

Original article

ERA: A new, fast, machine learning-based software to document rock paintings



Fabrice Monna^{a,*}, Tanguy Rolland^a, Jérôme Magail^b, Yury Esin^c, Benjamin Bohard^d, Anne-Caroline Allard^e, Josef Wilczek^{e,f}, Carmela Chateau-Smith^g

^a ARTEHIS, UMR CNRS 6298, Université de Bourgogne–Franche Comté, 6 Boulevard Gabriel, Bat. Gabriel, Dijon 21000, France

^b Musée D'anthropologie Préhistorique de Monaco, 56, boulevard du Jardin exotique, MC 98000, Monaco

^c Archaeological Research Center of the National University of Mongolia, n°202, 2nd Building, 1 Ikh Surguuli Str., Baga Toiruu, Ulaanbaatar 14200, Mongolia

^d Cadolès, 29 bis rue de l'Arquebuse, Dijon 21000, France

^e Centre André Chastel, UMR CNRS 8150, Faculté des Lettres de Sorbonne Université, 2, rue Vivienne, Paris 75002, France

^f Department of Archaeology, University of Hradec Králové, Rokitsanského 62, Hradec Králové 50003, Czech Republic

^g CPTC, EA4178, Université de Bourgogne, 4, boulevard Gabriel, Dijon 21000, France

ARTICLE INFO

Article history:

Received 16 September 2021

Accepted 16 September 2022

Keywords:

Colour channel
Machine learning
Whitening transformation
Colour space
Open source

ABSTRACT

The present study proposes a new software program to help researchers identify rock paintings from digital images, rapidly producing high-quality documentation, in a user-friendly way. The three RGB colour channels of the digital image are first decorrelated and then stretched, a well-known technique used by remote-sensing specialists for over thirty years. In contrast with the approaches previously developed specifically for rock art, several data-whitening algorithms are used at this step: (regular) principal component analysis, zero-phase component analysis, Cholesky decomposition, and independent component analysis. These transformations produce different arrangements of the colour information, which nevertheless share some important properties (e.g. the covariance matrix of the new channels equals the identity matrix). The decorrelated data, previously stretched and scaled to fit the RGB space, are then converted into various colour spaces (selected from among the most popular): XYZ, HLS, HSV, LAB (CIELAB), Luv, CMY(K), YCrCb, and YUV. The most subtle colour variations will be better perceived in some of these newly produced, contrasted, false-coloured images. The researcher can then take advantage of supervised machine learning algorithms to isolate painted figures. At this step, binary pixel classification is performed either by logistic regression, support vector machine, or *k*-nearest neighbours, possibly including confident learning. There is no need for complex tuning at any point during the procedure, which lasts a few minutes at most, while *a posteriori* cleaning of the produced document is minimal. The software, written in Python, is provided as a stand-alone executable program for Windows, for broader diffusion, and as open-source code, which can therefore be adapted to the evolving needs of the community.

© 2022 Elsevier Masson SAS. All rights reserved.

1. Introduction

One of the main difficulties encountered in the study of rock painting is that of clearly seeing painted patterns with the naked eye. This problem is particularly true when pigments have suffered from decay and fading over time [1]. Colour contrast between paintings and background may have almost completely vanished, so that recognition of shape, texture, contour, and position (i.e., elements essential for analysis) becomes problematic. The introduction of consumer-grade digital CMOS cameras in the late 1990s was a game-changer, as these tools rapidly became popular for the

non-invasive documentation of rock paintings (see [2] for a summary of method evolution). Common 24-bit colour images produced by cameras and processed by computers are basically built from three channels: red (R), green (G), and blue (B). Each is encoded in 8 bits, resulting in 256 integers (i.e. 2^8). This colour model is commonly represented by a cube [3], where each axis is a colour channel (with integer values ranging from 0 to 255), and where black and white, respectively correspond to the (0, 0, 0) and (255, 255, 255) triplets. The additive combination of these three channels forms a palette containing 16.7 million colours. Even though this colour palette is vast enough to cover the full range of human colour perception, estimated at 10 million different colours, the close examination of a raw digital image displaying a rather homogeneous painted panel, which may, in addition, have suffered

* Corresponding author.

E-mail address: Fabrice.Monna@u-bourgogne.fr (F. Monna).

from weathering, does not provide more information than a close inspection of the scene itself. Researchers have therefore taken advantage of the post-processing capabilities offered by raster graphics editors (e.g. rescaling colour, manipulating saturation, etc.) to artificially enhance the contrast between the painted figures and the background [4] (note that a similar approach that has recently been applied to rock art carving by manipulating 3D geometry in place of colour [5,6]). In 2005, the introduction by Jon Harman of DStretch® [7] led to a key breakthrough in methods for recognizing and recording painted rock art [8]. This software program benefitted from decorrelation stretching procedures previously developed by remote-sensing specialists to analyse satellite images [9,10]. Its underlying principle is quite simple: in the RGB space, colour channels tend to show high levels of correlation, so that the pixels composing the image occupy a reduced space in the RGB cube model. The goal is therefore to expand the volume of this envelope in order to artificially enhance the colour richness of the image. Such a boost is expected to facilitate differentiation of the painted parts from the background. According to the DStretch® documentation [7], the colour enhancement of the images operates in four steps. The first step relies on the decorrelation of the colour channels using a Karhunen-Loève transformation, also known as KLT or Hotelling transform, which is in fact a principal component analysis (PCA) [11], based on the covariance matrix [12], although the correlation matrix may also be used at this step [7]. The resulting variables are therefore uncorrelated in a new orthonormal basis in \mathbb{R}^3 . The second step is stretching, which rescales each component by normalizing the variances of the eigenvectors. The third step is a rotation of the coordinate system, which brings the transformed data back into their original (RGB) vector space. At this point, the transformed RGB values, remapped on to a 0–255 range, are converted into other colour spaces, including new ones specifically designed to emphasize yellow, red, or blue/black pigments. Once the painted figures appear, they can be isolated semi-automatically using a colour-picker tool, together with an appropriate tolerance value to determine how closely the pixels composing the entire image must match the colour picked in order to be finally selected [2]. They can also be traced manually with the help of a raster or vector graphics editor, using the false-coloured image thus produced as a background layer. The DStretch® program is distributed as a plugin compiled for ImageJ, an open-source multiplatform software for image processing [13]. More recently, Rogerio-Candelera and co-workers [14] used only a PCA to produce a false-colour image with three uncorrelated bands. These bands correspond to the three principal components (PCs) obtained after shifting and rescaling the values in a range compatible with 8-bit encoding (see also [1] for an example of PCA application). As the amount of information accumulated by the PCs decreases from the first to the last component, PC1 is expected to represent most of the variation contained within the original three bands. After close examination of several experiments, both Rogerio-Candelera and co-workers [14] and Domingo and co-workers [1] concluded that the minority components PC2 and PC3 are the most informative for rock art studies. Cerrillo-Cuenca and co-workers [15] took advantage of these observations and developed PyDRA, a distributable, open-source application, which later evolved by including a finely tunable, selective intensification of saturation or intensity [16]. Interestingly, these authors also introduced unsupervised pixel classification by applying a K-means algorithm that is expected to facilitate documentation.

Although no statistical analysis has investigated how rock art specialists preferentially process images, it is reasonable to affirm that DStretch® has been extensively adopted by the community [7,17–24], while a minority of scientists use a non-standard method based on empirical image manipulation with raster graphics editors, such as Adobe Photoshop® (see [25], and references in

[2]). Yet, as noted by Cerrillo-Cuenca and Sepúlveda [16], although DStretch® is generally efficient, its data treatment remains somewhat opaque. This is true not only for the decorrelation stretching phase, but also during the elaboration of new colour spaces. Numerous options are available during the workflow, but the reasons why and when they should be applied and tuned are not always sufficiently well explained. The plugin can be obtained on request from the author under a compiled form, so that any further inspection of the code requires a certain level of expertise.

The PyDRA application possesses several attractive features. Its mathematical basis is clearly described in two published papers [15,16], and the snippets written in Python 2.7 are readily available at Github (<https://github.com/ecerrillo/PyDRA>), under an open-source form. However, as no standalone version is provided, its use implies the installation of a Python interpreter, together with the necessary libraries, which is detrimental to dissemination among a broader audience. Furthermore, the 2.7 version of Python used for its development has now been discontinued; modifications will thus be required to run the code on the current Python version.

2. Research aim

In this study, we aim to provide the community with software (for Extraction of Rock Art, i.e. ERA), to identify rock paintings from digital images, and to produce high-quality documentation very quickly, using the information-retrieval capabilities of machine learning algorithms. Interestingly, the software is provided as a standalone executable for Windows, and as open-source snippets for Python 3.7 (and above). In contrast with previous approaches, decorrelation of the RGB channels uses four different whitening transformations. Once rescaled for 8-bit encoding, and reshaped in colour raster form, images are converted into a variety of colour spaces, selected from among those most commonly used. Optional tunings were carefully limited to a strict minimum, to simplify the process while maintaining high-level outputs. These options should efficiently counteract the deleterious effect of some particular distributions of colour channels on colour enhancement. Fundamental principles are also discussed, because a better understanding of the underlying concepts will help the researcher to optimize the production workflow from image capture to final documentation. The quality of outputs produced by supervised machine learning is evaluated.

3. Material and methods

3.1. Whitening the RGB channels

Principles. Whitening (or sphering) transformations belong to a family of methods for decorrelating and standardizing a set of variables [26]. They are applied, in our case, to enhance the colour richness of the image. An image can be seen as a data matrix, $\mathbf{X} \in \mathbb{R}^{n \times p}$, where each row, n , corresponds to a pixel, and $p = 3$ because there are three colour channels (RGB). Note, however, that pixel colour is encoded using a combination of integers. The first step is to centre the data:

$$\mathbf{X}_c = \mathbf{X} - \boldsymbol{\mu}, \quad (1)$$

where $\boldsymbol{\mu}$ corresponds to the vector mean of each channel. The resulting data matrix, $\mathbf{X}_c \in \mathbb{R}^{n \times p}$, possesses a zero mean and has a covariance matrix, $\text{cov}(\mathbf{X}_c) = \boldsymbol{\Sigma} \in \mathbb{R}^{p \times p}$:

$$\boldsymbol{\Sigma} = \frac{1}{n-1} \mathbf{X}_c \mathbf{X}_c^T \quad (2)$$

with n , the number of data samples (i.e. pixels). The goal is then to find a $p \times p$ whitening matrix, namely \mathbf{W} , satisfying:

$$\mathbf{Z} = \mathbf{W} \mathbf{X}_c, \quad (3)$$

with:

$$\text{cov}(\mathbf{Z}) = \mathbf{I}_p, \tag{4}$$

where \mathbf{I}_p is a $p \times p$ identity matrix, in other words a situation where all variables are decorrelated and possess a unit variance, and where \mathbf{Z} is the new, transformed data matrix. It can be demonstrated that the expressions above are equivalent to:

$$\mathbf{W}^T \mathbf{W} = \mathbf{\Sigma}^{-1} \tag{5}$$

The \mathbf{W} matrix is not unique, as any $p \times p$ orthogonal matrix \mathbf{Q} with $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}_p$, produces:

$$\mathbf{W} = \mathbf{Q} \mathbf{\Sigma}^{-1/2}, \tag{6}$$

where the \mathbf{W} matrix satisfies Eq. (5) above. Geometrically, this involves a multivariate rescaling by $\mathbf{\Sigma}^{-1/2}$, followed by a rotation by \mathbf{Q} . To go further, the \mathbf{W} matrix can be rewritten using the eigendecomposition of the covariance matrix, $\mathbf{\Sigma}$, in the eigenvector matrix, \mathbf{U} , and the eigenvalue matrix, $\mathbf{\Lambda}$:

$$\mathbf{\Sigma} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T \tag{7}$$

Eq. (6) can be rewritten as:

$$\mathbf{W} = (\mathbf{Q} \mathbf{U}) \mathbf{\Lambda}^{-1/2} \mathbf{U}^T \tag{8}$$

Finally, one obtains:

$$\mathbf{Z} = \mathbf{W} \mathbf{X}_c = (\mathbf{Q} \mathbf{U}) \mathbf{\Lambda}^{-1/2} \mathbf{U}^T (\mathbf{X} - \boldsymbol{\mu}) \tag{9}$$

To summarize, the data, centred using their vector mean $\boldsymbol{\mu}$, are rotated by \mathbf{U}^T to align with the eigenbasis, scaled by $\mathbf{\Lambda}^{-1/2}$, and (possibly) rotated again by $\mathbf{Q} \mathbf{U}$. Note that without this final rotational step by $\mathbf{Q} \mathbf{U}$, the data are already whitened as they are decorrelated, and each dimension possesses a unit variance.

Algorithms. The issue is therefore to find relevant configurations of \mathbf{W} , and thus an appropriate value for \mathbf{Q} . Five possible (natural) sphering procedures have been discussed [26]. The most common procedure, PCA whitening, simply uses $\mathbf{W} = \mathbf{\Lambda}^{-1/2} \mathbf{U}^T$. Note that, for PCA whitening, \mathbf{W} is not unique, as \mathbf{U} suffers from sign ambiguity. The ZCA whitening (aka Mahalanobis whitening, [27]) is unique and implies the use of $\mathbf{W} = \mathbf{\Sigma}^{-1/2}$; in other words, $\mathbf{Q} = \mathbf{I}_p$. In contrast with the above method, the data are rotated back by \mathbf{U} to their original coordinate system. The Cholesky whitening uses Cholesky factorization, well known in numerical computation [28]. It decomposes a positive-definite matrix into the product of a lower triangular matrix and its conjugate component. Here, the precision matrix, another name for the inverse of the covariance matrix, $\mathbf{\Sigma}^{-1}$, is decomposed as: $\mathbf{\Sigma}^{-1} = \mathbf{L} \mathbf{L}^T$. In this case, the whitening matrix \mathbf{W} is unique and equals \mathbf{L}^T . Two additional procedures were presented by Kessy and co-workers (2018) [26], which either maximally compress the original variables (PCA-cor) or are maximally similar to them (ZCA-cor). They were nevertheless discarded here, as both produced almost identical results to either PCA or ZCA whitening for all images tested. One other algorithm was considered: independent component analysis (ICA), as it was recently and successfully applied to rock paintings [29]. It starts from data whitened with PCA, which are rotated to maximize the non-gaussianity of their projection on to 3 new axes, thus producing a new space where the data are maximally independent (see [30] for more details). This transformation can be accomplished using the FastICA algorithm, here following the implementation found in [31]. As a final step, and whatever the procedure applied, the $\mathbf{Z} \in \mathbb{R}^{n \times p}$ matrix is converted into the shape of the original raster, and the values are scaled to an integer 8-bit format to fit the 0 - 255 range.

Optional further colour enhancement. Two different approaches, which are not mutually exclusive, can be applied to further enhance colour distribution. The first consists in rapidly selecting a part of the image by hand, taking care to focus on the area of interest. The vector $\boldsymbol{\mu}$ and the $\mathbf{\Sigma}$ and \mathbf{W} matrices are computed from

this subset of selected pixels, \mathbf{X}_s , and upon this basis \mathbf{Z} is drawn for the entire image. Importantly, the remapping of \mathbf{Z} in 8 bits takes into account the minimum and maximum values found in the subset, and not in the entire image, which is subsequently clipped of its values below 0 or above 255. The second approach consists in a linear contrast stretching with saturation, also known as percentage linear contrast stretch [32], where \mathbf{Z} is rescaled to include all values that fall within an interval corresponding to the $(\alpha/2)^{\text{th}}$ and $(100-\alpha/2)^{\text{th}}$ percentiles (with α configurable).

3.2. Converting the whitened data to other colour spaces

The four resulting images (one for each whitening procedure) are then converted into each of the eight following colour spaces: XYZ, HLS, HSV, LAB (i.e. CIE L^*a^*b), Luv (i.e. CIE L^*u^*v), CMY(K), YCrCb, and YUV, which were specifically chosen as they are among those most commonly used. Describing each of these colour spaces and their respective properties is outside the scope of the present study, but the reader will find this information in the vast specialized literature (e.g. [33,34]), while the mathematical description of the transformations performed by OpenCV is available at https://docs.opencv.org/3.4/de/d25/imgproc_color_conversions.html. As a result, 36 colour images are produced: 4 whitening procedures \times 9 colour spaces (i.e. one whitened RGB image plus its conversion into 8 other colour spaces). The reason for this abundance is to supply a large set of false-coloured images, through which the operator can easily navigate to identify the painted areas, without any fastidious manual tuning.

3.3. Machine learning for the automatic delineation of painted areas

Underlying idea. The integration of machine learning aims to produce documentation close to the best standards of publication, with minimum intervention by the operator. Rather than the unsupervised method previously applied [16], a supervised approach was preferred here, because the operator can train the model very specifically to fit the desired objective [35]. Note that Lerma [36] previously tested a similar form of supervised classification, taking as feature input the three visible spectral bands of an image plus one near infrared. In the present study, a simple binary classification (painted vs non-painted) is presented first, while situations with multi-coloured paintings will be discussed later.

Supervision. This step consists in labelling two classes: pixels belonging to the painted areas, and those belonging to the substrate. Ideally, supervision should be at pixel level to minimize the risk of erroneous labelling [37]. This pixel-wise task would be quite tedious, particularly since good representativity is sought. Consequently, supervision is performed by drawing two groups of curves (of adjustable thickness) on the image, each at best corresponding to one class. However, such rough-and-ready supervision will inevitably lead to some errors (i.e. pixels erroneously included in both classes).

Additional feature engineering and feature selection. To each pixel coordinate corresponds a vector containing 108 values (36 images \times 3 colour channels). Processing this entire dataset as input is certainly not the best option, as it includes variables without any discriminating power (i.e., channels where no difference can be noticed between painted areas and substrate). It would considerably slow down the training and prediction phases, and might even have a deleterious effect on output quality. Automatic preselection therefore retains the n most discriminating variables, possessing the n highest values of the t -statistic (n is configurable, with $n = 30$ channels as default value). It is worth mentioning that even if the distribution within each group is non-normal, variable-by-variable, the two-sample t -test can be used at this step, as the central limit theorem applies when sampling is sufficiently large

[38]. Using thick curves for supervision will always produce thousands of labelled pixels. The n variables thus retained are nevertheless strongly correlated, so that a PCA must be performed. This reduces the number of (decorrelated) principal components to m ($m < n$), allowing the total variance to be expressed at a given level (also configurable by the operator, with 97% as default value).

Algorithms. At this point, the aim is to build a mathematical model (i.e. a decision rule) allowing binary pixel classification. In other words, the two categories: y_c , (with $c \in \{0;1\}$ for background and painted pixels, respectively) must be predicted from \mathbf{x} , a vector describing the set of m previously built features, $\mathbf{x} = \{PC_1, \dots, PC_m\}$. There are several different techniques for solving such a classification problem, among them naïve Bayes, k -nearest neighbours, logistic regression, linear and quadratic discriminant analysis, artificial neural network, and random forest, to cite only the most popular. They can also be combined using voting classifiers. Many competitions open to both researchers and enthusiasts seek to find the technique producing the highest performance for a specific problem, using a given dataset. The winning workflow generally involves complex algorithms, fine hyperparameter tuning, and heavy calculation load. Even though its performance will formally surpass that of more classical algorithms, the gain is often modest. Here, the idea is not to find the best algorithm through a long and complex optimization process (as manual cleaning of the resulting image will undoubtedly be necessary at the end), but to provide good results quickly. This is particularly true in our case, since the best algorithm is likely to vary with the dataset (i.e. the image processed) and the choices made by the operator during supervision. For these reasons, only three of the simplest approaches are implemented in the software: logistic regression (LR), support-vector machine (SVM), and k -nearest neighbours (KNN).

For logistic regression, a new variable, z , is built from a linear combination of weights, \mathbf{w} , and sample features: $z = \mathbf{w}^T \mathbf{x}$. The probability of belonging to a particular class is provided by the logistic sigmoid function: $\Phi(z) = 1/(1 + e^{-z})$, with $\Phi(z) \in [0; 1]$; weights are optimized using the L-BFGS algorithm; regularisation is operated using an L2 penalty. The support-vector machine algorithm aims to maximize the margin between the decision boundary hyperplane and the closest training samples with the goal of separating the two classes. The SVM can also manage data that are not linearly separable using the so-called “kernel trick”, which transforms the data into a higher dimensional space. Here, a radial basis function (RBF) is applied. The algorithm provides the probabilities of belonging to each class. Finally, the KNN algorithm is a non-parametric machine learning technique, which determines the k -nearest neighbours (here $k = 31$) and simply assigns a class after majority voting.

For more details about these methods, the reader is invited to refer to the abundant specialized literature, e.g. [39–41]. Once the classifier is trained, in other words once the mapping function $f(\mathbf{x}) \rightarrow y_c$ is optimized, the model is applied to the entire image, and predictions are made for each of the pixels composing the image to identify its class: background or painted pixel? For the sake of speed, the traditional split into training and test datasets (the first used to train the model, and the second for performance evaluation), possibly including a grid search for hyperparameters, was not applied. Since the supervised dataset is very large and the complexity of classifiers low, models do not suffer from overfitting, and thus, from difficulties in successfully generalizing to unseen data [42]. Assessing performance scores would also be pointless in our case, since outputs are visually evaluated. By default, the LR model is applied to the entire set of supervised pixels. This model generally produces good results quite quickly. Its outputs may thus rapidly provide valuable information about how the operator could improve final documentation quality: e.g., by increasing the number of supervised pixels, thus focusing on a specific area of the

image, or by changing the default settings used for feature engineering (e.g., number of best channels retained, variance expressed by PCs, etc.).

Confident learning. Recent methods have been specifically developed to identify potential noise in the labelling mentioned above, and to integrate dealing with this common issue into the learning process (see [43] for more details). Noisy data are pruned using probabilistic thresholds, the level of noise is estimated, and examples to train the models are ranked with confidence. This confident learning approach is integrated into the ERA software to complement the more conventional learning processes.

Image blurring. In the output images, a few pixels may erroneously stray beyond the segmentation borders of painted areas. Applying a Gaussian blur to images may help to eliminate such false positives, by smoothing out image noise, but at the expense of detail. By default, a square 5×5 px kernel is applied, but its size can be modified.

3.4. Practical implementation

The pixel envelope volume within the RGB cube was calculated using the free R programming language (<https://www.r-project.org/>), with the tiff, misc3d, oce, and RGL libraries, using a three-dimension kernel density estimate, with a cut-off of 99.9%. The ERA software itself is written in Python 3.7, with the numpy, scipy, scikit-learn, qimage2ndarray (<https://github.com/hmeine/qimage2ndarray>), OpenCV, and PyQt5 (or PySide) libraries, together with the whiten function (modified after <https://gist.github.com/joelouismarino/ce239b5601fff2698895f48003f7464b>). It is made available, together with a standalone executable version for Windows, as Supplementary Material SM1, at the following internet address: https://gitlab.huma-num.fr/fmonna/era-extraction-from-rock-art/-/tree/Paper_Supp_Mat. Fig. 1 depicts the layout of the page dedicated to data preparation. General principles, options, and workflow are discussed in detail below, and in the accompanying user manual, included in Supplementary Material SM1.

3.5. Visual evaluation

During the elaboration of the ERA software, dozens of images were processed. Outputs were evaluated visually, by rock art experts, based on the time required in post-processing to attain the highest standards of archaeological documentation.

4. Results and discussion

4.1. Basic workflow

The rock art painting used here as the first example belongs to the Okunev culture, which existed in the South Siberian steppe during the Bronze Age (2600–1800 BC). It represents the face of a deity on a brownish rock, with orange lichen colonizing some parts of the surface, and dark cracks. The image (3888×2592 px, saved in high-quality JPG format) was captured at Tarpig Mountain, as orthogonally as possible to the panel surface, using a DSLR CANON EOS 400D camera (Fig. 2a). The 3 channels composing the image are strongly inter-correlated, as illustrated by their projection in the RGB cube (Fig. 2a). The envelope of the input picture occupies only about 7% of the full RGB colour space, explaining why the image appears so homogeneous. When whitening is applied, the envelope increases considerably, occupying more than 21% of the cube, whatever the procedure (Fig. 2b), with slight differences because stretching is performed after the rotational step. Colour richness is basically the same in all cases, but the resulting images are not all equally useful to identify the painted areas.

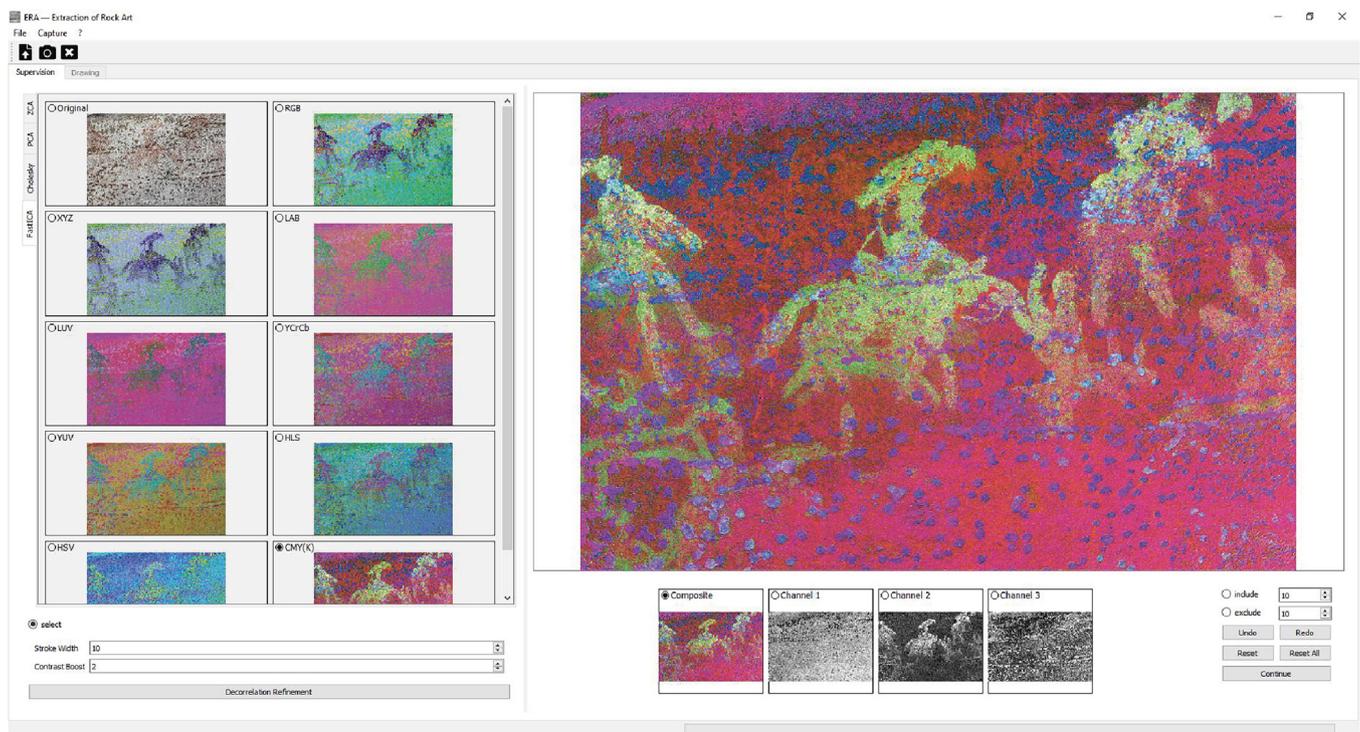


Fig. 1. Layout of the ERA software. The processed image represents several images of the Xiongnu - Xianbei period; in the middle, a rider with an eagle over an earlier image of an animal in a different shade of red (original colour). The image was captured at Kavkazskoe, South Siberia, Russia.

Here, the Cholesky and FastICA transforms seem to be the most appropriate. Note that FastICA produces results visually different at each run, due to its stochastic nature, but that the data structure remains unchanged (i.e. the three channels are sometimes rearranged and/or the values reversed). Interestingly, Cerrillo Cuenca and co-workers (2021) reported that decorrelation by FastICA often isolates paintings more sharply than PCA, probably because ICA seeks to separate superimposed components, while principal components are computed in order, according to the amount of variation they represent [29]. After colour conversion, the operator can rapidly identify the best colour image(s) and/or one specific grey channel by toggling between images, and can also supervise pixel labelling, with black for painted areas, and red for background (Fig. 2c). Features corresponding to the values of supervised pixels for each channel are extracted, the best channels are identified and selected, PCA is computed, and the machine learning model is trained. For the Okunev painting, applying the trained LR model to the entire set of pixels composing the image produces good results. The painted areas are emphasized in black or in their true colour with no (i.e. white) background, or in black on the original colour image (Fig. 2d). Another output is systematically produced using confident learning (not shown in Fig. 2). Most of the time, no great difference was observed between results with or without confident learning. In some situations, however, confident learning clearly produced better results, probably because the training set contained too many mislabelled pixels. During these tests, it was also noticed that the slower KNN model generally produced outputs similar to LR. The SVM model often provided outputs with a little bit less noise, but such improvements require more computation time (several minutes may be necessary). As each image possesses its own characteristics, the final choices are left to the researcher's appreciation, based on visual evaluation by the expert, for more versatility. The default parameters are nonetheless appropriate as a first guess, in most situations.

4.2. Optional colour enhancement

Depending on the distribution of the input pixels, the beneficial effect of the sphering transformation described above may be modest. Just a handful of pixels falling close to the corners or the edges of the RGB cube (unrelated to the painted area) would be sufficient to impair the colour enhancement of the entire image. The presence of outliers may have a drastic impact on the mean, the covariance matrix, and thus subsequently on the PCA transformation, but above all on the final rescaling in 8-bit format, which linearly remaps the Z matrix by making the minimum value coincide with 0, and the maximum with 255.

The Okunev painting from the Tarpig Mountain is used again here to show the effect of the two options available for further colour enhancement (Fig. 3), in comparison with the default processing depicted in Fig. 3a, all using the Cholesky whitening procedure. For subset-based enhancement, the operator needs to manually select areas covering both the painted areas and the substrate around the figure of interest (Fig. 3b). In other words, the orange lichen growing on the rock must be set aside, as must the cracks and vegetation behind the painted panel. As whitening and stretching parameters applied to the entire image are computed on the basis of this selection alone, the influence of troublesome elements in the rest of the image is somewhat minimized. Consequently, the pixels occupy a larger volume, $\sim 33\%$ of the RGB cube (versus $\sim 23\%$ for default processing). The painted figure is slightly better identified (Fig. 3b). However, although manual selection was performed carefully, some pixels were incorrectly included (e.g., tiny patches of lichen), and the contrast between the figure and its substrate is not optimally increased. Contrast boosting using linear stretching with saturation produces a much better output, at least in this example (Fig. 3c). With $\alpha = 2\%$, the deleterious effects of (outlier) pixels below the 1st and above the 99th percentiles are removed within each RGB channel. Since these troublesome pix-

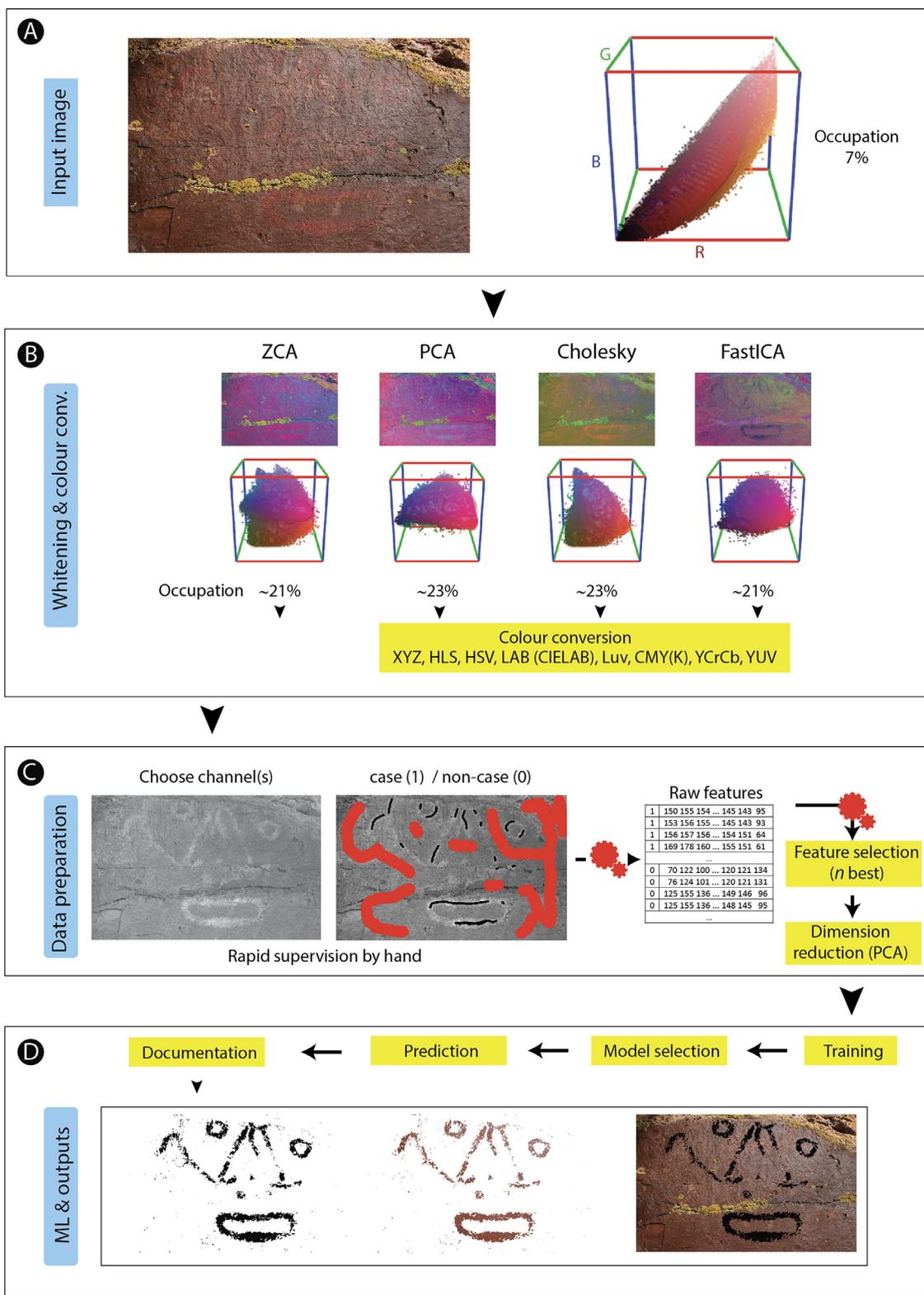


Fig. 2. Typical workflow with the ERA software. (a) Original image of an early Bronze Age rock painting from Tarpig Mountain (South Siberia, Russia), and the position of the pixels composing the image in the RGB cube; (b) whitening and colour conversion step, with the projection of pixels in the RGB colour space; (c) data preparation, including selection of one or several channels from which the supervision is operated; manual supervision on the image (painted areas in black, and substrate/background in red); feature engineering; (d) training, prediction, and production of the final documentation. Occupation, expressed as a percentage, represents the part of the RGB cube filled with image pixels.

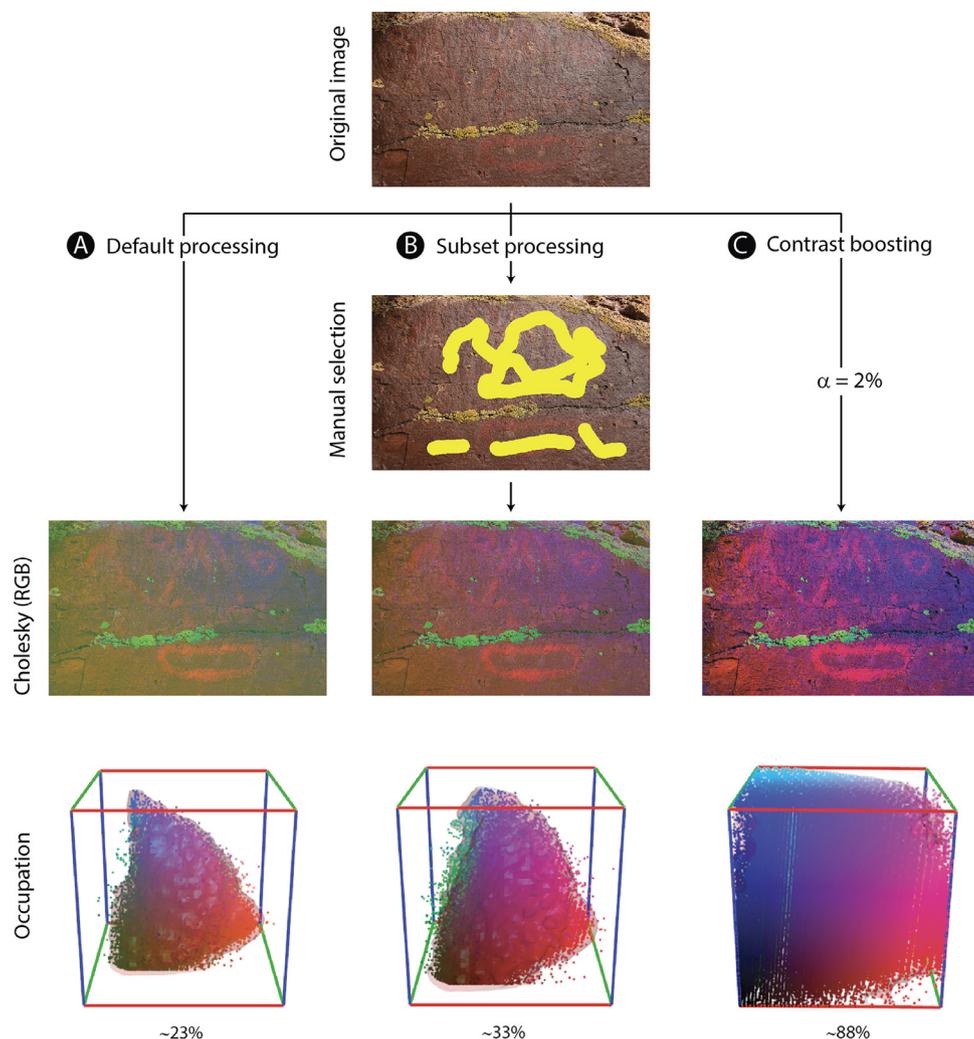


Fig. 3. The effect of optional further colour enhancement compared to the standard procedure. (a) default settings; (b) manual selection of pixels on which whitening parameters are computed; (c) contrast boosting by linear stretching with saturation. For each processing method, the resulting projection of the pixels in the RGB cube is provided, as well as the volume of their envelope, expressed as a percentage of the whole.

els are not very numerous, $\alpha = 2\%$ is sufficient to increase the pixel envelope in the RGB cube to 88%. The figure can then be seen much more clearly (Fig. 3c).

A synthetic image was constructed to investigate which colour enhancement method should be applied in a challenging context, where extraneous elements are included in the scene (Fig. 4a, left). The background was first filled with a pinkish RGB colour, set to (155, 85, 102). Two texts written in similar colours were then added in the foreground: 'ERA' in the bottom left using (151, 90, 100), and 'Rock Art' in the top right using (155, 86, 105). Noise was then added to the image to mask the contours of the letters, making the texts barely or even not at all distinguishable, especially for 'Rock Art', where the colour is very close to that of the background. Finally, four uniform relatively large rectangles were also added to mimic the possible presence of lichen (yellow), shadows (black), and bird droppings (whitish) on the rock substrate, or the inclusion of some sky (light blue) in the image. Note that such a configuration also takes into account the possible presence of a black and white scale, often integrated in field photographs. The pixels constituting the original image only take up ca. 1% of the RGB cube (Fig. 4a, left). The situation does not improve with the default processing, because the presence of the four coloured rectangles ruins the effect of the whitening procedure (Fig. 4b, left). The text 'ERA' is however distinguishable in several channels (e.g.,

the HSV channel 2 after Cholesky whitening; Fig. 4b, right), making manual supervision and extraction of the text possible, with only a low amount of noise. The 'Rock Art' text cannot be perceived. The same observations are made when linear stretching with saturation is applied. In fact, $\alpha = 2\%$ is not sufficient to remove the deleterious influence of the coloured rectangles, which cover too large a part of the image (Fig. 4c). Increasing α would be tempting, but not appropriate, since this would lead to saturation (and loss of information) of the pinkish pixels falling close to the cube edge, which remain of interest. Operating whitening from only a selected subset of pixels within the pinkish area is the best choice here. The pixel volume within the RGB cube increases dramatically to $\sim 59\%$ (Fig. 4d, left), 'ERA' is better seen, and interestingly, 'Rock Art' becomes visible on several channels (Fig. 4d, right). Extracting both inscriptions is straightforward, using two separate supervisions (one for each text): 'ERA' is isolated with the same level of precision as previously, while 'Rock Art' is now clearly visible, together with an unavoidable amount of noise, due to the colour proximity of the text with the background. Both isolated inscriptions can then be combined (e.g., in false colour) in a single image using a raster graphics editor (Fig. 4d, right). Superimposition of figures of different colours, such as those described in [44], can therefore be tackled with the ERA software by processing each painting independently.

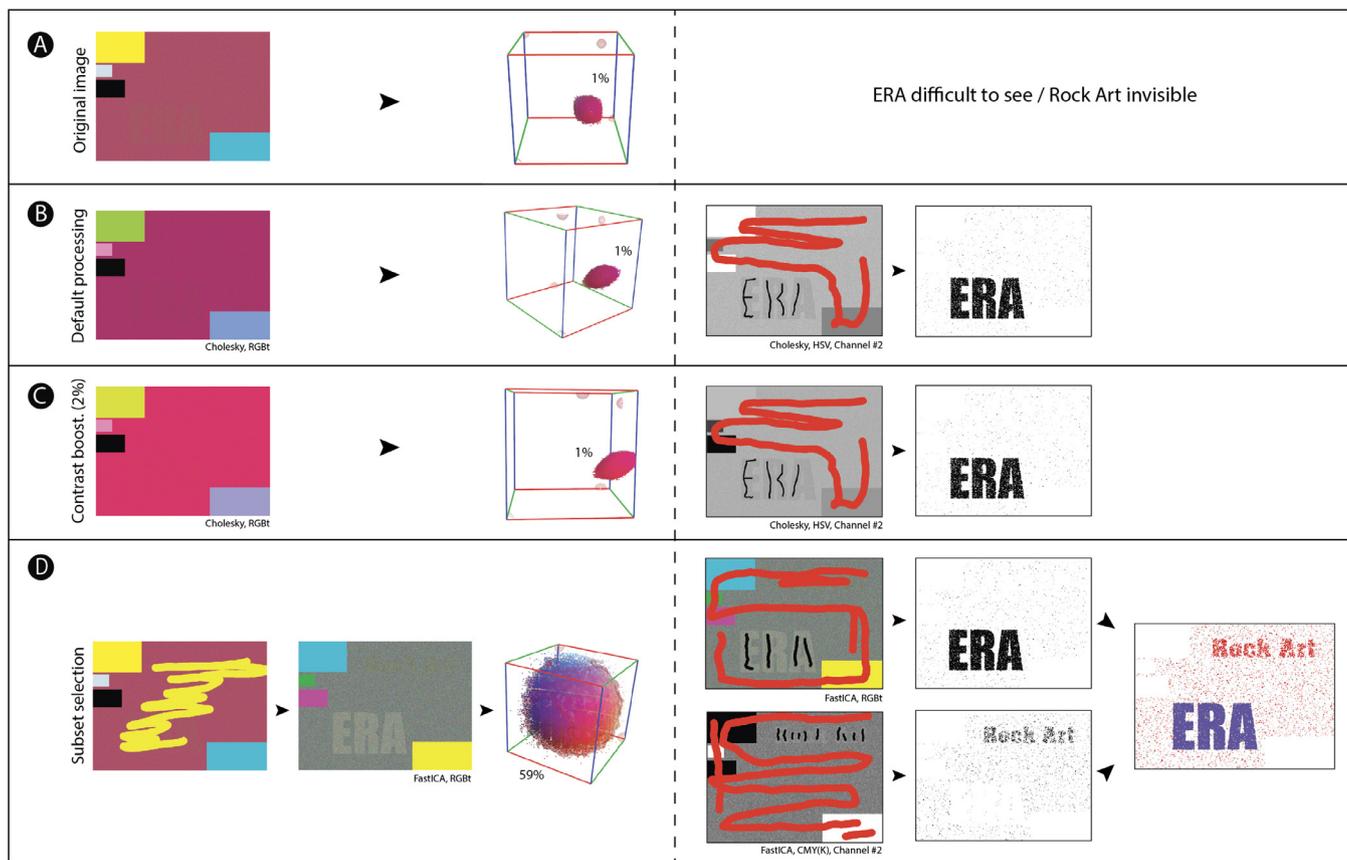


Fig. 4. Colour enhancement of a synthetically produced image, with extraneous elements. (a) the original image and its pixels in the RGB cube (text barely visible or completely invisible); (b) default processing (left); ‘ERA’ is visible and can be extracted by supervised machine learning algorithm (right); (c) contrast boosting (2%, left), and subsequent extraction of ‘ERA’ (right); (d) subset selection of pixels (left), and extraction of ‘ERA’ and ‘Rock Art’ by two different supervisions (right). The values, expressed as a percentage of the RGB cube, represent the volume of the pixel envelope.

4.3. Guidelines for fast and efficient processing

The aim here is not to offer a recipe that could be used in all circumstances, but rather to provide general guidelines to help the researcher obtain the most appropriate results rapidly. During the shooting step, care should be taken to seek the most homogeneous lighting for the scene, because the whitening procedure will be much more efficient in such a situation. Following the same principle, a colour chart, a scale, or anything that has nothing to do with the area to be treated should not be included in the image; such elements introduce outlier pixels. As an example, Fig. 5a was captured at the Shalabolino rock art site, South Siberia, Russia. This rock painting depicts a human-like face, with an oblique cross between eyes and mouth, which overlies a pre-existing rock carving representing a bovid animal. Fig. 5b demonstrates how the presence of a colour chart, which by nature integrates a large spectrum of colours, can ruin any colour enhancement of the painted panel, to such an extent that the original RGB photograph is more informative than the processed version. In such a case, the best choice is to focus on the area of interest, trying to eliminate any worrying elements, as in Fig. 5c. Unfortunately, it was not possible to eliminate the dark area behind the targeted substrate without cropping part of the painted zone. A subset of pixels was therefore selected, as well as contrast boosting, with α set to 1% (Fig. 5d). The face is clearly seen, for example with the ZCA whitening, transformed into the CMY(K) colour space (cf. Fig. 5e). Switching from one channel to another (e.g., those displayed in Fig. 5f–g) will avoid misinterpreting the painted parts and the background during the supervision phase (Fig. 5h). The result is obtained within seconds (Fig. 5i).

Unfortunately, it is difficult to compare this black and white output with that obtained with DStretch® and an associated raster graphics editor, as the rendered quality depends in great part on the operator’s skill in tuning the numerous settings of the DStretch® software, but also in handling the colour picker tool and its associated tolerance value to produce the final binary picture. The face will be as clearly visible with DStretch® (especially with the YRE colour space) as with ERA. However, despite several attempts to isolate the painted figure, the level of noise with DStretch® remained much higher than that observed with ERA. This result is probably due to the use of a single channel, or at best a combination of three (if working on a colour image), whereas ERA applies its machine learning algorithms to an array containing a very large set of carefully selected channels. With PyDRA, the problem is rather that classification is unsupervised, so that models are not designed for tasks as specific as those requiring supervision.

5. Limitations

Poor-quality JPEG images should be avoided whenever possible. It is well known that this popular lossy compression, based on a block-coding scheme, introduces visible artefacts in pixels and at block boundaries, together with notable loss of detail in lowly contrasted, homogeneous areas [45]. These artefacts will appear on several channels after processing, undoubtedly reducing recognition. Ideally, the photographs should therefore be captured in RAW format, and later saved in uncompressed TIF format before processing. If this is not possible for any reason, a high-quality camera should be used, and the photographs should be saved in the

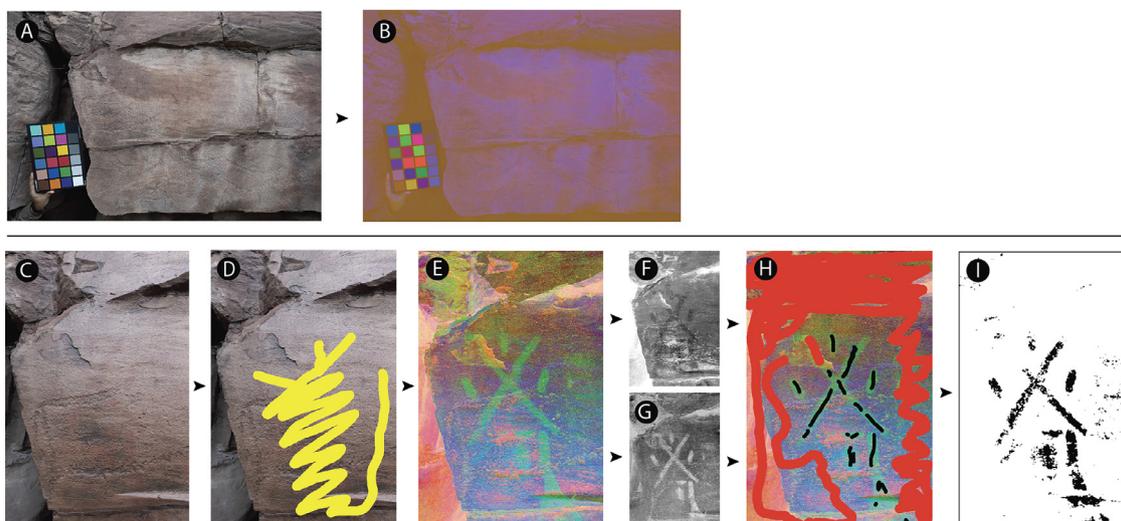


Fig. 5. Processing early Bronze Age rock art paintings from the Shalabolino site, South Siberia, Russia. Pictures were captured in RAW format with a SONY RX100M3, resized to 1500 px for the largest side and saved in TIF format, before processing. (a) extended view of a painted panel, including an X-RITE colour chart, and (b) its processing (Cholesky whitening, RGB); (c) a new picture, taken closer to the rock, avoiding troublesome elements; (d) pixels manually selected, from which the whitening parameters are computed, and (e) then applied to the whole image (ZCA whitening, CMY(K)); the two first channels of this colour image (f,g) are used for supervision (h); the final output using logistic regression (i).

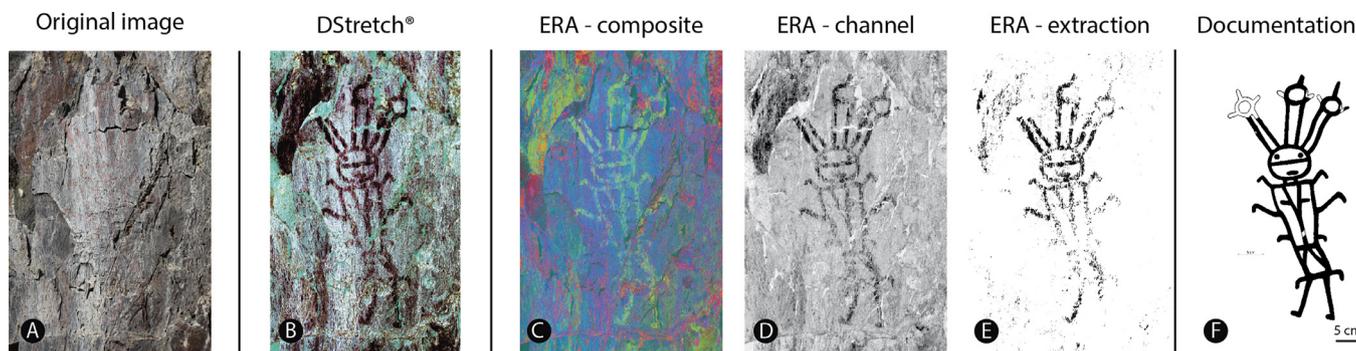


Fig. 6. Processing of an image representing an early Bronze Age human figure. It was captured with a Canon EOS 5D Mark II, at Oya river, South Siberia, Russia. The original picture was 3744 × 5616 px in size and was saved in high-quality JPEG format. For further treatment, it was rescaled to 2000 × 3000 px. (a) original image; (b) DStretch® processing with the YRE colour space; (c) false-coloured image produced by ERA using Cholesky decomposition and YCrCb colour space; (d) channel 1 of the CMY(K) colour space after Cholesky decomposition by ERA; (e) attempt to extract the figure after supervision with the LR algorithm of ERA; (f) final documentation produced manually from ERA outputs. ERA images were produced with a contrast boost of 2%.

highest possible JPEG quality. The ERA software is not demanding in terms of computational resources, since it can run on a consumer-grade computer, delivering outputs within seconds, or at most within a few minutes. Depending on the amount of memory installed, however, resizing very large pictures may be necessary, as more than 100 one-channel images are produced from a single RGB photograph.

The ERA software cannot accomplish miracles: results will depend above all on the degree of pigment alteration. Elements lost over the course of time will never be recoverable. The machine does not seek to replace the expert eye, as many crucial choices must still be made, in particular in terms of interpretation, and hence in the supervision, cleaning, and adjustments of the outputs. For example, when applying machine learning, one should be aware that the quality of the training set used for supervision is of vital importance: it must be truly representative of the whole, as otherwise the classification results will be poor. The noise present on the final documentation can be reduced by manual cleaning, or by increasing the blur radius, but at the price of lower recognition of the pixels composing the painted figure. Note that extraction by machine learning is always an optional step. The operator will often encounter situations where pigments have almost com-

pletely or partially faded, or where superimpositions are so complex that machine learning algorithms implemented in ERA will provide improper or incomplete results. In such cases, documentation can still be prepared with a raster/vector graphics editor, from an appropriate selection of RGB images or individual channels produced by ERA, perhaps by manually completing or cleaning the images. To illustrate such situations, two additional examples are reported, where ERA partly fails to produce adequate documentation by machine learning extraction. The first concerns an early Bronze Age human figure captured at Oya, South Siberia, Russia (Fig. 6). On the original image, the figure is almost invisible (Fig. 6a). Both DStretch® and ERA succeeded in enlightening the painted parts (Fig. 6b, c); they become even clearer when an appropriate single ERA channel is selected (e.g. channel 1 of the CMY(K) colour space after Cholesky decomposition, Fig. 6d). Unfortunately, painting has to some extent vanished over time and most of the pixels composing the figure are disconnected, so that the training set contains too many mislabelled pixels. As a result, the extraction is only partial (Fig. 6e), unusable for the final documentation, which must be performed by manual tracing (Fig. 6f). The second example is an image representing a part of a herd, captured in Oued In Djaren, Tadrat region, Tassili N'Ajjer National Park, Al-

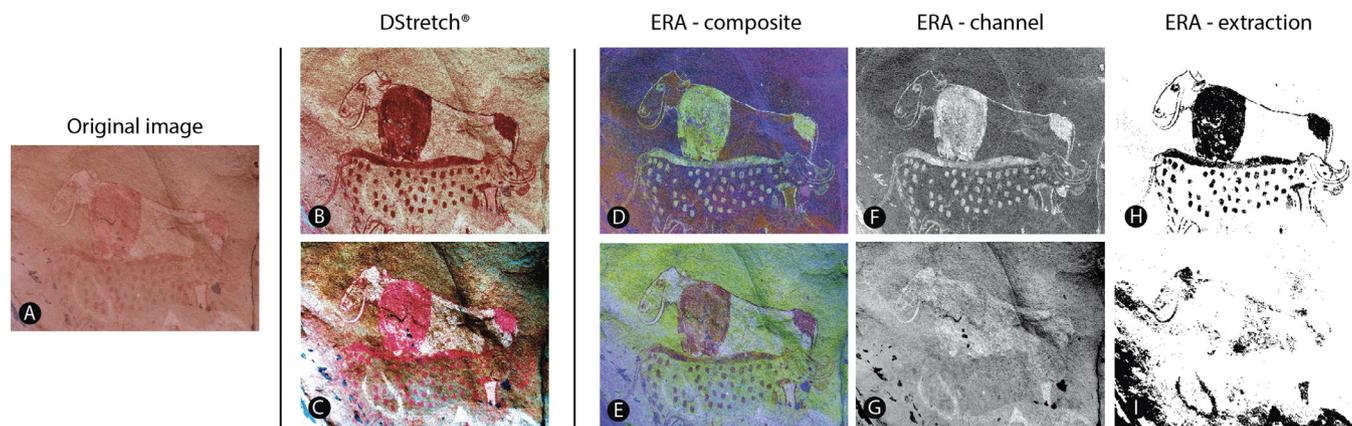


Fig. 7. Processing of an image representing a part of a herd, Oued In Djaren, Tadrat region, Tassili N'Ajjer National Park, Algeria. The picture captured with a SONY DSC-F828 in RAW format, at a resolution of 3264×2448 px, was converted into TIF. (a) original image; (b) DStretch® processing with the YRE colour space; (c) DStretch® processing with the LWE colour space; (d) false-coloured image produced by ERA using FastICA and XYZ colour space; (e) false-coloured image produced by ERA using FastICA and CMY(K) colour space; (f) channel 2 of the CMY(K) colour space after ZCA decorrelation by ERA; (g) channel 3 of the CMY(K) colour space after FastICA decomposition by ERA; (h) extraction of parts of the figures after supervision with the LR algorithm; (i) attempt to extract the bovid in the foreground after supervision with the LR algorithm. ERA images were produced with a contrast boost of 2%.

geria (Fig. 7). Three bovids organized on three distinct planes are distinguishable in some way on the original image (Fig. 7a). Again, processing by both DStretch® and ERA allows the figures to be much more clearly recognised. This is true for the YRE option of DStretch® (Fig. 7b) and the combination FastICA and XYZ colour space of ERA (Fig. 7d), for the two bovids in the background. The bovid present in the foreground is visible with the LWE option of DStretch® (Fig. 7c), as well as with the combination FastICA and CMY(K) colour space of ERA, which interestingly also marks the two other bovids. At least two groups of tints, reddish and whitish, with different shades, are present on this panel. The bovid in the foreground is whitish, the one in the middle is reddish, and the one in the background is composed of both tints. Using channel 2 of the CMY(K) colour space after ZCA decorrelation (Fig. 7f) for supervision, the reddish painting can be successfully extracted by machine learning (Fig. 7h). Note that the contours of the animals are well marked, even if they are thin. The whitish bovid in the foreground and inner parts of the one in the background are reasonably well rendered using channel 3 of the CMY(K) colour space after FastICA decomposition (Fig. 7g), but the automatic extraction of these areas falls short, except for one leg of the bovid in the background and parts of the bovid body in the foreground, where the whitish areas are very pronounced. This poor result comes with a number of undesired pixels. In such complex circumstances, when figures have partly and irregularly faded, when they overlap, and/or when they are drawn with different pigments, manual tracing from images produced by ERA remains a possible solution.

6. Conclusions

We demonstrate that the ERA software can be used easily, rapidly, and efficiently to identify painted areas in a rock art context presenting various colour schemes. The principles of whitening procedures that enhance colour are described comprehensively, as are the two options for boosting contrast between paintings and substrate. Tuning options are limited, to facilitate the integration of ERA in a routine workflow by researchers unfamiliar with image processing. The ERA software is made freely available in a compiled, straightforwardly executable version for Windows, and as an open-source Python snippet for those who want to improve the code (for example by integrating the unsupervised PyDRA procedure), or simply for those who desire complementary information about calculations. Interestingly, the Python version also runs on

Android, provided that a Python IDE (such as Pydroid 3, available on Google Play) and appropriate libraries are installed. Tests performed with a Samsung S5e tablet were conclusive. The images were processed quickly without any issues related to memory limitation, which can be a decisive asset in the field. Guidelines follow easy-to-understand principles, even without extensive knowledge of linear algebra, because they mainly rely on visual inspection and analysis of the elements composing the scene. Machine learning algorithms act as a background process, taking as input features a large set of channels, and not merely three (for RGB colour images), or even only one (for greyscale images). This is why the models trained by rough-and-ready supervision are often so efficient at producing documentation close to current standards in archaeology, with a low amount of noise (depending, of course, on the preservation of the painting and the quality of the training set used for supervision). In any case, if machine learning extraction of rock painting fails, manual tracing from images produced by ERA remains a potential option. Interestingly, panoramas or orthomosaics produced by photogrammetry can be seamlessly processed by ERA. Nothing prevents the integration of enhanced images produced by ERA into a photogrammetrical workflow to tackle the question of the relationship between rock art and rock surface, as previously proposed [46]. Finally, for the sake of traceability, it is advised to report details concerning the workflow applied and, if possible, to make available the original set of images used to produce the documentation. Such good practices would not only allow researchers to replicate the entire process, but also to compare existing and future technical solutions seeking to bring to light rock paintings.

Appendix A. Supplementary data

Supplementary material related to this article can be found, at https://gitlab.huma-num.fr/fmonna/era-extraction-from-rock-art/-/tree/Paper_Supp_Mat.

Acknowledgements

This research was funded by the Joint mission Mongolia – Monaco and the project ROSAS (uB-FC and RNMSH), and by the Observatoire des Patrimoines Sorbonne Université (A.-C. A. Ph.D. grant). We are grateful for comments by the anonymous reviewers, which have greatly improved the manuscript.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.culher.2022.09.018. https://gitlab.huma-num.fr/fmonna/era-extraction-from-rock-art/-/tree/Paper_Supp_Mat.

References

- [1] I. Domingo, B. Carrión, S. Blanco, J.L. Lerma, Evaluating conventional and advanced visible image enhancement solutions to produce digital tracings at el Carche rock art shelter, *DAACH* 2 (2015) 79–88.
- [2] I.D. Sanz, Rock art recording methods: from traditional to digital, in: C. Smith (Ed.), *Encyclopedia of Global Archaeology*, Springer, New York, 2014.
- [3] W. Burger, M.J. Burge, in: *Digital Image Processing: An Algorithmic Introduction Using Java*, Springer, 2016, p. 811.
- [4] L.M. Brady, Documenting and analyzing rock paintings from Torres Strait, NE Australia, with digital photography and computer image enhancement, *J. Field Archaeol.* 31 (2006) 363–379.
- [5] C. Horn, O. Ivarsson, C. Lindhé, R. Potter, A. Green, J. Ling, Artificial intelligence, 3D documentation, and rock art – approaching and reflecting on the automation of identification and classification of rock art images, *J. Archaeol. Method Theory* 29 (2022) 188–213.
- [6] T. Rolland, F. Monna, J.F. Buonocristiani, J. Magail, Y. Esin, B. Bohard, C. Chateau-Smith, Volumetric obscurity as a new tool to better visualize relief from digital elevation model, *Remote Sens.* 14 (2022) 941.
- [7] J. Harman, 2008. DStretch Algorithm description. Available at <http://www.dstretch.com/> (Accessed 30 June 2021).
- [8] N. Cortón Noya, Á. López García, F. Carrera Ramírez, Combining photogrammetry and photographic enhancement techniques for the recording of megalithic art in north-west Iberia, *DAACH* 2 (2015) 89–101.
- [9] A.R. Gillespie, A.B. Kahle, R.E. Walker, Color enhancement of highly correlated images. I. Decorrelation and HIS contrast stretch, *Remote Sens. Environ.* 20 (1986) 209–235.
- [10] R.E. Alley, Algorithm Theoretical Basis Document for Decorrelation Stretch, Version 2.2, 1996. NASA Jet Propulsion Laboratory, Pasadena California. http://eosps0.gsfc.nasa.gov/sites/default/files/atbd/ASTER_ATBD_99-2010.pdf.
- [11] I.T. Jolliffe, in: *Principal Component Analysis*, 2nd ed., Springer, New York, 2002, p. 488.
- [12] R.D. Dony, Karhunen-Loève transform, in: K.R. Rao, P.C. Yip (Eds.), *The Transform and Data Compression Handbook*, CRC Press LLC, Boca Raton, 2001.
- [13] Rasband W, ImageJ - Image processing and analysis in Java. <http://rsb.info.nih.gov/ij/>.
- [14] M.A. Rogerio-Candelera, V. Jurado, L. Laiz, C. Saiz-Jimenez, Laboratory and in situ assays of digital image analysis based protocols for biodeteriorated rock and mural paintings recording, *J. Archaeol. Sci.* 38 (2011) 2571–2578.
- [15] E. Cerrillo-Cuenca, P. Ortiz-Coder, J. Á. Martínez-del-Pozo, Computer vision methods and rock art: towards a digital detection of pigments, *Archaeol. Anthropol. Sci.* 6 (2014) 227–239.
- [16] E. Cerrillo-Cuenca, M. Sepúlveda, An assessment of methods for the digital enhancement of rock paintings: the rock art from the precordillera of Arica (Chile) as a case study, *J. Archaeol. Sci.* 55 (2015) 197–208.
- [17] D. Caldwell, U. Botzjorn, An historic sign, possible Mesolithic menhir, DStretch, and problems in dating rock art to the Sauveterrian in the Massif de Fontainebleau, *J. Archaeol. Sci.* 42 (2014) 140–151.
- [18] E. Kotoula, D.W. Robinson, C. Bedford, Interactive relighting, digital image enhancement and inclusive diagrammatic representations for the analysis of rock art superimposition: The main Pleito cave (CA, USA), *J. Archaeol. Sci.* 93 (2018) 26–41.
- [19] C. Defrasne, Digital image enhancement for recording rupestrian engravings: applications to an alpine rockshelter, *J. Archaeol. Sci.* 50 (2014) 31–38.
- [20] J.L. Le Quellec, J. Harman, C.I. Defrasne, F. Duquesnoy, D-Stretch® et l'amélioration des images numériques: applications à l'archéologie des images rupestres, *Les Cahiers de l'AARS* 16 (2013) 177–198.
- [21] E. Rodríguez González, S. Carbonell Pastor, J.R. Casals, Lost colours: Photogrammetry, image analysis using the DStretch plugin, and 3-D modelling of post-firing painted pottery from the south west Iberian Peninsula, *DAACH* 13 (2019) e00093.
- [22] E. Quesada, J. Harman, A step further in rock art digital enhancements. DStretch on Gigapixel imaging, *DAACH* 13 (2019) e00098.
- [23] L. Evans, A.L. Mourad, DStretch® and Egyptian tomb paintings: a case study from beni hassan, *J. Archaeol. Sci. Rep.* 18 (2018) 78–84.
- [24] E. Robert, S. Petrognani, E. Lesvignes, Applications of digital photography in the study of Paleolithic cave art, *J. Archaeol. Sci. Rep.* 10 (2016) 847–858.
- [25] L.M. Brady, A different look: comparative rock-art recording from the Torres Strait using computer enhancement techniques, *Aust. Aboriginal Stud.* 1 (2007) 98–115.
- [26] A. Kessy, A. Lewin, K. Strimmer, Optimal whitening and decorrelation, *Am. Stat.* 72 (2018) 309–314.
- [27] A.J. Bell, T.J. Sejnowski, The "Independent Components" of scenes are edge filters, *Vision Res.* 37 (1997) 3327–3338.
- [28] G.H. Golub, C.F. Van Loan, in: *Matrix computations*, 4th ed., Johns Hopkins University Press, 2013, p. 784.
- [29] E. Cerrillo Cuenca, M. Sepúlveda, Z. Guerrero-Bueno, Independent component analysis (ICA): a statistical approach to the analysis of superimposed rock paintings, *J. Archaeol. Sci.* 125 (2021) 105269.
- [30] A. Hyvärinen, Independent component analysis: recent advances, *Philos. Trans. A Math. Phys. Eng. Sci.* 371 (2013) 20110534.
- [31] A. Hyvärinen, E. Oja, Independent component analysis: algorithms and applications, *Neural Netw.* 13 (2000) 411–430.
- [32] S.S. Al-Amri, N.V. Kalyankar, S.D. Khamitkar, Linear and non-linear contrast enhancement image, *IJCSNS* 10 (2010) 139–143.
- [33] S.E. Han, B. Tao, T. Cooper, I. Tasi, Comparison between different color transformations for JPEG 2000, in: *Proceedings of the IS&T's 2000 PICS Conference*, 2000.
- [34] K. Dawson-Howe, in: *A Practical Introduction to Computer Vision with OpenCV*, Wiley, 2014, p. 235.
- [35] A.A. Patel, Hands-on unsupervised learning using Python, in: *How to Build Applied Machine Learning Solutions from Unlabeled Data*, O'Reilly, 2019, p. 400.
- [36] J.L. Lerma, Documentation and recovery of rupestrian paintings: an automatic approach, in: J. Albertz (Ed.), *Proceedings of the XVIII International Symposium of CIPA 2001 Postdam*, (Germany), 2002.
- [37] F. Monna, J. Magail, T. Rolland, N. Navarro, J. Wilczek, J.O. Gantulga, Y. Esin, L. Granjon, A.C. Allard, C. Chateau-Smith, Machine learning for rapid mapping of archaeological structures made of dry stones – example of burial monuments from the Khirgisuur culture, Mongolia –, *J. Cult. Herit.* 43 (2020) 118–128.
- [38] T.K. Kim, T test as a parametric statistic, *Korean J. Anesthesiol.* 68 (2015) 540–546.
- [39] S.Y. Kung, in: *Kernel Methods and Machine Learning*, Cambridge University Press, 2014, p. 572.
- [40] C.M. Bishop, in: *Pattern Recognition and Machine Learning*, Springer-Verlag NewYork Inc., 2006, p. 738.
- [41] B. Lantz, in: *Machine Learning With R*, 2nd ed., Packt Publishing, 2015, p. 452.
- [42] A. Géron, in: *Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow. Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed., O'Reilly, 2019, p. 600.
- [43] C.G. Northcutt, L. Jiang, I.L. Chuang, Confident Learning: estimating uncertainty in dataset labels, *J. Artif. Intell. Res.* 70 (2021) 1373–1411.
- [44] R.G. Gunn, C.L. Ogleby, D. Lee, R.L. Whear, A method to visually rationalize superimposed pigment motifs, *Rock Art Res.* (2010) 27–2.
- [45] A. Bovik, in: *Handbook of Image and video Processing*, Academic Press Inc., 2000, p. 891.
- [46] I. Domingo, V. Villaverde, E. López-Montalvo, J.L. Lerma, M. Cabrelles, Latest developments in rock art recording: towards an integral documentation of Levantine rock art sites combining 2D and 3D recording techniques, *J. Archaeol. Sci.* 40 (2013) 1879–1889.