In semi-supervised models, the discriminator component helps train the feature maps with unlabeled images as well as labeled ones [111,112]. To the best of our knowledge, no semi-supervised adversarial approach has yet been applied to microstructures.

To the best of our knowledge, the only semi-supervised method employed in the literature for metallography segmentation consists in applying a pseudo-labeling process in combination with self-paced learning [43], which implies learning only from sufficiently confident pseudo-labels. The authors test several options for a criterion by which to filter those pseudo-labels, including a dynamic threshold that allows more of them to be used as the training progresses, as well as a combination of cross-entropy and Dice losses.

Universal Semisupervised Semantic Segmentation (USSS) [69] combines both semi-supervised learning and multi-domain learning into a model able to combine information from two different domains while, at the same time, retrieving knowledge from unlabeled instances from those domains. This model fits into the information theory-based category, due to its choice of loss functions. Its main components include one encoder module for every type of input image from any domain, independent decoder modules for labeled images from each domain, and an entropy module for computation of two unsupervised loss functions out of the unlabeled images. These unsupervised loss functions consist of the entropy of a similarity score measured over different subsets of the data: one associated to the cross-dataset unlabeled image similarity and one associated to within dataset similarity. These loss functions are controlled by two parameters:

- α: controls whether the cross-domain unsupervised loss is being used.
- β: enables or disables the within domain unsupervised loss.

The loss function used by USSS is defined using the following similarity vector where i and j are dataset indices (equal for within-dataset similarity), $E \circ F$ is a label embedding obtained from the encoding of an image and ϕ is a similarity metric:

$$[v_{ij}]_k = \phi\left((E \circ F)\left(x_u^{(i)}\right), c_k^{(j)}\right) \quad \forall k = 1, \dots |Y_j| \tag{1}$$

The entropy of a discrete distribution computed as the softmax operator of the similarity vectors is used as the unsupervised loss.

The interest in the USSS model is due to the fact that it allows higher flexibility than other semi-supervised approaches, it does not require self-supervision and instead allows to directly train using labeled and unlabeled instances within the same process, using more than one data domain if required. This facilitates several experimentation schemes in order to verify which settings of semi-supervised learning can improve the performance of a model.

3.3.3. Supervised methods

In this section we describe different supervised approaches to semantic segmentation suitable for microstructures. In the literature, there are proposals with supervised methods based on classical techniques, such as support vector machines [27], Bayesian based classifiers [58] and Random Forest [57]. However, we focus our experiments on deep learning-based methods since they are currently the state of the art in computer vision. We detail differences and similarities between the main models proposed.

In computer vision problems such as image classification, object detection and semantic segmentation, state-of-the-art models are often based on CNNs. There are several well-known architectures for the classification task, such as VGG [113], ResNet [114] or EfficientNet [115] which exploit the topological properties of images. This CNN structure is used as a backbone to build different deep learning models, such as the models used for semantic segmentation. On Fig. 8 we show different semantic segmentation architectures based on the same backbone, which is highlighted in red.

Regardless of the underlying deep neural network employed as backbone, there are several techniques to reach a pixel-wise prediction, resulting in very different neural networks designs to perform the semantic segmentation task. In previous works, fully convolutional neural networks (FCNN) have been widely used to address segmentation in the metallography domain [31,32,55]. Also, segmentation models originally proposed to perform segmentation in other domains have been used on metallographs, such as U-Net [56], DeepLabV3 [54] or PixelNet [45]. Finally, a specific segmentation model for the metallography domain called DefectSegNet and based on the U-Net [59] and DenseNet [116] models is proposed [42].

The most relevant proposals we will use on microstructures segmentation can be grouped into three categories: those which employ encoder–decoder architectures; those that present a pyramidal structure; and those using per-pixel hypercolumns. A brief description of each one and its main proponents is presented below:

- Encoder–decoder architectures. These architectures try to condense the information via an encoder, namely the backbone, and then decode it with an additional decoder component. In order not to lose local information, the encoder layers are usually connected to the decoder layers of the same resolution. FCNN [117], U-Net [59], Linknet [61] and a more sophisticated U-Net++ [60] exploit this idea. Fig. 8a shows a schematic view of a network of this kind.
- Pyramidal structures. These segmentation models merge different resolution layers to extract more complex features. That is, the output is influenced directly by feature maps that detect different levels of detail from the image. Models like FPN [68], PSPNet [64], PAN [65] and Deeplabv3 [66] use this idea with small differences. One example of a pyramidal structure is displayed in Fig. 8c.
- Hypercolumns. A hypercolumn connects features of initial layers with features of final layers to obtain information at different resolutions. More specifically, the hypercolumn h_p for pixel p is defined as the concatenation of all the features of each feature map with less euclidean distance to the pixel p, $h_p = [c_1(p), c_2(p), \dots, c_n(p)]$ where $c_i(p)$ refers to the feature with less euclidean distance to p in the ith feature map. This hypercolumn is the input of a dense neural network which predicts the final class value. PixelNet [62], illustrated in Fig. 8b, uses this structure.

4. Deep learning-based ensembles for the segmentation of metallographic images

The fusion of information from different models is a very common practice in machine learning, where we can obtain knowledge based on various models with different behaviors [118]. In this section we present two novel deep learning-based ensemble approaches for the semantic segmentation task. First, we formally describe the mathematical terms on which we will rely on. Second, we describe a general Stacking Ensemble-based model fusion approach for our problem. Third, we present an information fusion method called Artificial MultiView Ensemble (AMVE) based not on predictions but on extracted features. Finally, we present Semantic Segmentation Ensemble, a fusion method that takes advantage of the specific image topology of the segmentation problem.

4.1. Definitions

In this subsection we name each of the elements involved in the following subsections.

Let P be the pixel set. Each $p \in P$ has a ground truth prediction y_p, also called label. For each pixel p we can define its k-sized window as $W_{p,k}$, which consists of the square window of side k pixels long centered on pixel p. Let $(m_1, \dots, m_M)$ be the models we want to fuse. For each model and each pixel, let $\hat{y}_p^{(i)}$ be the prediction of p by the model m_i. The prediction of a window $W_{p,k}$ will be called $\hat{y}_{W_{p,k}}^{(i)}$. Each model m_i predicts intermediate feature maps on which they base their predictions usually organized in different levels. Let call $F_l^{(i)}$ the feature map of the model m_i on level l. Also, for each pixel p there is a specific feature on each feature map $F_l^{(i)}$ whose Euclidean distance to this pixel p is the minimum. Let us call this feature $F_{l,p}^{(i)}$. We can also define $F_{l,p,k}^{(i)}$, which is the feature window centered on $F_{l,p}^{(i)}$ of size k.

4.2. Stacking Ensemble

The Stacking Ensemble approach is a classical method of model assembly [119]. It is based on building a meta-learning model that learns based on the predictions of the models we want to aggregate.

Based on the above definitions, the Stacking Ensemble approximation can be described as a parametric function f_θ which takes as input the predictions of the model m_i and is solution of the following minimization problem:

$$\arg\min_\theta \sum_{p \in P} \mathcal{L}\left(y_p, f_\theta\left(\hat{y}_p^{(1)}, \dots, \hat{y}_p^{(M)}\right)\right) . \tag{2}$$

Fig. 9 shows a diagram describing the Stacking Ensemble approach. Each model takes as input the original image. After processing each model output (light blue), the Stacking Ensemble approach (green) processes a second level prediction. For a certain pixel prediction, the Stacking Ensemble uses the predictions of the previous models for this certain pixel.

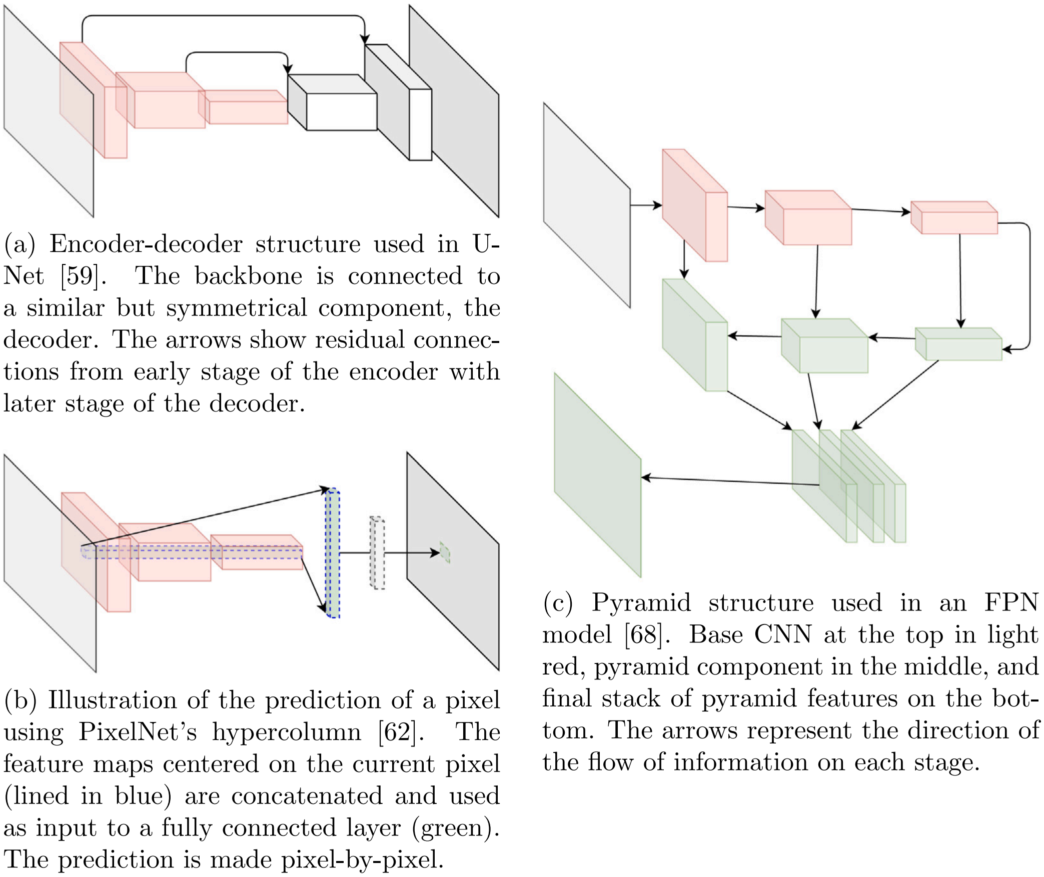
4.3. Artificial Multiview Ensemble

On this subsection, we present a new ensemble approach called Artificial MultiView Ensemble (AMVE). This method is a meta-learning model that learns from the features extracted by various models. This approach is inspired by Sun et al. [120], where the features learned by each model are interpreted as views of the same instance.

Based on the above definitions, the AMVE approximation can be formulated as a parametric function f_θ which takes as input the features $F_{p,l}^{(i)}$ of a certain layer l and is solution of the following minimization problem, which is similar to Eq. (2):

$$\arg\min_\theta \sum_{p \in P} \mathcal{L}\left(y_p, f_\theta\left(F_{p,l}^{(1)}, \dots, F_{p,l}^{(M)}\right)\right) . \tag{3}$$

Fig. 10 displays a diagram describing the AMVE approach. As with the previous structure, each model takes as input the original image. After processing each model output (light blue), the AMVE approach (green) takes a second level prediction based on not the pixel predictions but a certain feature map of the model (dark blue). For a certain



(a) Encoder-decoder structure used in U-Net [59]. The backbone is connected to a similar but symmetrical component, the decoder. The arrows show residual connections from early stage of the encoder with later stage of the decoder.

(b) Illustration of the prediction of a pixel using PixelNet's hypercolumn [62]. The feature maps centered on the current pixel (lined in blue) are concatenated and used as input to a fully connected layer (green). The prediction is made pixel-by-pixel.

(c) Pyramid structure used in an FPN model [68]. Base CNN at the top in light red, pyramid component in the middle, and final stack of pyramid features on the bottom. The arrows represent the direction of the flow of information on each stage.

Fig. 8. Schematic diagrams for each of the main families of supervised approaches. The original architecture of a CNN, highlighted in red, is used as backbone for each segmentation model. The arrows on each model represent the flow of data on each stage. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

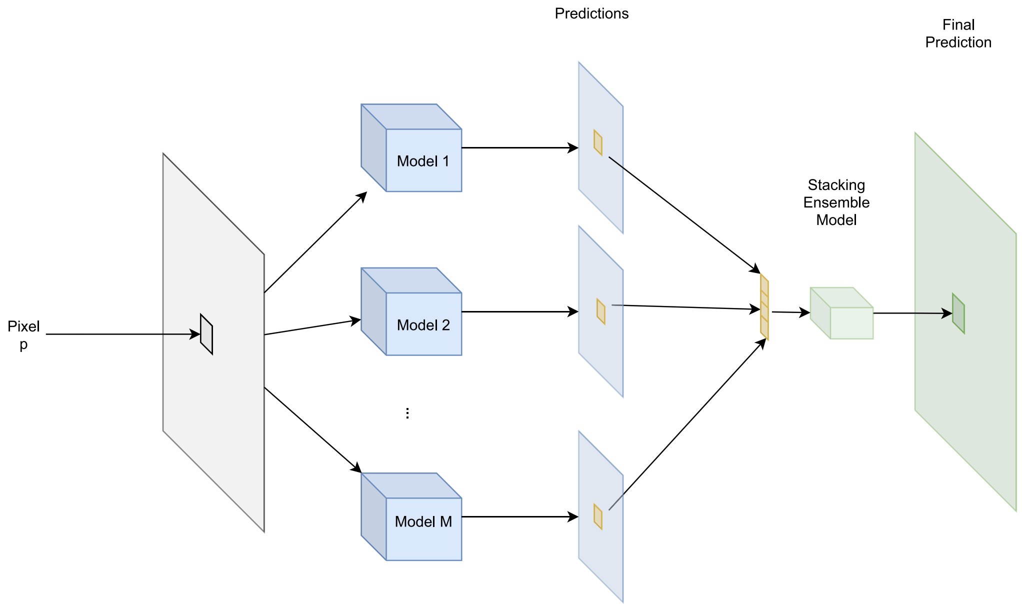


Fig. 9. Illustration of the Stacking Ensemble approach. In blue, the stacked models and their predictions. In yellow, the predictions of previous models used by the Stacking Ensemble method to infer the final pixel p prediction. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

pixel prediction, the AMVE bases the prediction on the feature with the least Euclidean distance to the pixel.

As can be seen, the main difference between the Stacking model and the AMVE model is that the Stacking approach takes the final prediction of models as inputs and AMVE approach takes features as inputs. In addition, we can discuss different pros and cons of one model versus the other:

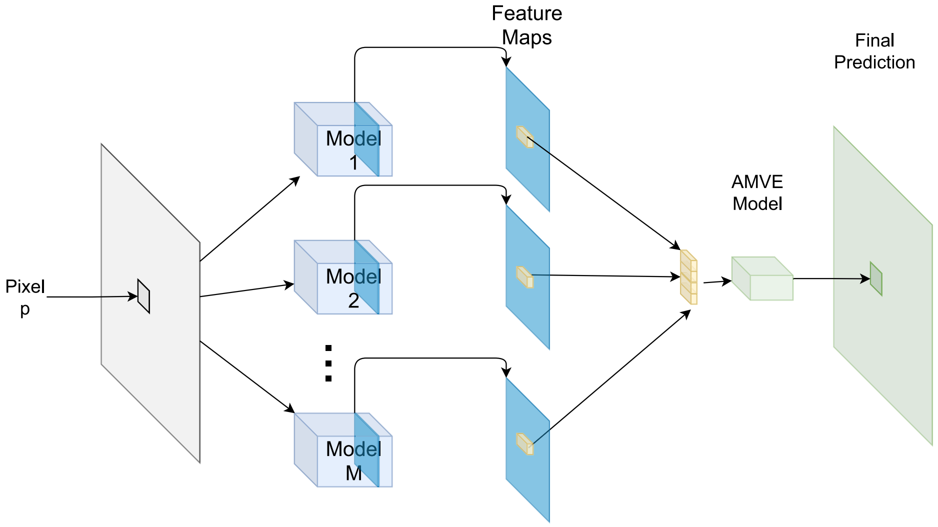


Fig. 10. Illustration of the AMVE approach. In light blue, the models we want to aggregate. In dark blue, the feature map used as input to the AMVE model. In yellow, the features of previous models used by AMVE to infer the final pixel prediction. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- In the Stacking Ensemble approach, we can assume that each pixel has an associated vector. In AMVE approach, we need to decide which layer of each model will be taken as inputs, so it is less generalizable. Also, the internal structure of each model could hinder the application of this technique, for example, if the structure of a model is not based on layers.
- In the AMVE approach, the prediction is based on features. In the Stacking Ensemble approach, the prediction is based on probability vectors V which have as constraints that $V_j \geq 0$ and $\sum V_j = 1$, so are able to contain less information. Due to those restrictions, the learned knowledge may be poorer.
- The final prediction of a pixel is always a probability vector of the same size. In an AMVE, each model could have a different amount of features. This causes models with more features to have more presence in the meta-learner's training, biasing its behavior.

4.4. Semantic Segmentation Ensembles

We introduce the Semantic Segmentation Ensemble (SSE) based not on pixel-wise prediction but on the prediction of windows. This approach is based on the idea that predictions of nearby pixels are related to each other. Therefore, using ensembles whose input is a window of predictions instead of just pixels should make more informed predictions.

Based on the above definitions, the SSE approximation can be formulated as a parametric function f_θ which takes as input the windows of predictions $\hat{y}^{(i)}_{W_{p,k}}$ to predict the centered pixel prediction y_p and is solution of the following minimization problem, which is similar to Eq. (2):

$$\arg\min_{\theta} \sum_{p \in P} \mathcal{L}\left(y_p, f_\theta\left(\hat{y}^{(1)}_{W_{p,k}}, \ldots, \hat{y}^{(M)}_{W_{p,k}}\right)\right) . \tag{4}$$

Fig. 11 shows a diagram to illustrate the SSE approach. For this approach, each base model processes the image (light blue). To apply the SSE approach, we use not only the pixel prediction (yellow) but the window prediction centered on its prediction (purple). The final prediction based on the windows is used to predict a single pixel.

Once the proposal has been described, we can comment the following points:

- We can notice that the use of windows as input for the SSE method is easy to generalize to feature maps. This consideration allows us to combine AMVE and SSE by using windows of features instead of single features.
- When $k = 1$, this strategy is essentially equivalent to the Stacking or the AMVE approach. Therefore, this is a generalization of the previous ensembles.
- We can also appreciate that windows on the borders are not defined when $k > 1$. On Fig. 11 the red window on the top left represents this problem. On these cases, a strategy to fill the empty predictions of the window is needed. To solve this, we fill the empty spaces using zero padding.

5. Experiments, results and discussion

Many semantic segmentation methods exist, but approaches dedicated to tackling micrographs are limited. The objective of this experimentation is to provide the reader with a better intuition on how the most relevant proposals perform at this task, and to extract some knowledge about the possible obstacles and shortcomings that may appear when dealing with this kind of images.

This section is organized as follows: the details about parameters and configuration are first explained, along with an explanation about the loss functions tested. Next, the results of each category of methods are shown and analyzed: baselines together with unsupervised techniques, a semi-supervised method, and fully supervised methods. A discussion drawing conclusions from these results is provided at the end.

5.1. Experimental setup

This subsection details the general experimental framework of our experimentation. We describe the validation protocol carried out, list the models and libraries used, the most relevant hyperparameters of the models and the choice of their values, and the performance metrics we used for the evaluation.

Validation strategy. The validation protocol adopted is Cross-Validation. Despite the additional computational cost compared to a more classical and simple approach such as Hold-Out, with cross-validation we obtain more robust results. In order to obtain comparable results with the experimentation carried out in [45] on the UHCS dataset, we performed a 6-fold Cross Validation, but unlike this study, we use a validation set (10% random of the training subset) independent of the test subset to choose when to stop the training process, so that our models in each fold are not over-fitted to the test subset, with which we calculate the performance metrics.

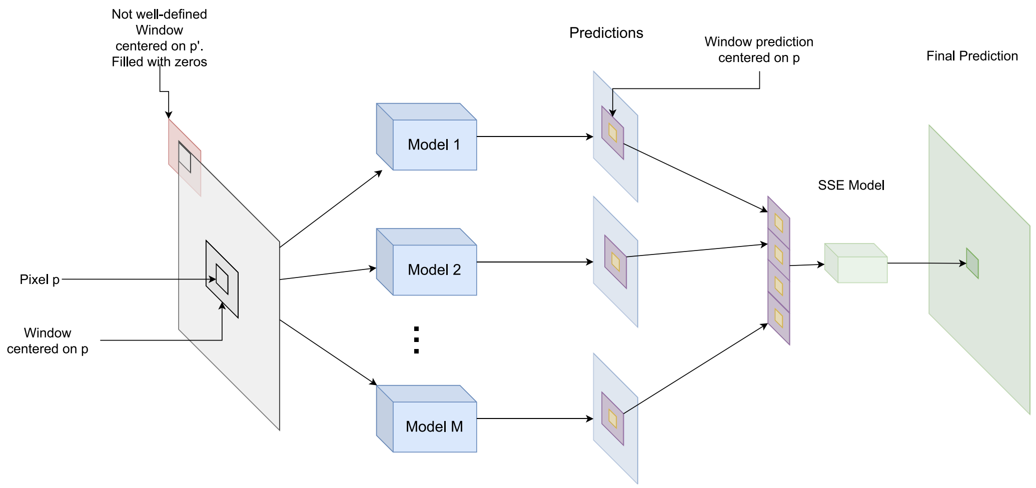


Fig. 11. Illustration of the SSE approach. In blue, the models stacked and their predictions. In yellow, the pixel p prediction. On purple, the window prediction centered on the pixel p prediction used by the semantic segmentation stacking model to predict the class of a certain pixel. On light red, a window centered on one of the borders pixels which is filled with zeros. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Performance metrics. The performance metrics used on this paper are accuracy and intersection over union (IoU). Accuracy is a general metric that indicates the overall ratio of correct predictions. However, in problems with class imbalance, it is not a good indicator of the performance of a model. In contrast, the IoU metric is specialized for segmentation problems and provides a more class-informed measure. We will use the mean IoU, averaging over all classes, as the final performance metric.

Data augmentation. Data augmentation [121] is a strategy to increase the level of labeled samples in a dataset by applying transformations to the existing, real samples, which do not alter their validity. For the purposes of this experimentation, data augmentation was relevant since datasets are usually small. The transformations used were: vertical flip, horizontal flip and vertical+horizontal flip (the labels were transformed accordingly). This generates 3 new images from each original one, allowing to work with 4 times as much images as before. There exist other transformations such as translations or deformations but they were not considered useful for this use case, so they were not applied.

Stochastic weight averaging. Stochastic Weight Averaging (SWA) [122] is a regularization mechanism which prevents overfitting by means of an aggregation of weights along different epochs during the training process of a neural network. This discards the need for an internal validation subset where the network does not train. SWA was shown to notably improve performance in semi-supervised scenarios [123]. We have thus tested it together with the semi-supervised method USSS. Nonetheless, some tests have been performed with the best-performing supervised models as well so as to measure its potential across different scenarios. However, we obtained identical results in supervised models using SWA and not using it so in the following we show only the results without using SWA.

Transfer learning. Although most experiments were carried out using models that were only pretrained over the well-known Imagenet dataset as a starting point for their weights, we also consider the use case where one may be able to obtain a publicly available micrograph dataset such as UHCS or MetalDAM, and transfer the knowledge extracted from that data onto a different dataset from the same field. We observed that, when pretraining a model over the UHCS dataset and then fine-tuning it with images from the MetalDAM dataset, the loss function starts already at a low value and converges relatively quickly. Meanwhile, a model with only the generic weights from Imagenet will take almost double the epochs to achieve similar loss values, as can be seen in Fig. 12.

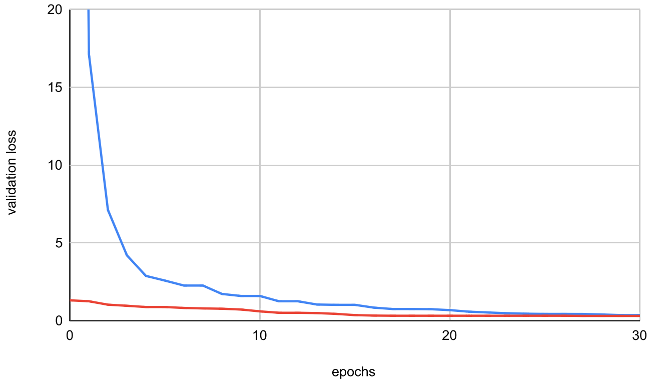


Fig. 12. Convergence of a model pretrained over UHCS (red) vs. a model initialized using Imagenet weights (blue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Soft labeling. Another approach we used in a previous experimentation is the use of soft labeling. This approach consists in smoothing the labels so that the model understands the uncertainty of the boundaries in images. Our approach uses Gaussian blur on each label so that the boundary labels are fuzzier than labels inside segments. However, this approach did not yield promising results so this idea was rejected on the work.

Loss functions. There exist several loss functions that can be used to optimize the parameters of a neural network. In general, one may choose the loss that fits best for the concrete task, but sometimes there can be more than one that makes sense. Jadon [124] describes different types of loss functions that can be used for the semantic segmentation task. Throughout this work, we have used cross-entropy loss, focal loss, Dice loss and Jaccard loss. The first two losses are distribution-based, and the remaining two are region-based.

In addition to these loss functions, the nature of the data suggests to add cohesion between the labels of each image. Kim et al. [125] propose a regularization term that imposes continuity on the nearest labels. In our experiments, we analyze the change in performance of our models when we add this term onto our loss function.

Our experimentation is performed on the two datasets described in Section 2.3, and it can be divided into four groups:

- First, the method used as a baseline (Multi-Otsu [75]) and the unsupervised method (k-means) are included in the same experimental group, due to their similarities [106]. Both methods are used individually, and in combination with a superpixel algorithm [97]. All these methods can be found implemented in widely used libraries in the field of computer vision, such as OpenCV [126] and scikit-image [127]. It is important to highlight the fact that these techniques do not have the capacity to classify the segmented regions into the different classes, due to their unsupervised nature. In order to obtain comparable results to the rest of the experimentation, we perform the mapping between segmented regions and labels that maximizes the mean IoU. In both methods it is necessary to specify the desired number of clusters. This parameter is set equal to the number of classes in the dataset.
- Secondly, the experiments performed using semi-supervised methods are shown. Different parameters of the USSS methodology are adjusted and all results are detailed, in order to analyze which kinds of behavior influence the performance of the model. The base hyperparameters are kept unchanged after preliminary experimentations: the backbone chosen is DRN-D-22 [128] pre-trained over ImageNet (due to USSS being implemented on top of DRN and the PyTorch library[3]) and the default parameters from the original proposal are maintained, using a SGD optimizer with learning rate of 0.001, 0.9 momentum and 10^{-4} weight decay. It has been trained over 200 epochs using a batch size of 4 images.
- Thirdly, the experimentation with the supervised deep-learning based models introduced in Section 3.3 is presented. Most of these models can be found implemented in the Segmentation Models PyTorch library [129], on which our experimental framework is based. Attempts have been made to optimize hyperparameters for these models with the Optuna software [130], but no satisfactory results have been obtained due to the long training time required for these models, and the high variability of results at an early stage of training, where the models have not yet converged. Because of this, we have opted for a base hyperparameter configuration for all the supervised models. These models have been trained for 200 epochs, with a batch size of 4. The optimization algorithm used is Adam, with a learning rate of 0.001, β_1 of 0.9 and β_2 of 0.999. EfficientNetB0 pre-trained on the ImageNet dataset [131] has been used as a backbone. We choose to train all models with the cross-entropy. The best performing models have also been trained with the focal, dice and jaccard loss functions, with and without the continuity regularization term.
- Lastly, the experimentation with the supervised ensemble proposed on Section 4 is presented. The models have been built on top of the PyTorch framework. Each model is composed by a convolutional layer, a dropout layer with rate 0.3 and finally a softmax activation function. If the ensemble model is Stacking, it takes as input the output of the models aggregated and is called "ST" in the result tables. If the ensemble model is AMVE, it takes as input the output of the last feature map of the models aggregated and is named "AMVE". To implement the SSE approximation shown in Section 4.4, we choose $k = 1$ and $k = 3$ for the kernel size of the convolutional layer and they will add "K1" or "K3" to the name of the model respectively. As we previously noted on the final considerations of Section 4.4, when "K1" is added, the method is equivalent to the original method, Stacking or AMVE approach. Most of these models have been trained for 200 epochs, except for some of them that have needed a few more epochs to converge, with a batch size of 4. The optimization algorithm used is Adam, with a learning rate of 0.001, β_1 of 0.9 and β_2 of 0.999.

Table 4
Results from the selected baselines over the UHCS and MetalDAM datasets.

Dataset	Model	ACC	IoU 0	IoU 1	IoU 2	IoU 3	Mean IoU
UHCS	Multi-Otsu	42.78	14.26	**20.30**	36.6	8.49	19.91
	Multi-Otsu+Superpixel	44.89	**14.43**	20.15	39.38	9.72	**20.92**
	k-means	42.47	13.57	19.99	36.31	8.48	19.59
	k-means+Superpixel	**44.96**	13.88	19.85	**39.58**	**9.79**	20.77
Dataset	Model	ACC	IoU 0	IoU 1	IoU 2	IoU 4	Mean IoU
MetalDAM	Multi-Otsu	67.21	**26.92**	45.24	12.18	**26.97**	27.83
	Multi-Otsu+Superpixel	63.17	25.82	37.64	**12.98**	23.82	25.07
	k-means	**67.28**	26.79	**45.66**	12.50	26.46	**27.85**
	k-means+Superpixel	63.24	26.18	37.41	12.15	25.82	25.39

Table 5
Basic USSS models with full datasets.

Dataset	Model	α	β	SWA	ACC	IoU 0	IoU 1	IoU 2	IoU 3	Mean IoU
UHCS	USSS	0	0	No	86.16	**45.94**	69.39	**88.71**	21.70	56.44
	USSS	0	0	Yes	**87.34**	44.33	**74.74**	88.60	**28.02**	**58.92**
Dataset	Model	α	β	SWA	ACC	IoU 0	IoU 1	IoU 2	IoU 4	Mean IoU
MetalDAM	USSS	0	0	No	78.04	59.44	74.05	18.04	**33.77**	**46.33**
	USSS	0	1	No	76.61	54.02	72.58	**25.34**	32.33	46.07
	USSS	0	0	Yes	78.32	58.38	74.48	20.89	29.31	45.77
	USSS	0	1	Yes	**78.77**	**59.55**	**75.02**	21.49	23.61	44.92

5.2. Results

In the following, all experiments performed with the different methodologies over both datasets are detailed and analyzed, along with a comparison among them. The source code developed throughout the experimentation process is available online.[4]

5.2.1. Baselines and unsupervised methods

The results obtained with image processing techniques and unsupervised methods are presented in this first section as baselines in our study. Due to their simplicity and the complexity of the metallographic images, these methods are not expected to have a good performance, but they are perfect candidates to establish the baseline results in order to compare and analyze the results obtained with much more sophisticated and computationally expensive methods in the following sections. Results from the baseline and unsupervised methods over the UHCS dataset and over the MetalDAM dataset are presented in Table 4.

5.2.2. Semi-supervised methods

The USSS technique can be trained over one or several datasets, with or without unlabeled examples. In the following, several results using the selected datasets, both independently and together, are discussed.

Single domain models. In this case, the portion of the UHCS dataset used does not include unlabeled images, since it consists of images with similar characteristics and same magnification. Consequently, we use UHCS as a starting point to observe the performance of the USSS implementation in supervised mode, in order to compare against the rest of methods and its own performance with the MetalDAM dataset. The results in Table 5, therefore, correspond to runs of USSS with the same 6-fold cross validation scheme and without the unsupervised loss functions enabled ($\alpha = 0$ and $\beta = 0$).

In the case of the MetalDAM dataset, USSS was able to train with unlabeled as well as labeled images. This allows to perform more combinations of experiments and increases the utility of the available parameters. More specifically, parameter β now controls the activation of the within-dataset unsupervised loss function.

[3] https://pytorch.org/.

[4] https://github.com/ari-dasci/S-metallograph-segmentation (available when paper is accepted).

Table 6
Mean IoU results of USSS over the UHCS dataset, removing the labels for a certain proportion of images and using them as unlabeled samples (first row) or not providing them to the model at all (second row).

% labeled # unlabeled	100%	75%	50%	25%
Rest	**57.16**	42.05	40.55	46.70
None		50.15	48.82	37.26

Table 7
Mean IoU results of USSS over the MetalDAM dataset, incorporating more unlabeled samples than labeled (first row), as many as labeled ones (second row) or none at all (third row). The multipliers in the first row indicate the number of times the unlabeled subset is larger than the labeled one.

% labeled # unlabeled	100%	50%	25%
More	(2x) **47.78**	(2x) 35.84	(4x) 26.77
Same	45.56	35.72	28.05
None	45.47	33.33	24.56

Table 8
USSS multi-domain models evaluated over UHCS and MetalDAM.

Dataset	Without SWA			With SWA		
	α	0	1		0	1
	β			β		
UHCS	0	51.51	51.17	0	**53.24**	50.73
	1	52.10	50.70	1	48.69	49.69
MetalDAM	0	45.48	39.34	0	55.03	54.83
	1	39.20	35.12	1	**55.06**	54.45

Ratio of unlabeled images. Since the main purpose of applying a semi-supervised approach is to add more knowledge from unlabeled samples, several tests have been performed in order to analyze whether these instances are actually helping the model, as well as the ratio of unlabeled to labeled images that appears to be most beneficial. It is important to notice that these tests were performed using a fixed random selection of images for the labeled and unlabeled subsets, so as to ensure that every version of the model was working under the same conditions. Table 6 shows the results related to the UHCS dataset and Table 7 does the same for the MetalDAM dataset.

In the case of the UHCS dataset, only the images with labels available have been used, in order not to mix various zoom levels and images with very different characteristics. As a result, the percentages in Table 6 are obtained over the total of 24 training images. For each case, one experiment has been carried out using the portion of unused images as unlabeled images and another one providing no unlabeled samples to the model.

Multi-domain models. One of our main interests in a semi-supervised multi-domain model was to analyze whether semantic segmentation, and specially microstructure segmentation, can benefit from unlabeled images and from a different but similar domain of images. This experiment is thus dedicated to models trained with both available datasets, UHCS and MetalDAM, at the same time.

As was explained before, USSS has two boolean hyperparameters which control the learning from unlabeled instances, α and β. When enabling β, each domain receives feedback from its own unlabeled samples (in this case, only MetalDAM dataset), and when α is enabled, each domain receives feedback from the rest of unlabeled images. Table 8 shows the results for these multi-domain models.

5.2.3. Supervised methods

In this section, results obtained with the set of supervised deep learning models presented in Section 3.3.3 applied in the selected datasets, with different loss functions, are shown and analyzed.

Table 9
Metrics of the main supervised architectures trained over UHCS and MetalDAM datasets.

Dataset	Model	ACC	IoU 0	IoU 1	IoU 2	IoU 3	Mean IoU
UHCS	DeepLabv3	87.76	49.40	76.14	88.65	34.43	62.15
	DeepLabv3+	90.18	58.23	83.00	90.16	47.36	69.69
	FPN	89.44	57.51	78.42	90.19	45.87	68.00
	Linknet	88.70	56.99	81.45	88.21	44.34	67.75
	PAN	88.40	55.59	72.31	89.65	41.15	64.68
	PSPNet	88.20	53.17	75.32	89.80	35.01	63.33
	Unet	**91.15**	61.89	84.51	**90.93**	52.48	72.45
	Unet++	**91.15**	**62.92**	84.58	90.83	**54.16**	**73.12**
	PixelNet	90.77	62.22	**86.69**	89.96	44.29	70.79
Dataset	Model	ACC	IoU 0	IoU 1	IoU 2	IoU 4	Mean IoU
MetalDAM	DeepLabv3	81.03	63.74	76.94	35.23	50.87	56.69
	DeepLabv3+	84.86	70.56	83.04	**39.27**	52.65	61.38
	FPN	85.29	70.36	83.21	38.24	**53.16**	61.24
	Linknet	84.69	72.03	83.20	33.96	32.27	55.36
	PAN	84.44	69.79	82.04	35.21	45.94	58.24
	PSPNet	79.09	60.39	74.12	35.80	43.46	53.44
	Unet	86.33	73.20	85.46	33.59	47.56	59.95
	Unet++	**87.04**	**74.28**	**86.13**	36.14	49.43	**61.49**
	PixelNet	84.07	70.13	82.75	16.21	36.55	51.41

Table 10
Mean IoU when applying different loss functions on the main supervised methods over UHCS and MetalDAM dataset.

Dataset	Model	CCE	Focal	Dice	Jaccard
UHCS	DeepLabV3+	70.25	69.7	70.26	71.31
	FPN	68.65	68.35	70.75	70.92
	Unet	72.9	70.86	73.91	74.39
	Unet++	73.93	70.66	**75.03**	74.6
MetalDAM	DeepLabV3+	61.38	59.68	59.50	61.37
	FPN	63.45	54.38	59.80	58.96
	Unet	60.51	61.67	60.20	61.00
	Unet++	61.75	60.02	60.54	**66.11**

Base models. First, a set of experiments is carried out with the different models in both datasets. In order to have a first comparison of the performance of the models in each of the datasets, all these models are trained with the same loss function. The categorical cross entropy loss function is selected for this first supervised experimentation. The results obtained are shown in Table 9 for the UHCS and MetalDAM datasets.

Loss functions and continuous regularizer term. Secondly, the models that showed the most promising performance in the previous experimentation, in general on the two datasets, are used in this second experimentation, in this case, with the different loss functions and with the continuity regularization term.

A summary of the results obtained with the different loss functions are shown in Table 10, for the UHCS and MetalDAM datasets, and the complete results with accuracy and the IoU metric for each class can be found in Appendix (Tables A.16 and A.19). These results show an improvement in the performance of some models with some of the loss functions, compared to the initial experimentation with the categorical cross-entropy (CCE) loss function.

5.2.4. Proposed ensembles

In this section, results obtained with ensembles proposed in Section 4 applied in the selected datasets with different configurations are shown and analyzed. This experimentation is made up of ensembles of the two and three best supervised models of the experimentation shown in the previous subsection, both with the Stacking and Multiview ensemble strategy, and with kernel size 1 and 3.

In the UHCS dataset, the selected base models are Unet++ with Dice Loss function, Unet++ with Jaccard Loss function and Unet with Jaccard loss. In the MetalDAM dataset, the selected base models are Unet++ with Jaccard loss, Unet++ with categorical cross entropy loss,

Table 11
Mean IoU performance of the proposed ensemble models on the UHCS and MetalDAM datasets.

Dataset	Ensemble	K1	K3
UHCS	Stacking 2 Best	76.08	75.91
	Stacking 3 Best	73.48	76.34
	AMVE 2 Best	76.12	76.30
	AMVE 3 Best	**76.71**	68.65
MetalDAM	Stacking 2 Best	67.47	53.17
	Stacking 3 Best	**67.77**	59.00
	AMVE 2 Best	61.85	52.86
	AMVE 3 Best	64.44	48.39

and FPN with categorical cross entropy loss. On Table 11 we show the performance of assembling those models with our proposals. The complete results with accuracy and the IoU metric for each class can be found in Appendix (Tables A.17 and A.18)

Fig. 13 shows graphically the performance of our models on an example of metallograph from the UHCS dataset. In particular, the prediction obtained with the best model in this dataset is shown. The example presented in Fig. 13(c) shows the robustness of our model against the problem of confusing the spheroidite particles class (red) with the ferritic matrix class (blue), a common problem in this dataset as explained in Section 2.3.1. However, the opposite case does occur. It can be seen how areas of ferritic matrix are predicted as spheroidite particles, especially at the edges of these regions. The ambiguity and imprecision present in the manual annotation process of this type of datasets may be one of the main causes of the failures obtained at the edges of the regions. Some areas of cementite network (yellow) are also confused by our models with spheroidite particles or ferritic matrix. A clear example of this can be seen in the lower right part of Fig. 13(c). In general, our models tend to generate false positives from the spheroidite particles class, because it is a largely majority class in this dataset. Finally, as expected, the models show more failures in the Widmanstätten laths class (green) than in the rest of the classes due to its large inferiority and fragmented shapes.

Fig. 14 shows graphically the performance of our models on an example of metallograph from the MetalDAM dataset. In particular, the prediction obtained with the best model in this dataset is shown. Our model presents a good performance in the segmentation of classes matrix (blue) and austenite (yellow), with the exception of some false positives obtained due to their majority. Regarding class defect (black), it is important to highlight its importance due to its nature, which makes the false negatives obtained for this class critical, as explained in Section 2.3.2. In addition, the numerical inferiority and the presence in few metallographs can complicate the segmentation task for this class. Nevertheless, it is the class that presents a greater difference in intensities with respect to the rest of the classes as can be seen in Fig. 14(a), which helps to obtain a good segmentation for this class. Finally, our models show to have difficulties in segmenting class martensite/austenite, obtaining a poor performance and a remarkable amount of false negatives. The fact that it is a minority class and the poor contrast that this class presents with respect to the majority classes make it the most difficult class to segment in the MetalDAM dataset.

5.3. Discussion

From the results presented above, it is noticeable that annotated images are fundamental for obtaining sufficiently performant segmentation models, in the metallography context.

Unsupervised methods. Unsupervised methods can only achieve around 20%–22% mean IoU, which may be useful if predictions are used as pre-annotations but is not enough to be used as standalone segmenters. In the UHCS dataset, the combination of Multi-Otsu with superpixels achieves the best performance within this category, while k-means by itself is the best option in the MetalDAM dataset. The results obtained with the Multi-Otsu and k-means methods are very similar, as expected due to their theoretical similarities [106]. On the other hand, the application of these methods in combination with superpixels presents differences in terms of results with respect to the individual application of each of the methods. It can be clearly seen that in the UHCS dataset, the use of superpixel provides an increase in the quality of the results, while in the MetalDAM dataset the opposite occurs, the use of Superpixel decreases the quality of the results. Therefore, in this specific domain of metallography, it seems that the use of superpixel algorithms as a previous step to the application of the segmentation algorithm, is more or less opportune depending on the dataset.

As expected, these results indicate poor performance of the baseline and unsupervised methods in both datasets. Section 3.2 discusses difficulties that these types of methods can have to handle complex images, and here it is empirically verified. These datasets do not have the presence of all the classes in all the images, which increases the difficulty to segment them satisfactorily, in particular, with these methods of an unsupervised nature, which do not have the ability to segment each image into an appropriate number of classes, they need the desired number of segmentation levels to be set in advance, so these methods segment classes not present in several images.

Semi-supervised methods. The semi-supervised method tested was not able to take much advantage of the extra information provided by unlabeled samples, but an improvement was seen when modeling MetalDAM using a multi-domain approach in combination with UHCS.

It is straightforward to notice that USSS benefited from a certain improvement in performance for the UHCS dataset thanks to SWA. This may be due to the learning process becoming more stable as well as to being able to use the whole training partition in each iteration, since no internal validation is needed for weight selection in the network. However, SWA did not help the model much when it was trained for MetalDAM. Although USSS underperforms in both datasets, it has more difficulties with the latter, being able to mostly predict adequately majority labels 0 and 1, and falling short in the rest of labels. This is expected but, as a result, it causes poor predictions of the most relevant class, class 2.

The results about the ratio of unlabeled images suggest that having more images can be beneficial in general, since the best result is obtained when having the full labeled dataset. In the case of UHCS, if there are less labeled images (12 to 18 images), the model appears to work better without unlabeled samples unless the amount of labeled data is very low (6 images in the 25% experiment), in which case the performance improves thanks to the unlabeled images. This may indicate that the real usefulness of this kind of semi-supervised methods comes only when labels are really scarce. With respect to the MetalDAM dataset, these experiments show a direct relation between the amount of selected labeled images and the performance of the model: the best performing model was the one with the maximum number of images (42 labeled and 84 unlabeled), followed by the rest of models using all available labeled images. Among the models trained with half the labeled images, there were very small differences between the ones using unlabeled samples, which were superior to the one that did not have them. A similar situation can be observed when using a quarter of the labeled images. The increase in performance provided by the unlabeled samples does not compensate the scarcity of labeled ones in general. It is also fundamental to note that minority classes suffer more than majority classes with the decrease in images, something that needs to be taken into account in MetalDAM, where the interesting class is a minority one (MA). This can be better observed in the Appendix A.1.

As to the multi-domain models, there is only small variability in most of the cases with the UHCS dataset. The best model for it is essentially a supervised one using SWA, but it achieves lower metrics than its single-model counterpart. MetalDAM, for its part, is best modeled when using a slightly different configuration, with both the

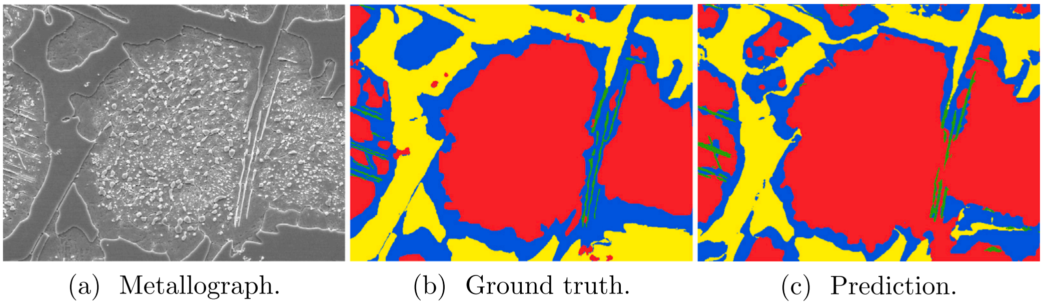


Fig. 13. Example of prediction on the UHCS dataset with the best model obtained AMVE 3 Best K = 1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

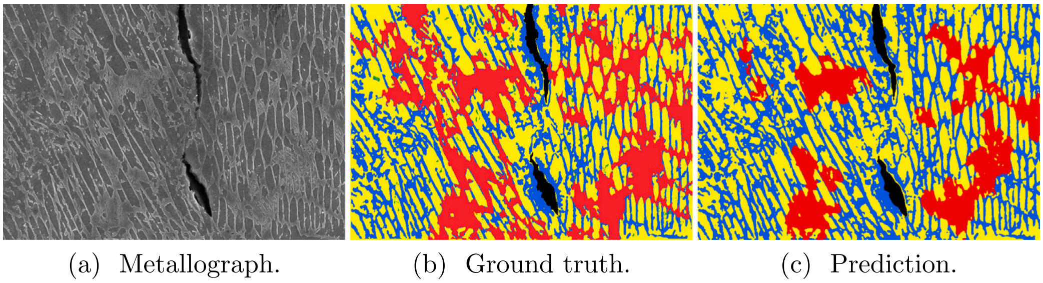


Fig. 14. Prediction example on the MetalDAM dataset with the best model obtained Stacking 3 Best K = 1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

β parameter and SWA enabled, and does get better results than the single-model version. This may indicate that there exists a certain trade-off between optimizing for the first dataset and for the second, which could mean that the domains are sufficiently different so that learning from each other cannot help the model be superior for both of them at the same time. Taking into account that unlabeled images were only provided for MetalDAM in these experiments, it could make sense that the best UHCS model does not benefit from within-dataset unsupervised information. In fact, the semi-supervised aspect seems not to be really useful for modeling UHCS better than the supervised version and does not affect metrics over MetalDAM much either, being SWA the factor that improves the most in this case.

Supervised methods. For their part, fully supervised methods did achieve better results with varying degrees of success. Unet++ shows the best performance on both datasets, so it seems to be a robust model to address the image segmentation problem in this specific domain. This better behavior could be explained by the encoding–decoding neural network design. The results obtained in general by the supervised models show a notable improvement with respect to the baselines and the semi-supervised methods.

It is important to highlight the results obtained with PixelNet and compare them with the rest of the models used, since this model is considered the state of the art in supervised segmentation of the UHCS dataset [45]. In the proposed experimentation, PixelNet is outperformed in both datasets by several of the methods used. In the UHCS dataset, PixelNet has a good performance, achieving third place in mean IoU with respect to the rest of the methods, however, in the MetalDAM dataset it suffers a notable performance reduction, being in the last position. In contrast, Unet++ has achieved the best results in both datasets, thus proving to be a robust and effective model in the metallographic domain.

With respect to the different loss functions, the best results in the UHCS dataset are obtained by the Unet++ model with the Dice loss function, improving the results of the initial experimentation. In the same way, Unet++ with Jaccard loss function improve the previous results obtained with CCE loss function in the MetalDAM dataset. Next, experimentation with the continuous regularizer term in combination with the different loss functions was carried out, however, a significant drop in the performance of the models was obtained. These results can be found in Appendix (Tables A.16 and A.19). The results seem to indicate that the use of the continuous regularizer term is not appropriate for this type of images. This unsupervised regularization encourages the classification under the same label for the pixels that are spatially continuous. In metallographic images, where there are many small and fragmented areas, this can be a disadvantage since the continuous regularizer term seems to confuse the model and join different spatially proximal areas of different classes.

Ensembles. Some of the newly proposed ensembles of the best methods notably outperformed the rest. The results show a better performance of the Multiview strategy in the UHCS dataset, with the model composed of the three best base models and a kernel size of 3. On the contrary, in the MetalDAM dataset, the Stacking strategy shows a better performance, specifically, the stacking model formed by the three best base models and a kernel size of 1.

This type of ensemble strategy is applied in this problem in order to take advantage of the diversity present in the set of base classifiers and to obtain an improvement in the performance of each one of them. The results show that the assembly models proposed in this paper have the ability to handle this advantage, and obtain the best results, in each of the datasets, of all the models applied in this experimental study. It should also be noted that the SSE methodology obtains good performance in most cases even if it does not achieve the absolute best result. For those reasons, we cannot draw direct conclusions on which ensembling strategy is best for microstructures.

Overall, the MetalDAM dataset has shown to be more complex and difficult to model than UHCS, a fact that encourages further research into these kinds of methods so as to continue improving performance in segmentation. Although many unlabeled images were available, the USSS method was unable to achieve competitive performance in spite of its flexibility and ability to learn from both datasets, which may indicate that other approaches such as pseudo-labeling may be more suitable for the purposes of microstructure segmentation.

6. Limitations, difficulties and challenges

The segmentation of microstructures in metallographic images is a very challenging task. This is caused by the inherent nature of these images, some of whose main visual characteristics are summarized below:

- high-resolution;
- extreme variability in terms of textures and shapes;
- highly fragmented images with fuzzy boundaries (that hinder the precise localization of each region);
- largely imbalanced classes (where some of them are commonly present, while others are practically residual);
- and absence of prior structural information (in opposition, for instance, to facial images, where one expects eyes, nose, mouth, and ears to be always located at certain relative positions).

These visual characteristics have severe consequences in the type of machine learning techniques that can be employed, as well as the segmentation strategies to follow, when tackling this problem present several limitations and difficulties:

- The manual annotation of these images is an expensive process, becoming really hard to construct a suitable pixel-wise ground truth for segmentation. As a consequence, there is a total absence of large datasets in the field, and the few existing available ones have very small size, not even reaching the hundred labeled images. This complicates the application of supervised techniques, since most state-of-the-art image segmentation methods are data hungry approaches, like deep networks.
- Due to the fact that the datasets are small, unlike many other computer vision problems (such as image classification or object detection), there are no deep pre-trained models for the segmentation of metallographic images, making it difficult to apply and improve existing learning-based techniques, and limiting the progress in the field.
- All existing methods are extremely *ad-hoc* and suffer from high generalization errors. A particular segmentation method will be effective in one single dataset, but it will perform very poorly on another one. Furthermore, it is usually necessary to apply specific pre- and post-processing techniques to improve the results obtained, and those that work in one dataset barely work well in another.

Finally, in correspondence to all aforementioned limitations, we consider the following research lines as the most promising challenges to fill the existing gaps in the field:

- Acquiring larger datasets and sharing them with the scientific community is a must, and would certainly accelerate progress in this field.
- The complex nature of these images has an impact on the evaluation of the segmentation quality. This comes to fruition in the fact that there is no clear metric or criterion to evaluate the results obtained, generally having a mismatch between the segmentation metric value and the visual quality of the segmented image. The search for a more reliable correspondence between quantitative and qualitative results represents one of the main future challenges.
- Another priority should be to speed up the annotation by human experts. The manual segmentation depends on the complexity of the images to be labeled, the knowledge and experience of the annotating expert, and the labeling tool employed. In this sense, semi-automatic, semi-supervised or weakly supervised tools able to provide sufficiently precise pre-annotations, shortening and facilitating the expert's work, would be very useful and appreciated by the scientific community in microstructural science. Surely, this would also have a direct impact on the availability of larger datasets.
- In the absence of large datasets, the way forward in metallographic image segmentation should be toward less reliance on annotations. According to this perspective, and since unsupervised methods seem to offer a quite limited performance in this problem, more research on semi-supervised approaches would be desirable as they appear as one of the most promising strategies to follow.
- There is a lack of pre-trained models to address metallographic image segmentation, and the efficacy of pre-training on images presenting relatively similar characteristics (such as histological or satellite images) has not been investigated yet. Taking advantage of the available knowledge on similar tasks is another relevant challenge, whose achievement would be of great benefit in tackling this problem.

Table A.12
USSS semi-supervised models with different unlabeled-to-labeled ratio over UHCS.

Model	Labeled	Unlabeled	ACC	IoU 0	IoU 1	IoU 2	IoU 3	Mean IoU
USSS	100% (24)	0	**87.18**	**47.84**	**73.47**	**89.17**	**18.16**	**57.16**
USSS	75% (18)	25% (6)	84.89	38.66	68.75	86.17	2.56	49.04
USSS	75% (18)	0	84.55	38.32	66.44	86.71	9.12	50.15
USSS	50% (12)	50% (12)	81.85	27.42	54.74	83.89	2.12	42.05
USSS	50% (12)	0	84.94	37.16	61.79	87.52	8.79	48.82
USSS	25% (6)	75% (18)	80.90	41.66	55.82	83.63	5.69	46.70
USSS	25% (6)	25% (6)	81.53	37.04	49.87	84.41	2.89	43.55
USSS	25% (6)	0	76.26	28.01	37.85	79.95	3.25	37.26

Table A.13
USSS semi-supervised models with different unlabeled-to-labeled ratio over MetalDAM.

Model	Labeled	Unlabeled	ACC	IoU 0	IoU 1	IoU 2	IoU 4	Mean IoU
USSS	100% (42)	2 × (84)	**79.59**	**61.42**	**75.53**	**25.49**	**28.70**	**47.78**
USSS	100% (42)	1 × (42)	78.51	58.41	74.73	22.62	26.46	45.56
USSS	100% (42)	0.5 × (21)	79.08	60.50	75.25	22.40	23.79	45.49
USSS	100% (42)	0	78.76	60.27	74.80	21.83	25.00	45.47
USSS	50% (21)	2 × (42)	71.11	42.06	68.12	13.07	0.00	35.84
USSS	50% (21)	1 × (21)	71.95	42.24	69.01	17.69	0.00	35.72
USSS	50% (21)	0	68.83	36.87	65.64	13.22	0.00	33.33
USSS	25% (10)	4 × (42)	62.56	18.12	60.25	6.42	0.00	26.77
USSS	25% (10)	1 × (10)	64.43	22.83	61.83	4.95	0.00	28.05
USSS	25% (10)	0	60.72	14.16	59.03	2.93	0.00	24.56

Table A.14
USSS multi-domain models evaluated over UHCS.

Model	α	β	SWA	ACC	IoU 0	IoU 1	IoU 2	IoU 3	Mean IoU
USSS	0	0	N	83.98	38.51	64.72	86.52	16.30	51.51
USSS	0	1	N	**85.40**	**42.44**	66.65	87.72	11.59	52.10
USSS	1	1	N	84.85	41.58	65.02	87.22	8.99	50.70
USSS	1	0	N	84.84	41.94	64.96	**87.45**	10.31	51.17
USSS	0	0	Y	84.87	33.79	**73.05**	85.21	**20.90**	**53.24**
USSS	0	1	Y	82.08	30.54	66.46	81.98	15.76	48.69
USSS	1	1	Y	83.12	29.85	66.71	83.42	18.78	49.69
USSS	1	0	Y	83.21	32.82	71.69	82.97	15.43	50.73

Table A.15
USSS multi-domain models evaluated over MetalDAM.

Model	α	β	SWA	ACC	IoU 0	IoU 1	IoU 2	IoU 4	Mean IoU
USSS	0	0	N	75.67	55.56	71.00	20.88	34.48	45.48
USSS	0	1	N	72.52	48.72	68.30	16.49	23.29	39.20
USSS	1	1	N	69.97	40.73	66.31	16.48	16.96	35.12
USSS	1	0	N	71.68	47.52	67.46	15.09	27.30	39.34
USSS	0	0	Y	80.26	61.95	**76.41**	28.29	**53.46**	55.03
USSS	0	1	Y	80.18	**62.07**	76.21	29.77	52.19	**55.06**
USSS	1	1	Y	**80.27**	61.92	76.10	**31.32**	48.47	54.45
USSS	1	0	Y	80.10	**62.07**	76.24	27.76	53.25	54.83

Table A.16
Extended results of base model with different loss functions and continuous regularizer term on the UHCS dataset.

Model	Loss	Continuity	ACC	IoU 0	IoU 1	IoU 2	IoU 3	Mean IoU
DeepLabV3+	CCE	No	90.22	59.77	81.39	90.62	48.52	70.07
		Yes	90.67	60.47	84.57	90.85	42.93	69.71
	Dice	No	90.36	59.15	82.80	90.44	48.66	70.26
		Yes	90.68	61.50	86.44	90.08	28.19	66.55
	Focal	No	90.06	59.62	80.72	90.40	48.08	69.70
		Yes	89.68	57.96	81.82	90.31	12.98	60.77
	Jaccard	No	90.32	60.06	81.15	90.58	50.21	70.50
		Yes	88.67	57.46	85.21	88.78	13.21	61.17
FPN	CCE	No	89.93	57.54	80.46	90.35	46.55	68.72
		Yes	90.92	61.62	85.17	90.91	35.09	68.20
	Dice	No	91.28	62.23	85.57	90.75	53.24	72.95
		Yes	90.10	58.72	84.71	89.72	33.81	66.74
	Focal	No	90.64	59.68	83.17	90.85	45.80	69.88
		Yes	88.68	55.55	76.89	89.80	18.02	60.07
	Jaccard	No	91.94	64.70	88.14	91.32	53.70	74.46
		Yes	91.00	61.65	86.04	90.63	37.87	69.05
Unet	CCE	No	90.80	61.39	83.88	90.55	53.20	72.25
		Yes	90.66	59.95	85.37	90.49	47.12	70.73
	Dice	No	92.45	67.43	86.37	92.38	58.42	76.15
		Yes	90.10	60.86	87.92	89.35	19.35	64.37
	Focal	No	91.35	63.90	84.48	91.34	53.02	73.19
		Yes	89.04	58.43	78.10	90.13	17.01	60.92
	Jaccard	No	91.79	66.21	82.91	92.18	58.04	74.83
		Yes	89.35	60.86	87.03	88.59	02.77	59.81
Unet++	CCE	No	91.59	63.83	84.59	91.65	53.90	73.49
		Yes	91.20	62.46	85.78	91.03	50.14	72.35
	Dice	No	92.51	66.49	88.47	91.87	59.01	76.46
		Yes	89.76	59.79	87.60	88.67	16.27	63.08
	Focal	No	91.48	63.21	86.96	91.04	55.17	74.10
		Yes	88.88	57.71	83.25	88.50	07.15	59.15
	Jaccard	No	92.01	64.02	88.43	91.34	57.51	75.33
		Yes	90.15	62.88	87.33	89.73	05.73	61.42

Table A.17
Extended results of the proposed ensemble models on the UHCS dataset.

Model	K	ACC	IoU 0	IoU 1	IoU 2	IoU 3	Mean IoU
Stacking 2-best	1	92.37	66.50	88.47	91.80	57.54	76.08
	3	92.29	66.01	88.51	91.68	57.43	75.91
Stacking 3-best	1	92.44	**66.73**	89.12	91.80	57.91	76.39
	3	92.45	66.69	89.25	91.80	57.60	76.34
AMVE 2-best	1	92.36	66.39	88.70	91.76	57.62	76.12
	3	92.43	66.62	88.90	91.81	57.99	76.33
AMVE 3-best	1	**92.49**	66.70	**89.40**	**91.84**	**58.91**	**76.71**
	3	91.84	65.17	89.49	91.37	28.55	68.65

Table A.18
Extended results of the proposed ensemble models on the MetalDAM dataset.

Model	K	ACC	IoU 0	IoU 1	IoU 2	IoU 4	Mean IoU
Stacking 2-best	1	87.53	74.90	86.27	39.67	**69.24**	67.52
	3	86.10	74.03	85.52	30.83	22.30	53.17
Stacking 3-best	1	**87.82**	75.46	**86.75**	40.56	68.34	**67.78**
	3	87.48	**75.83**	86.19	39.91	27.47	57.35
AMVE 2-best	1	87.11	74.51	85.79	40.00	43.00	60.83
	3	87.04	74.76	85.71	39.76	0.00	50.06
AMVE 3-best	1	87.38	74.97	86.22	**40.63**	53.79	63.90
	3	80.80	61.11	80.20	20.09	25.42	46.71

Table A.19
Extended results of base model with different loss functions and continuous regularizer term on the MetalDAM dataset.

Model	Loss	Continuity	ACC	IoU 0	IoU 1	IoU 2	IoU 4	Mean IoU
DeepLabV3+	CCE	No	84.86	70.56	83.04	**39.27**	52.65	61.38
		Si	84.28	68.16	81.86	37.80	42.93	57.69
	Dice	No	82.87	67.00	79.84	36.71	55.17	59.68
		Si	56.49	0.0	56.00	6.04	44.37	25.98
	Focal	No	85.44	70.73	83.63	36.70	43.39	59.50
		Si	63.23	22.68	58.81	28.17	51.79	39.93
	Jaccard	No	83.47	67.45	80.62	37.02	60.41	61.38
		Si	55.39	0.0	55.37	0.0	13.51	17.44
FPN	CCE	No	84.71	69.15	82.68	36.40	65.59	63.45
		Si	83.86	67.77	81.11	36.95	51.39	59.31
	Dice	No	83.00	68.01	79.98	36.73	32.82	54.38
		Si	56.30	0.0	55.91	8.83	27.44	23.04
	Focal	No	84.19	68.95	81.90	36.48	48.75	59.80
		Si	64.26	23.60	60.62	29.03	35.79	37.26
	Jaccard	No	84.29	68.91	82.25	35.98	48.73	58.97
		Si	56.71	0.0	56.14	9.77	41.60	26.27
Unet	CCE	No	86.28	72.87	85.43	36.55	47.21	60.51
		Si	84.71	71.40	83.03	36.21	43.76	58.60
	Dice	No	84.70	71.87	82.93	38.24	53.67	61.67
		Si	57.27	0.01	56.43	16.93	27.48	25.21
	Focal	No	86.41	73.00	85.56	36.65	47.50	60.20
		Si	57.65	0.99	57.02	19.80	37.34	28.76
	Jaccard	No	85.90	72.75	84.71	36.71	49.85	61.01
		Si	55.14	0.01	55.14	0.0	13.15	17.07
Unet++	CCE	No	**86.89**	**74.13**	**85.91**	38.67	48.31	61.75
		Si	86.84	73.54	85.46	38.46	46.99	61.11
	Dice	No	85.87	72.75	84.40	34.40	48.55	60.02
		Si	56.50	0.0	56.14	13.73	14.92	21.20
	Focal	No	86.36	73.53	84.66	37.23	43.47	60.54
		Si	59.73	11.77	58.58	6.04	45.24	30.41
	Jaccard	No	86.70	74.04	85.55	37.81	**67.05**	**66.11**
		Si	55.78	0.08	55.54	6.94	16.06	19.77

7. Conclusions

This tutorial shows that metallographic segmentation is a complex phenomenon with many potential approaches. Moreover, there exist many different techniques to address metallographic segmentation: our proposed taxonomy arranges them under two main typologies: image processing methods and learning-based approaches. In this paper, the state-of-the-art in metallographic segmentation methods is arranged under these categories, thus the reader can relate the existing approaches in an orderly manner. Moreover, we aim to provide a meaning for each category in a hypothetical practical scenario where, depending on the expert availability or the data source, different categories can be more suitable than others. The aggregation of different models is also proposed, exploiting the idea of similarity on nearby predictions in ensemble models.

The taxonomy is also coupled with a realistic experimental comparison, using two real-world datasets, one of them kindly provided by ArcelorMittal. These two datasets enable us to show how Image Processing Techniques are fast prototyping tools and can act as baseline results, whereas Learning-based approaches are needed to extract most information and knowledge from metallographic images, constituting an interesting start point for automated labeling that can be later refined by the expert in a shorter time. In particular, we deepen in the amount of human knowledge poured into the data and the impact such expert labeling will have in the performance attained by the actual models. From the latter, semi-supervised scenario, we may conclude that partial annotations can be used as guidelines for human experts, but the current semi-supervised techniques are far from being part of autonomous labeling procedures.

From this tutorial, we may also conclude that there are many open research questions related to metallographic segmentation and many avenues remain to be explored. The intrinsic characteristics of metallographic images pose a great challenge, both in the reduced size of datasets and the highly imbalanced label ratio. As a result, interesting pixels but from minor class labels are hard to emphasize for the models.

The imbalance among labels is worsened due to the lack of prior structural information, which cannot be exploited by the models and would allow generalized or pre-trained models. Thus, existing approaches are tailored to be very problem-dependent, making them poorly extensible to related segmentation tasks. We must also indicate that the lack of large, open datasets does not help to alleviate these problems: the availability of data would enable the generation of specific, pre-trained models that would yield better performance in the metallographic segmentation domain.

CRediT authorship contribution statement

Julián Luengo: Conceptualization, Methodology, Writing – original draft, Methodology, Writing – review & editing, Formal analysis, Visualization, Validation, Supervision. **Raúl Moreno:** Conceptualization, Methodology, Validation, Investigation, Resources, Writing – original draft, Writing – review & editing, Project administration. **Iván Sevillano:** Conceptualization, Methodology, Software, Formal analysis, Writing – original draft, Writing – review & editing, Visualization. **David Charte:** Conceptualization, Methodology, Software, Formal analysis, Writing – original draft, Writing – review & editing, Visualization. **Adrián Peláez-Vegas:** Conceptualization, Methodology, Software, Formal analysis, Writing – original draft, Writing – review & editing, Visualization. **Marta Fernández-Moreno:** Conceptualization, Methodology, Software, Formal analysis, Writing – original draft, Writing – review & editing, Visualization. **Pablo Mesejo:** Conceptualization, Methodology, Software, Formal analysis, Writing – original draft, Writing – review & editing , Visualization. **Francisco Herrera:** Validation, Supervision, Project administration, Funding acquisition, Resources.

Declaration of competing interest

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

Acknowledgments

This publication is supported by ArcelorMittal, Luxembourg Global R&D, specifically the project granted by ArcelorMittal Global R&D Digital Portfolio in collaboration with the Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada. This publication is supported by the Andalusian Excellence, Spain project P18-FR-4961 and SOMM17/6110/UGR. D. Charte is supported by the Spanish Ministry of Universities, Spain under the FPU program (Ref. FPU17/04069). Funding for open access charge: Universidad de Granada / CBUA.

Appendix. Results

A.1. Semi-supervised methods

See Tables A.12–A.15.

A.2. Supervised methods

See Tables A.16–A.19.

References

[1] Bo-hu Li, Bao-cun Hou, Wen-tao Yu, Xiao-bing Lu, Chun-wei Yang, Applications of artificial intelligence in intelligent manufacturing: a review, Front. Inf. Technol. Electron. Eng. 18 (1) (2017) 86–96.

[2] Ji Zhou, Peigen Li, Yanhong Zhou, Baicun Wang, Jiyuan Zang, Liu Meng, Toward new-generation intelligent manufacturing, Engineering 4 (1) (2018) 11–20.

[3] Ray Y. Zhong, Xun Xu, Eberhard Klotz, Stephen T. Newman, Intelligent manufacturing in the context of industry 4.0: a review, Engineering 3 (5) (2017) 616–630.

[4] Joon B. Park, Roderic S. Lakes, Characterization of Materials — I, Springer New York, New York, NY, 2007, pp. 41–81.

[5] Horacio Espinosa, Leonardo Pagnotta, Maria Pantano, Mechanical characterization of materials at small length scales, J. Mech. Sci. Technol. 26 (2012) 545–561.

[6] Yann LeCun, Yoshua Bengio, Geoffrey Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[7] Brian DeCost, Elizabeth Holm, A computer vision approach for automated analysis and classification of microstructural image data, Comput. Mater. Sci. 110 (2015) 126–133.

[8] Aritra Chowdhury, Elizabeth Kautz, Bülent Yener, Daniel Lewis, Image driven machine learning methods for microstructure recognition, Comput. Mater. Sci. 123 (2016) 176–187.

[9] Elizabeth A Holm, Ryan Cohn, Nan Gao, Andrew R Kitahara, Thomas P Matson, Bo Lei, Srujana Rao Yarasi, Overview: Computer vision and machine learning for microstructural characterization and analysis, Metall. Mater. Trans. A (2020) 1–15.

[10] Richard Szeliski, Computer Vision: Algorithms and Applications, Springer Science & Business Media, 2010.

[11] Claude R. Brice, Claude L. Fennema, Scene analysis using regions, Artificial Intelligence 1 (3–4) (1970) 205–226.

[12] Theodosios Pavlidis, Structural Pattern Recognition, Vol. 1, Springer, 1977.

[13] Edward M. Riseman, Michael A. Arbib, Computational techniques in the visual segmentation of static scenes, Comput. Graph. Image Process. 6 (3) (1977) 221–276.

[14] Dzung L. Pham, Chenyang Xu, Jerry L. Prince, Current methods in medical image segmentation, Annu. Rev. Biomed. Eng. 2 (1) (2000) 315–337.

[15] Mohammad Hesam Hesamian, Wenjing Jia, Xiangjian He, Paul Kennedy, Deep learning techniques for medical image segmentation: achievements and challenges, J. Digit. Imaging 32 (4) (2019) 582–596.

[16] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, Bernt Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3213–3223.

[17] Berthold Horn, Berthold Klaus, Paul Horn, Robot Vision, MIT Press, 1986.

[18] Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, Demetri Terzopoulos, Image segmentation using deep learning: A survey, IEEE Trans. Pattern Anal. Mach. Intell. (2021).

[19] Shijie Hao, Yuan Zhou, Yanrong Guo, A brief survey on semantic segmentation with deep learning, Neurocomputing 406 (2020) 302–321.

[20] Fahad Lateef, Yassine Ruichek, Survey on semantic segmentation using deep learning techniques, Neurocomputing 338 (2019) 321–348.

[21] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, Pablo Martinez-Gonzalez, Jose Garcia-Rodriguez, A survey on deep learning techniques for image and video semantic segmentation, Appl. Soft Comput. 70 (2018) 41–65.

[22] Idan Rosenthal, Adin Stern, Nachum Frage, Microstructure and mechanical properties of AlSi10Mg parts produced by the laser beam additive manufacturing (AM) technology, Metallogr. Microstruct. Anal. 3 (6) (2014) 448–453.

[23] LA Morales-Hernández, IR Terol-Villalobos, A Domínguez-González, F Manriquez-Guerrero, G Herrera-Ruiz, Spatial distribution and spheroidicity characterization of graphite nodules based on morphological tools, J. Mater Process. Technol. 210 (2) (2010) 335–342.

[24] Krzysztof Jan Kurzydlowski, Brian Ralph, The Quantitative Description of the Microstructure of Materials, Vol. 3, CRC Press, 1995.

[25] Thomas Hochrainer, Michael Zaiser, Peter Gumbsch, A three-dimensional continuum theory of dislocation systems: kinematics and mean-field formulation, Phil. Mag. 87 (8–9) (2007) 1261–1282.

[26] Wayne S. Rasband, et al., ImageJ, 1997.

[27] Mingchun Li, Dali Chen, Shixin Liu, Dinghao Guo, Online learning method based on support vector machine for metallographic image segmentation, Signal Image Video Process. (2020) 1–8.

[28] Kunihiko Fukushima, Sei Miyake, Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition, in: Competition and Cooperation in Neural Nets, Springer, 1982, pp. 267–285.

[29] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[30] Dan Ciregan, Ueli Meier, Jürgen Schmidhuber, Multi-column deep neural networks for image classification, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 3642–3649.

[31] Seyed Majid Azimi, Dominik Britz, Michael Engstler, Mario Fritz, Frank Mücklich, Advanced steel microstructural classification by deep learning methods, Sci. Rep. 8 (1) (2018) 1–14.

[32] Junmyoung Jang, Donghyun Van, Hyojin Jang, Dae Hyun Baik, Sang Duk Yoo, Jaewoong Park, Sungwook Mhin, Jyoti Mazumder, Seung Hwan Lee, Residual neural network-based fully convolutional network for microstructure segmentation, Sci. Technol. Weld. Join. 25 (4) (2020) 282–289.

[33] George F. Vander Voort, Gabriel M. Lucas, Elena P. Manilova, Metallography and microstructures of stainless steels and maraging steels[1], in: Metallography and Microstructures, ASM International, 2004.

[34] Xiaoxiao Li, Ali Ramazani, Ulrich Prahl, Wolfgang Bleck, Quantification of complex-phase steel microstructure by using combined EBSD and EPMA measurements, Mater. Charact. 142 (2018) 179–186, http://dx.doi.org/10.1016/j.matchar.2018.05.038, URL: https://www.sciencedirect.com/science/article/pii/S1044580317332357.

[35] George Vander Voort, Color metallography vol. 9 ASM handbook, Metallogr. Microstruct. (2004) 493–512.

[36] Michael Kampffmeyer, Arnt-Borre Salberg, Robert Jenssen, Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016, pp. 1–9.

[37] Ryan Hausen, Brant E. Robertson, Morpheus: A deep learning framework for the pixel-level analysis of astronomical image data, Astrophys. J. Suppl. Ser. 248 (1) (2020) 1–37.

[38] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, Antonio Torralba, Scene parsing through ADE20K dataset, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[39] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Scharwächter, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, Bernt Schiele, The cityscapes dataset, in: CVPR Workshop on the Future of Datasets in Vision, Vol. 2, 2015.

[40] Shiyue Zhang, Dali Chen, Shixin Liu, Pengyuan Zhang, Wei Zhao, Aluminum alloy microstructural segmentation method based on simple noniterative clustering and adaptive density-based spatial clustering of applications with noise, J. Electron. Imaging 28 (3) (2019) 033035.

[41] Brian DeCost, Matthew Hecht, Toby Francis, Bryan Webler, Yoosuf Picard, Elizabeth Holm, UHCSDB: Ultrahigh carbon steel micrograph database: Tools for exploring large heterogeneous microstructure datasets, Integr. Mater. Manuf. Innov. 6 (2017).

[42] Graham Roberts, Simon Y Haile, Rajat Sainju, Danny J Edwards, Brian Hutchinson, Yuanyuan Zhu, Deep learning for semantic segmentation of defects in advanced stem images of steels, Sci. Rep. 9 (1) (2019) 1–12.

[43] Dali Chen, Dingpeng Sun, Jun Fu, Shixin Liu, Semi-supervised learning framework for aluminum alloy metallographic image segmentation, IEEE Access 9 (2021) 30858–30867.

[44] Wenyuan Cui, Yunlu Zhang, Xinchang Zhang, Lan Li, Frank Liou, Metal additive manufacturing parts inspection using convolutional neural network, Appl. Sci. 10 (2020) 545.

[45] B.L. DeCost, T. Francis, E. Holm, High throughput quantitative metallography for complex microstructures using deep learning: A case study in ultrahigh carbon steel, Microsc. Microanal. 25 (1) (2019) 21–29.

[46] Asako Kanezaki, Unsupervised image segmentation by backpropagation, in: Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, 2018, pp. 1543–1547.

[47] Hoheok Kim, Junya Inoue, Tadashi Kasuya, Unsupervised microstructure segmentation by mimicking metallurgists' approach to pattern recognition, Sci. Rep. 10 (1) (2020) 1–11.

[48] W.B. Lievers, A.K. Pilkey, An evaluation of global thresholding techniques for the automatic image segmentation of automotive aluminum sheet alloys, Mater. Sci. Eng. A 381 (1–2) (2004) 134–142.

[49] Dongjae Kim, Sihyung Lee, Wooram Hong, Hyosug Lee, Seongho Jeon, Sungsoo Han, Jaewook Nam, Image segmentation for FIB-SEM serial sectioning of a Si/C–Graphite composite anode microstructure based on preprocessing and global thresholding, Microsc. Microanal. 25 (5) (2019) 1139–1154.

[50] Li Chen, Min Jiang, JianXun Chen, Image segmentation using iterative watersheding plus ridge detection, in: Proceedings of the 16th IEEE International Conference on Image Processing, 2009, pp. 4033–4036.

[51] Peter Stanley Jørgensen, Karin Vels Hansen, Rasmus Larsen, Jacob R Bowen, A framework for automatic segmentation in three dimensions of microstructural tomography data, Ultramicroscopy 110 (3) (2010) 216–228.

[52] Victor Hugo C de Albuquerque, Auzuir Ripardo de Alexandria, Paulo César Cortez, João Manuel RS Tavares, Evaluation of multilayer perceptron and self-organizing map neural network topologies applied on microstructure segmentation from metallographic images, NDT & E Int. 42 (7) (2009) 644–651.

[53] Dali Chen, Dinghao Guo, Shixin Liu, Fang Liu, Microstructure instance segmentation from aluminum alloy metallographic image using different loss functions, Symmetry 12 (2020) 639.

[54] Boyuan Ma, Xiaojuan Ban, Hai-You Huang, Yulian Chen, Wanbo Liu, Yonghong Zhi, Deep learning-based image segmentation for al-la alloy microscopic images, Symmetry 10 (2018) 107.

[55] Dali Chen, Pengyuan Zhang, Shixin Liu, YangQuan Chen, Wei Zhao, Aluminum alloy microstructural segmentation in micrograph with hierarchical parameter transfer learning method, J. Electron. Imaging 28 (2019) 1.

[56] Salah Ali, Sherry Mayo, Amirali K Gostar, Ruwan Tennakoon, Alireza Bab-Hadiashar, Thu MCann, Helen Tuhumury, Jenny Favaro, Automatic segmentation for synchrotron-based imaging of porous bread dough using deep learning approach, J. Synchrotron Radiat. 28 (2) (2021).

[57] Dmitry Bulgarevich, Susumu Tsukamoto, Tadashi Kasuya, Masahiko Demura, Makoto Watanabe, Pattern recognition with machine learning on optical microscopy images of typical metallurgical microstructures, Sci. Rep. 8 (2018).

[58] João Papa, Rodrigo Nakamura, V.H.C. Albuquerque, Alexandre Falcão, Joao Tavares, Computer techniques towards the automatic characterization of graphite particles in metallographic images of industrial materials, Expert Syst. Appl. 40 (2013) 590–597.

[59] Olaf Ronneberger, Philipp Fischer, Thomas Brox, U-Net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, 2015, pp. 234–241.

[60] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, Jianming Liang, UNet++: A nested U-Net architecture for medical image segmentation, in: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, 2018, pp. 3–11.

[61] Abhishek Chaurasia, Eugenio Culurciello, Linknet: Exploiting encoder representations for efficient semantic segmentation, in: 2017 IEEE Visual Communications and Image Processing (VCIP), 2017, pp. 1–4.

[62] Aayush Bansal, Xinlei Chen, Bryan Russell, Abhinav Gupta, Deva Ramanan, Pixelnet: Representation of the pixels, by the pixels, and for the pixels, 2017, arXiv:1702.06506.

[63] Alexander Kirillov, Ross Girshick, Kaiming He, Piotr Dollár, Panoptic feature pyramid networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6399–6408.

[64] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, Jiaya Jia, Pyramid scene parsing network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2881–2890.

[65] Hanchao Li, Pengfei Xiong, Jie An, Lingxue Wang, Pyramid attention network for semantic segmentation, 2018, arXiv:1805.10180.

[66] Liang-Chieh Chen, George Papandreou, Florian Schroff, Hartwig Adam, Rethinking atrous convolution for semantic image segmentation, 2017, arXiv:1706.05587.

[67] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, Hartwig Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 801–818.

[68] Selim Seferbekov, Vladimir Iglovikov, Alexander Buslaev, Alexey Shvets, Feature pyramid network for multi-class land segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 272–275.

[69] Tarun Kalluri, Girish Varma, Manmohan Chandraker, CV Jawahar, Universal semi-supervised semantic segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 5259–5270.

[70] Yuexing Han, Chuanbin Lai, Bing Wang, Hui Gu, Segmenting images with complex textures by using hybrid algorithm, J. Electron. Imaging 28 (1) (2019) 013030.

[71] Lvping Chen, Yu Han, Bo Cui, Yihong Guan, Yatao Luo, Two-dimensional fuzzy clustering algorithm (2DFCM) for metallographic image segmentation based on spatial information, in: 2015 2nd International Conference on Information Science and Control Engineering, 2015, pp. 519–521.

[72] Yuanyuan Chen, Wuyin Jin, Meng Wang, Metallographic image segmentation of GCr15 bearing steel based on CGAN, Int. J. Appl. Electromagn. Mech. (Preprint) (2020) 1–7.

[73] Mehmet Sezgin, Bülent Sankur, Survey over image thresholding techniques and quantitative performance evaluation, J. Electron. Imaging 13 (1) (2004) 146–165.

[74] P.K. Sahoo, S. Soltani, A.K.C. Wong, A survey of thresholding techniques, Comput. Vis. Graph. Image Process. 41 (2) (1988) 233–260.

[75] P. Liao, Tse-Sheng Chen, P. Chung, A fast algorithm for multilevel thresholding, J. Inf. Sci. Eng. 17 (2001) 713–727.

[76] Soham Sarkar, Swagatam Das, Sujoy Paul, Ritambhar Burman, Sheli Chaudhuri, Multi-level image segmentation based on fuzzy-Tsallis entropy and differential evolution, in: IEEE International Conference on Fuzzy Systems, 2013, pp. 1–8.

[77] Pedram Ghamisi, Micael Couceiro, Jon Benediktsson, N. Ferreira, An efficient method for segmentation of image based on fractional calculus and natural selection, Expert Syst. Appl. 39 (2012) 12407–12417.

[78] Diego Oliva, Erik Cuevas, Gonzalo Pajares, Daniel Zaldivar, Marco Cisneros, Multilevel thresholding segmentation based on harmony search optimization, J. Appl. Math. 2013 (2013).

[79] C.H. Li, C.K. Lee, Minimum cross entropy thresholding, Pattern Recognit. 26 (4) (1993) 617–625.

[80] N. Otsu, A threshold selection method from gray level histograms, IEEE Trans. Syst. Man Cybern. 9 (1979) 62–66.

[81] Jui-Cheng Yen, F. Chang, S. Chang, A new criterion for automatic multilevel thresholding, IEEE Trans. Image Process. 4 (3) (1995) 370–378.

[82] J. Kittler, J. Illingworth, Minimum error thresholding, Pattern Recognit. 19 (1986) 41–47.

[83] T.W. Ridler, Picture thresholding using an iterative selection method, IEEE Trans. Syst. Man Cybern. 8 (1978) 630–632.

[84] Wen-Hsiang Tsai, Moment-preserving thresolding: A new approach, Comput. Vis. Graph. Image Process. 29 (3) (1985) 377–393.

[85] J. Kapur, P. Sahoo, A. Wong, A new method for gray-level picture thresholding using the entropy of the histogram, Comput. Vis. Graph. Image Process. 29 (1985) 273–285.

[86] Prasanna Sahoo, Carrye Wilkins, Jerry Yeager, Threshold selection using Renyi's entropy, Pattern Recognit. 30 (1) (1997) 71–84.

[87] Ahmed S. Abutaleb, Automatic thresholding of gray-level pictures using two-dimensional entropy, Comput. Vis. Graph. Image Process. 47 (1) (1989) 22–32.

[88] A. Wong, P. Sahoo, A gray-level threshold selection method based on maximum entropy principle, IEEE Trans. Syst. Man Cybern. 19 (1989) 866–871.

[89] Nikhil R. Pal, Sankar K. Pal, Entropic thresholding, Signal Process. 16 (2) (1989) 97–108.

[90] Rolf Adams, Leanne Bischof, Seeded region growing, IEEE Trans. Pattern Anal. Mach. Intell. 16 (6) (1994) 641–647.

[91] Alain Tremeau, Nathalie Borel, A region growing and merging algorithm to color segmentation, Pattern Recognit. 30 (7) (1997) 1191–1203.

[92] Fernand Meyer, Topographic distance and watershed lines, Signal Process. 38 (1) (1994) 113–125.

[93] J.M. Gauch, S.M. Pizer, Multiresolution analysis of ridges and valleys in grey-scale images, IEEE Trans. Pattern Anal. Mach. Intell. 15 (6) (1993) 635–646.

[94] Demetri Terzopoulos, Kurt Fleischer, Deformable models, Vis. Comput. 4 (1988) 306–331.

[95] Michael Kass, Andrew Witkin, Demetri Terzopoulos, Snakes: Active contour models, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, 1988, pp. 321–331.

[96] Stanley Osher, Ronald Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Vol. 153, Springer Science & Business Media, 2006.

[97] Pedro F. Felzenszwalb, D. Huttenlocher, Efficient graph-based image segmentation, Int. J. Comput. Vis. 59 (2004) 167–181.

[98] Dhanachandra Nameirakpam, Khumanthem Singh, Yambem Chanu, Image segmentation using k -means clustering algorithm and subtractive clustering algorithm, Procedia Comput. Sci. 54 (2015) 764–771.

[99] Yizong Cheng, Mean shift, mode seeking, and clustering, IEEE Trans. Pattern Anal. Mach. Intell. 17 (8) (1995) 790–799.

[100] Xin Jin, Jiawei Han, K-means clustering, in: Claude Sammut, Geoffrey I. Webb (Eds.), Encyclopedia of Machine Learning, Springer US, 2010, pp. 563–564.

[101] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise, in: KDD, Vol. 96, 1996, pp. 226–231.

[102] Bhawnesh Kumar, Umesh Kumar Tiwari, Santosh Kumar, Vikas Tomer, Jasmeet Kalra, Comparison and performance evaluation of boundary fill and flood fill algorithm, Int. J. Innov. Technol. Explor. Eng. 8 (2020).

[103] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurélien Lucchi, Pascal Fua, Sabine Süsstrunk, SLIC Superpixels, Technical Report, EPFL, 2010.

[104] James C. Bezdek, Robert Ehrlich, William Full, FCM: The fuzzy c-means clustering algorithm, Comput. Geosci. 10 (2) (1984) 191–203.

[105] Teuvo Kohonen, The self-organizing map, Proc. IEEE 78 (9) (1990) 1464–1480.

[106] Dongju Liu, Jian Yu, Otsu method and K-means, in: 2009 Ninth International Conference on Hybrid Intelligent Systems, Vol. 1, 2009, pp. 344–349.

[107] Xiaojin Zhu, Andrew B. Goldberg, Introduction to semi-supervised learning, Synth. Lect. Artif. Intell. Mach. Learn. 3 (1) (2009) 1–130.

[108] Longlong Jing, Yingli Tian, Self-supervised visual feature learning with deep neural networks: A survey, IEEE Trans. Pattern Anal. Mach. Intell. (2020).

[109] Zhengyang Fenga, Qianyu Zhoua, Qiqi Gua, Xin Tana, Guangliang Chengb, Xuequan Luc, Jianping Shib, Lizhuang Maa, DMT: Dynamic mutual training for semi-supervised learning, 2020, arXiv:2004.08514.

[110] Yves Grandvalet, Yoshua Bengio, et al., Semi-supervised learning by entropy minimization, in: Conference d'Apprentissage CAp, 2005, pp. 281–296.

[111] Nasim Souly, Concetto Spampinato, Mubarak Shah, Semi supervised semantic segmentation using generative adversarial network, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5688–5696.

[112] Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, Ming-Hsuan Yang, Adversarial learning for semi-supervised semantic segmentation, 2018, arXiv:1802.07934.

[113] Karen Simonyan, Andrew Zisserman, Very deep convolutional networks for large-scale image recognition, 2015, arXiv:1409.1556.

[114] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, 2015, arXiv:1512.03385.

[115] Mingxing Tan, Quoc Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: Proceedings of the 36th International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.

[116] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, Y. Bengio, The one hundred layers tiramisu: Fully convolutional DenseNets for semantic segmentation, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 1175–1183.

[117] Jonathan Long, Evan Shelhamer, Trevor Darrell, Fully convolutional networks for semantic segmentation, 2015, pp. 3431–3440.

[118] Thomas G. Dietterich, Ensemble methods in machine learning, in: Multiple Classifier Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000, pp. 1–15.

[119] David H. Wolpert, Stacked generalization, Neural Netw. 5 (2) (1992) 241–259.

[120] Shiliang Sun, Liang Mao, Ziang Dong, Lidan Wu, Multiview Machine Learning, Springer, 2019.

[121] Connor Shorten, Taghi M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (1) (2019) 1–48.

[122] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, Andrew Gordon Wilson, Averaging weights leads to wider optima and better generalization, 2018, arXiv:1803.05407.

[123] Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, Andrew Gordon Wilson, There are many consistent explanations of unlabeled data: Why you should average, in: 7th International Conference on Learning Representations, 2018.

[124] Shruti Jadon, A survey of loss functions for semantic segmentation, in: 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), IEEE, 2020, pp. 1–7.

[125] Wonjik Kim, Asako Kanezaki, M. Tanaka, Unsupervised learning of image segmentation based on differentiable feature clustering, IEEE Trans. Image Process. 29 (2020) 8055–8068.

[126] G. Bradski, The OpenCV library, Dr. Dobb's J. Softw. Tools (2000).

[127] Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, Scikit-image: image processing in Python, PeerJ 2 (2014) e453.

[128] Fisher Yu, Vladlen Koltun, Thomas Funkhouser, Dilated residual networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 472–480.

[129] Pavel Yakubovskiy, Segmentation models PyTorch package, 2020, https://github.com/qubvel/segmentation_models.pytorch.

[130] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, Masanori Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019.

[131] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, Li Fei-Fei, ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.