

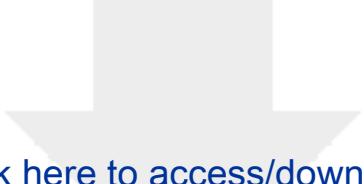
Engineering Applications of Artificial Intelligence
DIGITAL ROCK CHARACTERIZATION USING ARTIFICIAL INTELLIGENCE -
APPLICATION TO IMAGE SEGMENTATION
--Manuscript Draft--

Manuscript Number:	
Article Type:	Research paper
Keywords:	Artificial Intelligence; Digital Rock; Neural Networks
Corresponding Author:	João Marcelo Cardoso Carvalho State University of Norte Fluminense Macaé, RJ BRAZIL
First Author:	João Marcelo Cardoso Carvalho
Order of Authors:	João Marcelo Cardoso Carvalho André Duarte Bueno, Dr.
Abstract:	Abstract In the field of digital rock segmentation, the use of neural networks allows the classification or determination of solid and porous phases. The purpose of this work was to show how the use of such methods makes the binarization of images of reservoir rocks in solids and pores practical. For this, the color information of an image, collected through an annotation software developed in C++. The construction and training of neural networks were made with the PyTorch library through python scripts. Each training sequence (loading the dataset, separating it into training and test sets, and then into mini-batches, performing batch training and testing the result in each epoch) lasted about 12 milliseconds. The results were obtained with an error varying between 6% and 80% in comparison with laboratory measurements. It is believed that this occurs due to the points obtained from the regions of interest in the image, so a more careful collection could guarantee better results.
Suggested Reviewers:	Fernando Diogo de Siqueira siqueira@lenep.uenf.br Carlos Enrique Pico Ortiz capico@lenep.uenf.br

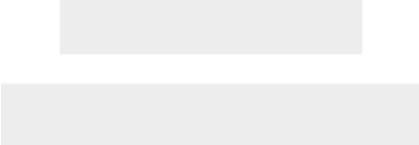


Click here to access/download
LaTeX Source Files
bibliografia.bib





Click here to access/download
LaTeX Source Files
rock-characterization-article.tex



Declaration of interests

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.

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

DIGITAL ROCK CHARACTERIZATION USING ARTIFICIAL INTELLIGENCE - APPLICATION TO IMAGE SEGMENTATION

J. M. C. Carvalho^{a,1}, A. D. Bueno^{a,1}

^a*Av. Alberto Lamego, 2000 – Parque Califórnia – Campos dos Goytacazes – RJ - Brazil – CEP: 28013-602*

Abstract

In the field of digital rock segmentation, the use of neural networks allows the classification or determination of solid and porous phases. The purpose of this work was to show how the use of such methods makes the binarization of images of reservoir rocks in solids and pores practical. For this, the color information of an image, collected through an annotation software developed in C++. The construction and training of neural networks were made with the PyTorch library through python scripts. Each training sequence (loading the dataset, separating it into training and test sets, and then into mini-batches, performing batch training and testing the result in each epoch) lasted about 12 milliseconds. The results were obtained with an error varying between 6% and 80% in comparison with laboratory measurements. It is believed that this occurs due to the points obtained from the regions of interest in the image, so a more careful collection could guarantee better results.

Keywords: Artificial Intelligence, Digital Rock, Neural Networks

1. Introduction

In this work, a study is developed about the application of artificial intelligence algorithms in the processing of images of reservoir rocks. More specifically, it is focused on the process of binarization of them, in order to facilitate the characterization of petrophysical properties.

1.1. Description of the Problem

It is increasingly common to apply artificial intelligence algorithms in the most different areas of research, especially in engineering. The use of intelligent algorithms makes it possible to perform complex classification and prediction tasks in a short time. Methods of such nature had been predicted for decades,

Email addresses: joao.carvalho@lenep.uenf.br (J. M. C. Carvalho),
bueno@lenep.uenf.br (A. D. Bueno)

but only with the exponential increase in the processing capacity of computers and the decrease in their cost has made then possible.

Recent advances in the field of imaging technology make rock analysis through digital imaging emergent. Microtomographs allow a non-destructive analysis of cores and help to build a more complete understanding of physical processes in porous media.

As shown in the work of Andrä et al. [1], Andrello Rubo et al. [2] the choice of a segmentation algorithm, filters and especially the threshold in gray-scale images are the key points for a good binarization of a reservoir rock image sample, so much that an unfortunate combination of these parameters can bring ambiguities for the interpretation process.

Rego and Bueno [13] showed that the use of artificial neural networks can bring good results to the image segmentation process, especially when compared to traditional methods. However, it is worth noting that there was still a difficulty in processing samples in which very light or dark grains were found. In addition, the work used traditional feed-forward neural networks, backpropagation algorithm, quadratic cost function and sigmoid activation function.

With the advancement of AI methods these choices have become obsolete over the years, and works like those of Sudakov et al. [15], Andrello Rubo et al. [2], van der Linden et al. [10], Saporetti et al. [14] has been showing good results on the processing of what can be called Digital Rock using, for example, convolutional neural networks combined with simpler machine learning algorithms.

These factors therefore contribute to the motivation of this work, which aims to use the experience obtained in previous works to advance knowledge in the field of research in Artificial Intelligence applied to petroleum reservoir engineering.

1.2. Artificial Neural Networks

Artificial neural networks are defined as non-linear models capable of mimicking the behavior of the human brain and its neuron structure [5]. Thus, ideally, an ANN is capable of performing the operations of learning, association, generalization and abstraction. These networks are composed of a series of elements called artificial neurons, highly interconnected and that perform simple operations and pass the information to the following elements of the structure. Each entry is represented by u_i and is weighted by a weight w_i with i ranging from 1 to n . The combination of these inputs is done by the sum Φ , to which a value of polarization or bias θ is added, as shown in Equation 1.

$$u = \Phi + \theta = \sum_1^n u_n w_n + \theta \quad (1)$$

The result of the operation shown in Equation 1 is applied as input to an activation function η , thus the output y of the neuron is shown in Equation 2. This activation function that is responsible for introducing non-linearity to the neuron model, and can appear in the form of a hyperbolic tangent, sigmoid function, $ReLU$, among others [3].

$$y = \eta(u) = \eta(\Phi + \theta) = \eta \left(\sum_1^n u_n w_n + \theta \right) \quad (2)$$

The ReLU activation function shown in Equation 3 has as its main advantage the fact that it does not immediately activate all neurons, as negative values of u end up being set to 0 [12].

$$\eta(u) = \begin{cases} u & u \geq 0 \\ 0 & u < 0 \end{cases} \quad (3)$$

The softmax function is similar to a sigmoid function and is widely used for classifying problems with more than two classes. It converts the values of the output layers into a probability function. Equation 4 shows its structure, where \vec{u} represents an entire input vector with k elements u_i . Here k is also the number of classes applied to the algorithm.

$$\eta(\vec{u})_i = \frac{e^{u_i}}{\sum_{j=1}^k e^{u_j}} \quad (4)$$

1.3. Feed-forward Neural Networks

Nielsen [12] defines feed-forward neural networks as structures divided in layers that are always fed with data in one direction. Any layer between these two is called deep or hidden layers.

The behavior of this neural network can be expressed in the form of Equation 5. In this notation w_{jk}^l denotes the weight of the connection between the k -th neuron of the $(l-1)$ -th layer with the j -th neuron of the l -th layer. Furthermore θ_j^l and y_j^l represent, respectively, the bias and activation of the j -th neuron in the l -th layer. The activation of the k -th neuron of the previous layer $(l-1)$ is used as input for the activation of the current neuron and is represented as y_k^{l-1} . Equation 6 shows the vectorized form of the activation.

$$y_j^l = \eta \left(\sum_k w_{jk}^l y_k^{l-1} + \theta_j^l \right) \quad (5)$$

$$y^l = \eta(w^l y^{l-1} + \theta^l) \quad (6)$$

According with Goodfellow et al. [4] the learning process of that form neural network relies on an algorithm called backpropagation, which the objective is to calculate how much it is necessary to modify the values of the weights and biases of the network after each output that the neural network returns. A cost function C is defined to how different from the network output y^L is the true value assigned to the input data at the input layer $y(x)$. The partial derivatives $\partial C / \partial \theta$ and $\partial C / \partial w$ are calculated regarding to the biases θ and weights w .

Normally, when a network is trained, the entire dataset is not used at once. It is common that before it is divided into smaller sets of input and label pairs,

selected randomly, called mini-batches. This approach is not only more computationally viable, as it avoids loading very large datasets into memory at once, but also helps in the network learning process [6]. The speed at which the network learns is also dictated by the learning rate.

According to Nielsen [12] the procedure for working with direct neural networks can be described under the following algorithm:

1. A set of data is entered in the input layer of the neural network.
2. For each training example, the following procedure is performed:
 - (a) Compute the outputs of neurons up to the last layer using Equation 5;
 - (b) Calculate the error $\delta^{x,L}$ using Equation 7;
 - (c) Backpropagate the error through the network using Equation 8 to calculate the error in each layer $l = L - 1, L - 2, \dots$;
3. Update the weights and biases of the network using Equations 9 and 10, where μ represents the learning rate of the network and m the size of the mini-batches.

$$\delta^{x,L} = \nabla_a C_x \odot \eta'(w^L y^{L-1} + \theta^L) \quad (7)$$

$$\delta^{x,l} = \left((w^{l+1})^T \delta^{x,l+1} \right) \odot \eta'(w^l y^{l-1} + \theta^l) \quad (8)$$

$$w^l \rightarrow w^l - \frac{\mu}{m} \sum_x \delta^{x,l} (a^{x,l-1})^T \quad (9)$$

$$b^l \rightarrow b^l - \frac{\mu}{m} \sum_x \delta^{x,l} \quad (10)$$

2. Materials and Methods

This section describes the development of the region annotation tool, the collecting data process from images obtained in the laboratory, and the CLI (Command-Line) application used for training the neural network and application of the model on these images. The Fig. 1 gives a whole process overview.

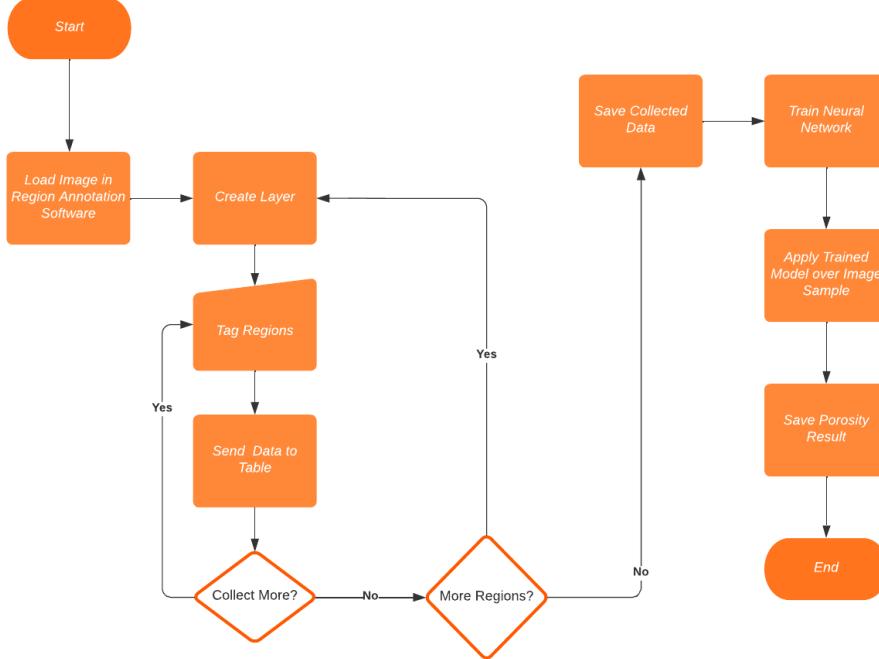


Figure 1: Process Overview.

2.1. Annotating Regions of Interest Software

A tool was developed that was capable of displaying an image to the user, and allowing the regions of interest to be associated with labels created by the professional who would be using it to carry out the study on the sample, it would be very useful to obtain material for feed the AI algorithms. The image to be studied would be loaded onto a base layer and as new labels were added new layers would overlap the base layer. The user could then, with some virtual annotation tool, such as a “pen”, mark the regions of interest in each layer. The RGB values of each pixel included in these marked regions can then be exported to a text file and used as a dataset in a neural network training script. This idea proved to be very simple to implement, in addition to having the potential to collect data for image segmentation of any nature. The Fig. 2 shows the tool Graphical User Interface (GUI).

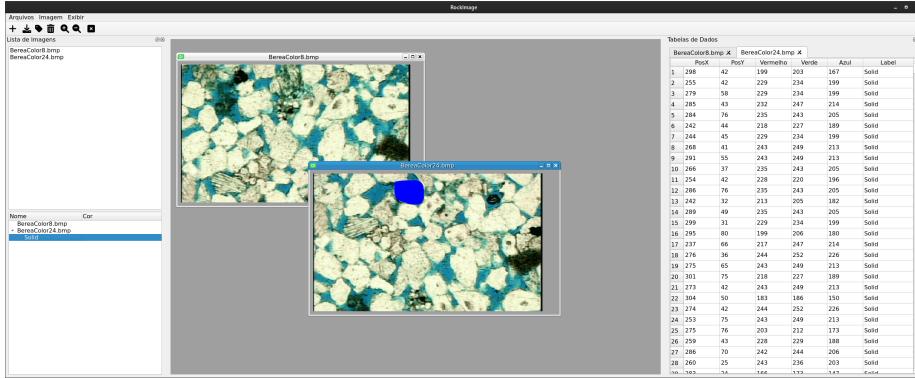


Figure 2: Annotation Tool GUI.

2.2. Data Collecting Process

The images used for data collection were classified into 5 different categories, as shown in Table [tab:Categories-of-Images], together with the estimated values of porosity and permeability in mD . These images represent rock samples of different porosity and permeability values, which implies different ways of performing the collection. However, actions such as avoiding contact with the edges were taken to obtain the best data quality.

Table 1: Images Categories.

Category	Porosity (%)	Permeability (mD)
Berea200		
Berea500		
P148_K2	14,8	2
P240_K104	24,0	104
P262_K441	26,2	441

In Figs. 3a, 3b, 3c, 3d, 3e it is possible to observe an example of data collection using the tool described above for each image type. During the collection, we also tried to maintain the same number of points for the regions of solids and pores. This was done so that it was possible to prevent the neural network from having its parameters biased during training.

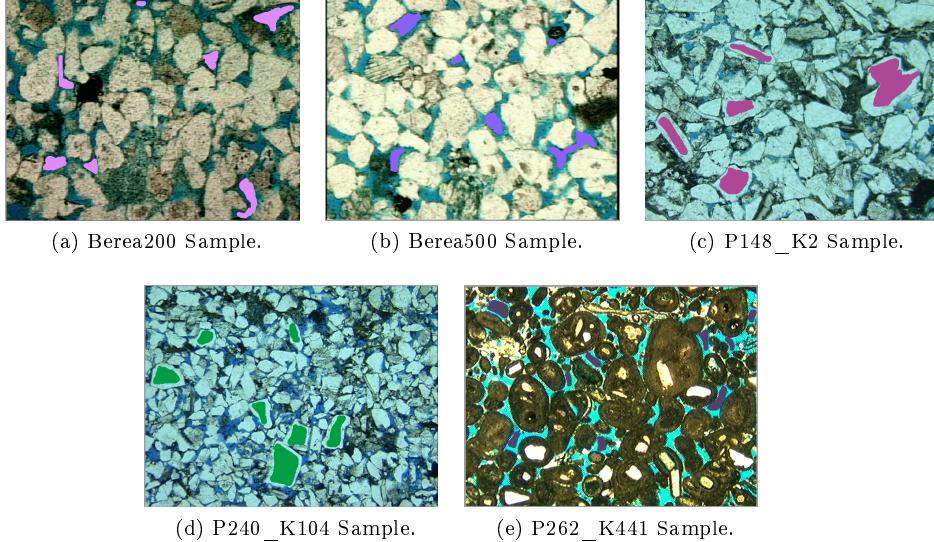


Figure 3: Data Collecting Examples.

2.3. Neural Network Training and Application Process

2.3.1. Model Description

Table 2 describes the neural network model used for training and binarizing the images. All internal layers of this network used ReLU as an activation function. A function called log_softmax was used to output the neural network, which, according to the library documentation, is recommended for classifier algorithms.

Table 2: Neural Networks Layers.

Layer Number	Number of Neurons	Activation Function
1	4	<i>relu</i>
2	4	<i>relu</i>
3	4	<i>relu</i>
4 (<i>output</i>)	2	<i>log_softmax</i>

The training process took place for each of the datasets used in the same way. First, they were loaded through the command line interface developed in python. Then they are shuffled and divided into training and test sets. After that, the training set was applied to the neural network in the form of mini batches. At the end of this step, the set test was submitted to the neural network to verify the accuracy. This process was then repeated for 5 epochs for each of the datasets.

2.3.2. Model Application

To apply the model, the CLI receives the path to the file that stores the model saved in pickle, in the training process, the path to the image that will be binarized, the path to the output binarized image and a flag to indicate whether the result of the binarized image should be saved or not.

Once the function is started, an instance of the neural network is created and the saved model is loaded into it. Then, the image is transformed into a PyTorch tensor and used to feed the AI model. The result of this event is the binararized image and the time elapsesed. The porosity of the sample image is then calculated and saved in a text file.

3. Results

In this section, the results obtained from the application of artificial intelligence methods over digital rocks samples will be shown along with the segmented images.

3.1. Porosity Values

This section presents the results obtained by applying the methods described in the previous chapter. The Table 3 shows the values obtained for porosity in each of the images and the time elapsed for application of the model.

Table 3: Porosity Results.

Sample	Porosity (%)	Elapsed Time (ms)
Berea200	25,5654	10,3312
Berea500	14,4561	9,9576
P148_K2	24,5020	10,2811
P240_K104	27,6058	22,7351
P262_K441	20,1338	9,2697

3.2. Output Images

The Figs. 4a, 4b, 4c, 4d, 4e shows the output images from the binarization process.

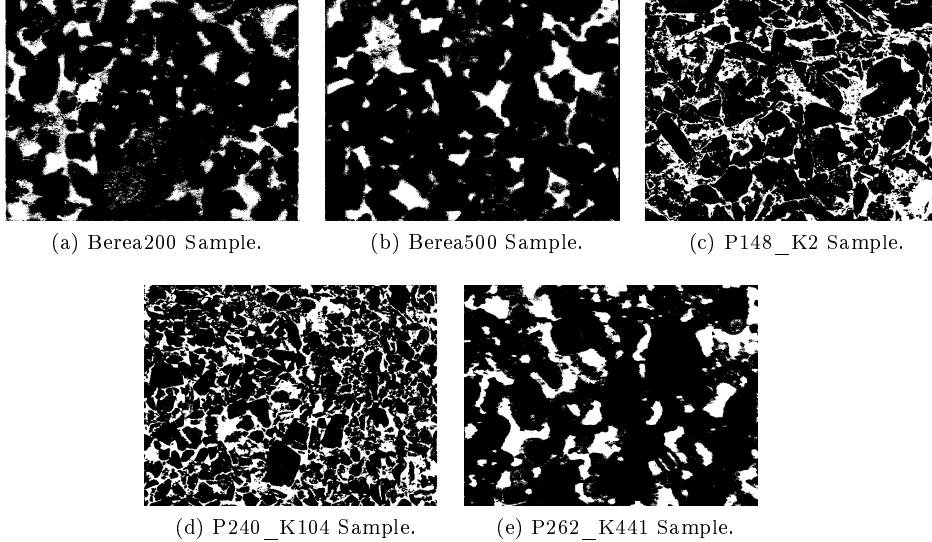


Figure 4: Output images from binarization process.

4. Discussion

As discussed earlier, the use of neural networks to solve problems related to image segmentation proves to be quite useful, especially when it comes to the study of digital rock samples. The results shown above corroborate those already shown in other works, such as Rego and Bueno [13], Sudakov et al. [15], Lin et al. [9], Leu et al. [8], Kuroda et al. [7], Ma and Gao [11].

What can be highlighted from the porosity values obtained is that there is a margin of error between the values obtained in the laboratory and those calculated by the algorithm. This may be related to the fact that in the laboratory part of the porosity value can be lost depending on the quality of the equipment used, while its electronic counterpart is capable of capturing the minutiae in the pores of smaller radius.

Another point to be considered is the impact that the quality of the images has on the segmented output. Figs. 4a, 4b and 4e have a result with a much lower visual quality compared to 4c and 4d images.

Finally, the 4c image has an error in the porosity value that can be justified by some regions that can be confused with oil stains, but which in reality can just be an error in the application of the resin. Figs. 5a and 5b highlights this region.

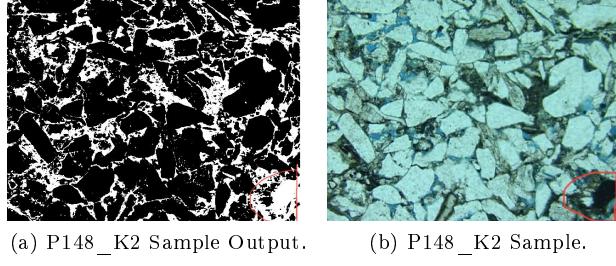


Figure 5: Data Collecting Examples.

5. Conclusion

As previously explained, the use of artificial intelligence algorithms is increasingly common in the most different areas of research, especially in the field of engineering. The use of such technology makes the execution of complex tasks, such as the classification and prediction model, in a short time possible.

More specifically in the field of reservoir engineering, the use of neural networks and computer vision allow analyzes to be performed to describe petrophysical characteristics of a sample without having to damage it. In this work, it was shown how absolute porosity values can be obtained using intelligent models that allow the segmentation of images in regions of pores and solid matrix, in a relatively shorter time and with precision similar to physical analysis. In addition, the development of these models took place using open source technologies, which makes it easier to reproduce the results described here.

The general objective of this work was to develop artificial intelligence methods capable of recognizing relevant patterns for the analysis of petrophysical properties in images of reservoir rocks. We tried to study different methods of machine learning and deep learning applied to image analysis, and to develop applications that could make possible the application of these models in real world examples. Among these applications would be python scripts for the use of artificial intelligence techniques and a tool for annotating regions of interest in digital rock samples.

Obtaining the results so that the objectives described above were achieved took place in the form of a procedure described in Figure 1, where the color information of the pixels of the regions of interest were collected and saved in a file in .dat format. This file is used as inputs for a neural network developed using the Machine Learning library PyTorch. The training script output is another file in .pt format, which contains the values of the biases and weights of the trained neural network. This saved template is then applied over the image to be binarized. Finally, the porosity value for the new image generated in the process is calculated. This entire procedure was repeated for images contained in the digital collection of Prof. DSc. André Duarte Bueno.

For this work, a single feed-forward neural network model was developed, with 3 deep layers, each with 4 neurons, as shown in Table 2. As activation

function, the ReLU function shown in Equation 3 was used. For the last layer, the softmax function was applied followed by the application of the natural logarithm of the value obtained. As optimizer, the Adam algorithm was chosen and, finally, the loss function NLL Loss, or, Negative Log-Likelihood Loss.

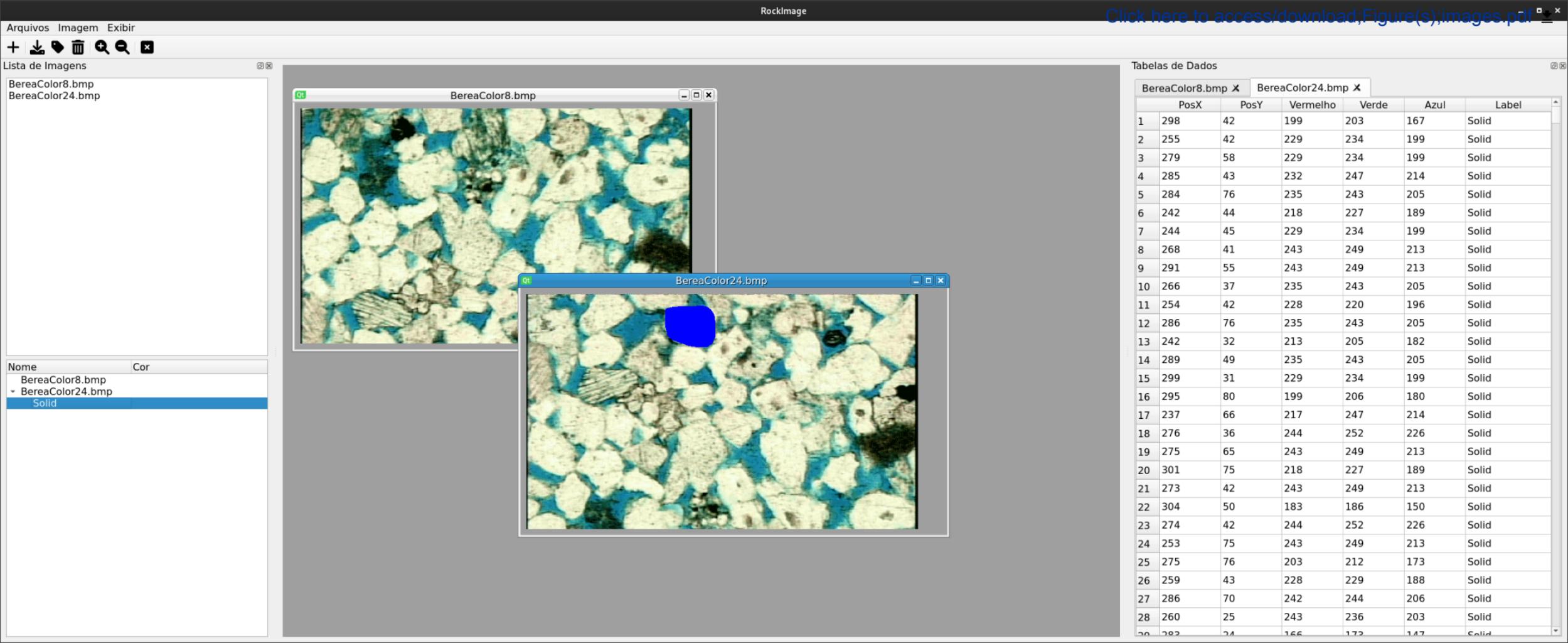
As a result, the duration of both training and application of the model lasted in the order of milliseconds, which shows the ability of these algorithms to deliver results quickly.

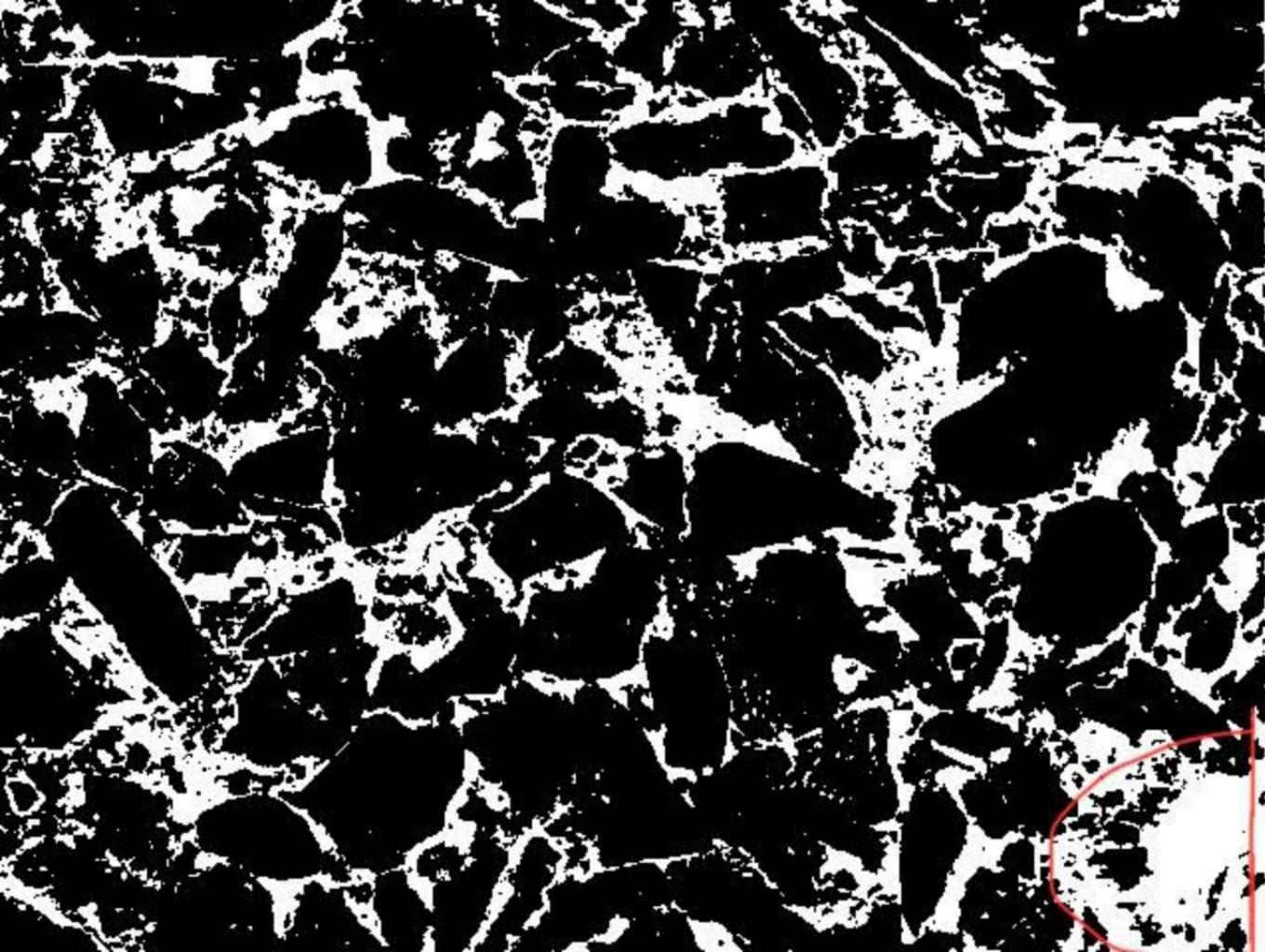
This work shows how technologies such as machine learning can be very useful in obtaining porosity values in samples of reservoir rocks. Additionally, the use of annotation tools can help make the data collection process less tedious and tedious.

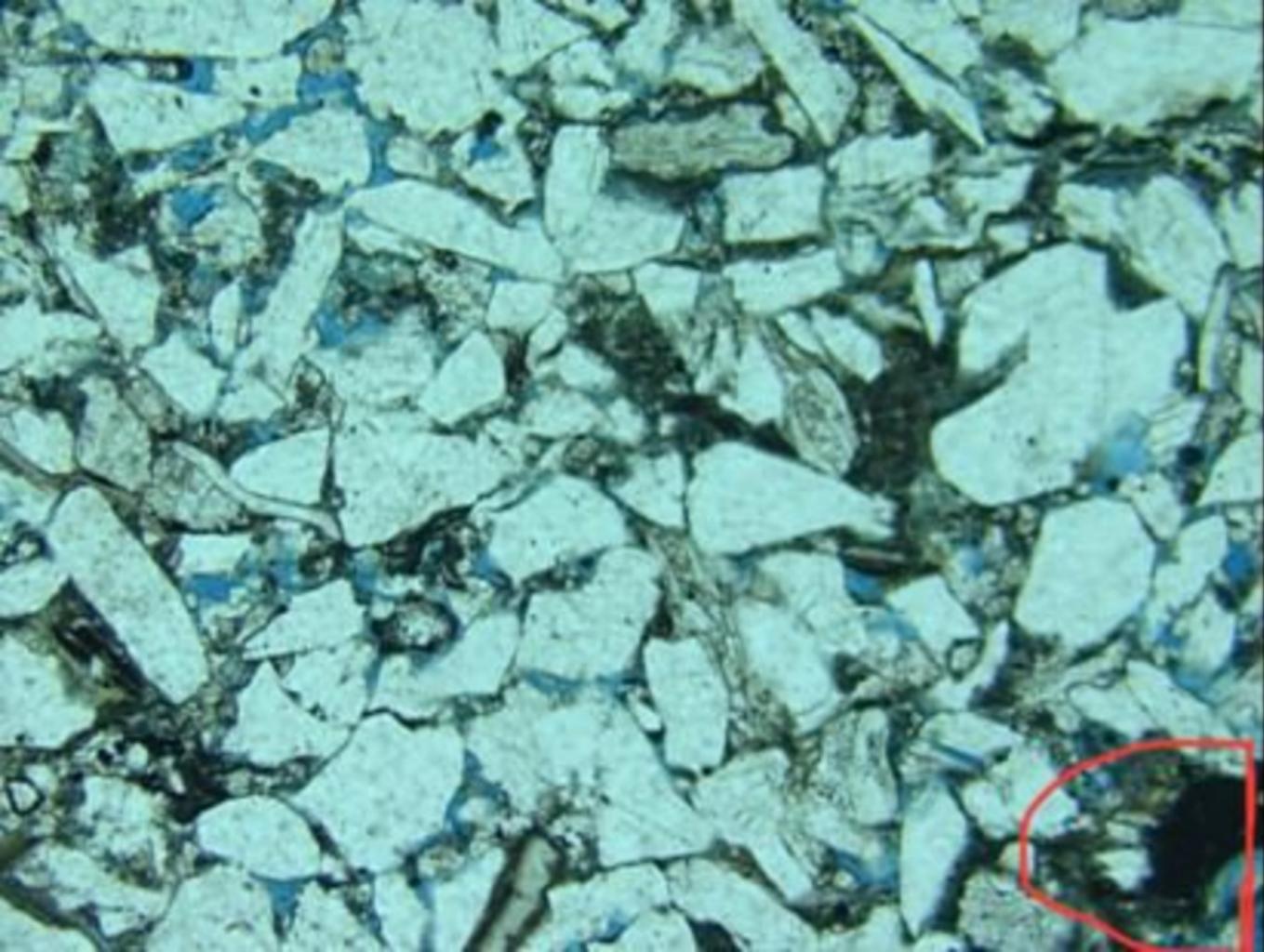
References

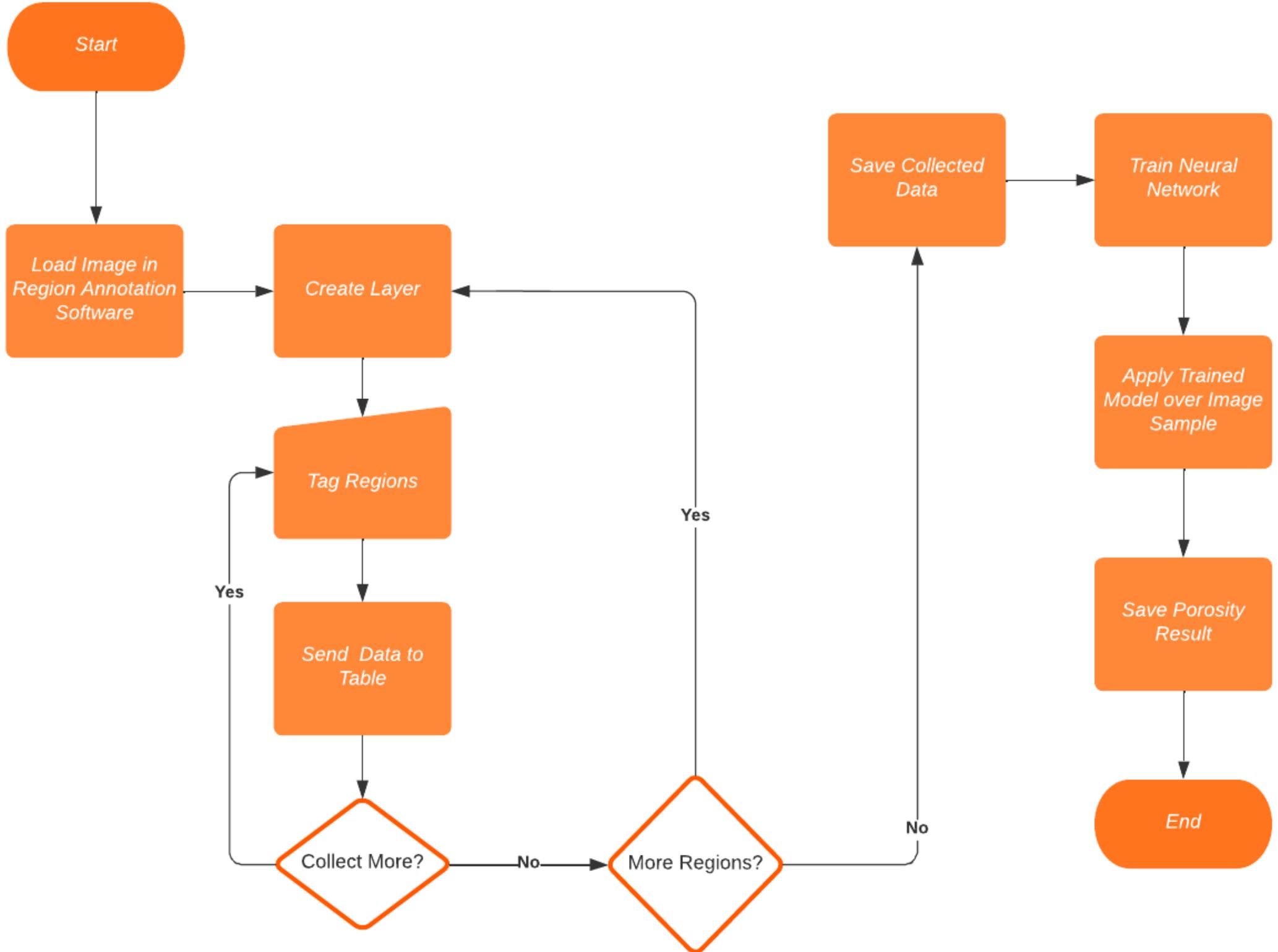
- [1] Andrä, H., Combaret, N., Dvorkin, J., Glatt, E., Han, J., Kabel, M., Keehm, Y., Krzikalla, F., Lee, M., Madonna, C., et al., 2013. Digital rock physics benchmarks apart i: Imaging and segmentation. *Computers & Geosciences* 50, 25–32.
- [2] Andrello Rubo, R., Carneiro, C., Michelon, M., Gioria, R., 2019. Digital petrography: Mineralogy and porosity identification using machine learning algorithms in petrographic thin section images. *Journal of Petroleum Science and Engineering* 183, 106382. doi:10.1016/j.petrol.2019.106382.
- [3] Chollet, F., et al., 2018. Deep learning with Python. volume 361. Manning New York.
- [4] Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y., 2016. Deep learning. volume 1. MIT press Cambridge.
- [5] Haykin, S., 2007. Redes neurais: princípios e prática. Bookman Editora.
- [6] James, G., Witten, D., Hastie, T., Tibshirani, R., 2013. An introduction to statistical learning. volume 112. Springer.
- [7] Kuroda, M.C., et al., 2016. Técnicas de aprendizagem de máquina bio-inspiradas aplicadas ao estudo de rochas reservatório .
- [8] Leu, L., Berg, S., Enzmann, F., Armstrong, R.T., Kersten, M., 2014. Fast x-ray micro-tomography of multiphase flow in berea sandstone: A sensitivity study on image processing. *Transport in Porous Media* 105, 451–469.
- [9] 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.

- [10] van der Linden, J.H., Narsilio, G.A., Tordesillas, A., 2016. Machine learning framework for analysis of transport through complex networks in porous, granular media: a focus on permeability. *Physical Review E* 94, 022904.
- [11] Ma, Z., Gao, S., 2017. Image analysis of rock thin section based on machine learning, in: International Geophysical Conference, Qingdao, China, 17-20 April 2017, Society of Exploration Geophysicists and Chinese Petroleum Society. pp. 844–847.
- [12] Nielsen, M.A., 2015. Neural networks and deep learning. volume 25. Determination press San Francisco, CA.
- [13] Rego, E.A., Bueno, O.A.D., 2010. Desenvolvimento de Metodo de Binarizacao para Analise de Rochas Reservatorio Tipicas da Bacia de Campos. Ph.D. thesis. Dissertacao (Msc Dissertation) UENF/LENEP.
- [14] Saporette, C.M., da Fonseca, L.G., Pereira, E., de Oliveira, L.C., 2018. Machine learning approaches for petrographic classification of carbonate-siliciclastic rocks using well logs and textural information. *Journal of Applied Geophysics* 155, 217–225.
- [15] Sudakov, O., Burnaev, E., Koroteev, D., 2019. Driving digital rock towards machine learning: Predicting permeability with gradient boosting and deep neural networks. *Computers & geosciences* 127, 91–98.

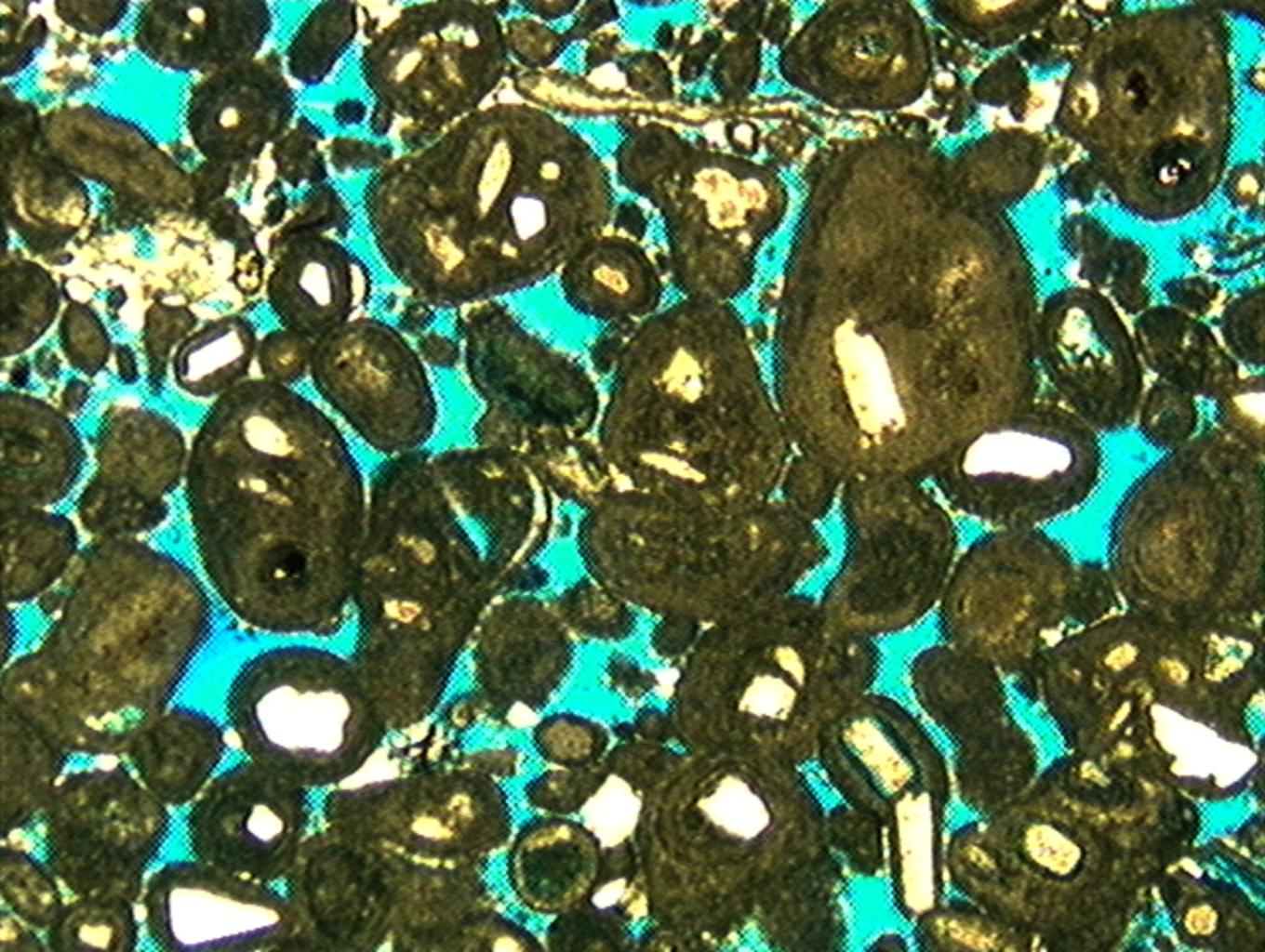




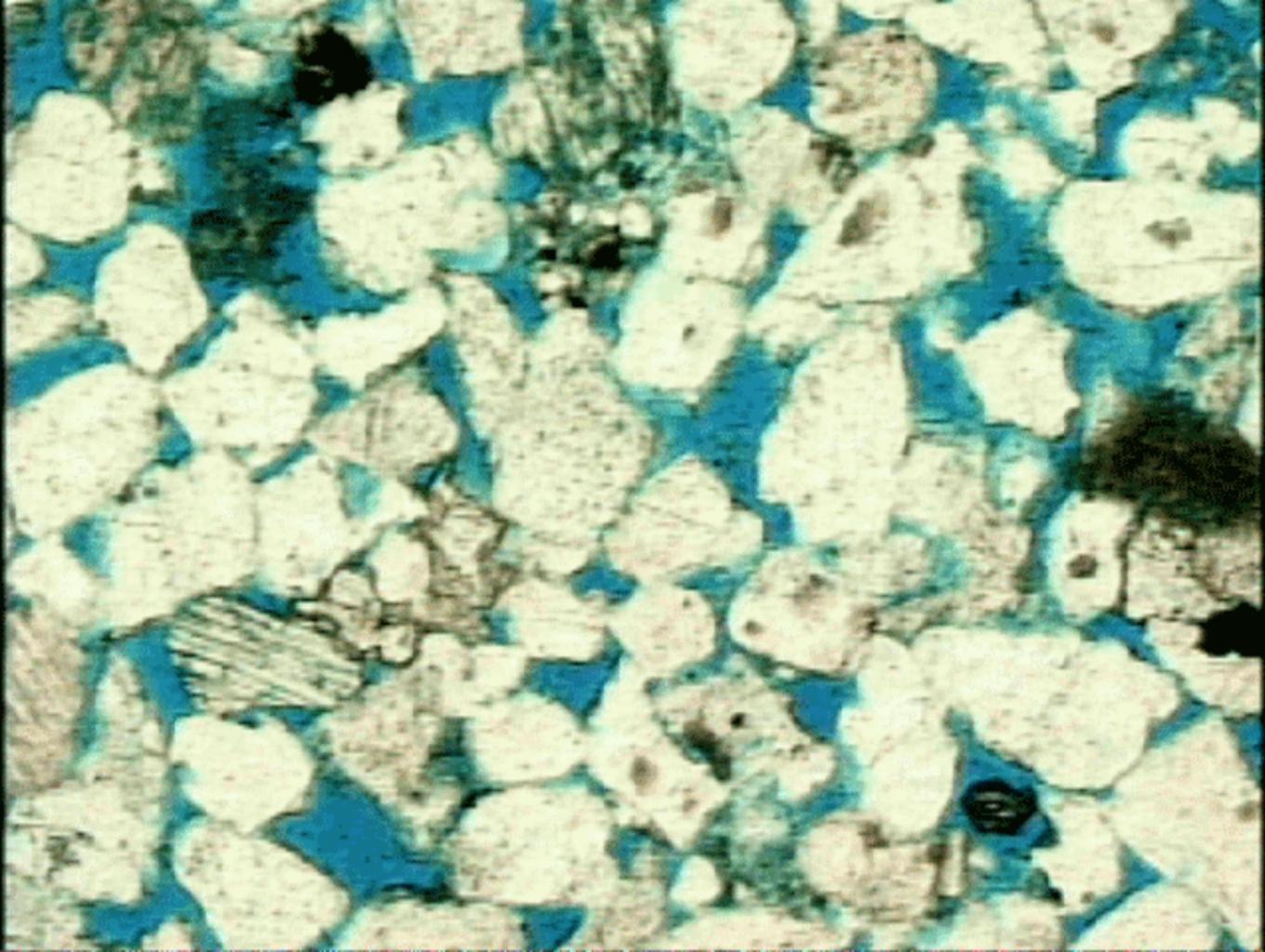




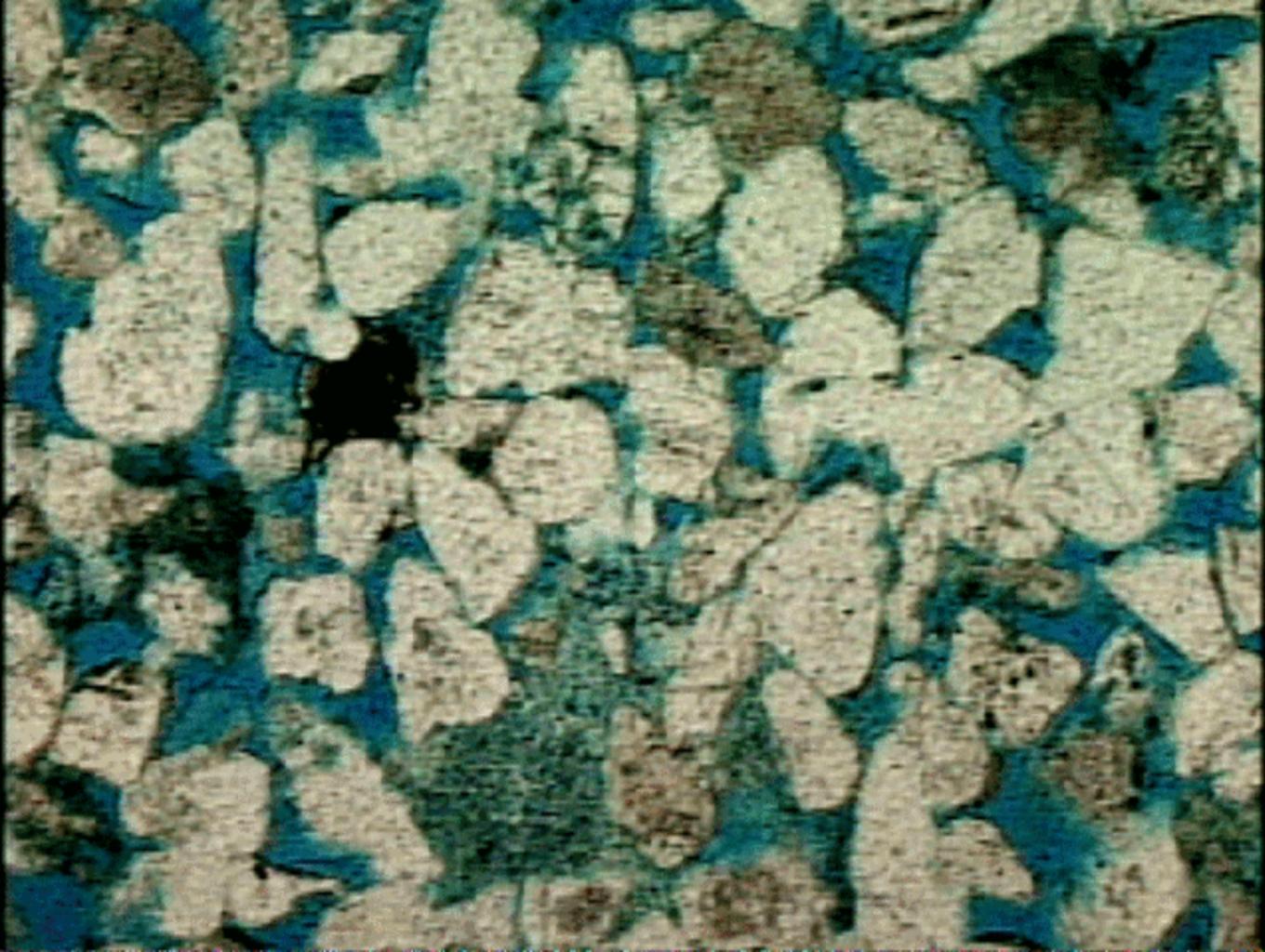




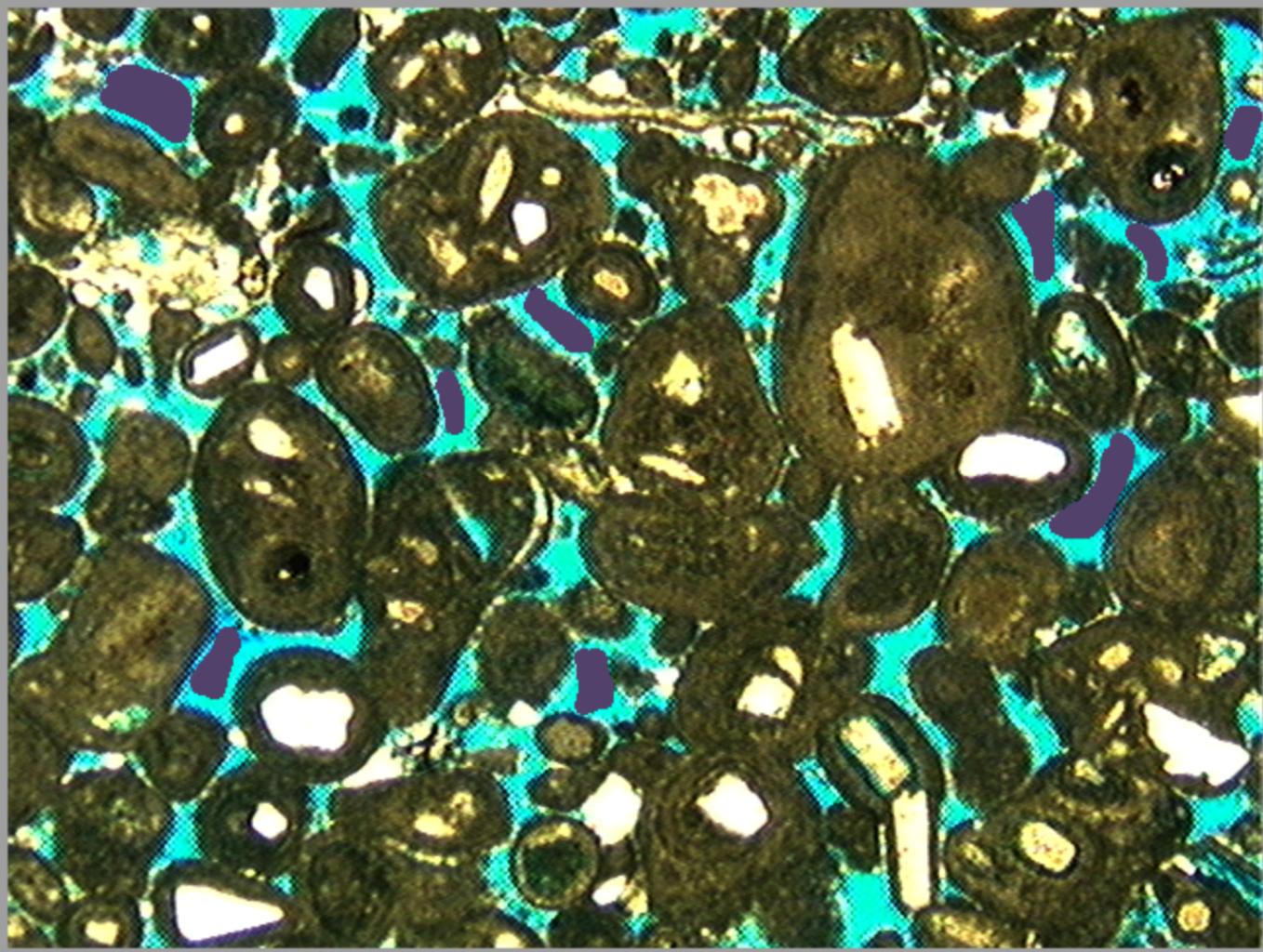


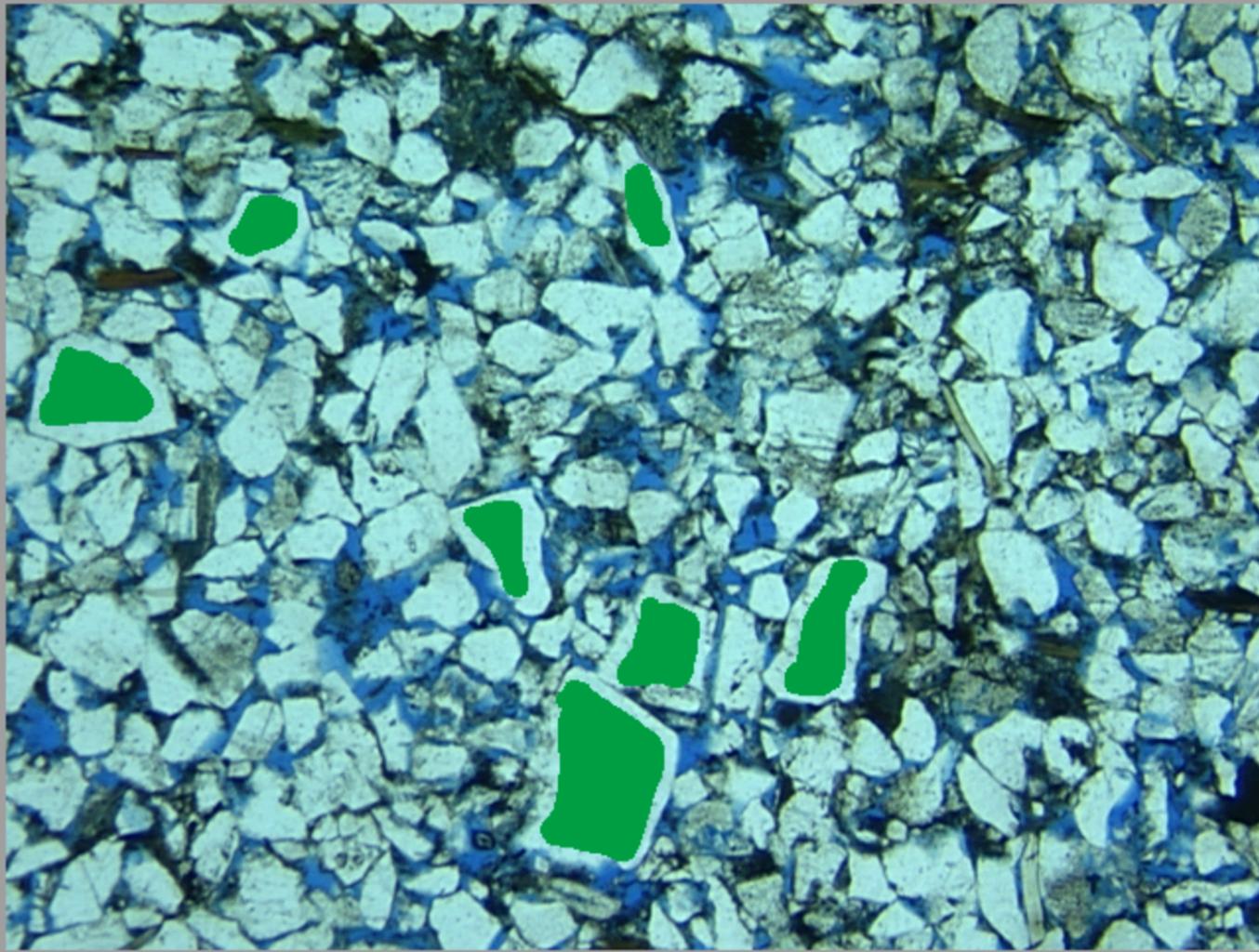


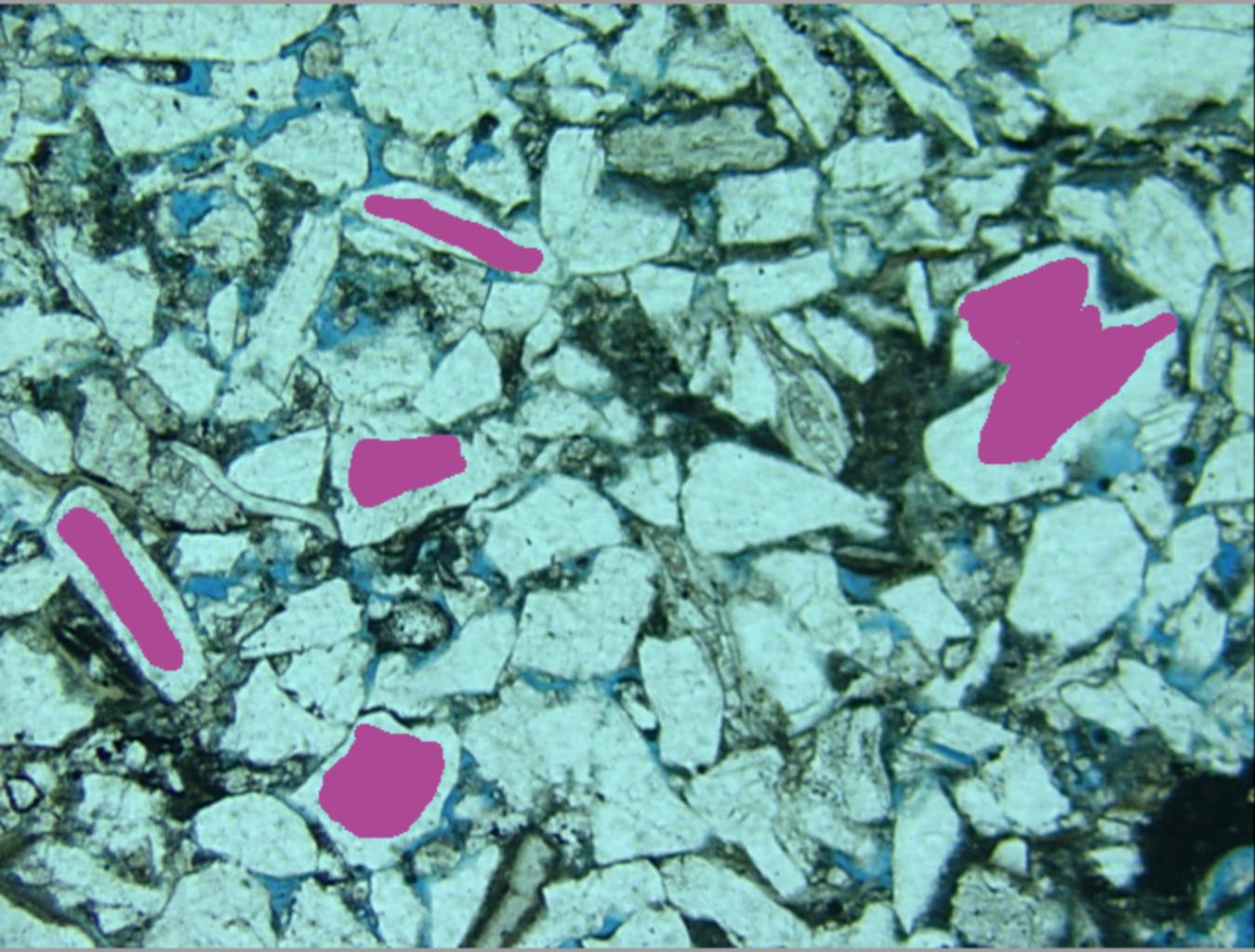


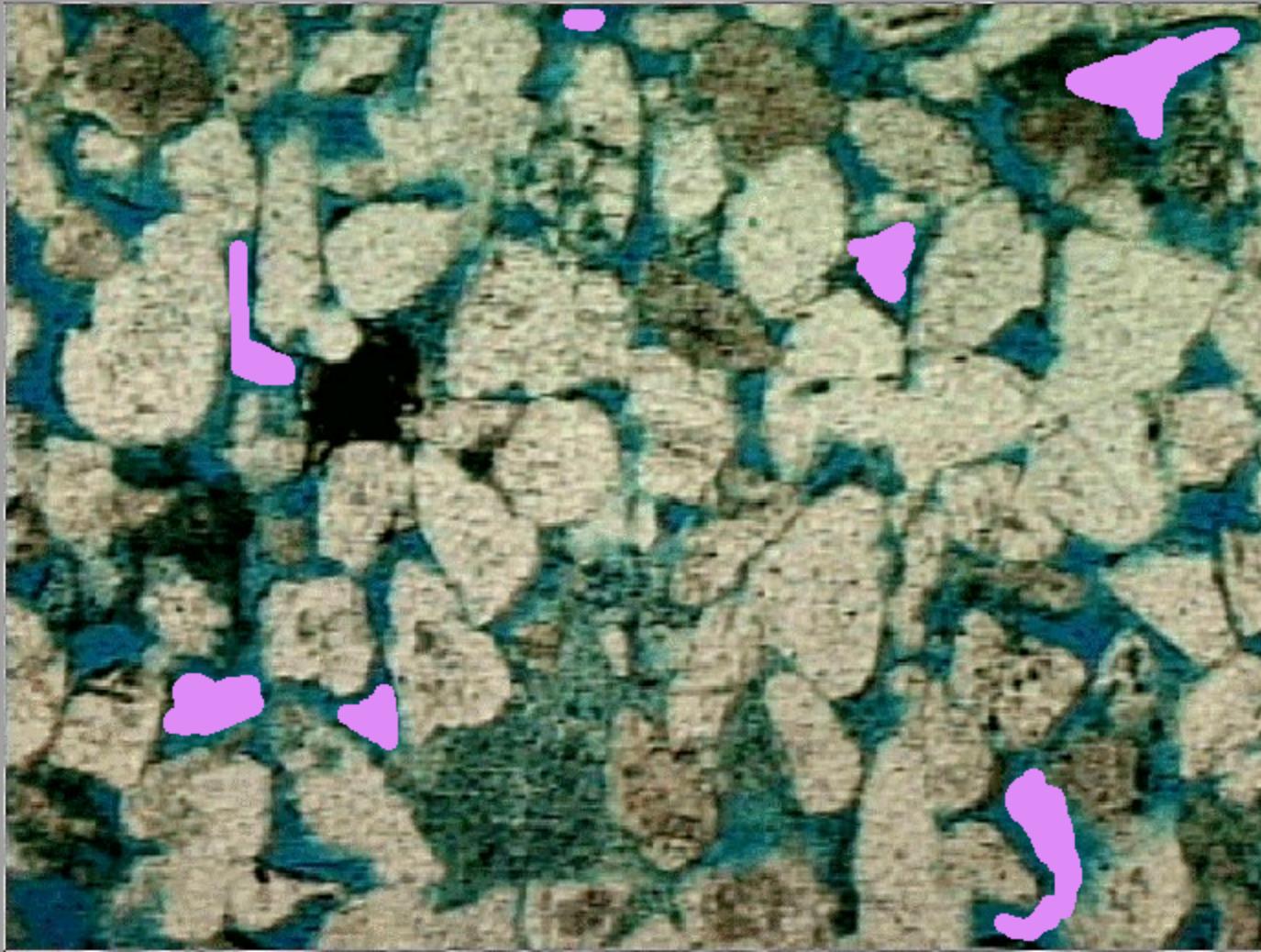


```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <ui version="4.0">
3      <class>RockImageUI::RockImageUI</class>
4      <widget class="QMainWindow" name="RockImageUI :: RockImageUI">
5          <property name="geometry">
6              <rect>
7                  <x>0</x>
8                  <y>0</y>
9                  <width>1300</width>
10                 <height>773</height>
11             </rect>
12         </property>
13         <property name="windowTitle">
14             <string>RockImage</string>
15         </property>
16         <property name="dockNestingEnabled">
17             <bool>false</bool>
18         </property>
19         <property name="dockOptions">
20             <set>QMainWindow::AllowTabbedDocks | QMainWindow::AnimatedDocks</set>
21         </property>
22         <widget class="QWidget" name="centralwidget">
23             <layout class="QGridLayout" name="gridLayout_2">
24                 <item row="0" column="1">
25                     <widget class="QMdiArea" name="openImagesArea">
26                         <property name="sizePolicy">
27                             <sizepolicy hszetype="Expanding" vsizetype="Expanding">
28                                 <horstretch>3</horstretch>
29                                 <verstretch>2</verstretch>
30                             </sizepolicy>
31                         </property>
32                     </widget>
33                 </item>
34             </layout>
35         </widget>
36         <widget class="QMenuBar" name="menubar">
37             <property name="geometry">
```









75	86	51	Solido
138	217	227	Solido
242	244	206	Solido
235	243	205	Solido
228	229	188	Solido
243	249	213	Solido
229	234	199	Solido
4	10	2	Solido
235	243	205	Solido

