Superpixel segmentations for thin sections: evaluation of methods to enable the generation of machine learning training data sets

Jiaxin Yu\textsuperscript{a}, Florian Wellmann\textsuperscript{a,*}, Simon Virgo\textsuperscript{a,e}, Marven von Domarus\textsuperscript{d}, Mingze Jiang\textsuperscript{a,b}, Joyce Schmatz\textsuperscript{b,c}, Bastian Leibe\textsuperscript{d}

\textsuperscript{a}RWTH-Aachen, Geoscience, Department of Computational Geoscience and Reservoir Engineering, Germany
\textsuperscript{b}MaP Microstructure and Pores GmbH, Germany
\textsuperscript{c}RWTH-Aachen, Lehr- und Forschungsgebiet für Geologie-Endogene Dynamik, Germany
\textsuperscript{d}RWTH-Aachen, Visual Computing Institute, Computer Vision group, Germany
\textsuperscript{e}Terranigma Solutions GmbH, Germany

This manuscript has been submitted for publication in Computers & Geosciences. The paper has not yet undergone peer-review. Subsequent versions of this manuscript may have slightly different content. If accepted, the final version of this manuscript will be available via the ‘Peer-reviewed Publication DOI’ link on the right-hand side of this webpage. Please feel free to contact any of the authors; we welcome feedback.

\textsuperscript{*}Corresponding author
Email address: florian.wellmann@cgre.rwth-aachen.de, @flohorovicic (Florian Wellmann)
Superpixel segmentations for thin sections: evaluation of methods to enable the generation of machine learning training data sets

Jiaxin Yu\textsuperscript{a}, Florian Wellmann\textsuperscript{a,\textcopyright}, Simon Virgo\textsuperscript{a,c}, Marven von Domarus\textsuperscript{d}, Mingze Jiang\textsuperscript{a,b}, Joyce Schmatz\textsuperscript{b,c}, Bastian Leibe\textsuperscript{d}

\textsuperscript{a}RWTH-Aachen, Geoscience, Department of Computational Geoscience and Reservoir Engineering, Germany
\textsuperscript{b}MaP Microstructure and Pores GmbH, Germany
\textsuperscript{c}RWTH-Aachen, Lehr- und Forschungsgebiet für Geologie-Endogene Dynamik, Germany
\textsuperscript{d}RWTH-Aachen, Visual Computing Institute, Computer Vision group, Germany
\textsuperscript{e}Terranigma Solutions GmbH, Germany

\textbf{Abstract}

Training data is the backbone of developing either Machine Learning (ML) models or specific deep learning algorithms. The paucity of well-labeled training image data has significantly impeded the applications of ML-based approaches, especially the development of novel Deep Learning (DL) methods like Convolutional Neural Networks (CNNs) in mineral thin section images identification. However, image annotation, especially pixel-wise annotation is always a costly process. Manually creating dense semantic labels for rock thin section images has been long considered as an unprecedented challenge in view of the ubiquitous variety and complexity of minerals in thin sections. To speed up the annotation, we propose a human-computer collaborative pipeline in which superpixel segmentation is used as a boundary extractor to avoid hand delineation of instances boundaries. The pipeline consists of two steps: superpixel segmentation using MultiSLIC, and superpixel labeling through a specific-designed tool. We use a cutting-edge methodology Virtual Petroscopy (ViP) for automatic image acquisition. Bentheimer sandstone sample is used to conduct performance testing of the pipeline. Three standard error metrics are used to evaluate the performance of Mul-
tiSLIC. The result indicates that MultiSLIC is able to extract compact superpixels with satisfying boundary adherence given multiple input images. According to our test results, large and complex thin section images with pixel-wise accurate labels can be annotated with the labeling tool more efficiently than in a conventional, purely manual work, and generate data of high quality.

**Keywords:** Mineral thin section images, Pixel-wise labeling, Image annotation, Superpixel segmentation

1. Introduction

The analysis of petrographic thin sections is a standard process in many geological and mineralogical studies. Traditionally, this analysis relies highly on the visual interpretation by experts through petrographic microscopes, and this aspect will remain important, as it allows to integrate wide-ranging expertise, different microscopic techniques, and the geological context of the investigated material. However, this conventional approach is time-consuming and therefore limited to selected samples. As geological surveys, resource companies and research institutes are increasingly digitizing petrographic thin sections, approaches for an automated quantitative analysis are now becoming a standard tool to investigate properties in thin sections (e.g. Marmo et al., 2005; Singh et al., 2010; Mlynarczuk et al., 2013; Thompson et al., 2001; Baykan and Yılmaz, 2010; Borges and de Aguiar, 2019; Ramil et al., 2018; Maitre et al., 2019). These approaches have been driven by the rapid developments in the field of visual image analysis and segmentation leading to considerable progress in computer-aided methods for the automated analysis of mineral thin section images.

Currently, most of these approaches are based on supervised classification methods, where a training data set, generated by human experts, is used as an input to train a machine learning model for subsequent prediction of other samples. In the specific case of visual object detection, this process is referred to as image annotation. Image annotation can be done at three levels, namely image-wise tagging (Deng et al., 2009; Krasin et al., 2017), object-wise segmentation (Lin
et al., 2014; Netzer et al., 2011) and pixel-wise labeling (Lin et al., 2014; Everingham et al., 2010). According to the level of annotation in the training data set, the subsequent machine learning models can be classified into image categorization, mineral identification, and pixel-based segmentation.

- In image classification (Fig. 1a), the input is usually a tagged image while the goal is to predict the correct class label of the entire image. In the case of petrographic thin section analyses, class labels are usually referring to lithology, rock type or texture (e.g. Marmo et al., 2005; Mlynarczuk et al., 2013; Singh et al., 2010).

- In mineral identification (Fig. 1b), a classifier is trained on top of images where target mineral grains are manually identified ahead of time. The output is usually a segmentation map where each type of mineral is indicated by a unique color-mode (Budennyy et al., 2017; Thompson et al., 2001; Baykan and Yılmaz, 2010; Borges and de Aguiar, 2019; Ramil et al., 2018). Compared to image tagging, generating training sets for grain segmentation and identification is more time-consuming, as it requires a detailed mineral detection and identification process.

- A full segmentation of thin sections (Fig. 1c) requires a detailed tracing of grain boundaries. The recent success of deep convolutional neural networks (CNNs) (Krizhevsky et al., 2012; Szegedy et al., 2015) has empowered the development of semantic segmentation (Long et al., 2015) using automatic feature hierarchy extraction (Long et al., 2015; Lin et al., 2016). First approaches using CNNs for mineral thin section analysis have been reported by Cheng and Guo (2017); Iglesias et al. (2019); Karimpouli and Tahmasebi (2019); Tang et al. (2017).

With the potential to consider both local (e.g. optical characteristics of mineral) and contextual information (rock texture and structure), full segmentation approaches are very promising for large-scale thin section analysis, with a potential to approach human-level analysis. At the
Figure 1: Illustration of the relationship between the level of annotation and required human intervention
same time, deep learning models also place higher requirements on data annotation \cite{Sun_et_al_2018}, as most widely used CNN structures \cite{Ronneberger_et_al_2015, Krizhevsky_et_al_2012, He_et_al_2016, Simonyan_and_Zisserman_2014} require a large amount of well-labeled categorical labels from images. In computer vision, the stunning achievements of machine learning and deep learning applications are strongly motivated by many large-scale open-source data sets such as MSCOCO \cite{Lin_et_al_2014}, Cityscapes \cite{Cordts_et_al_2016} and LabelMe \cite{Russell_et_al_2008}. Well annotated data sets are being used not only to learn classifiers, but to reliably identify and evaluate the promising methods \cite{Hradiš_et_al_2012}. In the field of petrographic thin section analyses, such a large annotated data set is not yet available. And as sketched in Fig. 1, a labeling of image domains and grain boundaries requires detailed and careful line draws—a work that quickly becomes infeasible for large thin section data sets. It is also worth noting that labeling of petrographic thin sections is a highly specialized task that cannot be outsourced easily to labeling services: whereas it is possible for almost everyone to identify a traffic light in an image \cite[e.g.][]{Von_Ahn_et_al_2004}, separating quartz from feldspar in a thin section requires specific training and expertise.

To overcome the problem of limited data sets for thin section analysis, previous approaches used pre-trained models using the principle of transfer learning \cite[e.g.][]{Zhang_et_al_2019} or data augmentation \cite[e.g.][]{Karimpouli_and_Tahmasebi_2019}. However, these approaches are also limited, as minerals in thin sections show specific characteristics that are different to most of the images in classical data sets \cite[for example those mentioned above]{} and the transfer is therefore limited. Also, even if augmentation methods can successfully be used to obtain more robust classification results, the possibility to identify features is still limited to the variability in the initial (small) data set.

Based on these preliminary considerations, we derive the premise that fully labeled data sets of thin sections are required to evaluate the full potential of novel machine learning algorithms. However, we also need methods to facilitate labeling by experts, making specific use of the characteristics in thin sections.
In this context, we propose the idea that such a data set including high-quality pixel-wise labels could be efficiently obtained through a human-computer collaborative annotation pipeline where computers extract grain boundaries by splitting images into superpixels. These superpixels are groups of pixels with similar properties and commonly determined with through unsupervised learning algorithms (e.g. Ren and Malik [2003], Stutz et al., 2018). In this paper, we first evaluate commonly used superpixel methods for their compatibility with the specific requirements to label petrographic thin sections. For this aim, we also revisit different evaluation measures.

One limitation of the existing algorithms is that they are designed for image data and therefore cannot consider all information from thin section data sets with plain-polarised and cross-polarized light images at multiple polarization angles. We address this issue with the extension of one of the most promising methods, SLIC, and propose a novel algorithm, MultiSLIC, that enables superpixel generation with multiple channels.

The paper is organised as follows: section 2 introduces the principles behind digital thin section generation and the dataset used in the following experiments. In section 3, we present the concept of superpixel segmentation in an image annotation pipeline and compare the existing segmentation algorithms. In addition, we present our own extension to consider multiple input channels, as these are often available for thin section data (for example when plane-polarized and cross-polarized images are available). In section 4, we present the results of a quantitative evaluation of these superpixel algorithms in the application to a real thin section and finally provide considerations for practical use in section 5.

2. Digital petrographic thin section data

2.1. Digital petrography

The variation of optical features of a mineral under different settings of a petrographic microscope is closely related to crystallographic characteristics that can be indicative of the mineral
In order to fully capture optical features in digital images, we use here a Virtual Petrography (ViP) system developed by Fraunhofer FIT in collaboration with the Institute of Structural Geology, Tectonics and Geomechanics at RWTH Aachen University (Virgo et al., 2016). The system is able to fully capture the information of entire rock thin-sections in a digital format under different settings, including objective zooming, stage rotation, and switching between plane polarized light (ppol) and crossed polarized light (xpol) (Virgo et al., 2016). The hardware of the system is an automated petrographic microscope that scans the thin sections in high resolution (up to 10^9 pixels per square centimeter with a 40x objective). The scanning process is performed sequentially along a predefined grid and automatically repeated for different rotation angles of crossed polarizers. Each scanned mosaic patch is an RGB image with 384 x 520 pixels, and patches are then seamlessly stitched to obtain a full thin section image. For a typical thin section with a size of 3 cm by 2 cm and a zoom level of 10x, we obtain a stitched image of 86,960 x 57,970 pixels.

High-resolution images at different rotation angles can be precisely matched on a pixel-level. This matching allows to determine the extinction behaviour at each pixel location, which can be interpolated with a smoothed function to compress information (Virgo et al., 2016). Based on the interpolated extinction information, a phase map can be produced to qualitatively show the mineral axis misorientations. This specific system also contains a dedicated displaying toolbox to allow users to adjust colouring of images and to evaluate extinction angles and behaviour. As shown in Fig. 2a, multiple scanning image layers form a digital thin section cube that captures the full optical characteristics of a thin section viewed under various polarisation and illumination conditions.

2.2. Thin section data set

The sample data set used in the paper is Bentheimer Sandstone (BS), which is one of the most well-known sandstone types in Europe. Locations with outcrops of this sandstone can be found
Figure 2: Ground-truth generation for digital thin section. (a) Ppol and all xpol layers form the labeling ground. (b) incomplete contour map generated based on a single layer in the image stack, in contrast, (c) shows the desired result containing the full shape of grains using the methodical approach; (d) Boundary maps delineated by different human annotators.
on the border between the Netherlands and Germany. It forms a significant reservoir rock for petroleum reservoirs and is characterized by loose compaction, simple mineral composition and a well-sorted grain and pore space network (Peksa et al., 2015). The sandstone is mainly composed of loosely packed detrital quartz grains, with additional 2-4% of altered Feldspar. Due to its properties and the block-scale homogeneity and lateral continuity, this rock type is widely used as a standard reservoir analogue. In the following, we use a sample of this sandstone which has some detrital chert fragments, which appear as speckled grain composed of coarsely crystalline quartz. Pores in the sample were impregnated with light blue-dyed epoxy resin.

2.3. Generating boundary maps for digital thin sections

Evaluating the quantitative performance of superpixel segmentations for a digital thin section (ViP data set) requires a ground truth, which is manually generated by an expert. In order to obtain comparable ground truth maps from multiple experts, we devised a tracing plan for the procedure to reduce a potential procedural bias. The detailed tracing plan consists of four steps:

1. Prepare a ground-truth layer in tracing software (for example QGis, vector graphics programme);
2. Start from one of the image layers and trace all clearly visible boundaries;
3. Go to the next image layer and trace new boundaries that emerged;
4. Repeat the step 3 until all image layers are utilized. Note that, none of the boundaries that have been traced according to the previous layer will be deleted or modified when going to the next layer.

An example of this procedure is shown in Fig. 2: given a digital thin section data set with one ppol and multiple xpol layers corresponding to different polarization angles (a), boundaries in every layer are sequentially traced to obtain one single ground truth. Fig. 2(b) shows an incomplete boundary map, with boundaries traced on the ppol layer. In contrast, Fig. 2(c) represents the
complete boundary map that includes all boundary information obtained from all image layers. 
As shown in Fig. 2(d), three different ground truth boundary maps are generated by three different 
domain experts to average the human bias. These boundary maps are subsequently used as ground 
truths for a quantitative superpixel evaluation in section 4. As there are three outputs for each 
error metric given three ground truth maps, the final quantitative results are averages of the three 
outputs.

3. Superpixel segmentation

3.1. Principle of a superpixel annotation pipeline

Superpixels are groups of pixels that are perceptually similar (Ren and Malik, 2003). Before 
discussing different algorithms to determine these similar regions, we briefly outline the general 
annotation procedure. The overall aim is to obtain a pixel-wise annotation for image data. The 
process can be separated into two steps:

1. **Boundary identification**: superpixels are used to determine regions with similar prop-
ties to simplify boundary identification. Different algorithms consider different aspects of 
similarity (see Sec. 3.2);

2. **Instance labeling**: the obtained superpixels are then annotated with a corresponding class 
   label, which is then assigned to all pixels within this superpixel.

A schematic example of the superpixel annotation process is provided in Fig. 3

Instead of providing a discrete representation of images, superpixels are better aligned with 
image edges and largely reduce the image complexity (Vargas et al., 2014). As shown in Fig. 4 
and pointed by Neubert (2015), if \( K \) represents the number of objects in the image, \( P = m \times n \) is 
the number of pixels of the input image where \( m, n \) is the height and width [in pixels] of the given 
image, then for the number of superpixels \( N \):
Figure 3: Superpixel annotation pipeline, indicated by solid lines: Given an image as input, superpixels are first extracted with an unsupervised algorithm. Human annotators will then associate superpixels with a corresponding class label. In the end, a pixel-wise annotation result can be obtained.

\[ K << N << P \]  \hspace{1cm} (1)

A good superpixel segmentation, especially with a suitable detection of grain boundaries, has the potential to significantly reduce labeling time, but still leaves the possibility to include expertise through human annotators, leading to a good compromise for the generation of large thin-section training data sets.

3.2. Superpixel algorithms: state of the art

There are plenty of ways to generate superpixels. Here we briefly review the categorization of top performing recent algorithms according to the extensive investigation of superpixel algorithms performed by Stutz et al. (2018). Readers can gain a basic understanding of different approaches.
from these categorizations without being perplexed by the implementation of each individual algorithm. The presented categorizations are graph-based, clustering-based, contour-evolution, and energy-optimization, which are based on the one proposed by Achanta et al. (2010) and extended by Stutz et al. (2018).

Graph-based algorithms treat an image as a weighted graph $G_w = (V, E, W)$ where each vertex $v_i \in V$ is associated to an image pixel $i$ that is connected to an adjacent image pixel $j$ by an edge element $e_k \in E$. $w_{ij} \in W$ is the edge weight representing the similarity between a pair of pixels $i, j$ (Neubert 2015). Based on the calculation of $w_{ij}$ for each pair of nodes, the graph will be partitioned into disjoint regions. ERS (Liu et al. 2011) is a representative of graph-based algorithm.

Clustering methods are commonly used to find groups in observations. Probably the most well-known clustering algorithm is K-means (Bishop 2006). K-means is simple, yet powerful. It is computationally fast and easy to implement. However, it has certain drawbacks, notably that the number of clusters $K$ has to be defined beforehand. Moreover, the clustering result highly depends on the initial partitions (Celebi et al. 2013). Such dependence can be migrated by repeating the
clustering with different (randomized) initial cluster centers. There are many modifications and variations of standard \( K \)-means proposed to address the problem regarding the initialization of cluster \cite{Celebi2013}, and measure of Euclidean distance as intra-cluster metric \cite{ParkJun2009, VelmuruganSanthanam2010}. Likewise, clustering-based superpixel algorithms initialize seed pixels as “cluster centers” and measure color and spatial distance as within-cluster metric \cite{Stutz2018}. Usually, the number of superpixels and their shapes and sizes can be controlled over input parameters. A famous example of a clustering algorithm is SLIC proposed by \cite{Achanta2010}.

Contour evolution algorithms segment images by growing seed pixels such that the dilated pixels adapt to local image structures \cite{Stutz2018}. Conceptually, they are rooted in the early active contour model or snake method \cite{Kass1988}. The basic idea of the snake method is to propagate or deform an initial contour surrounding the given object towards the boundary of the detected object \cite{ChanVese2001}. The classic deformable contour model is built on an energy functional that is specifically designed for smoothing and attracting contours. The contour will stop evolving at boundaries where the energy functional can be minimized \cite{Caselles1997}. However, snakes are not able to detect complex boundaries as they require an initial contour beforehand. To address this problem, geometric active contour models based on curve evolution approaches instead of energy minimization have been developed \cite{Caselles1997, McInerneyTerzopoulos2000, Caselles1997}. These models enable active contours to handle changes in the topology during the curve evolution using a level-set method \cite{OsherSethian1988, Sethian1999}. A recent example is ERGC \cite{Buyssens2014b} generating superpixels based on the eikonal equation.

Energy optimization algorithms start from an initial partition of images and progressively refine the boundary pixels at each iteration \cite{Conrad2013, VanBergh2012, Yao2015a}. Energy updating strategies ensure the connectivity of the final segmentation and preserve the region topology in the image. The exchange of boundary pixels or the refinement of
boundary pixels is achieved by iteratively optimizing the energy function formulated for the image. These algorithms differ in the selection of energy functions to be optimized as well as the strategy of optimization; commonly used energy optimization algorithms are CRS (Conrad et al., 2013), ETPS (Yao et al., 2015b) and SEEDS (Van den Bergh et al., 2012).

3.3. Qualitative evaluation of superpixel segmentation

When comparing superpixel segmentations, several aspects can be relevant. From an algorithm in the context of mineral or grain segmentation considered here, we would expect to correctly identify mineral or grain boundaries. This aspect is also related to the concepts of over- and undersegmentation: Oversegmentation describes the effect that a single object (mineral or grain, in our case) is segmented into more than one superpixel; whereas undersegmentation describes that single superpixels contain more than one object. Clearly, oversegmentation is preferable, as undersegmentation leads to a loss of information about the object boundaries. An additional aspect to consider is the compactness of superpixels, i.e., how similar they are to a circle. Superpixels with a very low compactness can have very thin and elongated shapes and then be difficult to select in a post-processing step. All of these aspects will be considered and evaluated in the following sections.

In a first step, we compare six state-of-art algorithms (Table A.1) recommended by Stutz et al. (2018) and described in section 3.2. We apply all algorithms to the sandstone sample and perform segmentations with different superpixel resolutions $K$ ($K = 200, 400, 600, \ldots, 3000$). The additional settings for all algorithms are provided in table A.1 in the appendix.

A subarea of digital thin section for Bentheimer sandstone is selected for the experiment. It contains 10 xpol image layers and 1 ppol layer with a uniform size of 1551x2171 pixels. Examples of superpixel segmentations using the aforementioned algorithms are shown in Fig. 4, with maps of superpixels (yellow lines) on a part of the thin section and superpixel maps with random colors to show distribution and superpixel shape. There is no single best algorithm that can fit all
purposes. We observe that none of the algorithms can fully capture all boundaries when the super-
pixel density is low. The ability of all algorithms to capture boundary details increases for larger
numbers of superpixels. SEEDS and ETPS generate highly irregular superpixels compared to the
other algorithms, in this case this becomes clearly visible when \( K \) becomes 800. In comparison to
SEEDS and ETPS, CRS, ERGC and ERS generate relatively compact superpixels (i.e., superpix-
els approaching a spherical shape) with \( K=25 \) and less irregularly shaped superpixels at higher
superpixel density levels. Only SLIC can maintain a regular oversegmentation regardless of super-
pixel density, while resulting in compact superpixels and these features motivate the adaptation
of SLIC to the specific aspects of thin section data sets in the following.

3.4. MultiSLIC

Segmenting thin section data sets requires specific considerations that are different from other
image segmentation tasks, as thin sections contain more than just a single image layers (see Fig. 2)a.
Due to the change of polarization mode, misorientations of mineral axes and extinction behavior
of anisotropic minerals, the single cross-polarised images can significantly differ from each other
in terms of color, brightness and visible rock texture, and all information should be considered
to obtain the best possible segmentation. Therefore, a superpixel algorithm to cope with this
additional dimensionality would be desirable.

In 2018 Stutz et al. (2018) conducted a detailed evaluation of 28 state-of-the-art superpixel
algorithms and showed that SLIC was performing among the best in regards to boundary adher-
ence and segmentation accuracy. We observed a similar behaviour in the qualitative evaluation,
presented above (Sec. 3.3). Considering that SLIC is a rather simple, yet very fast algorithm
that performs well on a variety of data as indicated by its high rank in the previously mentioned
benchmark, we chose SLIC as a candidate algorithm for an adaptation to the domain of thin
sections data sets, where one data set contains \( M \) images, corresponding to the total number of
cross-polarized image layers plus one plain-polarized image.
| Method | K = 25 | K = 200 | K = 800 |
|--------|--------|---------|---------|
| SLC    | ![SLC_25](image) | ![SLC_200](image) | ![SLC_800](image) |
| CRS    | ![CRS_25](image) | ![CRS_200](image) | ![CRS_800](image) |
| ERGC   | ![ERGC_25](image) | ![ERGC_200](image) | ![ERGC_800](image) |
| ERS    | ![ERS_25](image) | ![ERS_200](image) | ![ERS_800](image) |
| EPS    | ![EPS_25](image) | ![EPS_200](image) | ![EPS_800](image) |
| SEEDS  | ![SEEDS_25](image) | ![SEEDS_200](image) | ![SEEDS_800](image) |

**Figure 5:** Example of superpixel segmentation results on an excerpt from the ViP data set of the Bentheimer sandstone. Superpixel boundaries are drawn in yellow, the corresponding random color coded segmentation masks are placed on the right. From left to right, images are shown for $K \approx 25$, $K \approx 200$, $K \approx 800$, respectively.
For a better understanding of MultiSLIC, we provide here a brief description of the SLIC algorithm, followed by a detailed implementation of MultiSLIC.

The Simple linear iterative clustering algorithm (SLIC) as proposed by Achanta et al. (2010) is a superpixel algorithm that clusters pixels based on their spatial distance and color similarity. SLIC is, in principle, a special case of $k$-means that initialises the $k$ cluster centers $C_k = [l_k, a_k, b_k, x_k, y_k]^T$ at a regular grid interval $S$ and uses a novel distance measure $D$ on the 5D $[labxy]$ space. Here $[x, y]$ stands for the coordinate plane and $[l, a, b]$ for the CIELAB color space. The grid interval $S$ is given by the number of pixel $N$ and the number of cluster center $k$ as $S = \sqrt[2]{N/k}$.

Since all $K$ superpixel should be approximately equally-sized, each should contain approximately $S^2$ pixels. In the SLIC algorithm, the assumption is taken that any possible member pixel of a superpixel lies within a $2S \times 2S$ neighbourhood of its center. Accordingly, instead of calculating the distances for every pixel to every center, as is usually done in $k$-means, in SLIC one only calculates distances from pixels to cluster centers within this $2S \times 2S$ neighbourhood, and this approach significantly speeds up the clustering calculation.

The distance measure $D$ introduced by Achanta et al. (2010) is defined as $D = d_{lab} + \frac{r}{5} d_{xy}$. Here, $d_{lab}$ and $d_{xy}$ are the euclidean distances on the color space and coordinate space respectively, while $r$ a weighting term influencing the compactness of the superpixel.

SLIC therefore segments the single input image into $K$ approximately equally sized superpixels that are positioned in a regular manner, with a time complexity of $O(N)$. There is however no guarantee that each superpixel is connected. Depending on the application, it is therefore necessary to enforce connectivity in a post-processing step which is simply done by giving each connected area its own label.

The general operating principle of MultiSLIC is practically identical to SLIC. To accommodate
the multidimensional input, the distance measure $D_s$ differs from SLIC as follows:

$$
d_{m}^{d_{m}} = \sqrt{(l_k^m - l_i^m)^2 + (a_k^m - a_i^m)^2 + (b_k^m - b_i^m)^2}
$$

$$
d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}
$$

$$
D_s = r \left( \sum_{m} d_{m}^{d_{m}} \right)^{p} + \frac{r}{S} d_{xy}
$$

where $m$ denotes the specific dimension and therefore $d_{m}^{d_{m}}$ expresses the Euclidean distance between the CIELAB color of two pixels in the same $m$-th image. Thus $D_s$ is the sum of the $p$-norm over the $m$ separate color differences and the coordinate distance normalized by the initial regular grid step size $S$ multiplied by a weighting factor $r$. The algorithm is described in more detail in the Algorithm 1 box.

Note that $r$ and $p$ are control variables. The former gives control over the compactness of a superpixel, while $p$ influences the impact of dimensions with small color differences. For example, consider $p_1 = 1$ and $p_2 = \infty$, with $p_1$ the norm collapses into a simple sum over the dimensional differences in color, whereas with $p_2$ the norm is equal to the highest difference regardless of the other dimensions. Additionally, if there is only one dimension ($M = 1$) and the $p$-norm is chosen as $p = 1$, then MultiSLIC defaults to the original SLIC algorithm.

3.5. Visual comparison between SLIC and MultiSLIC

Fig. 6 shows the superpixels extracted by MultiSLIC as well as by its original version SLIC. Both algorithms can capture the most details in the background image when $K$ is around 3000. However, SLIC cannot detect the boundaries of “hidden” grains appearing dark due to the extinction. In contrast, MultiSLIC can extract most of the important region boundaries across different layers into one superpixel segmentation merely with $K \approx 400$. Intuitively, MultiSLIC generates a quite satisfying segmentation according to visual inspection and outperforms SLIC when $K$ is small.
Algorithm 1: MultiSLIC

**input**: Desired number of superpixel $K$ and $M$ images

**output**: A superpixel segmentation

1. Initialize cluster centers $C_k = [l_k^1, a_k^1, b_k^1, \ldots, l_k^M, a_k^M, b_k^M, x_k, y_k]^T$ at regular grid steps 
   $S = \sqrt{N/K}$.

2. Move cluster centers to the neighbouring pixel with the lowest gradient position.

3. repeat
   4. for $k \leftarrow 0$ to $K$
   5. Calculate distance $D_s$ (Eq: 2) to $C_k$ for all pixel from a $2S \times 2S$ neighbourhood around $C_k$.
   6. Assign each pixel to the closest center.
   7. Recalculate cluster centers and corresponding residual error $E$.
   8. until $E \leq$ threshold;

9. Enforce connectivity.
Figure 6: Qualitative result of segmentation generated by MultiSLIC. The same MultiSLIC superpixel segmentation is superimposed on xpol layer. In comparison, boundaries of SLIC superpixels are depicted as red. Each image contains two excerpts, the left and the right excerpt correspond to the result of $K \approx 400$ and $K \approx 2800$, respectively.

4. Quantitative evaluation of superpixel algorithms

After the visual inspection of the superpixel segmentation algorithms, we now compare the algorithms quantitatively. As ground truth, we use the boundary maps, generated by human annotators (Sec. 2.3).

4.1. Superpixel evaluation measures

Considering the high relevance of a precise boundary shape detection in the proposed human-computer collaborative annotation pipeline, we use boundary recall ($Rec$) (Martin et al. 2004) and undersegmentation error ($UE$) (Neubert and Protzel 2012) to score the boundary adherence of superpixel segmentations generated by MultiSLIC and the set of recent advanced superpixel algorithms described in section 3.2. Given an image $I$ with a total number of $N$ pixels, let $S = \{S_1, ..., S_m\}$ be the superpixel segmentation where $m$ is the number of superpixels, and $G = \{G_1, ..., G_n\}$ represents ground truth segmentations with $n$ being the number of disjoint
partition of ground truth. The definitions of $Rec$ and $UE$ used in Stutz et al. (2015) are adopted here:

Boundary recall $Rec$ is defined as

$$Rec(G, S) = \frac{TP(G, S)}{TP(G, S) + FN(G, S)}$$ (3)

where True Positives ($TP$) represent the number of boundary pixels in $G$ correctly detected by boundary pixels in $S$ within a certain tolerance. False Negatives ($FN$) are the number of boundary pixels in $G$ for which no boundary pixels in $S$ exists within a certain tolerance. Stutz et al. (2018) suggested a local neighborhood $h$ whose size is calculated as $h = (2r + 1)x(2r + 1)$ with $r = 0.0025 \times$ image diagonal as tolerance during the calculation. $Rec$ assesses how well the superpixel boundaries align with the ground-truth edges.

The quantitative measure for the undersegmentation error $UE$, as proposed by Levenshtein et al. (2009), measures the fraction of superpixel leaks with respect to the ground-truth segmented border. A superpixel is supposed to align with the boundary of one object and $UE$ punishes the superpixel segmentation if a superpixels boundary is crossing the boundary of a ground truth segment, meaning in other words the superpixel overlaps or leaks. As shown in Fig. 7, superpixels are divided into in and out parts by a ground truth segment $G$. There are various ways to compute $UE$, Levenshtein et al. (2009) proposed to use the sum of “leakage” of superpixel $S_j$ with respect to $G$ as the undersegmentation error metrics. For example, this would be $(|B_{out} + C_{out} + D_{out}|)/|G|$ in Fig. 7

$$UE(G, S) = \frac{1}{N} \sum_{G_i} \sum_{S_j \cap G_j} min \{|S_j \cap G_j, S_j - G_j|\}$$ (4)

However, in such a way, superpixels that slightly cover the segments will be overly penalized. As a result, we adopt Neubert (2015)’s formulation expressed in Eq.4 when a superpixel is only slightly crossing a ground truth boundary, it will not affect the $UE$ with the whole superpixel size, but only with the small overlapping part, in the example of Fig. 7 that is $(|B_{in} + C_{out} + D_{in}|)/|G|$.
One additional desiderata is compactness, a geometric property reflecting the regularity of a given shape as well as the boundary smoothness of the shape. Schick et al. (2012) suggested the use of an isoperimetric quotient (Osserman et al., 1978) to measure the compactness of the superpixel segmentation. The isoperimetric quotient $Q(S)$ is defined as the ratio of the area $A(S)$ of a superpixel $S$ to a circle that has the radius of $\frac{P(S)}{2\pi}$:

$$Q(S) = \frac{A(S)}{\pi \left(\frac{P(S)}{2\pi}\right)^2}$$  \hspace{1cm} (5)

where $P(S)$ is the perimeter of the superpixel. $Q_s$ takes a maximum value of one for a circle and value $\pi/4$ for a square. Then the compactness of superpixel segmentation is defined as the average of isoperimetric quotient weighted by the size of superpixel compared to the whole image:

$$CO(G, S) = \frac{1}{N} \sum_{S_j} |S_j| Q(S_j)$$  \hspace{1cm} (6)

Superpixel segmentations with a high $CO$ are considered to be more compact.
4.2. Quantitative evaluation of superpixel segmentations

We will now evaluate how these previously described superpixel algorithms perform in a quantitative evaluation on the Bentheimer sandstone data set described above (Sec. 2.2). As the conventional superpixel segmentation algorithms can only consider a single image, we compare the measures on four different layers of the thin section data set:

- The image of plane polarized light (ppol)
- One image with cross-polarized light at angle 0 (xpol0)
- An image with cross-polarized light at angle 144 (xpol144)
- An image of maximum pixel intensity for all cross-polarized image layers (max_intensity)

The quantitative results with regards to boundary recall Rec are shown in Fig. 8. Since any region where the pixel contrast is perceptually large is regarded as a boundary when generating the ground truth boundary map, we suggest a threshold of $Rec \geq 95\%$ to be used for practical use. It can be seen that all algorithms can refine the prior segmentation by increasing the number of superpixels and generally lead to a good boundary recall. SEEDS is the top performer regardless of the background images. SEEDS also grants an advantage at the very beginning when $K$ is small ($K=200$). Given either max_intensity layer or ppol layer as input image, ETPS offers a competitive result with SEEDS especially for large $K$. CRS always yields a bad $Rec$ for a lower number of superpixels ($K \leq 400$) while quickly increasing the boundary adherence as $K$ goes up and finally its $Rec$ reaches a value that is as good as the one yielded by ETPS. The rest algorithms fall behind SEEDS, ETPS and CRS. SLIC is the worst performer in terms of $Rec$ as it always lies lowermost in the figure when $K$ is around 3000. In comparison to other algorithms, MultiSLIC shows higher $Rec$ when $K \leq 1000$ in most scenarios, but this advantage gradually disappears as the superpixel density ($K$) increases.
Figure 8: Averaged boundary recall of the resulting superpixel segmentations on four different information layers of BS. Rec is plotted as a function of superpixel density. K represents the number of superpixels, K varies from 200 to 3000 in steps of 200. A higher Rec means better boundary adherence.

Interesting to note is also that the variation of the input image does not seem to affect the relative performance of the tested algorithms. Although MultiSLIC cannot compete with the best performing algorithms with regard to Rec, it provides a stable and good Rec as compared to others and shows an advantage in Rec for CRS, ERGC, ERS and SLIC when K is small.

The undersegmentation error UE is a ground truth-dependent metric. Since UE measures the leakage of superpixels across the region boundaries in the given ground truth, it also provides an assessment of how tightly superpixels adhere to the boundaries. The quantitative results of UE are shown in Fig. 9. It is worth mentioning that undersegmentation exists in all superpixel segmentations. It is straight-forward to identify the top performer and bottom performer in terms of UE. MultiSLIC outperforms the rest of the algorithms, showing the lowest UE in all tested scenarios. Both ERS and SEEDS show a consistently high UE, no matter how input images vary. The remaining algorithms show moderate performance and approximate a similar UE for high numbers of superpixels K.

Rec and UE provide an overview of the algorithms’ performances with respect to boundary accuracy. It is apparent that MultiSLIC provides considerably good performance regarding these
two error metrics.

In addition to the performance in these measures, MultiSLIC generally results in compact superpixels. This aspect is illustrated in an evaluation of the compactness measures, shown in Fig. 10. Both SLIC and MultiSLIC generally create compact superpixels, especially for larger values of $K$, where other algorithms often start to generate thin and elongated superpixels (see also Fig. 5). Both of them show a growing value of $CO$ as the superpixel density is being increased.

Also interesting to note is that SLIC performs worse on the ppol image, but better at others. This is likely to be related to the fact that MultiSLIC recovers more objects than SLIC and that, therefore, fewer domains without boundaries exist in the SLIC segmentation, leading to more compact superpixels—but at the cost of missing information (see also Fig. 6).

Overall, MultiSLIC performs well on both a qualitative and quantitative level and can be used as a pre-segmentation algorithm dealing with the high dimensionality input from digital thin sections.
Figure 10: Compactness on four different information layers of the ViP data set. $K$ represents the number of superpixels, $K$ varies from 200 to 3000 in steps of 200. $CO$ denotes Compactness. A higher $CO$ represents that the shape of superpixel more resembles a circle.

5. Considerations for practical use

The presented superpixel methods, and specifically the developed extension to multiple channels in MultiSLIC, are only relevant when they support the main aim stated in the introduction: facilitating a faster generation of fully labeled thin section training data sets for machine learning applications. As a first test of feasibility to use these methods in a thin section labeling workflow, we developed a simple labeling tool and compared labeling efficiency to a standard boundary tracing approach.

5.1. Labeling tool

We developed a prototype labeling tool in Jupyter-notebook using Bokeh, an open source Python library that provides the possibility of customizing fully interactive apps for data analysis and visualization. As shown in Fig. 11, the labeling tool contains a viewer for displaying on the left and a output window showing the labeling result on the right. This simple app is easy to use and can be plugged in new functionality and interactivity according to the research demands. The
detailed implementation is accessible online\[1\].

![Figure 11: The interface of superpixel labeling tool in Jupyter-notebook](image)

5.2. Notes on the efficiency of thin section annotations with superpixels

In order to identify the efficiency of the annotation pipeline, we performed two experiments using the same digital thin section. The result were evaluated on a pixel-wise classified map with same size as the input thin section, with approximately 80-90 grains (including sub-grains) in the image. We compared the efficiency of the standard approach (boundary tracing) with a superpixel approach:

- Experiment 1: Labeling the digital thin section using either a tablets and a stylus pen, using standard labeling tools or graphics editors (QGis, PhotoShop).
- Experiment 2: Labeling the digital thin section using a MultiSLIC superpixel segmentation with the labeling tool described above.

Timing results reported by the labeling experts indicated an increase of annotation speed by a factor of 5 to 8. It can be expected that increasing the size of input image will make this time

\[1\]https://github.com/yujiaxin666/Mineral
difference bigger, as superpixels provide a full segmentation of the image into non-overlapping segments.

Even if this is only a limited experiment, the results are very promising and suggest that superpixel annotation methods can indeed be of very important use in the generation of fully labeled thin section data sets, an important aspect of future work.

6. Discussion

Our results show that superpixel algorithms can efficiently segment thin section image data sets, as an important step to generating training data sets for machine learning applications. This aspect is especially relevant as thin section labeling requires expert knowledge, and this task can therefore not simply be outsourced, for example over crowdsourcing platforms (e.g. Paolacci et al., 2010). Especially for the purpose of full image segmentation (see Fig. 1), these approaches therefore open up the way to a generation of suitably large sizes of training data sets for a variety of machine learning applications.

Thin section data sets also pose specific requirements to superpixel segmentation algorithms. The most obvious aspect is that thin section data sets contain more than just a single image, due to the combined use of plane-polarized light views and cross-polarized views at different angles. This aspect is important, as superpixel algorithms are developed for classical image data sets (Stutz et al., 2018). Due to this limitation, we implemented the extension of an existing algorithm, SLIC, to use multiple image layers, resulting in the adapted algorithm MultiSLIC.

We evaluated several algorithms with respect to their successful use for the specific requirements of thin section data sets. In Fig. 8 we showed the performance of tested superpixel algorithms in terms of Boundary recall. According to the result, MultiSLIC cannot compete with SEEDS and ETPS, both of them successfully detect more than 99% boundaries simply using a ppol layer. However, an interesting observation can be obtained when analysing Fig. 8, 9, and 10 together: SEEDS show the highest Rec, especially for large superpixel density, but yields the worst UE at
the same time. A possible explanation is that SEEDS actually oversegments the image locally but undersegments the image globally. This can be illustrated by Fig. 12. SEEDS always generates highly irregular superpixels with complex boundaries in order to exhaustively detect the subtle changes in the image which is good for boundary recall, but at the same time leads to a dramatic decrease in compactness of superpixels. It can be noticed that the grain is wrapped by layered thin superpixels at the rims. ETPS also shows wrapping behaviour. In this context, SEEDS and ETPS merely provide high apparent $Recs$ but are actually not as suitable as a superpixel extractor for practical use when compared to MultiSLIC. It is also worth mentioning that ETPS provides a solid trade-off between compactness and boundary adherence. As it keeps superpixels compact for relatively homogeneous regions, promoting also the extension of this algorithm to use multiple image layers in future work.

![Figure 12: Zoom-in portions of segmentations generated by SEEDS, ETPS and MultiSLIC with superpixel density $K \approx 3000$.](image)

After generation of the superpixels, the next step in the annotation pipeline is the actual process of labeling. We demonstrate a simple example of a labeling tool above, but also a wide variety of tools exist for this purpose (Wigness, 2018; Anderberg and Liesén Gullmander, 2020). One additional post-processing step is to merge superpixels further according to similarity measures between superpixels.
A classical way to simplify the initial segmentation is to merge adjacent regions on the basis of color similarity and spatial proximity (Trémeau and Colantoni, 2000). Region merging can be carried out by constructing a map graph with each node associated with a region, and each pair of adjacent regions are connected by an edge representing the relationship between adjacent regions (Schettini, 1993). This map graph is called Regional Adjacency Graph (RAG) that provides a prior knowledge of image structure for region merging (Haris et al., 1998). For capturing the redundancy of superpixel segmentations, each node in the RAG corresponds to one superpixel and the edge between adjacent superpixels measures dissimilarity of their mean colors. Fig. B.14 shows the construction of a RAG, the color of the edges indicates how dissimilar regarding mean color two superpixels are. Brighter color means lower similarity. The edge would be black if adjacent superpixels having the same mean color. Pairs of superpixels similar in color in the constructed RAG will be progressively merged until no similar pairs remain. This is to say when a pair of superpixels are merged, a new node is created. The weights of nodes adjacent to this merged node will be recalculated and updated before proceeding to the next iteration.

Fig. 13 shows an example of RAG merging of the same superpixel segmentation using different threshold values. The threshold is used to control the merging process. Edges with color dissimilarity higher than the threshold will be retained. As shown in Fig. 13, using higher threshold values will result in a more aggressive merging of superpixels. However, the result is sensitive to the selection of threshold value. The efficiency of merging will be low if the threshold is too small, higher threshold will cause the loss of important boundaries. Additionally, the calculation of mean color for the set of pixels belonging to the same superpixel is highly dependent on the given background image. Intuitively, a satisfied merging can be obtained by using a background image with higher intensity contrast and vice versa.

We evaluated here the superpixel segmentation with a relatively simple data set, a thin section of the Bentheimer sandstone. The internal structure of this rock type is rather simple, with quartz as the dominating mineral and only low secondary alteration and limited differences grain shapes.
Figure 13: Result of RAG merging of initial superpixel segmentation using different threshold values. Using of higher threshold value will merge more superpixels.

In future work, it would be interesting to apply the best-performing algorithms to a variety of different rock types, for a detailed evaluation on segmentation of different rock types.

Superpixel segmentation results can be influenced by changing the input parameters to algorithms. The number of parameters for different algorithms varies from 2 to 6. However, it is impractical to exhaustively test all possible configurations of parameters. For a fair comparison, the default pre-optimized parameters according to Stutz et al. (2018) are used for each algorithm in the evaluation presented here. But such parameter selection cannot fully guarantee that the resulting segmentation for each implementation has the best quality, especially when considering other different types of rock.

7. Conclusion and Future works

In this paper, we proposed a human-computer collaborative pipeline to speed up the pixel-wise labeling of petrographic thin section images. In order to avoid subjective visual interpretations and hand delineations of the region boundaries, an algorithm will first splits images into superpixels. We have proposed a novel superpixel algorithm to cope with the high input dimensionality of digital thin sections. Both qualitative and quantitative evaluation studies show the good performance
of MultiSLIC in terms of boundary adherence and compactness of the resulting segmentation. Superpixels will then be labeled by human annotators with a specifically designed labeling tool. Tests with different domain experts indicate a dramatic increase in labeling speed using the superpixel labeling tool. Besides, the proposed pipeline has great generalization capacity allowing wide collaboration for labeling petrographic thin section images at the pixel level. In the future we will use the proposed pipeline integrated with the superpixel algorithm we developed to generate a consistent and sufficiently large training data set with pixel-wise annotations that can be used to develop novel ML and DL algorithms for intelligent analysis of petrographic thin section images.

8. Acknowledgement

The research in this project has been supported by the Exploratory Research Scheme of RWTH Aachen University under the Exploratory Research Scheme Prep Fund Project ”MINERALS: Machine learning training set for virtual microscopy” (PFSDS023).

Code availability section

- The code for superpixel segmentation (conventional methods) is available on [https://github.com/davidstutz/superpixel-benchmark](https://github.com/davidstutz/superpixel-benchmark).
- Processing of results is performed with Jupyter notebooks, available on [https://github.com/yujiaxin666/Mineral](https://github.com/yujiaxin666/Mineral).
- A sample data set of the Bentheimer sandstone is available on [https://www.digitalrocksportal.org/projects/371](https://www.digitalrocksportal.org/projects/371).

CRediT authorship contribution statement

**Jiaxin Yu:** Writing-Original draft, Formal analysis, Investigation, Visualization, Methodology, Software and Data curation. **Florian Wellmann:** Writing-Review & Editing, Conceptualization,
Appendix A. Parameters for superpixel algorithms

Appendix B. Superpixel merging

Figure B.14: Constructing Regional Adjacency Graph (RAG) for superpixel segmentation. (a) RAG for an oversegmentation of a sandstone image. (b) Zooming into the yellow outlined area where nodes and edges for superpixels are shown in detail.
Table A.1: Listing of superpixel algorithms and corresponding parameters used in each implementation.

| Algorithm                                          | Publication                        | Parameters             | Categorization      |
|----------------------------------------------------|------------------------------------|------------------------|---------------------|
| Simple Linear Iterative Clustering (SLIC)          | Achanta et al. (2012)              | Compactness=10         | Clustering-based    |
|                                                    |                                    |                        |                     |
| Contour Relaxed Superpixels (CRS)                  | Conrad et al. (2013)               | Compactness=0.001      | Energy optimization |
|                                                    | Mester et al. (2011)               | Clique-cost=0.3        |                     |
|                                                    |                                    | Iterations=3           |                     |
|                                                    |                                    | Color space=0          |                     |
| Eikonal Region Growing Clustering (ERGC)           | Buyssens et al. (2014b)            | Color space=1          | Contour evolution   |
|                                                    | Buyssens et al. (2014a)            | Perturb-seeds=0         |                     |
|                                                    |                                    | Compacity =0           |                     |
| Entropy Rate Superpixels (ERS)                     | Liu et al. (2011)                  | Lambda=0.5             | Graph-based         |
|                                                    |                                    | Sigma= 5               |                     |
| Extended Topology Preserving Segmentation (ETPS)   | Yao et al. (2015a)                 | Regularization Weight=0.01 | Energy optimization |
|                                                    |                                    | Length weight=0.1      |                     |
|                                                    |                                    | Size weight=1          |                     |
| Superpixels Extracted via Energy-Driven Sampling (SEEDS) | Van den Bergh et al. (2012)     | Bins=5                 | Energy optimization |
|                                                    |                                    | Prior=0                |                     |
|                                                    |                                    | Confidence=0.1          |                     |
|                                                    |                                    | Iterations=25           |                     |
|                                                    |                                    | Color space=1           |                     |
|                                                    |                                    | Means=1                |                     |
References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S., 2012. Slic superpixels compared to state-of-the-art superpixel methods. IEEE transactions on pattern analysis and machine intelligence 34, 2274–2282.

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S., 2010. Slic superpixels. Technical report, EPFL.

Anderberg, P., Liesén Gullmander, F., 2020. Annotation of image sequences using superpixels.

Baykan, N.A., Yılmaz, N., 2010. Mineral identification using color spaces and artificial neural networks. Computers & Geosciences 36, 91–97.

Van den Bergh, M., Boix, X., Roig, G., de Capitani, B., Van Gool, L., 2012. Seeds: Superpixels extracted via energy-driven sampling, in: European conference on computer vision, Springer. pp. 13–26.

Bishop, C.M., 2006. Pattern recognition and machine learning. Springer.

Borges, H.P., de Aguiar, M.S., 2019. Mineral classification using machine learning and images of microscopic rock thin section, in: Mexican International Conference on Artificial Intelligence, Springer. pp. 63–76.

Budennyy, S., Pachezhertsev, A., Bukharev, A., Erofeev, A., Mitrushkin, D., Belozerov, B., et al., 2017. Image processing and machine learning approaches for petrographic thin section analysis, in: SPE Russian Petroleum Technology Conference, Society of Petroleum Engineers.

Buysens, P., Gardin, I., Ruan, S., Elmoataz, A., 2014a. Eikonal-based region growing for efficient clustering. Image and Vision Computing 32, 1045–1054.

35
Buyssens, P., Toutain, M., Elmoataz, A., Lézoray, O., 2014b. Eikonal-based vertices growing and iterative seeding for efficient graph-based segmentation, in: 2014 IEEE International Conference on Image Processing (ICIP), IEEE. pp. 4368–4372.

Caselles, V., Catté, F., Coll, T., Dibos, F., 1993. A geometric model for active contours in image processing. Numerische mathematik 66, 1–31.

Caselles, V., Kimmel, R., Sapiro, G., 1997. Geodesic active contours. International journal of computer vision 22, 61–79.

Celebi, M.E., Kingravi, H.A., Vela, P.A., 2013. A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert systems with applications 40, 200–210.

Chan, T.F., Vese, L.A., 2001. Active contours without edges. IEEE Transactions on image processing 10, 266–277.

Cheng, G., Guo, W., 2017. Rock images classification by using deep convolution neural network, in: Journal of Physics: Conference Series, IOP Publishing. p. 012089.

Conrad, C., Mertz, M., Mester, R., 2013. Contour-relaxed superpixels, in: International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, Springer. pp. 280–293.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3213–3223.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee. pp. 248–255.
Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A., 2010. The pascal visual object classes (voc) challenge. International journal of computer vision 88, 303–338.

Haris, K., Efstratiadis, S.N., Maglaveras, N., Katsaggelos, A.K., 1998. Hybrid image segmentation using watersheds and fast region merging. IEEE Transactions on image processing 7, 1684–1699.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

Hradiš, M., Kolář, M., Láník, A., Král, J., Zemčík, P., Smrž, P., 2012. Annotating images with suggestions—user study of a tagging system, in: International Conference on Advanced Concepts for Intelligent Vision Systems, Springer. pp. 155–166.

Iglesias, J.C.Á., Santos, R.B.M., Paciornik, S., 2019. Deep learning discrimination of quartz and resin in optical microscopy images of minerals. Minerals Engineering 138, 79–85.

Karimpouli, S., Tahmasebi, P., 2019. Segmentation of digital rock images using deep convolutional autoencoder networks. Computers & geosciences 126, 142–150.

Kass, M., Witkin, A., Terzopoulos, D., 1988. Snakes: Active contour models. International journal of computer vision 1, 321–331.

Krasin, I., Duerig, T., Alldrin, N., Ferrari, V., Abu-El-Haija, S., Kuznetsova, A., Rom, H., Uijlings, J., Popov, S., Veit, A., et al., 2017. Openimages: A public dataset for large-scale multi-label and multi-class image classification. Dataset available from https://github.com/openimages 2, 2–3.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, pp. 1097–1105.
Levinshtein, A., Stere, A., Kutulakos, K.N., Fleet, D.J., Dickinson, S.J., Siddiqi, K., 2009. Turbopixels: Fast superpixels using geometric flows. IEEE transactions on pattern analysis and machine intelligence 31, 2290–2297.

Lin, G., Shen, C., Van Den Hengel, A., Reid, I., 2016. Efficient piecewise training of deep structured models for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3194–3203.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context, in: European conference on computer vision, Springer. pp. 740–755.

Liu, M.Y., Tuzel, O., Ramalingam, S., Chellappa, R., 2011. Entropy rate superpixel segmentation, in: CVPR 2011, IEEE. pp. 2097–2104.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440.

MacKenzie, W.S., Adams, A.E., Brodie, K.H., 2017. Rocks and minerals in thin section: A colour atlas. CRC Press.

Maitre, J., Bouchard, K., Bédard, L.P., 2019. Mineral grains recognition using computer vision and machine learning. Computers & Geosciences 130, 84–93.

Marmo, R., Amodio, S., Tagliaferri, R., Ferreri, V., Longo, G., 2005. Textural identification of carbonate rocks by image processing and neural network: Methodology proposal and examples. Computers & geosciences 31, 649–659.
Martin, D.R., Fowlkes, C.C., Malik, J., 2004. Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Transactions on Pattern Analysis & Machine Intelligence 5, 530–549.

McInerney, T., Terzopoulos, D., 2000. T-snakes: Topology adaptive snakes. Medical image analysis 4, 73–91.

Mester, R., Conrad, C., Guevara, A., 2011. Multichannel segmentation using contour relaxation: fast super-pixels and temporal propagation, in: Scandinavian Conference on Image Analysis, Springer. pp. 250–261.

Mlynarczuk, M., Górszczyk, A., Ślipek, B., 2013. The application of pattern recognition in the automatic classification of microscopic rock images. Computers & Geosciences 60, 126–133.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y., 2011. Reading digits in natural images with unsupervised feature learning.

Neubert, D.I.P., 2015. Superpixels and their application for visual place recognition in changing environments.

Neubert, P., Protzel, P., 2012. Superpixel benchmark and comparison, in: Proc. Forum Bildverarbeitung, pp. 1–12.

Osher, S., Sethian, J.A., 1988. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. Journal of computational physics 79, 12–49.

Osserman, R., et al., 1978. The isoperimetric inequality. Bulletin of the American Mathematical Society 84, 1182–1238.

Paolacci, G., Chandler, J., Ipeirotis, P.G., 2010. Running experiments on amazon mechanical turk. Judgment and Decision making 5, 411–419.
Park, H.S., Jun, C.H., 2009. A simple and fast algorithm for k-medoids clustering. Expert systems with applications 36, 3336–3341.

Peksa, A.E., Wolf, K.H.A., Zitha, P.L., 2015. Bentheimer sandstone revisited for experimental purposes. Marine and Petroleum Geology 67, 701–719.

Ramil, A., López, A., Pozo-Antonio, J., Rivas, T., 2018. A computer vision system for identification of granite-forming minerals based on rgb data and artificial neural networks. Measurement 117, 90–95.

Ren, X., Malik, J., 2003. Learning a classification model for segmentation, in: null, IEEE. p. 10.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, Springer. pp. 234–241.

Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T., 2008. Labelme: a database and web-based tool for image annotation. International journal of computer vision 77, 157–173.

Schettini, R., 1993. A segmentation algorithm for color images. Pattern Recognition Letters 14, 499–506.

Schick, A., Fischer, M., Stiefelhagen, R., 2012. Measuring and evaluating the compactness of superpixels, in: Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), IEEE. pp. 930–934.

Sethian, J.A., 1999. Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. volume 3. Cambridge university press.
Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Singh, N., Singh, T., Tiwary, A., Sarkar, K.M., 2010. Textural identification of basaltic rock mass using image processing and neural network. Computational Geosciences 14, 301–310.

Stutz, D., Hermans, A., Leibe, B., 2018. Superpixels: An evaluation of the state-of-the-art. Computer Vision and Image Understanding 166, 1–27.

Sun, T., Zhang, W., Wang, Z.J., Ma, L., Jie, Z., 2018. Image-level to pixel-wise labeling: From theory to practice., in: IJCAI, pp. 928–934.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9.

Tang, D., Spikes, K., et al., 2017. Segmentation of shale sem images using machine learning, in: 2017 SEG International Exposition and Annual Meeting, Society of Exploration Geophysicists.

Thompson, S., Fueten, F., Bockus, D., 2001. Mineral identification using artificial neural networks and the rotating polarizer stage. Computers & Geosciences 27, 1081–1089.

Trémeau, A., Colantoni, P., 2000. Regions adjacency graph applied to color image segmentation. IEEE Transactions on image processing 9, 735–744.

Vargas, J.E., Saito, P.T., Falcao, A.X., De Rezende, P.J., Dos Santos, J.A., 2014. Superpixel-based interactive classification of very high resolution images, in: 2014 27th SIBGRAPI Conference on Graphics, Patterns and Images, IEEE. pp. 173–179.

Velmurugan, T., Santhanam, T., 2010. Computational complexity between k-means and k-medoids
clustering algorithms for normal and uniform distributions of data points. Journal of computer
science 6, 363.

Virgo, S., Heup, T., Urai, J.L., Berlage, T., 2016. Virtual petrography (vip)-a virtual microscope
for the geosciences. EGUGA, EPSC2016–14669.

Von Ahn, L., Blum, M., Langford, J., 2004. Telling humans and computers apart automatically.
Communications of the ACM 47, 56–60.

Wigness, M., 2018. Superlabel: A superpixel labeling interface for semantic image annotation.
Technical Report. ARMY RESEARCH LAB ADELPHI MD ADELPHI United States.

Yao, J., Boben, M., Fidler, S., Urtasun, R., 2015a. Real-time coarse-to-fine topologically pre-
serving segmentation, in: Proceedings of the IEEE conference on computer vision and pattern
recognition, pp. 2947–2955.

Yao, J., Boben, M., Fidler, S., Urtasun, R., 2015b. Real-time coarse-to-fine topologically preserv-
ing segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern
Recognition, pp. 2947–2955.

Zhang, Y., Li, M., Han, S., Ren, Q., Shi, J., 2019. Intelligent identification for rock-mineral
microscopic images using ensemble machine learning algorithms. Sensors 19, 3914.