.

Automatic feature extraction and classification of Iberian ceramics based on deep convolutional networks

Celia Cintas[a,*], Manuel Lucena[b], José Manuel Fuertes[b], Claudio Delrieux[c], Pablo Navarro[d], Rolando González-José[d], Manuel Molinos[e]

[a]*IBM Research Africa, Nairobi, Kenya.*
[b]*Department of Computer Science, University of Jaén, Spain.*
[c]*Departamento de Ingeniería eléctrica y de Computadoras, Universidad Nacional del Sur, CONICET, Bahía Blanca, Argentina.*
[d]*Instituto Patagónico de Ciencias Sociales y Humanas. Centro Nacional Patagónico, CONICET, Puerto Madryn, Argentina.*
[e]*Research University Institute for Iberian Archeology, University of Jaén, Spain.*

Abstract

Accurate classification of pottery vessels is a key aspect in several archaeological inquiries, including documentation of changes in style and ornaments, inference of chronological and ethnic groups, trading routes analyses, and many other matters. We present an unsupervised method for automatic feature extraction and classification of wheel-made vessels. A convolutional neural network was trained with a profile image database from Iberian wheel made pottery vessels found in the upper valley of the Guadalquivir River (Spain). During the design of the model, data augmentation and regularization techniques were implemented to obtain better generalization outcomes. The resulting model is able to provide classification on profile images automatically, with an accuracy mean score of 0.9013. Such computation methods will enhance and complement research on characterization and classification of pottery assemblages based on fragments.

Keywords:
Deep learning, Convolutional networks, Pottery profiles, Typologies.

1. Introduction and previous work

Ceramics are one of the most frequently found archaeological artifacts, and are a short-lived material. This property helps researchers to document variations in style, materials employed, and manufacturing techniques. Therefore, ceramics can be used to distinguish between chronological and ethnic groups, and to reconstruct the economic history to show trading routes and cultural relationships. This is especially the case with ceramic vessels, where shape and decoration are exposed to constantly changing fashions. This fact gives us a basis for dating the archaeological strata, and provides evidence of local production, trade relations, consumer behavior of the local population, etc. (Orton et al., 1993; Kampel and Sablatnig, 2003). In this context, analyzing and comparing artifacts in a quantitative manner has become a very important topic in modern Archeology, due to the existence of more and more affordable digitizing tools, and the high availability of computers powerful enough to handle the high volumes of data generated.

In past years several typologies, attending on different criteria, have been proposed in order to better study the material (Pereira Sieso, 1989). Since the selection of these criteria has depended on each individual researcher, those typologies have not contributed to the homogenization of the analysis of pottery shapes. It is therefore interesting for the archaeologist to have coherent, non-subjective judgment standards for classification. Moreover, in recent years a number of automated tools have been proposed in order for researchers to handle large archaeological datasets (Karasik and Smilansky, 2011; Lucena et al., 2014, 2016).

There are many wheel-made ceramic characterization techniques available in the literature that use the vessel profile as a distinguishing feature (Shennan and Wilcock, 1975; Rice, 1987). In (Nautiyal et al., 2006), a parametric representation method is proposed, mainly for archiving and visualization purposes. Also, profile comparison methods have been put forward, based on some distance between two given profiles, such as the one proposed in (Mom, 2007a), in which a distance based on the overlap maximization between profiles is used. This method, however, does not give good results for vessels (Mom, 2007b), and is more appropriate for solid objects. Subsequent proposals (Maaten et al., 2009) are based on well known local features, such as Shape Context descriptors (Belongie et al., 2002), combined with a multivariate analysis to compare vessels. Other approaches (Saragusti et al., 2005; Karasik and Smilansky, 2011) rely on a continuous profile shape interpolation, characterizing each profile by means of its radius, tangent and curvature along the contour.

Deep Learning (DL) is slowly emerging as a general-purpose machine learning model which has been successfully applied in a broad variety of contexts (Ciodaro et al., 2012; Ma et al., 2015; Taigman et al., 2014; LeCun et al., 2015; Dieleman et al., 2015; Cintas et al., 2016). DL is aimed to elaborate computational models consisting on a set of processing layers, which are capable of devising representations in increasing abstraction levels. In that way, the overall learning procedure elaborates autonomous (hidden) representations of the features present in the input data. Therefore, each layer is able to represent progressively more complex characterizations, until the desired abstraction level is reached and the network produced a stable and accurate embodiment of the features of the overall dataset. Image recognition constitutes one of the most successful application contexts of DL, where layers are able to detect and understand from simpler image features (for instance, the presence of edges or corners in particular orientations), to more complex underlying structures (borders with typical shapes that surround the aimed targets), and so on until the final recognition is achieved satisfactorily (LeCun et al., 2015). Since the power of current technology is steadily growing (both hardware performance and easy-to-use libraries for designing and testing these models), then it is not uncommon to find DL architectures with dozens of layers and hence the "deep" aspect of this technique. Specifically, in the field of Archeology, (Wang et al., 2017) propose a two-step recognition method for modeling style of Bodhisattva head images based on VGGNet. Also, (Llamas et al., 2016) present automatic techniques for sorting tasks of digital documentation of architectural heritage.

This work is part of an initiative developed in collaboration with the Research Institute for Iberian Archeology of the University of Jaén, to be used on wheel-made Iberian pottery (S.VI B.C – I A.C) from the upper valley of Guadalquivir River (Spain), a well documented region with many excavated sites (Chapa et al., 1997; Ruiz Rodríguez et al., 1983, 1984). The studied pieces come from different archaeological settlements located in the provinces from Jaén, Granada and Córdoba. We propose a classification based on automatic feature extraction and classification by means of training a deep convolutional network (CNNs) with profile images of vessels manually classified by domain experts. The final model is able to provide automatic classification over raw contour images, with an accuracy mean score of 0.9013.

2. Methods and implementation

In order to make this paper self-contained, we include in this Section a brief description of convolutional neural networks in their use for image classification. More deeper descriptions are available in the cited references. Readers with knowledge in the subject may proceed to the next Section without loosing track of the aim of the paper. Convolutional neural networks (*CNNs*), in turn, are a special case of DL networks, and were initially proposed by (Fukushima, 1980), but the first breakthroughs were only feasible more than a decade later. (LeCun et al., 1998) pioneered the use of CNNs in computer vision problems, and since then, CNNs were shown to be very effective for large-scale computer vision problems like image classification, object recognition, and many other (Toshev and Szegedy, 2014; Krizhevsky et al., 2012; Dieleman et al., 2015;

*Corresponding author.
emails: celia.cintas@ibm.com (Celia Cintas), mlucena@ujaen.es (Manuel Lucena), jmf@ujaen.es (José Manuel Fuertes), cad@uns.edu.ar (Claudio Delrieux), pnavarro@cenpat-conicet.gob.ar (Pablo Navarro), rolando@cenpat-conicet.gob.ar (Rolando González-José), mmolinos@ujaen.es (Manuel Molinos)

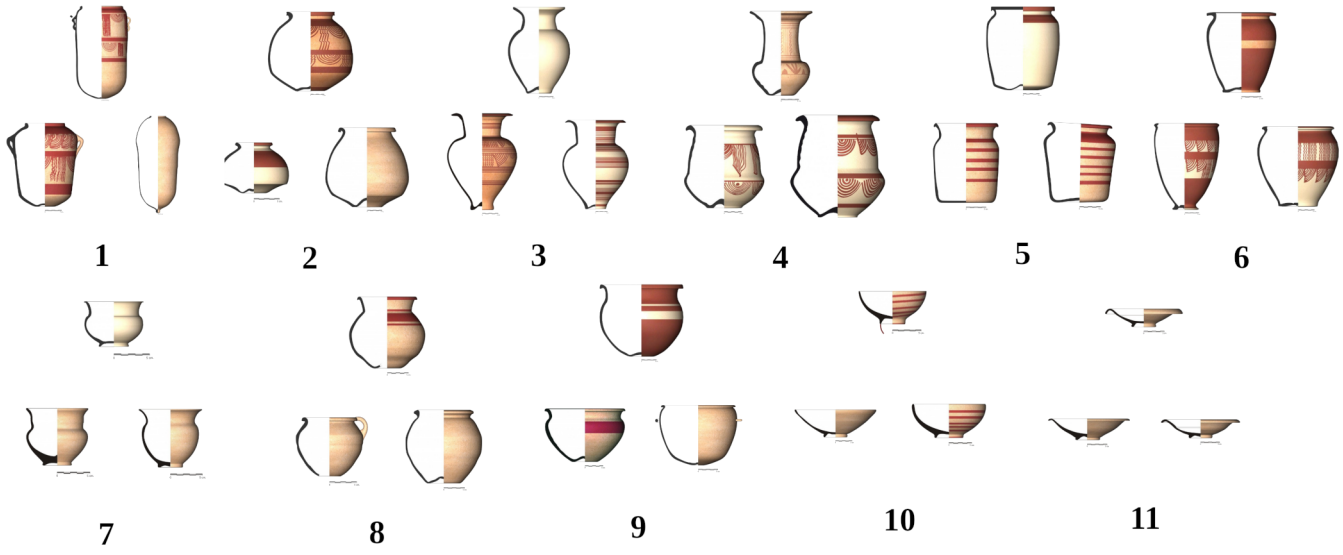

Figure 1: Example vessels from each of the database classes.

Table 1: Distribution of the dataset sample over the 11 classes

Type of Vessel	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
# of Samples	30	47	75	10	56	52	50	293	22	373	125

Tompson et al., 2015). A key aspect in computer vision is to ensure shift, scale and distortion invariance. For this, CNNs combine three features in their achitecture: local receptive fields, shared weights, and spatial sub-sampling (LeCun et al., 1998). In this way, the high spatial correlation, forming distinctive local patterns typically present in images, can be adequately represented. Another distinctive aspect of CNNs is that their topology is not fully connected. This drastically reduces the number of model parameters compared to a fully connected network, which is a key aspect during the training stage of the model to avoid overfitting (when the network has enough independent parameters to just memorize the training examples, without appropriate generalization). This connectivity reduction is achieved applying convolutional operations only in small disjoint regions of the input space, and also by sharing parameters between regions (Angermueller et al., 2016).

Consider a neural network with N layers. Input and output are represented respectively by vectors X_0, and X_N, where the vector X_{n-1} is the input to layer n (with $n = 1, \ldots, N$) (in our case, a two-dimension array with one channel). The output of layer X_n, can be represented as:

$$X_n = f(W_n X_{n-1} + b_n), \tag{1}$$

where W_n is a matrix of weights, b_n is a vector of biases, and f is the activation function. Even though f is expected to be derivable everywhere (for reasons that will become obvious below), a more efficient and still correct choice is the linear rectification $f(x) = \max(x, 0)$, The matrices of weights W_n and the vectors of biases b_n along all the layers constitute the *parameter space* of the model, which contains all the possible behaviors of the model. For the network to be able to learn a particular problem, these weights and biases must be updated according to a specific criterion. In our case, during the training stage, the learning algorithm aims to minimize a *loss function* that takes as inputs $\{W_n, b_n\}$ and returns the lowest possible learning error. In classification, the target value is categorical (nominal), as is the case in our work, and therefore the classification error may be defined as the proportion of misclassified items. Other learning purposes (regression, clustering, etc.) may require different error definitions. Cross Entropy Loss (Eq. 2) is particularly adequate when classes are unbalanced (the available data is not evenly distributed among all the classes). This is the case in our specific context (See Table 1). since the measure is averaged across observations for each batch.

$$\text{loss}(x, k) = -\log\left(\frac{\exp(x[k])}{\sum_j \exp(x[j])}\right) = -x[k] + \log\left(\sum_j \exp(x[j])\right) \tag{2}$$

A common way to compute Eq. 2 is to iterate the *gradient-based descent*, which is the generalization of the perceptron rule. In this method, the error gradient is estimated (in a given misclassified example, or averaged in a batch of examples), and the weight matrices are updated adding to each term a given factor η of the error gradient. η is the *learning rate*, a hyperparameter controlling the stride towards convergence. This procedure is iterated (along the so called *epochs*) until an error criterion is achieved and the training stage is considered finished. This amounts to traverse the parameter space from an arbitrary initial condition (even a purely random one) using the error gradient to define a trajectory that eventually leads to an adequate final condition. The use of a constant η factor has been proven to be

inefficient, and also the recomputation of the error gradient in every epoch may turn into undesired quasi-oscillatory behavior. For this reason, an inertial or *momentum*-like behavior is simulated, in which the tendency during training is to avoid abrupt changes in the trajectory. In our case we use Adaptive Moment Estimation (ADAM) (Kingma and Ba, 2015), which computes adaptive learning rates for each parameter, in addition to storing an exponentially decaying average of past squared gradients, behaving in a similar way as an inertia momentum (Ruder, 2016; Heusel et al., 2017). If X_{true} is the expected output corresponding to the network input X_0, then the goal of the training stage can be understood as finding parameter values that make the output X_n to be X_{true} for all the inputs. The *prediction error* is $e(X_N, X_{true})$ and the gradient of $e(X_N, X_{true})$ is then computed with respect to the weights and biases $\{W_n, b_n\}$. As explained above, according to the perceptron rule, the parameter values of each layer are then modified by repeatedly taking controlled steps in the direction opposite to the gradient:

$$W_n \leftarrow W_n - \eta \frac{\partial e(X_N, X_{true})}{\partial W_n}, \tag{3}$$

and

$$b_n \leftarrow b_n - \eta \frac{\partial e(X_N, X_{true})}{\partial b_n}, \tag{4}$$

where, η is the learning rate, a hyperparameter controlling the stride towards convergence. In CNNs in particular, the parameters represent learnable convolution filters, where initial regions of the previous layer are taken as input and then a convolution is applied to produce a stack of output feature maps. The input of layer n can be unfolded as a set of K matrices $X_{n-1}^{(k)}$, with $k = 1, \ldots, K$. Each of these matrices represents a different input feature map. The output feature maps $X_n^{(l)}$, with $l = 1, \ldots, L$ are represented as follows:

$$X_n^{(l)} = f\left(\sum_{k=1}^{K} W_n^{(k,l)} * X_{n-1}^{(k)} + b_n^{(l)}\right), \tag{5}$$

where $*$ represents the two-dimensional convolution operation. The matrices $W_n^{(k,l)}$ represent the coefficients of the convolution filters of layer n, and $b_n^{(l)}$ represents the bias for feature map l. Note that a feature map $X_n^{(l)}$ is obtained by computing a sum of K convolutions with the feature maps of the previous layer (Dieleman et al., 2015). To this general structure, some optimizations can be performed. One is to only connect each unit to a local subset of the units in the layer below, and each unit is replicated across the entire input. This achieves better generalization performances. Second, pooling layers may be located between convolutional layers, computing some aggregation function (typically the maximum or the mean) across small local regions of the input (Boureau et al., 2010), thus reducing significantly the dimensionality of the feature map.

3. Automated Vessel Identification

Given the limitations of the current vessel identification proposals mentioned in Section 1, we propose a method, based on novel deep learning algorithms. A set of 1133 profile images of vessels was selected as input data to train a CNN, using several learning techniques (detailed in this Section) to achieve a high generalization performance rate.

3.1. Dataset

The raw information belong to binary profile images, corresponding to Iberian wheel made pottery from various archaeological sites of the upper valley of the Guadalquivir River (Spain). Reference classification has been done by an expert group, based on morphological criteria, taking into account the presence or absence of certain parts, such as lip, neck, body, base and handles, and the ratios between their corresponding sizes. According to these criteria, vessels can be classified as belonging to one of 11 different classes (Fig. 1), each one with different number of elements (Table 1). Nine of them correspond to closed shapes, and the two remaining correspond to open shapes (Lucena et al., 2014). The available images consist of a profile view of the pottery, where image resolutions (in pixels), corresponding to size scale, may vary according to the acquisition settings. Images without prior expert classification were excluded. The resulting dataset is then composed of 1133 classified images, which was split into a training subset containing 793 images (70%), a validation set of 113 (10%) and a testing set of 227 images (20% of the total dataset), which were selected with a random permutation cross-validation iterator.

The feature representation of the raw data distribution can be seen at Fig. 2 using t-distributed Stochastic Neighbor Embedding (t-SNE) (Van Der Maaten and Hinton, 2008), a variation of Stochastic Neighbor Embedding (Hinton and Roweis, 2002), which is an enhanced method for representing high dimensional data by giving each data point a location in a three-dimensional map.

3.2. CNN architecture and training

Several experiments were performed with different CNN architectures, training hyper-parameters and data augmentation techniques for automatic classification, *i.e.*, to extract features and identify vessels in raw image datasets. By means of hyper-parameter optimization (Bergstra et al., 2011), the experiments were performed with several variations of regularization techniques (dropout, batch normalization, etc.), number of convolutional layers and their respective kernel size, gradient methods and learning rates.

The input for the designed CNN consist of a single-channel vessel image of size 64×64 pixels, all images were resized during training on batch mode employing bilinear down sampling, with brightness values scaled to $[0, 1]$, see Fig. 4 for examples.

In Fig. 3 the chosen architecture is depicted. The basic substructure used to build the network architecture consists of two convolution layers with square filters, followed by a batch normalization layer. The convolutional layers *Conv2d-1*, *Conv2d-3* have 64 and 128 filters of size 4×4. This substructure refers to the feature extraction step, if we move forward in the architecture (deeper layers), we will find the subset of layers responsible for classification, this task is implemented my means

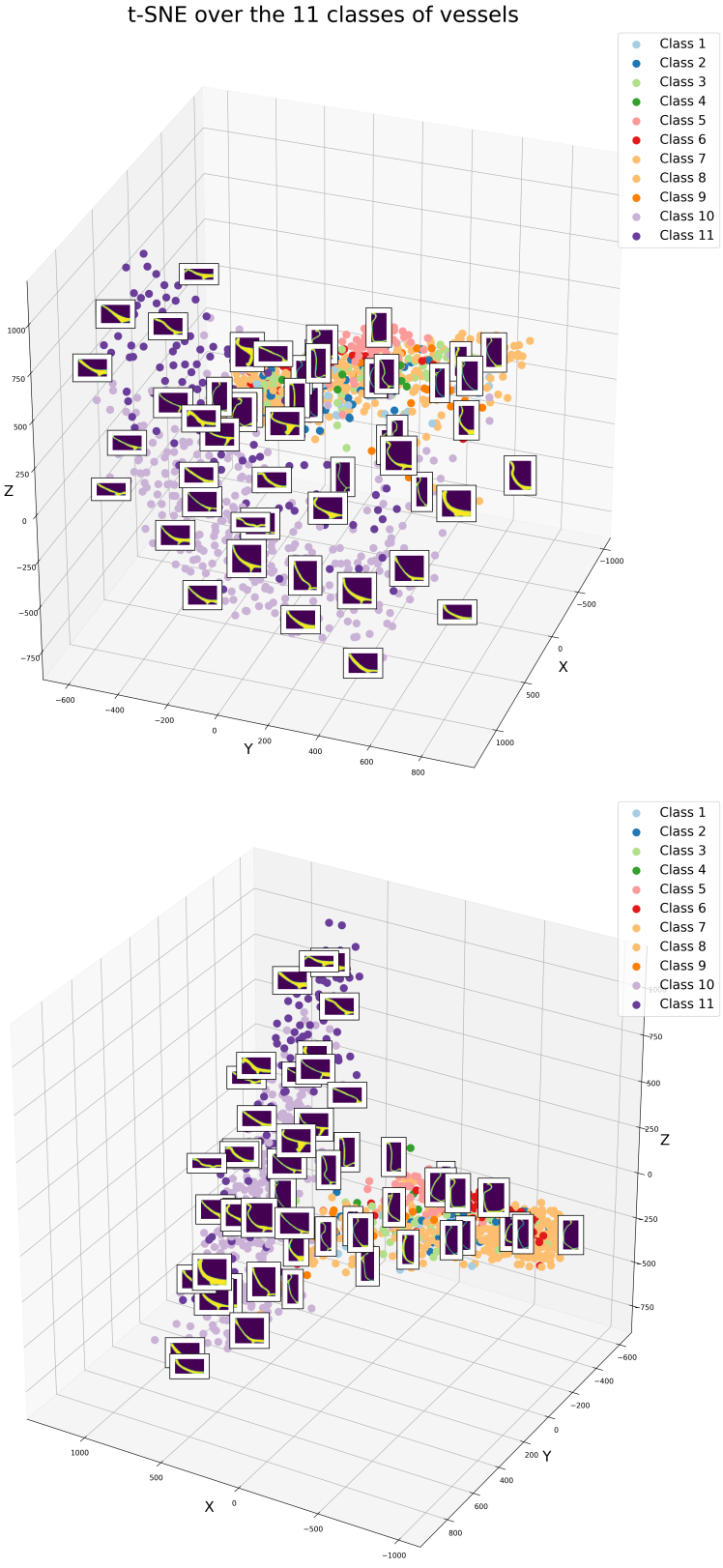

Figure 2: The feature representation of the raw data distribution using t-SNE is depicted. The colors represent the class assigned to the element defined by the expert. We can observe two principal clusters formed in the high dimensional space, which belong to closed shapes (classes 1 to 9) and the second cluster to open shapes (class 10 and 11), refer to Fig. 1 for a detail description of the pottery vessels morphology.

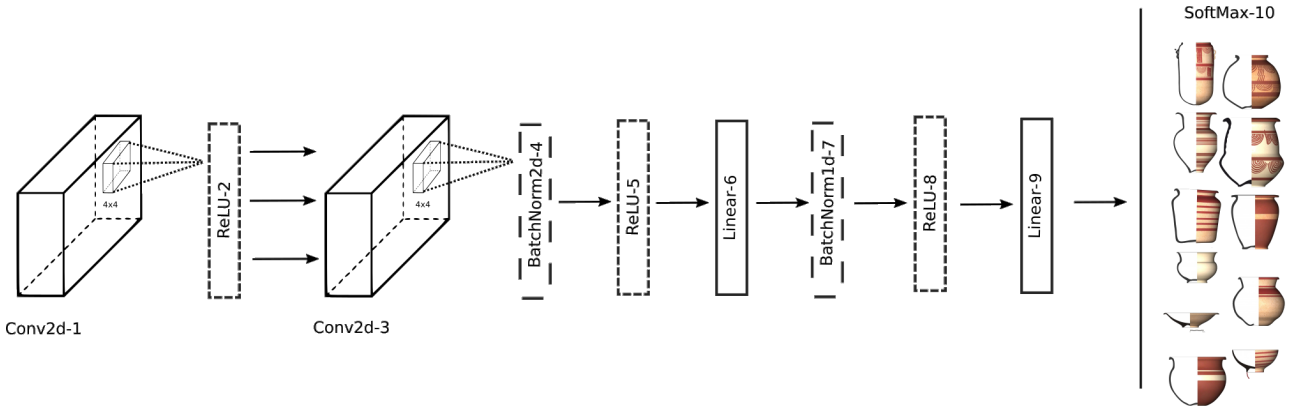

Figure 3: Structure used for building the CNN for automatic classification of Iberian Vessels.

of a second substructure consisting of two fully connected layers with 1024 units each (*Linear-6* and *Linear-9* in Fig. 3), and a normalization layer in between (*BatchNorm1d*). The output layer (*SoftMax-10*) is composed of 11 output units for the predicted class. The coefficients used for computing averages of gradient for the optimization algorithm Adam (described previously in Section 2) were $\beta_1 = 0.5$ and $\beta_2 = 0.999$, obtained using grid search for hyper parameter optimization. The implementation used Python and the PyTorch library (Paszke et al., 2017). This allows the use of GPU acceleration for the training stage. The code of the network architecture, training and validation is available at [1]. The training phase of the CNN took approximately 6 minutes using NVIDIA Titan Xp Pro cards provided by the "Signal Processing in Telecommunication Systems" research group (TIC-188).

3.3. Data Augmentation and Regularization

CNNs usually have a large quantity of trainable parameters, 33.701.323 in the case of the selected model for the vessel classification. Due to the restricted amount of samples in the training dataset, overfitting is an issue that must be addressed. The network will tend to remember the training examples rather than extracting inherent abstractions of the dataset, because available memory in the form of trainable parameters is enough to do so. This is a serious drawback since the classifier will not generalize well enough to classify new data adequately. To reduce overfitting we use two basic techniques: data augmentation during training, and batch normalization.

For data augmentation, the dataset was artificially enlarged applying random, label-preserving transformations to the original images. Randomly selected images were mirrored over the x axis and added to the dataset during training (see examples in Fig. 4 (d)) with a probability of 0.5. Also some of the images were scaled with a factor interval of $(0.7, 1.0)$, as shown in Fig. 4 (b, c, d, e). Batch normalization, in turn, standardizes the inputs of every layer to zero mean and unit variance.

[1] https://github.com/celiacintas/vasijas/blob/master/CNN_vessels.ipynb

Table 2: Score table per vessel class and average totals over test subset.

# class	precision	recall	f1-score	# images
1	0.80	1.00	0.89	4
2	0.95	0.97	0.96	73
3	0.97	0.91	0.94	33
4	0.38	0.75	0.50	4
5	0.92	0.63	0.75	19
6	0.50	0.50	0.50	2
7	1.00	0.88	0.93	8
8	0.80	0.89	0.84	9
9	0.67	0.86	0.75	7
10	0.92	0.89	0.91	66
11	0.67	1.00	0.80	2
Total average	**0.91**	**0.89**	**0.90**	**227**

This greatly reduces the amount by what the hidden unit values shift around during learning, allowing faster learning rates and achieving higher invariance with initialization. It also acts as a regularizer, in some cases eliminating the need for dropouts, as in the presented solution for vessel classification. In training with batch normalization, a training example is seen in conjunction with other examples in mini-batches, and the network no longer produces deterministic values for a given training example (Ioffe and Szegedy, 2015).

4. Experiments and analysis

To validate the obtained results, the automated typification was evaluated regarding standard classification metrics such as precision, recall, and f1-score with previously unseen images in the test set (see Table 2). The whole cross validation process (partitioning into training, test, and validation sets, and performance evaluation) was performed randomly five times, producing an accuracy mean score equal to 0.9013 with a standard deviation of 0.0059.

The metrics calculated for the chosen architecture over the unseen test dataset can be seen in Table 2. The classification metrics were calculated using the library *scikit-learn* (Pedregosa

Figure 4: Examples of images 64×64 with affine transformations applied randomly during the training stage.

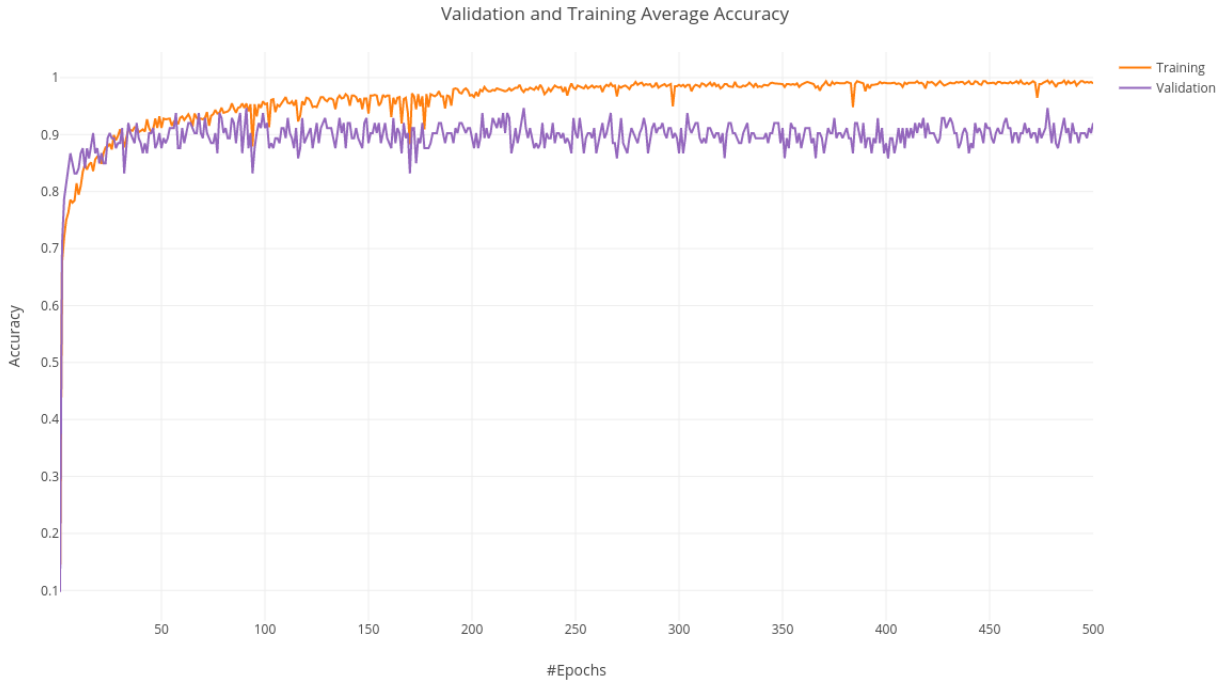

Figure 5: Average accuracy curves over validation and training set of the CNN detailed in Fig 3. The orange line depict the error on the training subset, and the violet line the validation error for the best performing CNN.

et al., 2011). Also, the accuracy outlines depicting the training and validation error of the network is shown in Fig. 5, furthermore, the noise present in the validation curve can be explained due to the small size of the validation set ($n = 113$), nonetheless both curves respect the same growth pattern. The Fig. 6 shows the normalized confusion matrix over the test dataset ($n = 227$), to avoid misleading visualization due to unbalanced classes. The code for training the CNN, structure, metrics and testing of the CNN can be seen in `https://github.com/celiacintas/vasijas`.

5. Discussion and conclusion

Our new technique presents a novel method for fully automatic pottery profile classification based on profile images and deep learning algorithms, outperforming our previous classification methods (Lucena et al., 2014, 2016, 2017). The method uses CNNs for automatic feature extraction and typification. After training the network with images of profile pottery tagged by domain experts, the resulting algorithm is able to classify correctly the type of vessel presented in the image. The presented pipeline is entirely automatic, with no supervised fine-tunning steps or previous *ad-hoc* features extraction tasks. The technique was implemented using open source libraries, and the source code is available on the already mention public repository, a quick guide of the pipeline is presented on the form of a *Jupyter Notebook*[2] (Kluyver et al., 2016). The proposed solution is suitable to perform typification over other type of archaeological material, such as the outline of specific stone projectiles (spears, darts, arrows, etc). Therefore it may be used for several other studies, including the analysis of fragments produced by the original design and production of projectile points or its retouching and reactivation process, and many other topics. Regarding future improvements, the worst performing classes for this problem over the test subset are the number 4 and 9 (see Fig. 6), possibly due to the size of the samples in the respective classes (see Table 1 for numerical

[2] `https://github.com/celiacintas/vasijas/blob/master/CNN_vessels.ipynb`

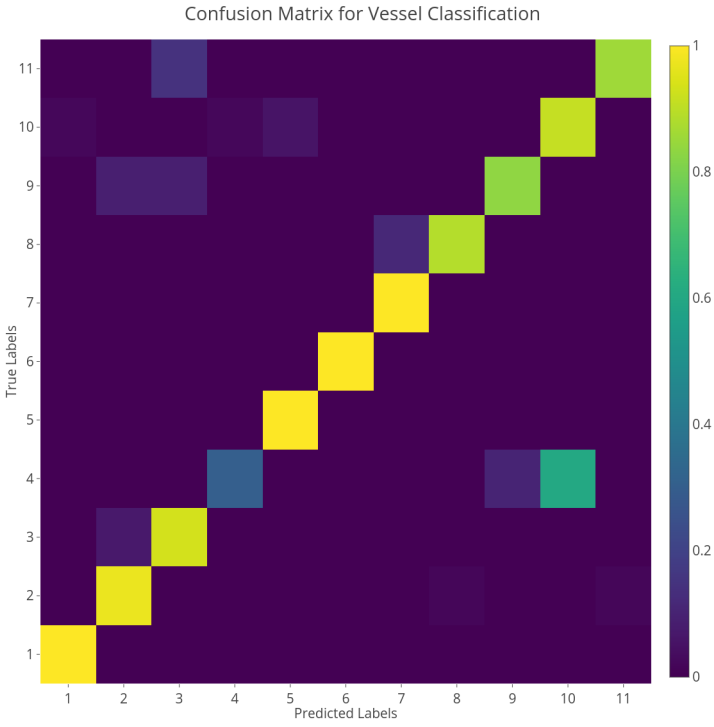

Figure 6: Normalized confusion Matrix of the predicted results of the network over a randomized test dataset of vessels images (n = 227). We can see that the network perform correctly in most of the examples except in the classes 4 and 9.

reference), a possible solution is to obtain new elements corresponding to these classes or to artificially enlarge the dataset applying GAN networks (Goodfellow et al., 2014) to generate Iberian vessels, that preserve geometric information. In a similar vein, deep learning algorithms are being tested for 3D vessel generation based on 3D GANs (Wu et al., 2016) and 2D GANs with solid revolution methods for fragment detection and reconstruction.

Acknowledgements

This work has been supported by Ibero-American University Association of Postgraduate (AUIP) and the Ministry of Economy and Knowledge of the Junta de Andalucía as sponsors of the Ibero-American Postdoctoral Mobility Program of the AUIP, the Center for Advanced Studies in Information and Communication Technologies (CEATIC) and the Research University Institute for Iberian Archeology of the University of Jaén.

References

Angermueller, C., Pärnamaa, T., Parts, L., Stegle, O., 2016. Deep learning for computational biology. Molecular Systems Biology 12 (7), 878.
URL: https://onlinelibrary.wiley.com/doi/abs/10.15252/msb.20156651
DOI: 10.15252/msb.20156651

Belongie, S., Malik, J., Puzicha, J., April 2002. Shape matching and object recognition using shape contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (4), 509–522.

Bergstra, J. S., Bardenet, R., Bengio, Y., Kégl, B., 2011. Algorithms for hyper-parameter optimization: Advances in neural information processing systems. pp. 2546–2554.

Boureau, Y.-L., Ponce, J., Lecun, Y., 2010. A Theoretical Analysis of Feature Pooling in Visual Recognition. Proceedings of the 27th International Conference on Machine Learning (2010), 111–118.

Chapa, T., Pereira, J., Madrigal, A., Mayoral, V., 1997. La Necrópolis ibérica de Castellones de Ceal (Hinojares, Jaén). Consejeria de Cultura. Junta de Andalucia.

Cintas, C., Quinto-Sánchez, M., Acuña, V., Paschetta, C., De Azevedo, S., Silva de Cerqueira, C., Ramallo, V., Gallo, C., Poletti, G., Bortolini, M. C., Canizales-Quinteros, S., Francisco, R., Bedoya, G., Ruiz-Linares, A., Gonzaléz-José, R., Delrieux, C., dec 2016. Automatic ear detection and feature extraction using Geometric Morphometrics and Convolutional Neural Networks. IET Biometrics.
DOI: 10.1049/iet-bmt.2016.0002

Ciodaro, T., Deva, D., de Seixas, J. M., Damazio, D., jun 2012. Online particle detection with Neural Networks based on topological calorimetry information. Journal of Physics: Conference Series 368 (1), 012030.
DOI: 10.1088/1742-6596/368/1/012030

Dieleman, S., Willett, K. W., Dambre, J., 2015. Rotation-invariant convolutional neural networks for galaxy morphology prediction. Monthly Notices of the Royal Astronomical Society 450 (2), 1441–1459.
DOI: 10.1093/mnras/stv632

Fukushima, K., 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics 36 (4), 193–202.
DOI: 10.1007/BF00344251

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative Adversarial Nets. Advances in Neural Information Processing Systems 27, 2672–2680.
DOI: 10.1017/CBO9781139058452

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Klambauer, G., Hochreiter, S., 2017. Gans trained by a two time-scale update rule converge to a nash equilibrium. CoRR abs/1706.08500.

Hinton, G. E., Roweis, S. T., 2002. Stochastic neighbor embedding. Advances in neural information processing systems, 833–840.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. CoRR abs/1502.03167.

Kampel, M., Sablatnig, R., 2003. An automated pottery archival and reconstruction system. Journal of Visualization and Computer Animation 14 (3), 111–120.

Karasik, A., Smilansky, U., 2011. Computerized morphological classification of ceramics. Journal of Archaeological Science 38 (10), 2644 – 2657.

Kingma, D. P., Ba, J. L., 2015. Adam: a Method for Stochastic Optimization. International Conference on Learning Representations 2015, 1–15.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C., 2016. Jupyter notebooks – a publishing format for reproducible computational workflows: Loizides, F., Schmidt, B. (Eds.), Positioning and Power in Academic Publishing: Players, Agents and Agendas. IOS Press, pp. 87 – 90.

Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. Advances In Neural Information Processing Systems, 1–9.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.
DOI: 10.1038/nature14539

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86 (11), 2278–2323.
DOI: 10.1109/5.726791

Llamas, J., Lerones, P. M., Zalama, E., Gómez-García-Bermejo, J., 2016. Applying deep learning techniques to cultural heritage images within the INCEPTION project: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Vol. 10059 LNCS. pp. 25–32.
DOI: 10.1007/978-3-319-48974-2_4

Lucena, M., Fuertes, J. M., Martinez-Carrillo, A. L., Ruiz, A., Carrascosa, F., 2016. Efficient classification of Iberian ceramics using simplified curves. Journal of Cultural Heritage 19, 538–543.
DOI: 10.1016/j.culher.2015.10.007

Lucena, M., Fuertes, J. M., Martínez-Carrillo, A. L., Ruiz, A., Carrascosa, F., Oct 2017. Classification of archaeological pottery profiles using modal analysis. Multimedia Tools and Applications 76 (20), 21565–21577.
URL: https://doi.org/10.1007/s11042-016-4076-9
DOI: 10.1007/s11042-016-4076-9

Lucena, M., Martínez-Carrillo, A. L., Fuertes, J., Carrascosa, F., Ruiz, A., 2014. Decision support system for classifying archaeological pottery profiles based on mathematical morphology. Multimedia Tools and its Applications.
DOI: 10.1007/s11042-014-2063-6

Ma, J., Sheridan, R. P., Liaw, A., Dahl, G. E., Svetnik, V., feb 2015. Deep neural nets as a method for quantitative structure-activity relationships. Journal of chemical information and modeling 55 (2), 263–74.
DOI: 10.1021/ci500747n

Maaten, L., Lange, G., Boon, P., 2009. Visualization and automatic typology construction of pottery profiles: Frischer, B. (Ed.), Making history interactive: computer applications and quantitative methods in archaeology (CAA). Vol. 2079 of BAR International Series. Archaeopress, Oxford u.a., pp. 356–362.

Mom, V., 2007a. SECANTO - The SECtion Analysis TOol: Figueiredo, A., Velho, G. L. (Eds.), The world is in your eyes. CAA2005. Computer Applications and Quantitative Methods in Archaeology. Tomar, Portugal, pp. 95–101.

Mom, V., 2007b. Where did i see you before... a holistic method to compare and find archaeological artifacts: Decker, R., Lenz, H. J. (Eds.), Advances in Data Analysis. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 671–680.

Nautiyal, V., Kaushik, V. D., Pathak, V. K., Dhande, S., Nautiyal, S., Naithani, M., Juyal, S., Gupta, R. K., Vasisth, A. K., Verna, K. K., Singh, A., 2006. Geometric modeling of indian archaeological pottery: A preliminary study: Clark, J., Hagemeister, E. (Eds.), Exploring New Frontiers in Human Heritage. CAA2006. Computer Applications and Quantitative Methods in Archaeology. Fargo, United States.

Orton, C., Tyers, P., Vinci, A., 1993. Pottery in Archaeology. Cambridge University Press, United Kingdom.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., 2017. Automatic differentiation in pytorch.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.

Pereira Sieso, J., 1989. La cerámica ibérica de la cuenca del Guadalquivir. Trabajos de Prehistoria 46, 149–159.

Rice, P. M., 1987. Pottery Analysis. University of Chicago Press, Chicago.

Ruder, S., 2016. An overview of gradient descent optimization algorithms. CoRR abs/1609.04747.

Ruiz Rodríguez, A., Hornos Mata, F., Choclán, C., Cruz Garrido, J., 1984. La necrópolis ibérica Finca Gil de Olid (Puente del Obispo-Baeza). Cuadernos de Prehistoria de la Universidad de Granada (9), 195–234.

Ruiz Rodríguez, A., Molinos, M., López, J., Crespo, J., Choclán, C., Hornos, F., 1983. El horizonte ibérico antiguo del Cerro de la Coronilla (Cazalilla, Jaén). Cortes A y F. Cuadernos de Prehistoria de la Universidad de Granada (8), 251–295.

Saragusti, I., Karasik, A., Sharon, I., Smilansky, U., 2005. Quantitative analysis of shape attributes based on contours and section profiles in artifact analysis. Journal of Archaeological Science 32 (6), 841 – 853.

Shennan, S., Wilcock, J., 1975. Shape and style variation in central german bell beakers. Science and Archaeology 15, 17–31.

Taigman, Y., Yang, M., Ranzato, M., Wolf, L., 2014. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. Conference on Computer Vision and Pattern Recognition (CVPR), 8.
DOI: 10.1109/CVPR.2014.220

Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C., June 2015. Efficient object localization using convolutional networks: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 648–656.
DOI: 10.1109/CVPR.2015.7298664

Toshev, A., Szegedy, C., 2014. DeepPose: Human Pose Estimation via Deep Neural Networks: Computer Vision and Pattern Recognition (CVPR). pp. 1653–1660.
DOI: 10.1109/CVPR.2014.214

Van Der Maaten, L. J. P., Hinton, G. E., 2008. Visualizing high-dimensional data using t-sne. Journal of Machine Learning Research 9, 2579–2605.
DOI: 10.1007/s10479-011-0841-3

Wang, H., He, Z., Huang, Y., Chen, D., Zhou, Z., 2017. Bodhisattva head images modeling style recognition of Dazu Rock Carvings based on deep convolutional network. Journal of Cultural Heritage 27, 60–71.
DOI: 10.1016/j.culher.2017.03.006

Wu, J., Zhang, C., Xue, T., Freeman, W. T., Tenenbaum, J. B., 2016. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. CoRR abs/1610.07584.